ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0070

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Igor COLIN

le 24 novembre 2016

Adaptation des méthodes d'apprentissage aux U-statistiques

Directeur de thèse : **Stephan CLÉMENÇON**
Co-encadrement de la thèse : **Joseph SALMON**

Jury

M. Stephan CLÉMENÇON, Professeur, Télécom ParisTech	Directeur
M. Joseph SALMON, Maître de conférences, Télécom ParisTech	Co-directeur
M. Alexandre D'ASPREMONT, Professeur, École Normale Supérieure	Rapporteur
M. Mikael JOHANSSON, Professeur, KTL Electrical Engineering	Rapporteur
M. Peter RICHTARIK, Maître de conférences, University of Edinburgh	Examinateur
M. Pascal BIANCHI, Professeur, Télécom ParisTech	Examinateur

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

Abstract

With the increasing availability of large amounts of data, computational complexity has become a keystone of many machine learning algorithms. Stochastic optimization algorithms (*e.g.*, stochastic gradient descent) and distributed/decentralized methods have been widely studied over the last decade and provide increased scalability for optimizing an empirical risk that is separable in the data sample. Yet, in a wide range of statistical learning problems such as ranking, clustering or metric learning among others, the risk is accurately estimated by U-statistics, *i.e.*, functionals of the training data with low variance that take the form of averages over d-tuples. This thesis is dedicated to extending methods developed for sample mean empirical risk to U-statistics. We first tackle the problem of sampling for the empirical risk minimization problem. We show that empirical risks can be replaced by drastically computationally simpler Monte-Carlo estimates based on $O(n)$ terms only, usually referred to as incomplete U-statistics, without damaging the learning rate. We establish uniform deviation results describing the error made when approximating a U-process by its incomplete version under appropriate complexity assumptions. Numerical examples are displayed in order to provide strong empirical evidence that such approach largely surpasses more naive subsampling techniques. We then focus on the decentralized estimation topic, where the data sample is distributed over a connected network. We introduce new synchronous and asynchronous randomized gossip algorithms which simultaneously propagate data across the network and maintain local estimates of the U-statistic of interest. We establish convergence rate bounds of $O(1/t)$ and $O(\log t/t)$ for the synchronous and asynchronous cases respectively, where t is the number of iterations, with explicit data and network dependent terms. Beyond favorable comparisons in terms of rate analysis, numerical experiments provide empirical evidence the proposed algorithms surpasses the previously introduced approach. Finally, we deal with the decentralized optimization of functions that depend on pairs of observations. Similarly to the estimation case, we introduce a method based on concurrent local updates and data propagation. Our gossip methods are based on dual averaging and aims at solving such problems both in synchronous and asynchronous setting. The proposed framework is flexible enough to deal with constrained and regularized variants of the optimization problem. Our theoretical analysis reveals that the proposed algorithms preserve the convergence rate of centralized dual averaging up to an additive bias term. Our simulations illustrate the practical interest of our approach on AUC maximization and metric learning problems.

Résumé de la thèse

L'explosion récente des volumes de données disponibles a fait de la complexité algorithmique un élément central des méthodes d'apprentissage automatique. Les algorithmes d'optimisation stochastique, comme la descente de gradient stochastique, ainsi que les méthodes distribuées et décentralisées ont été largement développés durant les dix dernières années. Ces méthodes ont permis de faciliter le passage à l'échelle pour optimiser des risques empiriques dont la formulation est séparable en les observations associées. Pourtant, dans de nombreux problèmes d'apprentissage statistique, allant de l'ordonnancement au partitionnement en passant par l'apprentissage de métrique, l'estimation précise du risque s'effectue à l'aide de U-statistiques, des fonctions des données prenant la forme de moyennes sur des d-uplets. Cette thèse vise à étendre aux U-statistiques des méthodes développées spécifiquement pour un risque sous forme de moyenne empirique. Nous nous intéressons tout d'abord au problème de l'échantillonage pour la minimisation du risque empirique. Nous montrons que le risque peut être remplacé par un estimateur de Monte-Carlo, intitulé U-statistique incomplète, basé sur seulement $O(n)$ termes et permettant de conserver un taux d'apprentissage du même ordre. Nous établissons des bornes sur l'erreur d'approximation du U-processus par sa version incomplète sous certaines hypothèses de complexité. Les simulations numériques mettent en évidence l'avantage d'une telle technique d'échantillonnage sur une approche plus naïve. Nous portons par la suite notre attention sur l'estimation décentralisée, où les observations sont désormais distribuées sur un réseau connexe. Nous élaborons des algorithmes dits *gossip*, dans des cadres synchrones et asynchrones, qui diffusent les observations tout en maintenant des estimateurs locaux de la U-statistique à estimer. Nous démontrons que ces algorithmes convergent à des vitesses de $O(1/t)$ et $O(\log t/t)$, respectivement pour les versions synchrones et asyncrhones, avec des dépendances explicites en les données et la topologie du réseau. Les simulations numériques confirment la supériorité de ces méthodes sur l'état de l'art. Enfin, nous traîtons de l'optimisation décentralisée de fonctions dépendant de paires d'observations. De même que pour l'estimation, nos méthodes sont basées sur la concomitance de la propagation des observations et l'optimisation local du risque. Ces algorithmes sont fondés sur la méthode du *dual averaging* et peuvent être formulées dans des cadres aussi bien synchrones qu'asynchrones. Notre analyse théorique souligne que ces méthodes conservent une vitesse de convergence du même ordre que dans le cas centralisé, à un terme de biais près. Les expériences numériques confirment l'intérêt pratique de notre approche sur des problèmes tels que l'apprentissage de métrique ou la maximisation de l'aire sous la courbe ROC.

Contents

List of Figures

List of Tables

List of Symbols

General syntax

$\mathcal{X}$	Set.
$\mathbf{x}$	Vector.
$\mathbf{X}$	Matrix.
X	Random variable.
$:=$	Definition, *e.g.*, $x := 2$.

Probability and statistics

n	Sample size.
d	Feature space dimension.
$\mathcal{X}$	Feature space ($\mathcal{X} \subseteq \mathbb{R}^d$).
μ, ν	Probability measures.
$\mathbb{E}_\mu$	Expectation with respect to probability measure μ [1].
$\widehat{\cdot}$	Empirical estimator.
$R(\cdot)$	True risk, *i.e.*, $R(\theta) = \mathbb{E}_{(X,Y)}[\mathbb{1}_{\{\theta(X) \neq Y\}}]$.
$R_n(\cdot)$	Empirical risk, *i.e.*, $R_n(\theta) = (1/n) \sum_{i=1}^n \mathbb{1}_{\{\theta(X_i) \neq Y_i\}}$, for a given sample $(X_i, Y_i)_{1 \leq i \leq n}$.
$O(\cdot)$	Big O.
$O_{\mathbb{P}}(\cdot)$	Probabilistic big O.
$\mathbb{1}_{\mathcal{A}}$	Indicator function of event $\mathcal{A}$, *i.e.*, $\mathbb{1}_{\mathcal{A}} = 1$ if $\mathcal{A}$ is true, 0 otherwise.
$\lVert \cdot \rVert_{TV}$	Total variation norm.

Linear algebra

$[\mathbf{X}]_{ij}$	Coefficient at row i and column j of matrix $\mathbf{X}$.
$\lVert \cdot \rVert$	ℓ_2-norm.
$\lVert \cdot \rVert_p$	ℓ_p-norm.
$\lvert\lvert\lvert \cdot \rvert\rvert\rvert$	Spectral norm.
$\mathbf{X}^\top$	Transpose of matrix $\mathbf{X}$, *i.e.*, $[\mathbf{M}^\top]_{ij} = [\mathbf{M}]_{ji}$.
$\mathbf{I}_n$	Identity matrix of size n.
$\mathbf{M}(t:s)$	Product $\mathbf{M}(t) \dots \mathbf{M}(s+1)$ for $0 < s < t$ and a sequence $(\mathbf{M}(r))_{r>0}$, with convention $\mathbf{M}(t:t) = \mathbf{I}_n$.
$\otimes$	Kronecker product.

Sets

$\lvert\mathcal{A}\rvert$	Cardinal of finite set $\mathcal{A}$.
Δ_n	Simplex in $\mathbb{R}^n$, *i.e.*, $\Delta_n = \{\xi \in \mathbb{R}^n_+, \lVert\xi\rVert_1 = 1\}$.
$[k]$	Set of integers from 1 to k, *i.e.*, $[k] = \{1, \dots, k\}$.

Networks

$\mathcal{G} = ([n], \mathcal{E})$	Undirected graph with n nodes and set of edges $\mathcal{E} \subseteq [n] \times [n]$.
$\mathbf{A}^{\mathcal{G}}$	Adjacency matrix of graph $\mathcal{G}$, *i.e.*, $[\mathbf{A}^{\mathcal{G}}]_{ij} = 1$ if $(i,j) \in \mathcal{E}$, 0 otherwise [2].

1. The probability measure will be ommited when clear from context.
2. The $\mathcal{G}$ exponent will be ommited when clear from context.

$\mathbf{L}^{\mathcal{G}}$	Laplacian matrix of graph $\mathcal{G}$, *i.e.*, $\mathbf{L}^{\mathcal{G}} = \mathbf{D}^{\mathcal{G}} - \mathbf{A}^{\mathcal{G}}$ [3].
$\mathbf{D}^{\mathcal{G}}$	Degree matrix of graph $\mathcal{G}$, *i.e.*, $\mathbf{D}^{\mathcal{G}}$ is diagonal and $[\mathbf{D}^{\mathcal{G}}]_{ii} = \lvert\{j \in [n], (i,j) \in \mathcal{E}\}\rvert$ [3].

Miscellaneous

L_f	Lipschitz constant of function f [4].
∇f	Gradient of function f.
$[\cdot]_+$	Hinge loss, *i.e.*, $[x]_+ = \max(0, 1-x)$.

3. The $\mathcal{G}$ exponent will be ommited when clear from context.
4. The f index will be ommited when clear from context.

Chapter 1

Résumé en français

1.1 Échantillonnage de U-statistiques

En classification et en régression, les estimateurs du risque empirique sont des moyennes statistiques sur un échantillon, c'est-à-dire de la forme

$$R_n(\boldsymbol{\theta}; X_1, \dots, X_n) = \frac{1}{n} \sum_{i=1}^{n} s(\boldsymbol{\theta}; X_i), \tag{1.1}$$

où $(X_i)_{1 \leq i \leq n}$ est l'échantillon d'observations. La théorie de minimisation du risque empirique (ou ERM pour *Empirical Risk Minimization*) fût à l'origine développée dans ce contexte et est une fondation de nombreuses méthodes d'apprentissage automatique, incluant l'optimisation stochastique et distribuée. Dans ce travail, nous portons notre attention sur des risques empiriques impliquant des U-statistiques. Pour $d > 0$, une U-statistique d'ordre d est une statistique impliquant des d-uplets de l'échantillon d'observations, c'est-à-dire

$$U_n(H_{\boldsymbol{\theta}}) = \frac{1}{\binom{n}{d}} \sum_{1 \leq i_1 < \dots < i_d \leq n} H_{\boldsymbol{\theta}}(X_{i_1}, \dots, X_{i_d}),$$

où $H_{\boldsymbol{\theta}}$ est une fonction symétrique mesurable de d éléments. Quand $d = 1$, nous retrouvons la moyenne décrite précédemment. Pour $d = 2$, la U-statistique est une moyenne sur toutes les paires possibles. Cette formulation est utilisée dans un vaste panel de problèmes d'apprentissage automatique ; par exemple, en apprentissage de métrique, pour un échantillon $(X_i)_{1 \leq i \leq n} \in (\mathbb{R}^p)^n$ donné et pour les étiquettes associées $(Y_i)_{1 \leq i \leq n} \in \{-1, +1\}^n$, le but est de trouver la distance minimisant le risque suivant :

$$R_n(\boldsymbol{\theta}; X_1, \dots, X_n) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} [Y_i Y_j (b - (X_i - X_j)^\top \boldsymbol{\theta} (X_i - X_j))]_+,$$

qui est évidemmment une U-statistique de degré 2. D'autres exemples incluent le *clustering*, l'ordonnancement ou l'apprentissage sur des graphes.

1.1.1 Les U-statistiques incomplètes

La plupart des problèmes d'apprentissage statistique peuvent être formulé de manière équivalente comme la recherche d'un certain paramètre $\boldsymbol{\theta}^*$ dans une classe Θ minimisant un risque $R(\boldsymbol{\theta}) = \mathbb{E}_\nu[s(\boldsymbol{\theta}; X)]$, pour une certaine distribution ν. Cette distribution est souvent inconnue en pratique et le paradigme de l'ERM suggère de remplacer le risque par sa contrepartie empirique, comme définie dans (2.1). La théorie de l'ERM repose essentiellement sur l'étude des déviations maximales entre ces moyennes empiriques et leurs espérances, sous des hypothèses adéquates de complexité sur l'essemble des paramètres candidats.

Lorsque le risque empirique est formulé comme une U-statistique, il est possible de montrer que dans le cadre asymptotique usuel, le taux d'apprentissage est de l'ordre de $O_{\mathbb{P}}(\sqrt{\log n / n})$. Cependant, bien que ces statistiques possèdent d'intéressantes propriétés comme une variance réduite, elles requièrent un budget de calcul souvent prohibitif pour être optimisées ou même estimées : le nombre de termes à moyenner est de l'ordre

de $\binom{n}{d}$ pour une U-statistique de degré d. Afin de remédier à ce problème, le concept de U-statistique incomplète fût introduit dans la contribution de BLOM, 1976.

L'idée derrière les U-statistiques incomplètes est d'estimer la U-statistique complète en effectuant un échantillonnage avec remplacement sur l'ensemble des d-uplets d'indices. Pour $B > 0$, une U-statistique incomplète d'ordre d basée sur B termes est de la forme :

$$\tilde{U}_B(H_{\boldsymbol{\theta}}) = \frac{1}{B} \sum_{(I_1,\ldots,I_d)\in\mathcal{D}_B} H_{\boldsymbol{\theta}}(X_{I_1},\ldots,X_{I_d}), \tag{1.2}$$

où $\mathcal{D}_B$ est un ensemble de taille B construit en échantillonnant avec remplacement dans l'ensemble $\Lambda := \{(i_1,\ldots,i_d), 1 \leq i_1 < \ldots < i_d \leq n\}$.

Nous étudions ici la précision avec laquelle un U-processus, c'est-à-dire une collection de U-statistiques, peut être estimé par une appproche Monte-Carlo (que nous appellerons U-processus incomplet dans cette thèse) impliquant beaucoup moins de termes, pourvu que l'on ait un certain contrôle sur la complexité des noyaux en jeu.

En pratique, B devrait évidemment être bien plus petit que $\binom{n}{d}$ de manière à résoudre le problème computationnel précédent. Il est à noter que la distribution d'une U-statistique complète construite à partir d'un sous-échantillon de taille réduite n' et tiré uniformément est différente de celle d'une U-statistique incomplète basée sur $B = \binom{n'}{d}$, bien qu'elles impliquent toutes les deux le même nombre de termes à moyenner.

En tant qu'estimateur du risque R, la statistique (2.2) est également non biaisée, autrement dit $\mathbb{E}[\widetilde{U}_B(H_{\boldsymbol{\theta}})] = R(\boldsymbol{\theta})$. Cependant, sa variance est naturellement plus grande que celle de la U-statistique complète $U_n(H_\theta)$. Plus spécifiquement, sa variance peut être écrite

$$\mathrm{Var}(\widetilde{U}_B(H_{\boldsymbol{\theta}})) = \left(1 - \frac{1}{B}\right)\mathrm{Var}(U_n(H_{\boldsymbol{\theta}})) + \frac{1}{B}\mathrm{Var}(H_{\boldsymbol{\theta}}(X_1,\ \ldots,\ X_d)). \tag{1.3}$$

Ainsi, la différence de variance disparaît à une vitesse en $1/B$ et une question naturelle est de savoir si cette variance ne détériore pas excessivement les vitesses d'apprentissage. Nous proposons le résultat suivant, basé sur la VC dimension de $\mathcal{H}$.

Theorem 1. (MAXIMAL DEVIATION) *Soit $\mathcal{H} := \{H_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta\}$ une collection de noyaux symétriques bornés tels que*

$$\mathcal{M}_{\mathcal{H}} := \sup_{(H_{\boldsymbol{\theta}},x)\in\mathcal{H}\times\mathcal{X}} |H_{\boldsymbol{\theta}}(x)| < +\infty. \tag{1.4}$$

On suppose également que $\mathcal{H}$ est une classe de fonctions de VC *dimension finie $V < +\infty$. Alors, les propositions suivantes sont vérifiées :*

(i) Pour tout $\delta \in (0,1)$, avec probabilité au moins $1-\delta$, on a : pour tout $B \geq 1$ et pour tout $n \in \mathbb{N}^*$,

$$\sup_{H_{\boldsymbol{\theta}}\in\mathcal{H}} \left|\widetilde{U}_B(H_{\boldsymbol{\theta}}) - U_n(H_{\boldsymbol{\theta}})\right| \leq \mathcal{M}_{\mathcal{H}} \times \sqrt{2\frac{V\log(1+|\Lambda|) + \log(2/\delta)}{B}}$$

(ii) Pour tout $\delta \in (0,1)$, avec probabilité au moins $1-\delta$, on a : $\forall n \in \mathbb{N}^*$, $\forall B \geq 1$,

$$\frac{1}{\mathcal{M}_{\mathcal{H}}} \sup_{H_{\boldsymbol{\theta}} \in \mathcal{H}} \left| \widetilde{U}_B(H_{\boldsymbol{\theta}}) - R(\boldsymbol{\theta}) \right| \leq 2\sqrt{\frac{2V \log(1+N)}{N}} + \sqrt{\frac{\log(2/\delta)}{N}} + \sqrt{2\frac{V \log(1+|\Lambda|) + \log(4/\delta)}{B}},$$

where $N = \lfloor n/d \rfloor$.

La première proposition du Théorème 6 permet de contrôler les écarts entre la U-statistique et sa contrepartie incomplète, de manière uniforme sur la classe $\mathcal{H}$. Quand le nombre de termes B augmente, cet écart diminue à une vitesse $O(1/\sqrt{B})$. La deuxième proposition du Théorème 6 donne une déviation maximale en fonction de $R(\boldsymbol{\theta})$. Il est à noter en particulier que, dans le contexte asymptotique précédemment spécifié, $\log(|\Lambda|) = O(\log n)$ lorsque $n \to +\infty$. De plus, il est possible d'obtenir une borne sur l'excès de risque des noyaux minimisant la version incomplète du risque empirique basé sur B termes et montrer que lorsqu'un U-statistique incomplète contient seulement $B = O(n)$ termes, la vitesse d'apprentissage du minimiseur correspondant est du même ordre que celle du minimiseur du risque complet, dont le calcul nécessite le moyennage de $O(n^d)$ termes. En comparaison, la minimisation d'une U-statistique complète impliquant $O(n)$ termes, obtenue en sous-échantillonant $n' = O(n^{1/d})$ observations de manière uniforme, mène a une vitesse d'apprentissage en $O(\sqrt{\log(n)/n^{1/d}})$, ce qui est bien plus lent.

Ces résultats montrent qu'il est préférable, en termes de vitesse d'apprentissage, d'estimer le risque avec la version incomplète lorsque l'opportunité se présente.

1.1.2 Expériences numériques

Nous avons effectué des expériences numériques sur le problème de l'apprentissage de métrique (voir Section 3.2.2). Comme dans une grande partie de la littérature sur l'apprentissage de métrique, nous avons restreint notre attention à la famille de pseudo-distances $D_{\boldsymbol{\theta}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ définie par

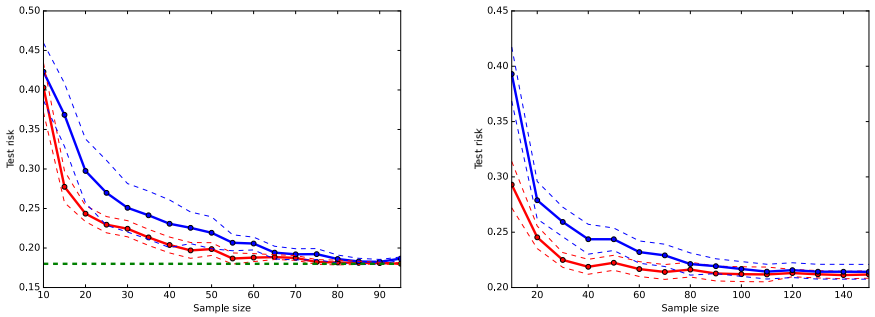
$$D_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top \boldsymbol{\theta} (\mathbf{x} - \mathbf{x}'),$$

où $\boldsymbol{\theta} \in \mathbb{S}_+^d$, et $\mathbb{S}_+^d$ est le cône des matrices $d \times d$ symétriques semi-définies positives.

Pour un échantillon d'observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ où $\mathbf{x}_i \in \mathbb{R}^d$ et $y_i \in \{1, \ldots, C\}$, soit $y_{ij} = 1$ si $y_i = y_j$ et 0 sinon pour toute paire d'observations. Pour un seuil donné $b \geq 0$, nous définissons le risque empirique comme suit :

$$R_n(\boldsymbol{\theta}) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \left[y_{ij} (b - D_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)) \right]_+, \tag{1.5}$$

où $[u]_+ = \max(0, 1-u)$ est la fonction de perte dite *hinge loss*. Notre but est de trouver le minimiseur du risque empirique parmi notre famille de distances. Dans nos expériences, nous avons utilisé deux jeux de données : un jeu de données synthétiques générées selon un mélange de 10 gaussiennes dans $\mathbb{R}^{40}$ et le jeu de données MNIST — voir Section 3.5 pour des détails sur



(A) Données synthétiques. (B) MNIST.

FIGURE 1.1: Risque de test pour l'estimateur complet (bleu) et incomplet (rouge).

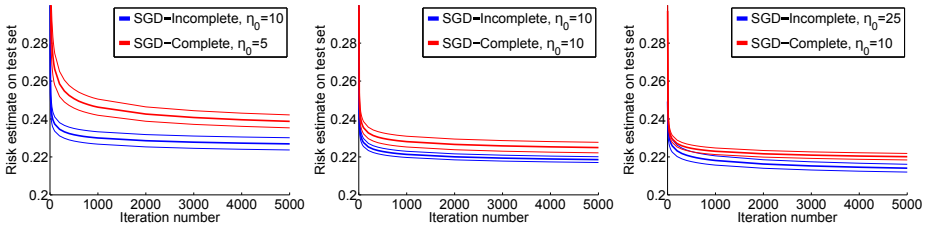


FIGURE 1.2: Descente de gradient stochastique sur MNIST pour différentes tailles de mini batch.

les jeux de données. Ces jeux de données contiennent respectivement 50000 et 60000 observations, calculer le risque empirique complet pour seulement un candidat $\boldsymbol{\theta}$ demanderait donc de moyenner 10^9 paires. Nous avons effectué deux types d'expériences. Tout d'abord, nous sous-échantillonons les données avant d'apprendre et nous évaluons la performance du minimiseur sur le sous-échantillon. Ensuite, nous utilisons la descente de gradient stochastique pour trouver le minimiseur du risque empirique sur l'échantillon original, en utilisant des sous-échantillons à chaque itération pour estimer le gradient. Nous utilisons p indices tirés aléatoirement pour l'estimateur complet et $p(p-1)/2$ paires pour l'estimateur incomplet, de telle manière que les deux estimateurs requièrent le même nombre de termes à moyenner. Pour chaque stratégie, nous utilisons une méthode de descente de gradient projeté afin de minimiser (2.8), en utilisant plusieurs valeurs de p et en moyennant les résultats sur 50 essais aléatoires. Étant donné que le jeu de test est grand, nous évaluons le risque de test sur 100 000 paires tirées aléatoirement.

La figure 1.1a montre le risque de test du minimiseur du risque empirique en fonction de la taille de l'échantillon p pour les deux estimateurs sur le jeu de données synthétiques. Comme prédit par notre analyse théorique, la stratégie incomplète fournit des performances bien supérieures en moyenne. Par exemple, elle s'approche à 5% d'erreur du vrai minimiseur empirique pour seulement $p = 50$, alors que la stratégie complète a besoin de $p > 80$ pour atteindre ce taux d'erreur. Les mêmes conclusions peuvent être tirées du jeu de données MNIST, comme le montre la Figure **??**.

Nous nous tournons désormais vers un nouveau type de contrainte :

comme mentionné précédemment, les méthodes distribuées et décentralisées sont requises dans un nombre croissant d'applications. Dans de tels contexts, le risque empirique lui même n'est pas calculable — où à un coup prohibitif — rendant nécessaire l'adaptation des méthodes usuelles à de telles contraintes.

1.2 Les protocoles *gossip*

Les méthodes que nous présentons pour estimer et optimiser des risques empiriques basés sur des U-statistiques reposent sur les protocoles *gossip*. De tels algorithmes sont parfaitement adaptés à notre contexte car ils reposent uniquement sur des communications pair-à-pair : chaque agent échange de l'information avec un voisin à la fois. Ainsi, avant de présenter nos méthodes décentralisées, nous décrivons brièvement les bases des méthodes *gossip* et fournissons des détails supplémentaires à propos de deux notions essentielles : les modèles temporels et le laplacian d'un graphe.

1.2.1 Contexte

Les algorithmes *gossip* ont été développés pour résoudre une large gamme de problèmes d'apprentissage automatique, de l'aggrégation de données sur un réseau de capteur (HEDETNIEMI, HEDETNIEMI, and LIESTMAN, 1988; DIMAKIS, SARWATE, and WAINWRIGHT, 2008; KAR and MOURA, 2009) à l'optimisation décentralisée multi-agent (NEDIĆ, 2011; DUCHI, AGARWAL, and WAINWRIGHT, 2012; TSIANOS, LAWLOR, and RABBAT, 2015). Bien que ces méthodes soient conçues pour s'attaquer à des problèmes très variés, elles partagent généralement un socle commun de contraintes :

(i) il n'y a pas de nœud central assurant une synchronisation temporelle ou une aggrégation globale des données du réseau

(ii) les capacités de calcul et de stockage de chaque nœud est particulièrement limitée

(iii) la communication d'agent à agent est coûteuse.

Dans certaines méthodes *gossip*, seules les contraintes (ii) et (iii) sont considérées, menant à des méthodes distribuées dites synchrones ou partiellement asynchrones (KARP et al., 2000; KEMPE, DOBRA, and GEHRKE, 2003; RAM, NEDIĆ, and VEERAVALLI, 2010). D'autres méthodes satisfont les trois contraintes et produisent des algorithmes dits decentralisés et complètement asynchrones (BOYD et al., 2006; NEDIĆ, 2011; LEE and NEDIĆ, 2015).

Exemple 1 (voitures connectées) Considérons des voitures connectées circulant dans une ville. Ces voitures peuvent contenir de précieuses informations sur le trafic alentour, des données météorologiques ou sur le comportement du conducteur. Cependant, le volume important du flux de données peut rendre difficile voire impossible une centralisation continue des informations sur un serveur central, afin d'appliquer des méthodes d'apprentissage automatique. Ainsi, une possibilité est de tirer avantage du fait que les voitures sont mobiles dans la ville afin d'exiger qu'elles effectuent des calculs locaux peu

coûteux en utilisant les données contenues localement et qu'elles échangent leurs résultats avec d'autres voitures dès qu'elles sont assez proches ou ont fini leurs calculs locaux. Une modèle adapté à ce schéma de communication est un graphe géométrique. Un graphe $\mathcal{G} = ([n], \mathcal{E})$ est dit géométrique de rayon r pour une distance d si pour toute paire de nœuds $\{i, j\} \in [n]^2$, $\mathcal{E}$ contient l'arrête (i, j) si et seulement si $d(i, j) < r$. L'asynchronie globale du réseau et les incertitudes de communications sont ensuite naturellement modélisées par des protocoles *gossip* complètement asynchrones. Remarquons que dans ce cas particulier, le graphe sous-jacent utilisé pour modéliser les capacités de communication du réseau est dynamique : il faudrait considérer une famille $(\mathcal{G}(t))_{t\geq 0}$ plutôt qu'un unique graphe $\mathcal{G}$. Heureusement, les méthodes que nous considérons peuvent aisément être étendues à de tels contextes.

Exemple 2 (téléphones mobiles) Dans le contexte d'applications pour des téléphones mobiles, chaque nœud du réseau est un téléphone et une communication est effectuée dès qu'un message est envoyé. Il est possible d'utiliser les messages des utilisateurs pour effectuer de la recommandation ou de la modélisation de *topic* par exemple. Toutefois, regrouper de telles informations sur un unique serveur central peut se révéler problématique pour différentes raisons : protection de la vie privée, volume de données potentiellement énorme, *etc.* De plus, même si les besoin computationnels des calculs locaux sont raisonnables par rapport aux capacités des téléphones actuels, l'établissement répété de communications peut avoir un impact non négligeable sur la batterie, rendant particulièrement adaptée l'utilisation de protocoles *gossip* pour envoyer les résultats des calculs locaux en même temps que des messages.

L'idée générale derrière les algorithmes *gossip* est d'alterner deux étapes : des mises à jours locales (par exemple une descente de gradient) et des étapes de communication (par exemple une moyenne). Dans le cas de l'estimation, les étapes de communication consistent généralement en le moyennage des estimateurs des nœuds concernés. Par exemple, dans BOYD et al., 2006, un nœud moyenne son estimateur dès qu'il établit une communication avec un de ses voisins. Il existe également quelques exceptions : dans PELCKMANS and SUYKENS, 2009, la mise à jour locale rajoute un élément à l'estimateur de la U-statistique et l'étape de communication est un échange d'observations. En optimisation *gossip*, les étapes de communication sont principalement des moyennages et les mises à jour locales correspondent à une étape d'un algorithme — centralisé — d'optimisation (descente de gradient, *dual averaging*, *etc.*).

1.2.2 Modèle temporel

Nous avions mentionné précédemment que, selon les contraintes considérées, les algorithmes *gossip* peuvent s'appliquer dans un contexte synchrone ou asynchrone. Dans le cas synchrone, nous considérons que les nœuds ont accès à une horloge commune. De cette manière, ils peuvent tous effectuer une mise à jour locale à chaque pas de temps. Ce modèle n'est pas toujours réaliste en pratique, mais permet d'analyser plus simplement

la convergence d'un algorithme car chaque nœud contribue de manière équivalente à la convergence vers l'objectif global. C'est pourquoi, dans cette thèse, nous utilisons l'analyse synchrone pour fournir des vitesses de convergence détaillées et comme tremplin pour l'analyse complètement asynchrone.

Dans un contexte complètement asynchrone, les nœuds n'ont pas accès à une horloge commune ; chaque nœud possède sa propre horloge locale. Une manière répandue de modéliser les horloges locales est de considérer des horloges indépendantes et identiquement distribuées, rythmées par un processus de Poisson de paramètre 1. Ainsi, un modèle équivalent est constitué par une horloge globale rythmée par un processus de Poisson de paramètre n et un tirage aléatoire d'arrête à chaque itération (comme dans le cas synchrone). Cependant, à une itération données, l'étape de mise à jour locale ne fait plus intervenir que la paire de nœuds actifs. Ainsi, les nœuds doivent conserver un estimateur de l'itération en cours pour assurer une convergence. L'estimateur du nombre d'itérations que nous utilisons dans nos méthodes est défini comme suit. Soit $\mathcal{G} = ([n], \mathcal{E})$ un graphe non orienté. Pour $k \in [n]$, on note p_k la probabilité que le nœud k soit tiré à une itération de l'algorithme. Si les arrêtes sont tirées de manière uniformément aléatoire, alors $p_k = 2d_k/|\mathcal{E}|$, où d_k représente le degré du nœud k dans le graphe $\mathcal{G}$. Par simplicité, nous nous concentrons uniquement sur ce cas, mais notre analyse peut facilement être étendue à un contexte plus général. On définit $(\delta_k(t))_{t\geq 1}$ pour tout $t \geq 1$ comme suit :

$$\delta_k(t) = \begin{cases} 1 & \text{if node } k \text{ is picked at iteration } t \\ 0 & \text{otherwise} \end{cases}.$$

On remarque immédiatement que les variables $(\delta_k(t))_{t>0}$ sont indépendantes et identiquement distribuées suivant une loi de Bernouilli de paramètre p_k. Soit $(m_k(t)) \geq 0$ défini pour tout $t > 0$ par :

$$m_k(t) = \frac{1}{p_k} \sum_{s=1}^{t} \delta_k(s).$$

Puisque les $(\delta_k(t))_{t>0}$ sont des variables aléatoires de Bernouilli, $m_k(t)$ est un estimateur sans biais du temps t. Ainsi, en effectuant l'hypothèse que les nœuds connaissent leurs degrés respectifs ainsi que le nombre total d'arrête dans le réseau, les estimateurs du nombre d'itérations sont non biaisés.

Le lecteur peut se référer à BOYD et al., 2006 pour plus de détails sur les modèles temporels synchrones et asynchrones.

1.2.3 Laplacien d'un graphe

Dans des contextes distribués et décentralisés, la dépendance d'une méthode envers la topologie du réseau est étroitement liée aux valeurs propres d'une matrix appelée le laplacian du graphe CHUNG, 1997. Soit $\mathcal{G} = ([n], \mathcal{E})$ un graphe non orienté et soit $\mathbf{A}^{\mathcal{G}}$ sa matrice d'adjacence, autrement dit pour tout couple $(i, j) \in [n]^2$, $[\mathbf{A}^{\mathcal{G}}]_{ij} = 1$ si et seulement si $(i, j) \in \mathcal{E}$. Le laplacien du graphe $\mathbf{L}^{G}$ est défini comme suit :

$$\mathbf{L}^{\mathcal{G}} = \mathbf{D}^{\mathcal{G}} - \mathbf{A}^{\mathcal{G}},$$

où $\mathbf{D}^{\mathcal{G}}$ est la matrix des degrées, c'est-à-dire $\mathbf{D}^{\mathcal{G}} = \mathrm{diag}(\mathbf{A}^{\mathcal{G}}\mathbf{1}_n)$. Le laplacien d'un graphe possède plusieurs propriétés intéressantes comme la positivité et la symétrie, mais la plus remarquable d'entre elles est qu'il correspond (à une renormalisation près) à la matrice de transition d'une marche aléatoire sur le graphe $\mathcal{G}$. Sa plus petite valeur propre non nulle, aussi appelée le trou spectral, caractérise la capacité du graphe à diffuser de l'information. En effet, il est possible de mettre en relation le laplacien avec l'analyse markovienne usuelle pour montrer que si le réseau est connecté et non bipartie, alors la marche aléatoire est respectivement irréductible et apériodique, assurant sa convergence vers une distribution uniforme à une vitesse géométrique, dans la raison est lié au trou spectral.

Afin d'illustrer nos propos, considérons la méthode d'estimation *gossip* décrite dans BOYD et al., 2006. Dans ce scénario, la borne d'erreur après t itérations est de la forme λ_2^t, où $\lambda_2 := 1 - \beta_{n-1}/|\mathcal{E}|$ et β_{n-1} est le trou spectral. Un autre exemple est la version distribuée du *dual averaging* présentée dans AGARWAL, WAINWRIGHT, and DUCHI, 2010. Leur borne supérieure sur l'erreur est égale à

$$\left(c + \frac{c'}{1-\lambda_2}\right)\frac{1}{\sqrt{t}},$$

où c et c' ne dépendent que du conditionnement du problème d'optimisation considéré.

Nous nous tournons désormais vers l'estimation décentralisée des U-statistiques à la fois dans le cas synchrone et asynchrone.

1.3 Estimation décentralisée d'une U-statistique

L'estimation décentralisée possède de nombreuses applications dans les réseaux de capteurs ou pair-à-pair, ainsi que pour extraire de l'information depuis des graphes de données volumineux tels que des documents web interconnectés ou des médias sociaux en ligne. Les algorithmes opérants sur de tels réseaux doivent souvent obéir à des contraintes strictes : les nœuds formant le réseau ne peuvent se reposer sur une entité centrale pour la communication et la synchronisation, n'ont pas de connaissances détaillées sur la topologie du réseau et ont des ressources limitées (mémoire, énergie, puissance computationnelle). Les algorithmes gossip (TSITSIKLIS, 1984; SHAH, 2009; DIMAKIS et al., 2010), où chaque nœud n'échange de l'information qu'avec un de ses voisins à la fois, se sont montrés de parfaits candidats pour ce problème.

De tels algorithmes ont été significativement utilisés dans le contexte de moyennage décentralisé sur des réseaux, où l'objectif est de calculer la moyenne de n réels ($\mathcal{X} = \mathbb{R}$):

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\mathbf{x}^\top \mathbf{1}_n. \tag{1.6}$$

Un des plus anciens travaux sur ce problème général provient de TSITSIKLIS, 1984, mais des algorithmes plus efficaces ont récemment été proposés, par exemple dans KEMPE, DOBRA, and GEHRKE, 2003; BOYD et al., 2006. Nous nous penchons ici plus particulièrement sur la méthode introduite dans BOYD et al., 2006, qui constitue un algorithme gossip permettant de

calculer la moyenne empirique (2.9) dans un contexte où les nœuds sont activés de manière asynchrone et moyennent simplement leur estimateur avec celui d'un voisin tiré aléatoirement. Les probabilités de communication sont données par une matrice stochastique $\mathbf{P}$, où $[\mathbf{P}]_{ij}$ est la probabilité qu'un nœud i choisisse le voisin j pour établir une communication. Comme expliqué dans la Section 2.2.3, les estimateurs locaux convergent vers (2.9) à une vitesse géométrique, dont la raison dépend du trou spectral du réseau. De telles méthodes peuvent être étendues pour calculer d'autres fonctions comme le maximum ou le minimum, ou également des sommes de la forme $\sum_{i=1}^n f(x_i)$ pour une fonction $f : \mathcal{X} \to \mathbb{R}$ donnée (comme fait par exemple dans les travaux de MOSK-AOYAMA and SHAH, 2008). Certains travaux se sont également penchés sur le développement d'algorithmes *gossip* plus rapides dans le cas de réseaux mal connectés, en supposant sur les nœuds connaissent (partiellement) leur emplacement géographique (DIMAKIS, SARWATE, and WAINWRIGHT, 2008; LI, DAI, and ZHANG, 2010). Plus récemment, LOIZOU and RICHTÁRIK, 2016 a développé une nouvelle perspective sur l'analyse des algorithmes *gossip* en utilisant des *Randomized Block Kaczmarz* et en étudiant l'optimisation duale. Pour des éléments plus détaillés sur la littérature sur les algorithmes *gossip*, le lecteur peut se référer à SHAH, 2009; DIMAKIS et al., 2010.

Nous nous penchons ici sur un problème d'optimisation décentralisée, où la quantité d'intérêt est une U-statistique d'ordre 2. C'est-à-dire que nous souhaitons estimer une quantité de la forme suivante :

$$\hat{U}_n(h) = \frac{1}{n^2} \sum_{i,j=1}^{n} h(\mathbf{x}_i, \mathbf{x}_j), \tag{1.7}$$

où $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est une fonction symétriuqe et $(\mathbf{x}_k)_{1 \leq k \leq n} \in \mathcal{X}^n$ sont des observations dans un espace $\mathcal{X}$. Cette formulation s'écarte quelque peu de la définition usuelle des U-statistiques car les termes diagonaux sont inclus ; cela simplifie notre analyse tout en incluant également le cas $h(\mathbf{x}, \mathbf{x}) = 0$. Nous supposons que les observations sont réparties sur un réseau $\mathcal{G} = ([n], \mathcal{E})$ et que chaque nœud $i \in [n]$ du réseau contient exactement une observation $\mathbf{x}_i$.

Les algorithmes *gossip* existants ne peuvent être utilisés pour calculer efficacement (2.10) car l'objectif dépend maintenant de paires d'observations. Pour autant que nous sachions, ce problème a seulement été traité dans PELCKMANS and SUYKENS, 2009. Leur méthode, appelée U2-GOSSIP, converge à une vitesse $O(1/t)$ mais présente plusieurs inconvénients. Tout d'abord, chaque nœud doit conserver deux observations auxiliaires et deux paires de nœuds doivent échanger leurs observations à chaque itération. Pour des problèmes en grande dimension (grand d), cela mène à des charges de communication et de mémoire significatives. Ensuite, l'algorithme ne présente pas de fonctionnement asynchrone car chaque nœud doit mettre à jour son estimateur à chaque itération. En conséquence, les nœuds doivent avoir accès à une horloge globale, ce qui est souvent peu réaliste en pratique. Dans la prochaine section, nous présentons de nouveaux algorithmes, synchrones et asynchrones, offrant une vitesse de convergence améliorée ainsi que des coûts de stockage et communication réduits à chaque itération.

Algorithm 1 GoSta-sync: un algorithme gossip synchrone pour calculer une U-statistique

Require: Chaque nœud k possède une observation $\mathbf{x}_k$

Chaque nœud k initialise son observation auxiliaire $\mathbf{y}_k = \mathbf{x}_k$ et son estimateur $z_k = 0$
for $t = 1, 2, \ldots$ **do**
  for $p = 1, \ldots, n$ **do**
    Fixer $z_p \leftarrow \frac{t-1}{t} z_p + \frac{1}{t} h(\mathbf{x}_p, \mathbf{y}_p)$
  end for
  Tirer (i, j) selon une loi uniforme sur $\mathcal{E}$
  Fixer $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
  Échanger les observations auxiliaires des nœuds i et j: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
end for

1.3.1 Les algorithmes GoSta

La méthode proposée repose sur l'observation que la U-statistique à estimer peut être réécrite $\hat{U}_n(h) = (1/n)\sum_{i=1}^n \overline{h}_i$, où $\overline{h}_i = (1/n)\sum_{j=1}^n h(\mathbf{x}_i, \mathbf{x}_j)$, et nous noterons par la suite $\overline{\mathbf{h}} = (\overline{h}_1, \ldots, \overline{h}_n)^\top$. Notre objectif reformulé ainsi est désormais similaire au problème habituel de la moyenne décentralisée (2.9), une différence notable étant que chaque valeur locale $\overline{h}_i$ est elle-même une moyenne dépendant de l'intégralité de l'échantillon des observations. Par conséquence, nos méthodes vont combiner deux étapes à chaque itération : une étape de propagation des données afin que chaque nœud i puisse estimer $\overline{h}_i$, et une étape de moyennage pour s'assurer de la convergence vers la valeur souhaitée $\hat{U}_n(h)$.

Cas synchrone

Dans le cas synchrone, nous supposons que les nœuds ont accès à une horloge commune. De cette manière, ils mettent à jour leurs estimateurs à chaque pas de temps. Nous insistons sur le fait que les nœuds n'ont pas besoin de connaître la topologie détaillée du réseau car ils n'interagissent qu'avec leur voisins direct sur le graphe. On note $z_k(t)$ l'estimateur (local) de $\hat{U}_n(h)$ au nœud k et à l'itération t. Afin de propager les données au travers du réseau, chaque nœud k détient une observation auxiliaire $\mathbf{y}_k$, initialisée à $\mathbf{x}_k$. Notre algorithme, dénommé GoSta pour *gossip* U-statistique, procède comme suit. À chaque itération, chaque nœud k met à jour son estimateur local en effectuant une moyenne glissante de $z_k(t)$ et de $h(\mathbf{x}_k, \mathbf{y}_k)$. Ensuite, une arrête du réseau est tiré uniformément parmi l'ensemble des arrêtes, et la paire de nœuds correspondants moyenne ses estimateurs et échange les observations auxiliaires. Ainsi, les observations suivent des marches aléatoires (bien que couplées) sur le graphe associé au réseau. La procédure complète est décrite dans l'Algorithme 5.

Nous établissons la vitesse de convergence de cette algorithme dans le théorème suivant.

Theorem 2. *Soient $\mathcal{G} = ([n], \mathcal{E})$ un graphe connexe et non bipartie, $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ un échantillon d'observations et $(\mathbf{z}(t))$ la suite d'estimateurs*

générés par l'Algorithme 5. Pour tout $k \in [n]$, on a:

$$\lim_{t\to+\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1\leq i,j\leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n(h).$$

De plus, pour tout $t > 0$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n \right\| \leq \frac{1}{ct} \left\| \overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n \right\| + \left(\frac{2}{ct} + e^{-ct} \right) \left\| \mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top \right\|,$$

où pour tout $1 \leq i, j \leq n$, $[\mathbf{H}]_{ij} = h(\mathbf{x}_i, \mathbf{x}_j)$, $c = c(\mathcal{G}) := \beta_{n-1}/|\mathcal{E}|$, et β_{n-1} est la deuxième plus petite valeur propre du laplacien du graphe $\mathbf{L}^{\mathcal{G}}$.

Le Théorème 7 montre que les estimateurs locaux générés par l'Algorithme 5 convergent vers $\hat{U}_n(h)$ à une vitesse $O(1/t)$. De plus, les constantes révèlent la dépendance de la vitesse de convergence à l'instance du problème. En effet, les deux normes constituent des termes dépendants des données et quantifient la difficulté du problème d'estimation au travers d'une mesure de dispersion. D'autre part, $c(\mathcal{G})$ est un terme dépendant du réseau puisque β_{n-1} est la deuxième plus petite valeur propre du laplacien du graphe $\mathbf{L}^{\mathcal{G}}$, aussi appelée le trou spectral de $\mathcal{G}$. Par conséquent, nous attendons des graphes mieux connectés une convergence plus rapide ; cela sera mis en évidence dans les applications numériques.

Afin d'estimer $\hat{U}_n(h)$, U2-GOSSIP (PELCKMANS and SUYKENS, 2009) n'utilise pas de moyennage. À la place, chaque nœud k stocke deux observations auxiliaires $\mathbf{y}_k^{(1)}$ et $\mathbf{y}_k^{(2)}$ qui sont toutes les deux initialisées à $\mathbf{x}_k$. À chaque itération, chaque nœud $k \in [n]$ met à jour son estimateur local en effectuant une moyenne glissante entre z_k et $h(\mathbf{y}_k^{(1)}, \mathbf{y}_k^{(2)})$. Ensuite, deux arrêtes sont tirées aléatoirement : les nœuds associés à la première (respectivement seconde) arrête échangent leurs premières (respectivement secondes) observations auxiliaires. En appliquant notre analyse de convergence à U2-GOSSIP, nous obtenons la vitesse de convergence suivante :

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n \right\| \leq \frac{\sqrt{n}}{t} \left(\frac{2}{1-\tilde{\lambda}} \left\| \overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n \right\| + \frac{1}{1-\tilde{\lambda}^2} \left\| \mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top \right\| \right), \tag{1.8}$$

où $1 - \tilde{\lambda} = 2c(\mathcal{G})$. L'avantage de propager deux observations dans U2-GOSSIP est observable dans le terme $1/(1-\tilde{\lambda}^2)$. Cependant, l'absence de moyennage mène à un facteur supplémentaire en $\sqrt{n}$. Intuitivement, ce terme provient du fait que les nœuds ne bénéficient pas des autres estimateurs du réseau. En pratique, $\tilde{\lambda}$ est proche de 1 pour des réseaux de taille raisonnable (par exemple, $\tilde{\lambda} = 1-2/n$ pour le graphe complet), et les termes au carré n'apportent qu'un gain négligeable. Ainsi, le facteur $\sqrt{n}$ domine dans (2.11) et nous nous attendons à ce que U2-GOSSIP converge plus lentement que GOSTA, ce qui est confirmé par les résultats numériques.

Cas asynchrone

Nous retirons désormais l'hypothèse sur l'horloge commune. En utilisant les estimateurs du nombre d'itérations décrits dans la Section 2.2.2, nous pouvons désormais établir une version asynchrone de GOSTA, comme

Algorithm 2 GoSta-async: un algorithme gossip asynchrone pour calculer une U-statistique

Require: Chaque nœud k détient une observation $\mathbf{x}_k$ et $p_k = 2d_k/|\mathcal{E}|$
1: Chaque nœud k initialise $\mathbf{y}_k = \mathbf{x}_k$, $z_k = 0$ et $m_k = 0$
2: **for** $t = 1, 2, \ldots$ **do**
3: Tirer (i, j) selon une loi uniforme sur $\mathcal{E}$
4: Fixer $m_i \leftarrow m_i + 1/p_i$ et $m_j \leftarrow m_j + 1/p_j$
5: Fixer $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
6: Fixer $z_i \leftarrow (1 - \frac{1}{p_i m_i})z_i + \frac{1}{p_i m_i} h(\mathbf{x}_i, \mathbf{y}_i)$
7: Fixer $z_j \leftarrow (1 - \frac{1}{p_j m_j})z_j + \frac{1}{p_j m_j} h(\mathbf{x}_j, \mathbf{y}_j)$
8: Échanger les observations auxiliaires entre les nœuds i et j: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
9: **end for**

établi dans l'Algorithme 6. Remarquons que l'étape de mise à jour locale diffère légèrement de celle du cas synchrone : cela est dû au fait que chaque nœud doit contribuer de manière équivalente à la statistique à estimer. Ainsi, les nœuds qui sont moins souvent activés doivent mettre un plus gros poids sur leur contribution.

Afin de montrer que les estimateurs locaux convergent vers $\hat{U}_n(h)$, nous utilisons un modèle similaire au cas synchrone. La dépendance temporelle de la matrice de transition est plus complexe ; il en va de même pour la borne supérieure.

Theorem 3. *Soit $\mathcal{G} = ([n], \mathcal{E})$ un graphe connexe et non bipartie, $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ un échantillon d'observations et $(\mathbf{z}(t))$ la suite d'estimateurs générée par l'Algorithme 6. Pour tout $k \in [n]$, on a:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n(h).$$

De plus, il existe une constante $c'(\mathcal{G}) > 0$ telle que, pour tout $t > 1$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n \right\| \leq c'(\mathcal{G}) \cdot \frac{\log t}{t} \|\mathbf{H}\|.$$

1.3.2 Expériences

Nous présentons dans cette section deux applications sur des jeux de données réels. La première application concerne l'estimation de la dispersion intra-cellule (Clémençon, 2011) qui mesure la qualité d'une partition $\mathcal{P}$ de l'espace $\mathcal{X}$ à partir de la distance moyenne entre points d'une même cellule $\mathcal{C} \in \mathcal{P}$. Cette dispersion est de la forme (2.10) avec

$$h_{\mathcal{P}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| \cdot \sum_{\mathcal{C} \in \mathcal{P}} \mathbb{1}_{\{(\mathbf{x}, \mathbf{x}') \in \mathcal{C}^2\}}. \tag{1.9}$$

Nous étudions également la mesure de l'AUC (Hanley and McNeil, 1982). Pour un échantillon d'observations $(\mathbf{x}_1, \ell_1), \ldots, (\mathbf{x}_n, \ell_n)$ données sur $\mathcal{X} \times \{-1, +1\}$, l'AUC d'un classifieur linéaire $\boldsymbol{\theta} \in \mathbb{R}^{d-1}$ est donnée par :

$$\mathrm{AUC}(\boldsymbol{\theta}) = \frac{\sum_{1 \leq i,j \leq n} (1 - \ell_i \ell_j) \mathbb{1}_{\{\ell_i (\boldsymbol{\theta}^\top x_i) > -\ell_j (\boldsymbol{\theta}^\top x_j)\}}}{4 \left(\sum_{1 \leq i \leq n} \mathbb{1}_{\{\ell_i = 1\}} \right) \left(\sum_{1 \leq i \leq n} \mathbb{1}_{\{\ell_i = -1\}} \right)}. \tag{1.10}$$

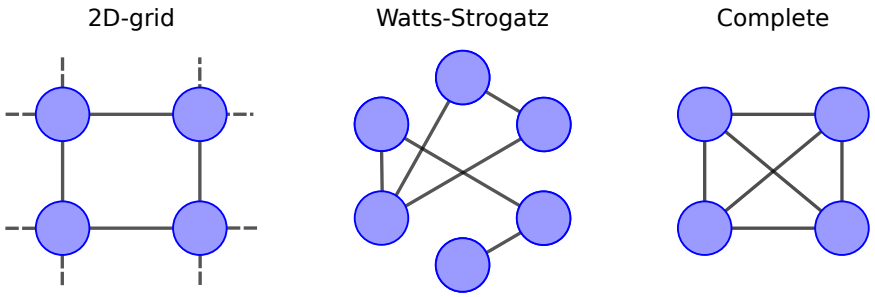


FIGURE 1.3: Exemples de réseaux.

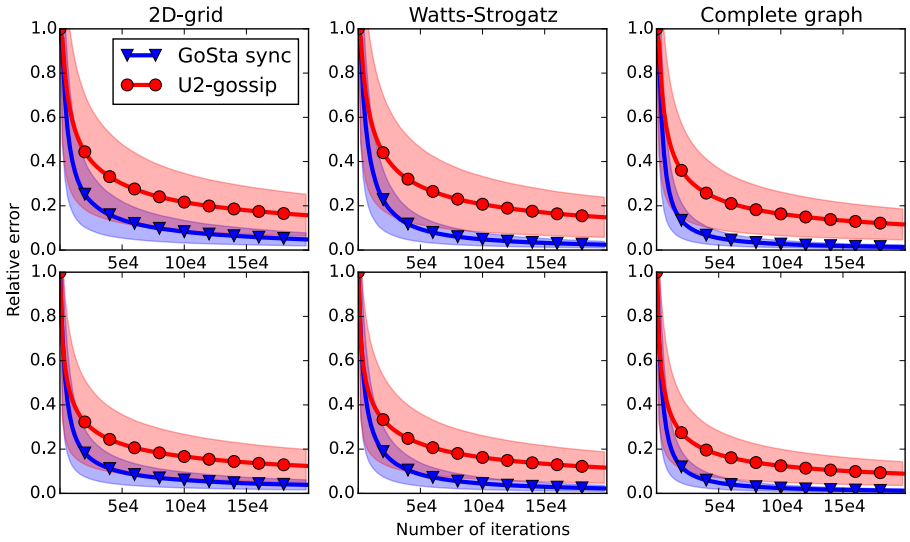


FIGURE 1.4: Évolution de l'erreur relative moyenne (ligne continue) et de son écart-type (surface pleine) en fonction des itérations pour U2-GOSSIP (rouge) et GOSTA-SYNC (bleu) sur le jeu de données SVMguide3 (ligne supérieure) et le jeu de données Wine Quality (ligne inférieure).

Ce score correspond à la probabilité d'un classifieur d'ordonner une paire d'observation dans le bon ordre.

Nous effectuons nos simulations sur trois types de réseaux, décrits ci-dessous.

• *Graphe complet :* Cela correspond au cas où tous les nœuds sont connectés les uns aux autres. Cette situation est idéale dans notre cadre d'application, puisque toutes les paires de nœuds peuvent communiquer directement.

• *Grille bidimensionnelle :* Ici, les nœuds sont placés sur une grille en deux dimensions et chaque nœud est connecté à ses quatre voisins sur la grille. Ce réseau offre une structure de graphe régulière ainsi qu'une communication isotrope. Cependant, son diamètre ($\sqrt{n}$) est plutôt élevé, en particulier par rapport aux réseaux invariants d'échelle.

• *Watts-Strogatz :* Cette technique de génération aléatoire de graphe est présentée dans WATTS and STROGATZ, 1998 et permet de créer des réseaux avec des propriétés de communication variables. Dans notre cas, nous ajustons les paramètres afin d'obtenir un compromis (en termes de connectivité) entre le graphe complete et la grille bidimensionnelle.

La Figure 2.3 présente des exemples de tels réseaux.

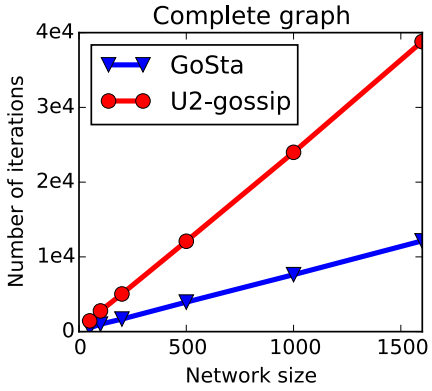


FIGURE 1.5: Temps pour atteindre 20% d'erreur.

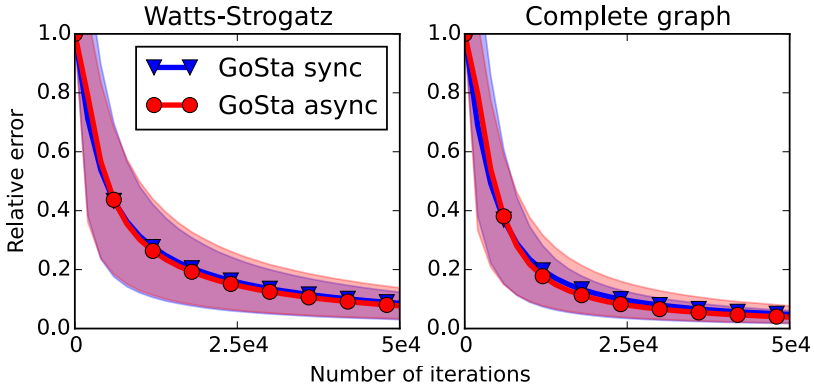


FIGURE 1.6: Erreur relative (ligne continue) et écart-type associé (zone pleine) des versions synchrone (bleu) et asynchrone (rouge) de GOSTA.

Mesure de l'AUC Nous utilisons le jeu de données de classification binaire SMVGUIDE3, contenant $n = 1260$ points en $d = 23$ dimensions [1] et nous fixons θ à la différence entre les moyennes des classes. La ligne supérieure de la Figure 2.4 montre l'évolution de l'erreur relative et de l'écart-type (entre les nœuds) associé en fonction des itérations, pour les deux méthodes sur chaque type de réseau. En moyenne, GOSTA-SYNC offre de meilleures performances que U2-GOSSIP sur chaque réseau. La variance des estimateurs entre les nœuds est également plus faible grâce au moyennage. Il est également intéressant de noter que l'écart de performance entre les deux algorithmes se creuse énormément au départ, ce qui peut possiblement être expliqué par le fait que le terme exponentiel dans la borne de convergence de GOSTA-SYNC n'est significatif que lors des premières itérations.

Inertie intra-cellule Nous utilisons le jeu de données Wine Quality, contenant $n = 1599$ points en $d = 12$ dimensions, avec un total de $K = 11$ classes. [2] Nous nous concentrons sur la partition $\mathcal{P}$ dont les cellules sont les classes associées aux centroïdes. Les résultats sont représentés sur la

1. Ce jeu de données est accessible à l'adresse suivante http://mldata.org/repository/data/viewslug/svmguide3/

2. Ce jeu de données est disponible à l'adresse suivante https://archive.ics.uci.edu/ml/datasets/Wine

ligne inférieure de la Figure 2.4. Comme dans le cas de l'AUC, GoSta-sync offre de meilleures performances sur tous les types de réseaux, que ce soit en termes d'erreur ou en termes de variance. La Figure 2.5, montre le temps moyen nécessaire pour atteindre une erreur relative de 0.2 sur un graph complet, avec un nombre de nœuds variant de $n = 50$ à $n = 1599$. Comme prédit par notre analyse, l'écart de performance se creuse en faveur de GoSta lorsque la taille du graphe augmente. Finalement, nous comparons les performances de GoSta-sync et GoSta-async (Algorithme 6) sur la Figure 2.6. Malgré la vitesse de convergence théorique légèrement détériorée pour GoSta-async, les deux algorithmes montrent des performances similaires en pratique.

Nous nous tournons désormais vers le cas où les objectifs mentionnés précédemment doivent être minimisés, toujours dans un contexte décentralisé.

1.4 Optimisation décentralisée pour des fonctions de paires

L'optimisation décentralisée est particulièrement bien adaptée pour s'attaquer aux défis posés par l'avancée du Big Data et de l'Internet des objets. Par exemple, dans l'apprentissage automatique à grande échelle, l'objectif est de minimiser une fonction de perte sur un immense jeu de données distribué sur plusieurs machines dans une ferme de calculs ou une plateforme de *cloud-computing*. D'autres applications proviennent des réseaux avec ou sans fils, où les agents locaux doivent se coordonner afin de minimiser une fonction objectif commune. Les stratégies usuelles pour résoudre de tels problèmes d'optimisation reposent sur les algorithmes *gossip*, comme dans le cas de l'estimation. Ces algorithmes ont retenu beaucoup d'attention grâce à leur simplicité ainsi que leur abilité à opérer sur des réseaux pair-à-pair où une coordination centralisée peut se révéler trop coûteuse ou même tout simplement impossible.

Un des problèmes centraux de l'optimisation décentralisée est la recherche d'un vecteur de paramètres $\boldsymbol{\theta}$ minimisant un risque empirique prenant la forme d'une moyenne de fonctions convexes $(1/n)\sum_{i=1}^{n} f(\boldsymbol{\theta}; \mathbf{x}_i)$, où chaque donnée $\mathbf{x}_i$ n'est connu que de l'agent i. Un large panel d'algorithmes *gossip* ont été proposé pour résoudre ce problème : certains sont basés sur la descente de gradient (Nedić and Ozdaglar, 2009; Johansson, Rabi, and Johansson, 2010; Ram, Nedić, and Veeravalli, 2010; Bianchi and Jakubowicz, 2013), d'autres sur la méthode ADMM (Wei and Ozdaglar, 2012; Wei and Ozdaglar, 2013; Iutzeler et al., 2013) ou encore le *dual averaging* (Duchi, Agarwal, and Wainwright, 2012; Yuan et al., 2012; Lee, Nedić, and Raginsky, 2015; Tsianos, Lawlor, and Rabbat, 2015). Dans ces méthodes, chaque agent cherche à minimiser son objectif partiel en effectuant des mises à jours locales (par exemple des descentes de gradient) tout en échangeant de l'information avec ses voisins pour s'assurer d'une convergence vers un consensus.

Dans ce travail, nous nous penchons sur le problème plus complexe de la minimisation d'une moyenne de fonctions de *paires* des observations :

$$\min_{\boldsymbol{\theta}} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j). \tag{1.11}$$

Pour ce faire, nous utilisons une approche similaire au cas de l'estimation, en combinant des calculs locaux avec une propagation des données sur le réseau. Des observations auxiliaires nous permettrons de calculer des estimateurs — biaisés — des gradients.

1.4.1 Définition du problème

Soit $\mathcal{X}$ un espace de paramètres, $d > 0$ et soit $f : \mathbb{R}^d \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ une fonction convexe et différentiable par rapport à sa première variable. Nous supposons que pour toute paire de paramètres $(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2$, il existe une constante $L_f > 0$ telle que $f(\cdot\,; \mathbf{x}, \mathbf{x}')$ est L_f-Lipschitz (par rapport à la norme euclidienne $\|\cdot\|$). Soit $\psi : \mathbb{R}^d \to \mathbb{R}^+$ une fonction convexe (potentiellement non régulière) telle que, par soucis de simplicité, $\psi(0) = 0$. Pour un échantillon d'observations donné $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{X}^n$, notre objectif est de résoudre le problème d'optimisation suivant :

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) + \psi(\boldsymbol{\theta}). \tag{1.12}$$

Dans une application d'apprentissage automatique, le Problème (2.15) est typiquement un problème de minimisation de risque empirique (pénalisé) et $\boldsymbol{\theta}$ correspond aux paramètres du modèle à apprendre. Dans un tel contexte, la fonction $f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j)$ est une fonction de perte par paire, mesurant la performance du modèle $\boldsymbol{\theta}$ sur la paire d'observations $(\mathbf{x}_i, \mathbf{x}_j)$, tandis que $\psi(\boldsymbol{\theta})$ représente un terme de régularisation, pénalisant la complexité du modèle $\boldsymbol{\theta}$. Ce terme de régularisation peut prendre une variété de formes : fonction indicatrice sur un ensemble convexe fermé (contraintes d'optimisation), norme $\|\cdot\|_1$ afin d'assurer la parcimonie du modèle, *etc.*

De nombreux problèmes d'intérêt peuvent être formulé comme le Problème (2.15). Par exemple, il est possible d'effectuer la maximisation mentionnée précédemment de l'AUC en utilisant une perte logistique (par soucis de régularité)

$$f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) = \mathbb{1}_{\{\ell_i > \ell_j\}} \log\left(1 + \exp((\mathbf{x}_j - \mathbf{x}_i)^\top \boldsymbol{\theta})\right),$$

et un terme de régularisation $\psi(\boldsymbol{\theta})$ peut être fixé à la norme ℓ_2 de $\boldsymbol{\theta}$ (ou la norme ℓ_1 si un modèle parcimonieux est préférable). D'autres exemples courants d'application du Problème (2.15) peuvent être trouvés dans l'apprentissage de métrique ou similarité (Bellet, Habrard, and Sebban, 2015), l'ordonnancement (Clémençon, Lugosi, and Vayatis, 2008), l'inférence supervisée de graphe (Biau and Bleakley, 2006) et l'apprentissage de noyaux multiples (Kumar et al., 2012).

Par soucis de clarté dans les notations, nous noterons f_i la fonction partielle associée à i, ainsi $f_i := (1/n) \sum_{j=1}^n f(\cdot; \mathbf{x}_i, \mathbf{x}_j)$ et $f = (1/n) \sum_{i=1}^n f_i$. Le

Problème (5.3) peut alors être reformulé comme suit :

$$\min_{\theta \in \mathbb{R}^d} R_n(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \psi(\boldsymbol{\theta}). \tag{1.13}$$

Remarquons que la fonction f est L_f-Lipschitz, puisque toutes les fonctions partielles f_i le sont également.

1.4.2 *Dual averaging gossip* pour les fonctions de paires

Dans nos méthodes, nous utilisons l'algorithme du *dual averaging* pour les mises à jour locales. L'algorithme du *dual averaging* (NESTEROV, 2009) construit une suite $(\boldsymbol{\theta}(t))_{t>0}$ dans l'espace dit *primal* $\mathcal{X}$ ainsi qu'une suite $(\mathbf{z}(t))_{t\geq 0}$ de variables *duales* qui collectent les sommes des gradients aperçus jusqu'à l'itération t. À chaque pas de temps t, la variable duale $\mathbf{z}$ est mise à jour comme suit :

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \nabla f(\boldsymbol{\theta}(t)).$$

La variable primale est elle générée à partir de la règle suivante :

$$\boldsymbol{\theta}(t+1) = \pi_t(\mathbf{z}(t+1)) := \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ -\mathbf{z}^\top \boldsymbol{\theta} + \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t)} \right\},$$

pour une suite de pas $(\gamma(t))_{t\geq 0}$ donnée. Ce choix est guidé par le fait que la structure des mises à jour rend l'analyse du cas distribué bien plus aisée que pour la descente de gradient stochastique, dès que le problème est contraint ou régularisé. Cela est dû au fait que l'algorithme du *dual averaging* maintient une simple somme de gradients, tandis que l'opérateur de projection (non linéaire) est appliqué séparemment — voir Section 5.3 pour des détails au sujet du *dual averaging*.

Notre travail est fondé sur l'analyse présentée dans DUCHI, AGARWAL, and WAINWRIGHT, 2012, où une version distribuée du *dual averaging* est proposée. Leur objectif est d'optimiser une moyenne de fonctions univariées $f(\cdot\,; \mathbf{x}_i)$, dans laquelle chaque nœud calcule un estimateur non biaisé de $\nabla f(\cdot\,; \mathbf{x}_i)$ qui sera ensuite moyenné sur le réseau. Cependant, dans notre contexte, un nœud ne peut fournir un estimateur sans biais de $\nabla f(\cdot\,; \mathbf{x}_i, \mathbf{x}_j)$, même en utilisant une propagation des observations similaires à GOSTA ; nous utiliserons plutôt les observations auxiliaires pour fournir des estimateurs *biaisés* des gradients. En nous basant sur le fait que la contribution de ce biais décroît exponentiellement vite en fonction du nombre d'itérations, nous montrons que la convergence du *dual averaging* est préservée. Nous présentons et analysons tout d'abord notre algorithme dans le cas synchrone et nous tournons par la suite vers le cas plus pointilleux de l'analyse asynchrone.

Cas synchrone

Dans le cas synchrone, chaque nœud a accès à une horloge globale. Ainsi, chaque nœud effectue une mise à jour locale (une étape du *dual averaging*) à chaque itération. L'étape de communication combine un moyennage des variables duales des nœuds sélectionnés et un échange d'observations auxiliaires similaire à GOSTA. La procédure est détaillée

Algorithm 3 *Dual averaging gossip* pour des fonctions de paires dans le cas synchrone.

Require: Pas $(\gamma(t))_{t\geq 1} > 0$.
Chaque nœud i initialise $\mathbf{y}_i = \mathbf{x}_i$, $\mathbf{z}_i = \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}}_i = 0$.
for $t = 1, \dots, T$ **do**
  Tirer (i, j) suivant une loi uniforme sur $\mathcal{E}$
  Fixer $\mathbf{z}_i, \mathbf{z}_j \leftarrow \frac{\mathbf{z}_i + \mathbf{z}_j}{2}$
  Échanger les observations auxiliaires : $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
  for $k = 1, \dots, n$ **do**
    Mettre à jour $\mathbf{z}_k \leftarrow \mathbf{z}_k + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k; \mathbf{x}_k, \mathbf{y}_k)$
    Calculer $\boldsymbol{\theta}_k \leftarrow \pi_t(\mathbf{z}_k)$
    Moyenner $\bar{\boldsymbol{\theta}}_k \leftarrow \left(1 - \frac{1}{t}\right) \bar{\boldsymbol{\theta}}_k + \frac{1}{t} \boldsymbol{\theta}_k$
  end for
end for
return Chaque nœud k possède $\bar{\boldsymbol{\theta}}_k$, pour $k = 1, \dots, n$

dans l'Algorithme 7, et le théorème suivant établit une borne de sa vitesse de convergence.

Theorem 4. *Soit $\mathcal{G} = ([n], \mathcal{E})$ un graphe connexe et non bipartie et soit $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} R_n(\boldsymbol{\theta})$. Soit $(\gamma(t))_{t\geq 1}$ une suite positive décroissante. Pour tout $i \in [n]$ et tout $t \geq 0$, soit $\mathbf{z}_i(t) \in \mathbb{R}^d$ et $\bar{\boldsymbol{\theta}}_i(t) \in \mathbb{R}^d$ générés selon l'Algorithme 7. Alors, pour tout $i \in [n]$ et $T > 1$, on a :*

$$\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i) - R_n(\boldsymbol{\theta}^*)] \leq C_1(T) + C_2(T) + C_3(T),$$

où

$$\begin{cases} C_1(T) = \dfrac{1}{2T\gamma(T)} \|\boldsymbol{\theta}^*\|^2 + \dfrac{L_f^2}{2T} \displaystyle\sum_{t=1}^{T-1} \gamma(t), \\ C_2(T) = \dfrac{3L_f^2}{T\left(1 - \sqrt{\lambda_2}\right)} \displaystyle\sum_{t=1}^{T-1} \gamma(t), \\ C_3(T) = \dfrac{1}{T} \displaystyle\sum_{t=1}^{T-1} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)], \end{cases}$$

où $1 - \lambda_2 = \beta_{n-1}/|\mathcal{E}| > 0$ et β_{n-1} est la seconde plus petite valeur propre du laplacien $\mathbf{L}^{\mathcal{G}}$ du graphe $\mathcal{G}$.

La vitesse de convergence est divisée en trois parties : $C_1(T)$ est un terme dépendant des données, qui correspond à la vitesse de convergence de la version centralisée du *dual averaging*, tandis que $C_2(T)$ est un terme dépendant du réseau, lié au trou spectral β_{n-1} du graphe $\mathcal{G}$. $C_3(T)$ dépend du biais des estimateurs des gradients $\bar{\boldsymbol{\epsilon}}$ qui devrait vraisemblablement décroître rapidement : le schéma de propagation est une marche aléatoire, ainsi la distribution des observations tend vers une uniforme à une vitesse exponentielle. Comme dans le cas de l'estimation, le trou spectral du réseau $\mathcal{G}$ est essentiel pour établir une borne sur l'erreur de notre méthode.

La borne supérieure énoncée dans le Théorème 9 n'assure pas la convergence de notre algorithme : le terme de biais, bien que borné, ne présente, *sous cette forme*, aucune garantie de convergence vers 0. En étendant l'analyse ergodique de la *mirror descent* de DUCHI et al., 2012, nous étudions dans cette thèse le cas du *dual averaging* avec des gradients biaisés

Algorithm 4 *Dual averaging gossip* pour des fonctions de paires dans le cas asynchrone.

Require: Pas $(\gamma(t))_{t\geq 0} > 0$, probabilités $(p_k)_{k\in[n]}$.
Chaque nœud i initialise $\mathbf{y}_i = \mathbf{x}_i$, $\mathbf{z}_i = \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}}_i = 0$, $m_i = 0$.
for $t = 1, \dots, T$ **do**
  Tirer (i, j) suivant une loi uniforme sur E
  Échanger les observations auxiliaires : $y_i \leftrightarrow y_j$
  for $k \in \{i, j\}$ **do**
    Fixer $\mathbf{z}_k \leftarrow \frac{\mathbf{z}_i + \mathbf{z}_j}{2}$
    Mettre à jour $\mathbf{z}_k \leftarrow \frac{1}{p_k} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k; \mathbf{x}_k, \mathbf{y}_k)$
    Incrémenter $m_k \leftarrow m_k + \frac{1}{p_k}$
    Calculer $\boldsymbol{\theta}_k \leftarrow \pi_{m_k}(\mathbf{z}_k)$
    Moyenner $\bar{\boldsymbol{\theta}}_k \leftarrow \left(1 - \frac{1}{m_k p_k}\right) \bar{\boldsymbol{\theta}}_k$
  end for
end for
return Chaque nœud k possède $\bar{\boldsymbol{\theta}}_k$

afin de prouver la convergence de notre méthode, avec peu d'impact sur la vitesse en comparaison avec le cas non biaisé.

Cas asynchrone

En utilisant l'estimateur du nombre d'itérations défini dans la Section **??**, nous pouvons maintenant adapter l'Algorithme 7. Comme pour le problème de l'estimation, les mises à jours doivent être pondérées en fonction des probabilités de réveil des nœuds. Le résultat suivant est analogue au Théorème 9 dans le cas asynchrone.

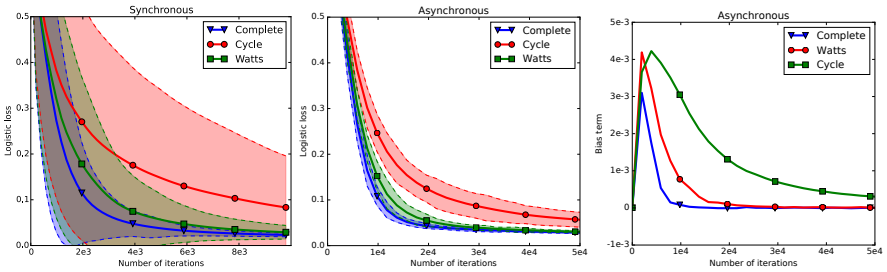
Theorem 5. *Soit $\mathcal{G} = ([n], \mathcal{E})$ un graphe connexe et non bipartite. Soit $(\gamma(t))_{t\geq 1}$ une suite définie par $\gamma(t) = c/t^{1/2+\alpha}$ pour des constantes $c > 0$ et $\alpha \in (0, 1/2)$. Pour $i \in [n]$, soient $(\mathbf{d}_i(t))_{t\geq 1}$, $(\mathbf{g}_i(t))_{t\geq 1}$, $(\boldsymbol{\epsilon}_i(t))_{t\geq 1}$, $(\mathbf{z}_i(t))_{t\geq 1}$ et $(\boldsymbol{\theta}_i(t))_{t\geq 1}$ des suites générées tel qu'indiqué dans l'Algorithme 8. Alors, il existe une constante $C < +\infty$ telle que, pour $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}' \in \mathbb{R}^d} R_n(\boldsymbol{\theta}')$, $i \in [n]$ et $T > 0$,*

$$R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*) \leq C \max(T^{-\alpha/2}, T^{\alpha-1/2}) + \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)].$$

Dans le cas asynchrone, aucune vitesse de convergence n'était connue, même pour l'algorithme du *dual averaging* distribué de DUCHI, AGARWAL, and WAINWRIGHT (2012), qui traite le problème plus simple de la minimisation de fonctions univariées. Les arguments utilisés pour établir le Théorème 10 peuvent être adaptés pour obtenir une vitesse de convergence (sans terme de biais) pour une version asynchrone de leur algorithme.

1.4.3 Expériences numériques

Pour étudier l'influence de la topologie du réseau, nous effectuons nos simulations sur trois types de réseaux différents : le graphe complet, le graphe cyclique et un graphe de Watts-Strogatz.



(A) Synchrone. (B) Asynchrone. (C) Terme de biais (asynchrone).

FIGURE 1.7: Maximisation de l'AUC.

Nous avons initialisé chaque $\boldsymbol{\theta}_i$ à 0 et pour chaque réseau, nous avons lancé 50 fois les Algorithmes 7 et 8 avec$\gamma(t) = 1/\sqrt{t}$.[3] La Figure 2.7a montre l'évolution de la fonction objectif et de l'écart-type associé (entre les nœuds) avec le nombre d'itérations dans le cas synchrone. Comme attendu, la vitesse de convergence moyenne est bien plus élevée sur le graphe complet et le graphe de Watts-Strogatz que sur le graphe cyclique. L'écart-type diminue également lorsque la connectivité du réseau augmente.

La Figure 2.7b montre les résultats pour le cas asynchrone. La vitesse de convergence est plus faible que dans le cas synchrone du point de vue du nombre d'itérations (environ 5 fois). Il est à noter cependant qu'un nombre bien plus faible d'étapes de *dual averaging* ont été effectuées : par exemple, sur le graphe de Watts-Strogatz, 210 000 calculs de gradients (partiels) sont nécessaires dans le cas synchrone pour atteindre 10% de précision, contre seulement 25 000 dans le cas asynchrone. De plus, l'écart-type des itérés est bien plus faible dans le cas asynchrone. Cela est dû à l'équilibre entre communication et optimisation (une étape d'optimisation pour chaque variable duale moyennée), là où l'optimisation prévaut largemnent dans le cas synchrone.

Les bons résultats de convergence en pratique de notre méthode vient du fait que le terme de biais $\bar{\boldsymbol{\epsilon}}(t)^\top \boldsymbol{\omega}(t)$ disparaît rapidement. La Figure 2.7c montre qu'il converge (en moyenne) vers 0 pour tous les réseaux. De plus, son ordre de grandeur est négligeable comparé à la fonction objectif.

1.5 Conclusion

Nous avons développé des garanties théoriques pour différents schémas d'échantillonnage pour des minimisations de risque empirique basées sur des U-statistiques, ainsi que de solides résultats numériques. Nous avons aussi développé des méthodes pour estimer et optimiser des fonctions de paires d'une manière *gossip*, de nouveau avec des garanties théoriques et de solides résultats numériques.

3. Même si cette suite ne rempli pas les hypothèses du Théorème 24 pour le cas asynchrone, la vitesse de convergence est acceptable en pratique.

Chapter 2

Introduction

The computational complexity of machine learning algorithms has become critical as such methods are required to handle exploding volumes of data, both in terms of sample size and feature space dimension, making the empirical risk estimation or optimization a challenge. Stochastic optimization algorithms, such as stochastic gradient descent, have significantly improved the speed of convergence — even the feasibility — of many machine learning problems. In such methods, one can use a small subsample of observations at each step while limiting the deterioration of the convergence rate in comparison to deterministic methods. Similarly, the downtrend in the improvements of processors computational capabilities has stimulated the development of distributed and parallelized algorithms. They offer efficient alternatives for minimizing an empirical risk involving a huge amount of observations and can usually be extended to decentralized constraints, where the data distribution is not controlled by a central *master* node.

However, in a wide variety of machine learning problems (*e.g.,* clustering, image recognition, ranking, learning on graphs), natural estimates of the risk are not basic sample means but take the form of averages of d-tuples, usually referred to as U-statistics in Probability and Statistics, see LEE, 1990a. In CLÉMENÇON, LUGOSI, and VAYATIS, 2005 for instance, ranking is viewed as pairwise classification and the empirical ranking error of any given prediction rule is a U-statistic of order 2, just like the *within cluster point scatter* in cluster analysis (see CLÉMENÇON, 2014) or empirical performance measures in metric learning (refer to CAO, GUO, and YING, 2012 for instance). Because empirical functionals are computed by averaging over tuples of sampling observations, they exhibit a complex dependence structure, which is the price to pay for low variance estimates. While the Empirical Risk Minimization (ERM) theory based on minimization of U-statistics is now consolidated (see CLÉMENÇON, LUGOSI, and VAYATIS, 2008), this approach generally leads to significant computational difficulties that are not sufficiently well documented in the machine learning literature. In many concrete cases, the mere computation of the risk involves a summation over an extremely high number of tuples and runs out of time or memory on most machines.

The goal of this work is to provide an analysis on several machine learning problems involving U-statistics. We first study the influence of the sampling schemes of a U-statistic for optimizing an empirical risk. Then, we present new and efficient methods for estimating a U-statistic in a decentralized setting. Finally, we investigate the decentralized optimization of convex functions that are separable in the pairs of obersvations.

2.1 U-statistics sampling

In classification and regression, empirical risk estimates are sample mean statistics, *i.e.,* of the form

$$R_n(\boldsymbol{\theta}; X_1, \ldots, X_n) = \frac{1}{n} \sum_{i=1}^{n} s(\boldsymbol{\theta}; X_i), \tag{2.1}$$

where $(X_i)_{1 \leq i \leq n}$ is the observation sample. The theory of ERM has been originally developed in this context and is at the root of many machine

learning techniques, including stochastic and distributed optimization. We focus on empirical risk formulations involving U-statistics. For $d > 0$, a U-statistic of order d is a statistic involving d-uplet of the observation sample, that is

$$U_n(H_{\boldsymbol{\theta}}) = \frac{1}{\binom{n}{d}} \sum_{1 \leq i_1 < \ldots < i_d \leq n} H_{\boldsymbol{\theta}}(X_{i_1}, \ldots, X_{i_d}),$$

where $H_{\boldsymbol{\theta}}$ is a symmetric and mesurable function of d elements. When $d = 1$, we recover the sample mean described above. For $d = 2$, the U-statistic is an average over all possible pairs. This formulation is used in a wide variety of machine learning problems; for instance, in the metric learning problem, for a given data sample $(X_i)_{1 \leq i \leq n} \in (\mathbb{R}^p)^n$ and the associated labels $(Y_i)_{1 \leq i \leq n} \in \{-1, +1\}^n$, one aims at finding the distance minimizing the following risk:

$$R_n(\boldsymbol{\theta}; X_1, \ldots, X_n) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} [Y_i Y_j (b - (X_i - X_j)^\top \boldsymbol{\theta} (X_i - X_j))]_+,$$

which is obviously a U-statistic of degree 2. Other examples include clustering, ranking or learning on graphs.

2.1.1 Incomplete U-statistics

Most statistical learning problems can be formulated as finding a certain parameter $\boldsymbol{\theta}^*$ in a class Θ which minimizes a true risk $R(\boldsymbol{\theta}) = \mathbb{E}_\nu[s(\boldsymbol{\theta}; X)]$, for a given distribution ν. This distribution is often unavailable in practice, so the ERM paradigm in statistical learning suggests to replace the true risk by the empirical risk, as defined in (2.1). The ERM theory essentially relies on the study of maximal deviations between these empirical averages and their expectations, under adequate complexity assumptions on the set of prediction rule candidates, relevant tools being mainly concentration inequalities for empirical processes.

When the empirical risk estimate is formulated as a U-statistic, one can show that in the usual asymptotic framework, the statistical learning rate is of order $O_{\mathbb{P}}(\sqrt{\log n / n})$. However, while yielding interesting properties such as reduced variance, such statistic requires largely superior processing capabilities to be computed or optimized, since the number of terms to be averaged is of order $\binom{n}{d}$, where d is the order of the U-statistic. As a remedy to this computational issue, the concept of *incomplete U-statistic* has been introduced in the seminal contribution of Blom, 1976. The idea behind incomplete U-statistics is to estimate the full U-statistic by performing a sampling with replacement over the $\binom{n}{d}$ sets of d-tuples of indices. Let $B > 0$, an incomplete U-statistic of order d based on B terms is of the form:

$$\tilde{U}_B(H_{\boldsymbol{\theta}}) = \frac{1}{B} \sum_{(I_1, \ldots, I_d) \in \mathcal{D}_B} H_{\boldsymbol{\theta}}(X_{I_1}, \ldots, X_{I_d}), \tag{2.2}$$

where $\mathcal{D}_B$ is a set of cardinality B built by sampling with replacement in the set $\Lambda := \{(i_1, \ldots, i_d), 1 \leq i_1 < \ldots < i_d \leq n\}$.

For this purpose, we investigate to which extent a U-process, that is a collection of U-statistics, can be accurately approximated by a Monte-Carlo version (which shall be referred to as an *incomplete U-process* throughout

this thesis) involving much less terms, provided it is indexed by a class of kernels of controlled complexity. In practice, B should obviously be chosen much smaller than $\binom{n}{d}$ in order to overcome the computational issue aforementioned. Note that the distribution of a complete U-statistic built from a subsample of reduced size n' drawn uniformly at random is quite different from that of an incomplete U-statistic based on $B = \binom{n'}{d}$ terms sampled with replacement in Λ, although they involve the summation of the same number of terms.

As an estimator of R, the statistic (2.2) is still unbiased, *i.e.* $\mathbb{E}[\widetilde{U}_B(H_{\boldsymbol{\theta}})] = R(\boldsymbol{\theta})$. However, its variance is naturally larger than that of the complete U-statistic $U_n(H_\theta)$. Precisely, its variance can be written

$$\mathrm{Var}(\widetilde{U}_B(H_{\boldsymbol{\theta}})) = \left(1 - \frac{1}{B}\right)\mathrm{Var}(U_n(H_{\boldsymbol{\theta}})) + \frac{1}{B}\mathrm{Var}(H_{\boldsymbol{\theta}}(X_1, \ldots, X_d)). \quad (2.3)$$

Therefore, the difference vanishes at a rate $1/B$ and a natural follow-up is to check that this additional variance does not damage excessively the learning rates. We propose the following result, based on the VC dimension of $\mathcal{H}$.

Theorem 6. (MAXIMAL DEVIATION) *Let $\mathcal{H} := \{H_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta\}$ be a collection of bounded symmetric kernels such that*

$$\mathcal{M}_{\mathcal{H}} := \sup_{(H_{\boldsymbol{\theta}}, x) \in \mathcal{H} \times \mathcal{X}} |H_{\boldsymbol{\theta}}(x)| < +\infty. \quad (2.4)$$

Suppose also that $\mathcal{H}$ is a Vapnik-Chervonenkis major class of functions with finite VC *dimension $V < +\infty$. Then, the following assertions hold true:*

(i) For all $\delta \in (0,1)$, with probability at least $1-\delta$, we have: for all $B \geq 1$ and for all $n \in \mathbb{N}^*$,

$$\sup_{H_{\boldsymbol{\theta}} \in \mathcal{H}} \left|\widetilde{U}_B(H_{\boldsymbol{\theta}}) - U_n(H_{\boldsymbol{\theta}})\right| \leq \mathcal{M}_{\mathcal{H}} \times \sqrt{2\frac{V\log(1+|\Lambda|) + \log(2/\delta)}{B}}$$

(ii) For all $\delta \in (0,1)$, with probability at least $1-\delta$, we have: $\forall n \in \mathbb{N}^*$, $\forall B \geq 1$,

$$\frac{1}{\mathcal{M}_{\mathcal{H}}} \sup_{H_{\boldsymbol{\theta}} \in \mathcal{H}} \left|\widetilde{U}_B(H_{\boldsymbol{\theta}}) - R(\boldsymbol{\theta})\right| \leq 2\sqrt{\frac{2V\log(1+N)}{N}} + \sqrt{\frac{\log(2/\delta)}{N}} + \sqrt{2\frac{V\log(1+|\Lambda|) + \log(4/\delta)}{B}},$$

where $N = \lfloor n/d \rfloor$.

The first assertion of Theorem 6 provides a control of the deviations between the U-statistic and its incomplete counterpart uniformly over the class $\mathcal{H}$. As the number of terms B increases, this deviation decreases at a rate of $O(1/\sqrt{B})$. The second assertion of Theorem 6 gives a maximal deviation result with respect to $R(\boldsymbol{\theta})$. Observe in particular that, with the asymptotic settings previously specified, $\log(|\Lambda|) = O(\log n)$ as $n \to +\infty$. In addition one may straightforwardly deduce a bound on the excess risk of kernels minimizing the incomplete version of the empirical risk based on B terms and show that when an incomplete U-statistic contains $B = O(n)$

terms only, the learning rate for the corresponding minimizer is of the same order as that of the minimizer of the complete risk, whose computation requires to average $|\Lambda| = O(n^d)$ terms. Minimizing such incomplete U-statistics thus yields a significant gain in terms of computational cost while fully preserving the learning rate. In contrast, the minimization of a complete U-statistic involving $O(n)$ terms, obtained by drawing subsamples of sizes $n' = O(n^{1/d})$ uniformly at random, leads to a rate of convergence of $O(\sqrt{\log(n)/n^{1/d)}})$, which is much slower.

These results ensure that it is preferable — in terms of learning rate — to estimate the risk with the incomplete version of the U-statistic when given the opportunity.

2.1.2 Application to Stochastic Gradient Descent

We now tackle the problem of *finding* an empirical minimizer; we investigate the benefits of computing incomplete U-statistics in iterative schemes for statistical learning over subsampled complete U-statistics. In particular, we analyze Stochastic Gradient Descent (SGD), as it is used in a wide range of machine learning methods, such as SVM, DEEP NEURAL NETWORKS or SOFT K-MEANS.

Let $\boldsymbol{\theta}$ be some parameter space and $H : \mathcal{X}^d \times \Theta \to \mathbb{R}$ be a loss function which is convex and differentiable in its last argument. For all $\boldsymbol{\theta} \in \Theta$, we aim at minimizing

$$R(\boldsymbol{\theta}) = \mathbb{E}[H_{\boldsymbol{\theta}}(X_1, \ldots, X_d)].$$

As previously mentioned, since the real risk is often unavailable, we will aim at minimizing the empirical risk associated to a sample $(X_1, \ldots, X_n)$:

$$R_n(\boldsymbol{\theta}; X_1, \ldots, X_n) = \frac{1}{\binom{n}{d}} \sum_{1 \leq i_1 < \ldots < i_d \leq n} H_{\boldsymbol{\theta}}(X_{i_1}, \ldots, X_{i_n}).$$

In the SGD scheme, we use the following update rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \tilde{g}(\boldsymbol{\theta}), \tag{2.5}$$

where $\tilde{g}(\boldsymbol{\theta})$ is an unbiased estimate of $R_n(\boldsymbol{\theta}; X_1, \ldots, X_n)$ and the step size $\eta_t \geq 0$ is such that $\sum_{t=1}^{+\infty} \eta_t = +\infty$ and $\sum_{t=1}^{+\infty} \eta_t^2 < +\infty$.

A natural approach consists in replacing the true gradient by a complete U-statistic constructed from subsamples of reduced sizes $n' \ll n$ drawn uniformly at random, leading to the following gradient estimate:

$$\widetilde{g}_{n'}(\boldsymbol{\theta}) = \frac{1}{\binom{n'}{d}} \sum_{(i_1, \ldots, i_d) \in \Lambda'} \nabla_{\boldsymbol{\theta}} H(X_{i_1}, \ldots, X_{i_d};\, \boldsymbol{\theta}), \tag{2.6}$$

where Λ' is the set of all d-tuples $1 \leq i_1 < \ldots < i_d \leq n$ related to a subset of n' indexes in $[n]$. This is the naive approach described in the previous section.

We propose an alternative strategy involving a gradient estimate in the form of an *incomplete* U-statistic:

$$\widetilde{g}_B(\boldsymbol{\theta}) = \frac{1}{B} \sum_{(I_1, \ldots, I_d) \in \mathcal{D}_B} \nabla_{\boldsymbol{\theta}} H(X_{I_1}, \ldots, \mathbf{X}_{I_d};\, \boldsymbol{\theta}), \tag{2.7}$$

where $\mathcal{D}_B$ is built by sampling with replacement in the set Λ.

It is well-known that the variance of the gradient estimate negatively impacts on the convergence of SGD. Recent works have focused on variance-reduction strategies for SGD when the risk estimates are basic sample means (see for instance LE ROUX, SCHMIDT, and BACH, 2012; JOHNSON and ZHANG, 2013). Therefore, in order to quantify the efficiency of a sampling scheme, we look at the variance associated to the gradient estimate; this leads to the following result.

Proposition 1. *Let $B = \binom{n'}{d}$ for $n' \ll n$. In the asymptotic framework, we have:*

$$\mathrm{Var}[\widetilde{g}_{n'}(\boldsymbol{\theta})] = O\left(\frac{1}{n'}\right), \quad \mathrm{Var}[\widetilde{g}_B(\boldsymbol{\theta})] = O\left(\frac{1}{\binom{n'}{d}}\right).$$

With Proposition 1, we show that the convergence rate of $\mathrm{Var}[\widetilde{g}_B(\boldsymbol{\theta})]$ is faster than that of $\mathrm{Var}[\widetilde{g}_{n'}(\boldsymbol{\theta})]$. Thus the expected improvement in objective function at each SGD step is larger when using a gradient estimate in the form of (2.7) instead of (2.6), although both strategies require to average over the same number of terms. This is confirmed by the experimental results reported in the next section.

2.1.3 Numerical Experiments

We performed numerical experiments on the metric learning problem (see Section 3.2.2). As done in much of the metric learning literature, we restrict our attention to the family of pseudo-distance functions $D_{\boldsymbol{\theta}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ defined as

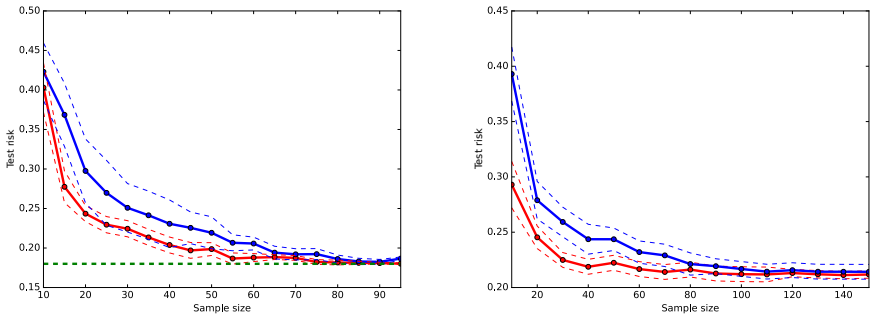
$$D_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top \boldsymbol{\theta} (\mathbf{x} - \mathbf{x}'),$$

where $\boldsymbol{\theta} \in \mathbb{S}_+^d$, and $\mathbb{S}_+^d$ is the cone of $d \times d$ symmetric positive-semidefinite (PSD) matrices.

Given a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$, let $y_{ij} = 1$ if $y_i = y_j$ and 0 otherwise for any pair of samples. Given a threshold $b \geq 0$, we define the empirical risk as follows:

$$R_n(\boldsymbol{\theta}) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \left[y_{ij} (b - D_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)) \right]_+, \tag{2.8}$$

where $[u]_+ = \max(0, 1 - u)$ is the hinge loss. Our goal is the find the empirical risk minimizer among our family of distance functions In our experiments, we used two data sets: a synthetic data set generated from a mixture of 10 gaussians in $\mathbb{R}^{40}$ and the MNIST data set — see Section 3.5 for details about datasets. These datasets contain respectively 50000 and 60000 training samples, so computing the full empirical risk for only one candidate $\boldsymbol{\theta}$ would require an averaging of 10^9 pairs. We conduct two types of experiment. First, we subsample the data before learning and evaluate the performance of the ERM on the subsample. Then, we use Stochastic Gradient Descent to find the ERM on the original sample, using subsamples at each iteration to estimate the gradient. We use p indices picked at random for the complete sampling scheme and $p(p-1)/2$ pairs for the incomplete one, so every scheme requires the computation of the same amount



(A) Synthetic dataset. (B) MNIST data set.

FIGURE 2.1: Test risk with respect to the sample size p when using complete (blue) or incomplete (red) U-statistics. Solid lines represent means and dashed ones represent standard deviation. The green dotted line represents the performance of the true risk minimizer.

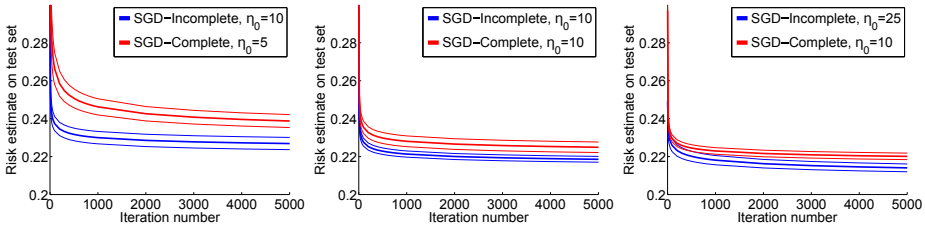


FIGURE 2.2: SGD results on the MNIST data set for various mini-batch size. Bold and thin lines respectively shows the means and standard deviations over 50 runs.

of terms. For each strategy, we use a projected gradient descent method in order to minimize (2.8), using several values of p and averaging the results over 50 random trials. As the testing sets are large, we evaluate the test risk on 100,000 randomly picked pairs.

Figure 2.1a shows the test risk of the ERM with respect to the sample size p for both sampling strategies on the synthetic data set. As predicted by our theoretical analysis, the incomplete U-statistic strategy achieves a significantly smaller risk on average. For instance, it gets within 5% error of the true risk minimizer for $p = 50$, while the complete U-statistic needs $p > 80$ to reach the same performance. The same conclusions hold for the MNIST data set, as can be seen in Figure 2.1b.

Figure 2.2 shows the comparison in SGD scheme for three mini-batch sizes, where we plot the evolution of the test risk with respect to the iteration number. For all mini-batch sizes, SGD-Incomplete achieves significantly better test risk than SGD-Complete. The best learning rate is often larger for SGD-Incomplete than for SGD-Complete: this confirms that gradient estimates from the former strategy are generally more reliable. This is further supported by the fact that even though larger learning rates increase the variance of SGD, in these two cases SGD-Complete and SGD-Incomplete have similar variance. SGD-Incomplete again performs significantly better on average and also has smaller variance. Lastly, as one should expect, the gap between SGD-Complete and SGD-Incomplete reduces as

the size of the mini-batch increases; note however that in practical implementations, the relatively small mini-batch sizes are generally those which achieve the best error/time trade-off.

We now turn to a new type of constraint: as previously mentioned, distributed and decentralized methods are required in an increasing number of applications. In such settings, the empirical risk itself is not computable — or at a prohibitive cost — so usual methods have to be adapted in order to stay efficient.

2.2 Gossip protocols

Our methods for estimating and optimizing U-statistic-based empirical risks rely on gossip protocols. Such algorithms are tailored to this setting as they only rely on simple peer-to-peer communication: each agent only exchanges information with one neighbor at a time. Thus, before introducing our decentralized methods, we briefly review the basics of gossip methods and provide additional details about two key notions: clock modelling and graph Laplacian.

2.2.1 Background

Gossip algorithms have been developped for solving a large variety of machine learning problems, from data aggregation over sensor networks (HEDETNIEMI, HEDETNIEMI, and LIESTMAN, 1988; DIMAKIS, SARWATE, and WAINWRIGHT, 2008; KAR and MOURA, 2009) to decentralized multi-agent optimization (NEDIĆ, 2011; DUCHI, AGARWAL, and WAINWRIGHT, 2012; TSIANOS, LAWLOR, and RABBAT, 2015). Despite being designed for operating on diverse problems, gossip algorithms usually share the same core constraints. Namely, one aims at estimating or optimizing some function depending on data samples that are partitioned over a connected network under at least one of the following constraints:

(i) there is no central node ensuring time-synchrony or global data aggregation among the network,

(ii) the computation and storage capabilities of each node are strongly limited,

(iii) agent-to-agent communication is expensive.

In some gossip methods, only constraints (ii) and (iii) are considered, leading to synchronous or partially asynchronous distributed algorithms (KARP et al., 2000; KEMPE, DOBRA, and GEHRKE, 2003; RAM, NEDIĆ, and VEERAVALLI, 2010), while other methods satisfy all three constraints, that is fully asynchronous decentralized algorithms (BOYD et al., 2006; NEDIĆ, 2011; LEE and NEDIĆ, 2015).

Example 1 (connected cars) Let us consider connected vehicles being driven in a city. These cars may hold valuable information such as surrounding traffic, weather data or driver behavior. However, due to the data stream volume, it may be hard or unreliable to continuously centralize information on a global server in order to perform machine learning techniques. Therefore, one could take advantage

from the cars being driven around the city to require them to perform — cheap — local computations using data they hold and to exchange calculation results with other cars whenever they are close enough and have finished their local computations. One convenient way to model this communication scheme is to use a geometric graph. A geometric graph $\mathcal{G} = ([n], \mathcal{E})$ of radius $r > 0$ is an undirected graph such that given a distance d over the nodes attributes space, for any pair of nodes $(i, j) \in V^2$, we have $(i, j) \in \mathcal{E}$ if and only if $d(i, j) < r$. Global network asynchrony and uncertainty of communications are then naturally modelled by fully asynchronous gossip protocols. Note that in this case, the underlying graph used for modelling the communication capability of the network is dynamic: one should consider a sequence $(\mathcal{G}(t))_{t\geq 0}$ rather than just one graph $\mathcal{G}$. Hopefully, the methods we consider can be easily extended to such settings.

Example 2 (mobile phones) In the mobile phones applications context, each node of the network is a mobile phone and a communication occurs when one phone sends a message to another. One may exploit users' messages to perform clustering or topic modelling to name a few examples. Nevertheless, gathering such data on a central server can also be delicate for various reasons: privacy, potentially huge data volume, *etc.* Moreover, while local computations can be handled by nowadays mobile phones, the impact of repeated communication establishment on the phone battery can be a serious bottleneck, hence the suitability of a gossip-type protocol sending local calculation results alongside messages.

The general idea behind gossip algorithms is to alternate two steps: (optional) local updates (*e.g.*, a gradient descent step) and communication steps (*e.g.*, averaging). In the estimation case, communication steps often consist in averaging the estimates of the selected nodes. For instance, in BOYD et al., 2006, a node averages its estimate whenever it establishes communication with another node. There exist some exceptions: in PELCKMANS and SUYKENS, 2009, the local update adds an element to the U-statistic estimate while the communication step is an observation swap. In gossip optimization, communications steps are again mostly averaging steps while local updates corresponds to one step of a — centralized — optimization algorithm (*e.g.*, gradient descent, dual averaging).

2.2.2 Clock modelling

We mentioned earlier that, depending on the set of constraints considered, gossip algorithms can perform in synchronous or asynchronous setting. In the synchronous setting, we consider that the nodes have access to a global clock. That way, they can all perform local updates at each time instance. This is not always a realistic assumption, but it allows for easier analysis and implementation, since learning rate is uniform over the network and every node contributes equivalently to the global objective. Therefore, in this work, synchronous analysis will be used for both providing detailed convergence rate and serving as a stepping stone toward fully asynchronous analysis.

In a fully asynchronous setting, nodes do not have access to a global clock; instead, each node has a local clock. One common way to model local clocks is to consider i.i.d. clocks ticking at a rate 1 Poisson process, so one can use an equivalent model with a global clock ticking at a rate n Poisson process and a random edge draw at each iteration, as in synchronous setting. However, at a given iteration, the local update step now only involves the selected pair of nodes. Therefore, the nodes need to maintain an estimate of the current iteration number to ensure convergence. The time estimate we use in our methods is defined as follows. Let $\mathcal{G} = ([n], \mathcal{E})$ be an undirected graph. For $k \in [n]$, let p_k denote the probability for the node k to be picked at any iteration. If the edges are picked uniformly at random, then one has $p_k = 2d_k/|\mathcal{E}|$, where d_k is the degree of node k. For simplicity, we focus only on this case, although our analysis holds in a more general setting. Let us define $(\delta_k(t))_{t\geq 1}$ such that for any $t \geq 1$,

$$\delta_k(t) = \begin{cases} 1 & \text{if node } k \text{ is picked at iteration } t \\ 0 & \text{otherwise} \end{cases}.$$

One can immediately see that $(\delta_k(t))_{t>0}$ are i.i.d. random variables following a Bernouilli distribution with parameter p_k. Let us define $(m_k(t)) \geq 0$ such that for $t > 0$:

$$m_k(t) = \frac{1}{p_k} \sum_{s=1}^{t} \delta_k(s).$$

Since $(\delta_k(t))_{t>0}$ are Bernoulli random variables, $m_k(t)$ is an unbiased estimate of the time t. Therefore, given that every node *knows its degree* and the *total number of edges* in the network, the iteration estimates are unbiased.

One may refer to BOYD et al., 2006 for more details about synchronous and asynchronous time models.

2.2.3 Graph Laplacian

In distributed and decentralized setting, the network dependency of a method is tightly bound to the eigenvalues of a matrix called the graph Laplacian CHUNG, 1997. Let $\mathcal{G} = ([n], \mathcal{E})$ be an undirected graph and let $\mathbf{A}^{\mathcal{G}}$ be its adjacency matrix, that is for any $(i, j) \in [n]^2$, $[\mathbf{A}^{\mathcal{G}}]_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$. The graph Laplacian $\mathbf{L}^{G}$ is defined as follows:

$$\mathbf{L}^{\mathcal{G}} = \mathbf{D}^{\mathcal{G}} - \mathbf{A}^{\mathcal{G}},$$

where $\mathbf{D}^{\mathcal{G}}$ is the degree matrix, *i.e.*, $\mathbf{D}^{\mathcal{G}} = \mathrm{diag}(\mathbf{A}^{\mathcal{G}}\mathbf{1}_n)$. The graph Laplacian holds several interesting properties such as symmetry and positiveness, but the more remarkable one is that, up to a renormalization, it corresponds to the transition matrix of a random walk over $\mathcal{G}$. Its smallest non-zero eigenvalue, also called *spectral gap*, characterizes the diffusion capability of a network. Indeed, one can link the Laplacian to the standard markovian analysis and show that if the network is connected and non-bipartite then the random walk is respectively irreducible and aperiodic, ensuring its convergence towards a uniform at a geometric rate whose ratio is tied to the spectral gap.

As an illustration, one can consider the gossip sample mean estimation described in BOYD et al., 2006. In this case, the error bound after t iterations is of the form λ_2^t, where $\lambda_2 := 1 - \beta_{n-1}/|\mathcal{E}|$ and β_{n-1} is the spectral gap. Another example lies in the distributed dual averaging of AGARWAL, WAINWRIGHT, and DUCHI, 2010. Their error upper-bound is equal to

$$\left(c + \frac{c'}{1-\lambda_2}\right)\frac{1}{\sqrt{t}},$$

with c and c' only depending on the conditioning of the optimization problem.

We now turn to decentralized estimation of U-statistics for both synchronous and asynchronous settings.

2.3 Decentralized estimation of U-statistics

Decentralized estimation have many applications in sensor and peer-to-peer networks as well as for extracting knowledge from massive information graphs such as interlinked Web documents and on-line social media. Algorithms running on such networks must often operate under tight constraints: the nodes forming the network cannot rely on a centralized entity for communication and synchronization, may not be aware of the global network topology and/or have limited resources (computational power, memory, energy). Gossip algorithms (TSITSIKLIS, 1984; SHAH, 2009; DIMAKIS et al., 2010), where each node exchanges information with at most one of its neighbors at a time, have emerged as a simple yet powerful technique for distributed computation in such settings.

Such algorithms have been extensively studied in the context of decentralized averaging in networks, where the goal is to compute the average of n real numbers ($\mathcal{X} = \mathbb{R}$):

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\mathbf{x}^\top \mathbf{1}_n. \tag{2.9}$$

One of the earliest work on this canonical problem is due to TSITSIKLIS, 1984, but more efficient algorithms have recently been proposed, see for instance KEMPE, DOBRA, and GEHRKE, 2003; BOYD et al., 2006. Of particular interest to us is the work of BOYD et al., 2006, which introduces a randomized gossip algorithm for computing the empirical mean (2.9) in a context where nodes wake up asynchronously and simply average their local estimate with that of a randomly chosen neighbor. The communication probabilities are given by a stochastic matrix $\mathbf{P}$, where $[\mathbf{P}]_{ij}$ is the probability that a node i selects neighbor j at a given iteration. As explained in Section 2.2.3, the local estimates converge to (2.9) at a geometric rate, with a ratio depending on the spectral gap of the network. Such algorithms can be extended to compute other functions such as maxima and minima, or sums of the form $\sum_{i=1}^n f(x_i)$ for some function $f : \mathcal{X} \to \mathbb{R}$ (as done for instance in MOSK-AOYAMA and SHAH, 2008). Some works have also gone into developing faster gossip algorithms for poorly connected networks, assuming that nodes know their (partial) geographic location (DIMAKIS,

Algorithm 5 GoSta-sync: a synchronous gossip algorithm for computing a U-statistic

Require: Each node k holds observation $\mathbf{x}_k$

Each node k initializes its auxiliary observation $\mathbf{y}_k = \mathbf{x}_k$ and its estimate $z_k = 0$

for $t = 1, 2, \ldots$ **do**
  for $p = 1, \ldots, n$ **do**
    Set $z_p \leftarrow \frac{t-1}{t} z_p + \frac{1}{t} h(\mathbf{x}_p, \mathbf{y}_p)$
  end for
  Draw (i, j) uniformly at random from $\mathcal{E}$
  Set $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
  Swap auxiliary observations of nodes i and j: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
end for

Sarwate, and Wainwright, 2008; Li, Dai, and Zhang, 2010). More recently, Loizou and Richtárik, 2016 developed a new perspective on gossip algorithms analysis, using Randomized Block Kaczmarz and studying optimization duality. For a detailed account of the literature on gossip algorithms, we refer the reader to Shah, 2009; Dimakis et al., 2010.

Here, we tackle the problems of decentralized estimation, where the quantity of interest is a U-statistic of order 2. That is, we are interested in estimating the quantity:

$$\hat{U}_n(h) = \frac{1}{n^2} \sum_{i,j=1}^{n} h(\mathbf{x}_i, \mathbf{x}_j), \tag{2.10}$$

where $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is some symmetric function and $(\mathbf{x}_k)_{1 \leq k \leq n} \in \mathcal{X}^n$ are observations in some feature space $\mathcal{X}$ [1]. We assume the observations are distributed over a network $\mathcal{G} = ([n], \mathcal{E})$ and each node $i \in [n]$ of this network contains exactly one observation $\mathbf{x}_i$.

Existing gossip algorithms cannot be used to efficiently compute (2.10) as it depends on *pairs* of observations. To the best of our knowledge, this problem has only been investigated in Pelckmans and Suykens, 2009. Their algorithm, coined U2-gossip, achieves $O(1/t)$ convergence rate but has several drawbacks. First, each node must store two auxiliary observations, and two pairs of nodes must exchange an observation at each iteration. For high-dimensional problems (large d), this leads to a significant memory and communication load. Second, the algorithm is not asynchronous as every node must update its estimate at each iteration. Consequently, nodes must have access to a global clock, which is often unrealistic in practice. In the next section, we introduce new synchronous and asynchronous algorithms with faster convergence as well as smaller memory and communication cost per iteration.

2.3.1 GoSta Algorithms

We base the proposed method on the observation that the U-statistic can be rewritten $\hat{U}_n(h) = (1/n) \sum_{i=1}^{n} \overline{h}_i$, with $\overline{h}_i = (1/n) \sum_{j=1}^{n} h(\mathbf{x}_i, \mathbf{x}_j)$, and we

1. This formulation deviates from the usual U-statistic definition since diagonal terms are included. This will however simplify the analysis while including the particular case where $h(\mathbf{x}, \mathbf{x}) = 0$.

write $\overline{\mathbf{h}} = (\overline{h}_1, \dots, \overline{h}_n)^\top$. The goal is thus similar to the usual distributed averaging problem (2.9), with the key difference that each local value $\overline{h}_i$ is itself an average depending on the entire data sample. Consequently, our algorithms will combine two steps at each iteration: a data propagation step to allow each node i to estimate $\overline{h}_i$, and an averaging step to ensure convergence to the desired value $\hat{U}_n(h)$.

Synchronous Setting

In the synchronous setting, we assume that the nodes have access to a global clock so that they can all update their estimate at each time instance. We stress that the nodes need not to be aware of the global network topology as they will only interact with their direct neighbors in the graph.

Let us denote by $z_k(t)$ the (local) estimate of $\hat{U}_n(h)$ by node k at iteration t. In order to propagate data across the network, each node k maintains an auxiliary observation $\mathbf{y}_k$, initialized to $\mathbf{x}_k$. Our algorithm, coined GoSta for gossip U-statistic, goes as follows. At each iteration, each node k updates its local estimate by taking the running average of $z_k(t)$ and $h(\mathbf{x}_k, \mathbf{y}_k)$. Then, an edge of the network is drawn uniformly at random, and the corresponding pair of nodes average their local estimates and swap their auxiliary observations. The observations are thus each performing a random walk (albeit coupled) on the network graph. The full procedure is described in Algorithm 5.

We state the convergence rate of this algorithm in the next theorem.

Theorem 7. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non-bipartite graph, $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ a data sample and $(\mathbf{z}(t))$ the sequence of estimates generated by Algorithm 5. For all $k \in [n]$, we have:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n(h).$$

Moreover, for any $t > 0$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n \right\| \leq \frac{1}{ct} \left\| \overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n \right\| + \left(\frac{2}{ct} + e^{-ct} \right) \left\| \mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top \right\|,$$

where for any $1 \leq i, j \leq n$, $[\mathbf{H}]_{ij} = h(\mathbf{x}_i, \mathbf{x}_j)$, $c = c(\mathcal{G}) := \beta_{n-1}/|\mathcal{E}|$, and β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}^{\mathcal{G}}$.

Theorem 7 shows that the local estimates generated by Algorithm 5 converge to $\hat{U}_n(h)$ at a rate $O(1/t)$. Furthermore, the constants reveal the rate dependency on the particular problem instance. Indeed, the two norm terms are *data-dependent* and quantify the difficulty of the estimation problem itself through a dispersion measure. In contrast, $c(\mathcal{G})$ is *network-dependent* since β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}^{\mathcal{G}}$, *i.e.*, the spectral gap of $\mathcal{G}$. Therefore, we expect graph with better connectivity to converge faster; this will be evidenced in the numerical experiments.

To estimate $\hat{U}_n(h)$, U2-gossip (Pelckmans and Suykens, 2009) does not use averaging. Instead, each node k requires two auxiliary observations $\mathbf{y}_k^{(1)}$ and $\mathbf{y}_k^{(2)}$ which are both initialized to $\mathbf{x}_k$. At each iteration, each node

Algorithm 6 GoSta-async: an asynchronous gossip algorithm for computing a U-statistic

Require: Each node k holds observation $\mathbf{x}_k$ and $p_k = 2d_k/|\mathcal{E}|$
1: Each node k initializes $\mathbf{y}_k = \mathbf{x}_k$, $z_k = 0$ and $m_k = 0$
2: **for** $t = 1, 2, \dots$ **do**
3: Draw (i, j) uniformly at random from $\mathcal{E}$
4: Set $m_i \leftarrow m_i + 1/p_i$ and $m_j \leftarrow m_j + 1/p_j$
5: Set $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
6: Set $z_i \leftarrow (1 - \frac{1}{p_i m_i})z_i + \frac{1}{p_i m_i} h(\mathbf{x}_i, \mathbf{y}_i)$
7: Set $z_j \leftarrow (1 - \frac{1}{p_j m_j})z_j + \frac{1}{p_j m_j} h(\mathbf{x}_j, \mathbf{y}_j)$
8: Swap auxiliary observations of nodes i and j: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
9: **end for**

$k \in [n]$ updates its local estimate by taking the running average of z_k and $h(\mathbf{y}_k^{(1)}, \mathbf{y}_k^{(2)})$. Then, two random edges are selected: the nodes connected by the first (resp. second) edge swap their first (resp. second) auxiliary observations. Applying our convergence analysis to U2-gossip, we obtain the following refined rate:

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n \right\| \leq \frac{\sqrt{n}}{t} \left(\frac{2}{1-\tilde{\lambda}} \left\| \overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n \right\| + \frac{1}{1-\tilde{\lambda}^2} \left\| \mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top \right\| \right), \tag{2.11}$$

where $1 - \tilde{\lambda} = 2c(\mathcal{G})$. The advantage of propagating two observations in U2-gossip is seen in the $1/(1-\tilde{\lambda}^2)$ term, however the absence of averaging leads to an overall $\sqrt{n}$ factor. Intuitively, this is because nodes do not benefit from each other's estimate. In practice, $\tilde{\lambda}$ is close to 1 for reasonably-sized networks (for instance, $\tilde{\lambda} = 1 - 2/n$ for the complete graph), so the square term does not provide much gain and the $\sqrt{n}$ factor dominates in (2.11). We thus expect U2-gossip to converge slower than GoSta, which is confirmed by the numerical results.

Asynchronous Setting

We now remove the global clock assumption. Using the time estimate described in Section 2.2.2, we can now give an asynchronous version of GoSta, as stated in Algorithm 6. Note that the update step slightly differs from the synchronous setting: this is due to the fact that each node needs to contribute equivalently in the statistic estimate. Therefore, nodes activated less often must have a bigger weight on their contributions.

To show that local estimates converge to $\hat{U}_n(h)$, we use a similar model as in the synchronous setting. The time dependency of the transition matrix is more complex ; so is the upper bound.

Theorem 8. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph, $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ a data sample and $(\mathbf{z}(t))$ the sequence of estimates generated by Algorithm 6. For all $k \in [n]$, we have:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n(h).$$

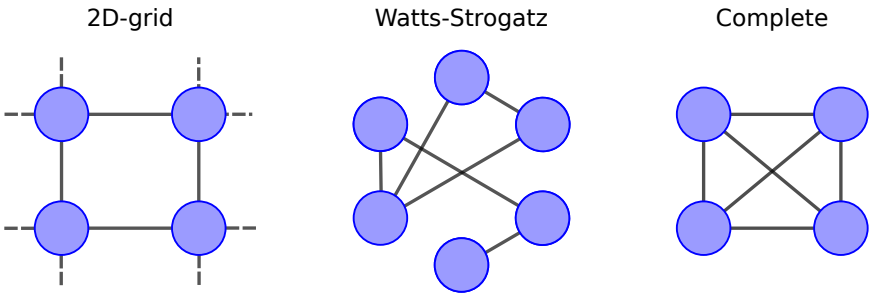


FIGURE 2.3: Network examples

Moreover, there exists a constant $c'(\mathcal{G}) > 0$ such that, for any $t > 1$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n \right\| \leq c'(\mathcal{G}) \cdot \frac{\log t}{t} \|\mathbf{H}\|.$$

2.3.2 Experiments

We present two applications on real datasets. The first one is the within-cluster point scatter (CLÉMENÇON, 2011), which measures the clustering quality of a partition $\mathcal{P}$ of $\mathcal{X}$ as the average distance between points in each cell $\mathcal{C} \in \mathcal{P}$. It is of the form (2.10) with

$$h_{\mathcal{P}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| \cdot \sum_{\mathcal{C} \in \mathcal{P}} \mathbb{1}_{\{(\mathbf{x}, \mathbf{x}') \in \mathcal{C}^2\}}. \tag{2.12}$$

We also study the AUC measure (HANLEY and MCNEIL, 1982). For a given sample $(\mathbf{x}_1, \ell_1), \ldots, (\mathbf{x}_n, \ell_n)$ on $\mathcal{X} \times \{-1, +1\}$, the AUC measure of a linear classifier $\boldsymbol{\theta} \in \mathbb{R}^{d-1}$ is given by:

$$\mathrm{AUC}(\boldsymbol{\theta}) = \frac{\sum_{1 \leq i,j \leq n} (1 - \ell_i \ell_j) \mathbb{1}_{\{\ell_i (\boldsymbol{\theta}^\top x_i) > -\ell_j (\boldsymbol{\theta}^\top x_j)\}}}{4 \left(\sum_{1 \leq i \leq n} \mathbb{1}_{\{\ell_i = 1\}} \right) \left(\sum_{1 \leq i \leq n} \mathbb{1}_{\{\ell_i = -1\}} \right)}. \tag{2.13}$$

This score is the probability for a classifier to rank a positive observation higher than a negative one.

We perform our simulations on the three types of network described below.

• *Complete graph:* This is the case where all nodes are connected to each other. It is the ideal situation in our framework, since any pair of nodes can communicate directly.

• *Two-dimensional grid:* Here, nodes are located on a 2D grid, and each node is connected to its four neighbors on the grid. This network offers a regular graph with isotropic communication, but its diameter ($\sqrt{n}$) is quite high, especially in comparison to usual scale-free networks.

• *Watts-Strogatz:* This random network generation technique is introduced in WATTS and STROGATZ, 1998 and allows us to create networks with various communication properties. Here, we tune parameters in order to achieve a connectivity compromise between the complete graph and the two-dimensional grid.

One may refer to Figure 2.3 for examples of such networks.

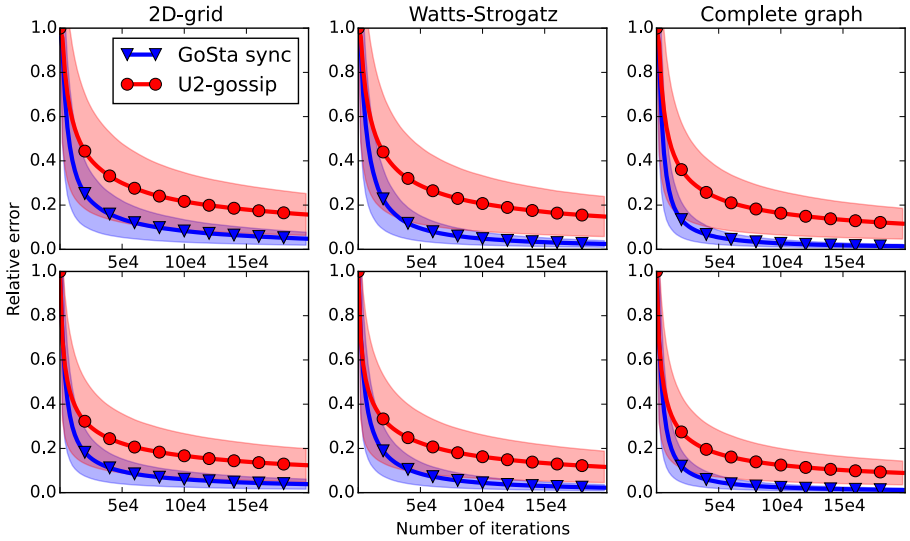


FIGURE 2.4: Evolution of the average relative error (solid line) and its standard deviation (filled area) with the number of iterations for U2-GOSSIP (red) and GOSTA-SYNC (blue) on the SVMguide3 dataset (top row) and the Wine Quality dataset (bottom row).

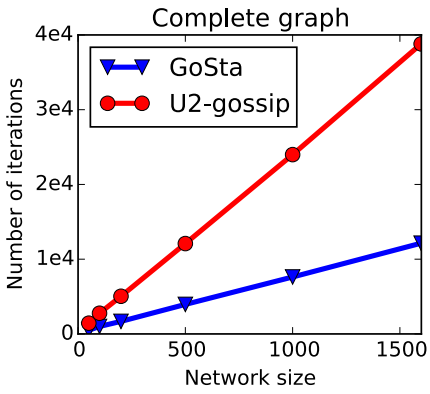


FIGURE 2.5: 20% error reaching time.

AUC measure We use the SMVGUIDE3 binary classification dataset which contains $n = 1260$ points in $d = 23$ dimensions[2] and we set θ to the difference between the class means. The top row of Figure 2.4 shows the evolution over time of the average relative error and the associated standard deviation *across nodes* for both algorithms on each type of network. On average, GOSTA-SYNC outperforms U2-GOSSIP on every network. The variance of the estimates across nodes is also lower due to the averaging step. Interestingly, the performance gap between the two algorithms is greatly increasing early on, presumably because the exponential term in the convergence bound of GOSTA-SYNC is significant in the first steps.

2. This dataset is available at http://mldata.org/repository/data/viewslug/svmguide3/

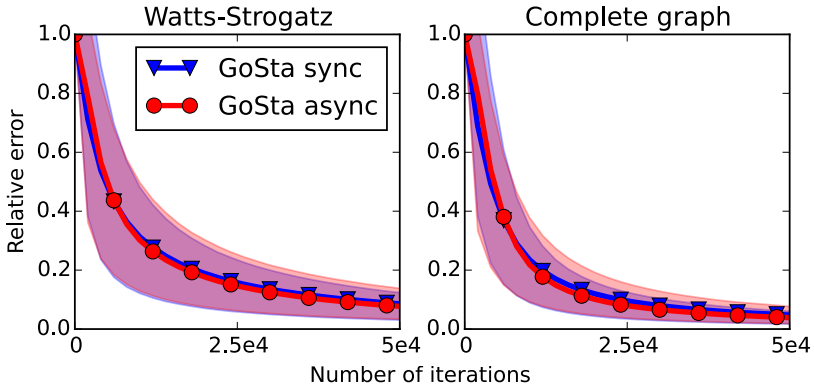


FIGURE 2.6: Relative error (solid line) and its standard deviation (filled area) of synchronous (blue) and asynchronous (red) versions of GOSTA.

Within-cluster point scatter We use the Wine Quality dataset which contains $n = 1599$ points in $d = 12$ dimensions, with a total of $K = 11$ classes. [3] We focus on the partition $\mathcal{P}$ associated to class centroids . The results are shown in the bottom row of Figure 2.4. As in the case of AUC, GOSTA-SYNC achieves better perfomance on all types of networks, both in terms of average error and variance. In Figure 2.5, we show the average time needed to reach a 0.2 relative error on a complete graph ranging from $n = 50$ to $n = 1599$. As predicted by our analysis, the performance gap widens in favor of GOSTA as the size of the graph increases. Finally, we compare the performance of GOSTA-SYNC and GOSTA-ASYNC (Algorithm 6) in Figure 2.6. Despite the slightly worse theoretical convergence rate for GOSTA-ASYNC, both algorithms have comparable performance in practice.

We now turn to the case where previously mentioned objectives need to be minimized, while still in a decentralized setting.

2.4 Decentralized optimization for pairwise functions

Decentralized optimization is particularly well suited to address the challenges posed by the advent of big data and the "Internet of Things". For instance, in large-scale machine learning, one aims at finding a model that minimizes a loss function over a massive dataset distributed across several machines in a commodity cluster or cloud computing platform. Other prominent applications come from wired and wireless networks, where local agents must coordinate in order to minimize a global objective function. Common strategies to solve such optimization problems rely on *gossip algorithms*, as for decentralized estimation. These algorithms have attracted a lot of interest due to their simplicity and their ability to operate in peer-to-peer networks where centralized coordination may be prohibitively expensive or even unavailable.

One of the flagship problems in decentralized optimization is to find a parameter vector $\boldsymbol{\theta}$ which minimizes an empirical risk expressed as average of convex functions $(1/n)\sum_{i=1}^{n} f(\boldsymbol{\theta}; \mathbf{x}_i)$, where the data $\mathbf{x}_i$ is only known to agent i. Various gossip algorithms based on (sub)gradient descent (NEDIĆ

3. This dataset is available at https://archive.ics.uci.edu/ml/datasets/Wine

and OZDAGLAR, 2009; JOHANSSON, RABI, and JOHANSSON, 2010; RAM, NEDIĆ, and VEERAVALLI, 2010; BIANCHI and JAKUBOWICZ, 2013), ADMM (WEI and OZDAGLAR, 2012; WEI and OZDAGLAR, 2013; IUTZELER et al., 2013) and dual averaging (DUCHI, AGARWAL, and WAINWRIGHT, 2012; YUAN et al., 2012; LEE, NEDIĆ, and RAGINSKY, 2015; TSIANOS, LAWLOR, and RABBAT, 2015) have been proposed to solve this problem, possibly including constrained and regularized terms. In these methods, each agent seeks to minimize its local function by applying local updates (*e.g.*, gradient steps) while exchanging information with neighbors to ensure convergence to the consensus value.

Here, we tackle the more challenging problem of minimizing an average of *pairwise* functions of the agents' data:

$$\min_{\boldsymbol{\theta}} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j). \tag{2.14}$$

To do so, we use an approach similar to the estimation case, combining local computations and data propagation over the network. Auxiliary observations will allow us to compute — biased — gradient estimates.

2.4.1 Problem Statement

Let $\mathcal{X}$ be some parameter space, $d > 0$ and let $f : \mathbb{R}^d \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a differentiable and convex function with respect to the first variable. We assume that for any $(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2$, there exists $L_f > 0$ such that $f(\cdot\,; \mathbf{x}, \mathbf{x}')$ is L_f-Lipschitz (with respect to the Euclidean norm $\|\cdot\|$). Let $\psi : \mathbb{R}^d \to \mathbb{R}^+$ be a convex function (possibly non-smooth) such that, for simplicity, $\psi(0) = 0$. Given a data sample $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{X}^n$, we aim at solving the following optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) + \psi(\boldsymbol{\theta}). \tag{2.15}$$

In a typical machine learning scenario, Problem (2.15) is a (regularized) empirical risk minimization problem and $\boldsymbol{\theta}$ corresponds to the model parameters to be learned. In such a context, the function $f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j)$ is a pairwise loss measuring the performance of the model $\boldsymbol{\theta}$ on the data pair $(\mathbf{x}_i, \mathbf{x}_j)$, while $\psi(\boldsymbol{\theta})$ represents a regularization term penalizing the complexity of $\boldsymbol{\theta}$. Common examples of regularization terms include indicator functions of a closed convex set to model explicit convex constraints, or norms enforcing specific properties such as sparsity (a canonical example being the $\|\cdot\|_1$ norm).

Many interesting problems can be cast as Problem (2.15). For instance, one can perform the aforementioned AUC maximization using the logistic loss

$$f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) = \mathbb{1}_{\{\ell_i > \ell_j\}} \log\left(1 + \exp((\mathbf{x}_j - \mathbf{x}_i)^\top \boldsymbol{\theta})\right),$$

and the regularization term $\psi(\boldsymbol{\theta})$ can be set to be the ℓ_2-norm of $\boldsymbol{\theta}$ (or the ℓ_1-norm when a sparse model is desirable). Other popular instances of Problem (2.15) include metric/similarity learning (BELLET, HABRARD, and SEBBAN, 2015), ranking (CLÉMENÇON, LUGOSI, and VAYATIS, 2008),

supervised graph inference (BIAU and BLEAKLEY, 2006) and multiple kernel learning (KUMAR et al., 2012).

For notational convenience, we denote by f_i the partial function associated to i, that is $f_i := (1/n)\sum_{j=1}^n f(\cdot; \mathbf{x}_i, \mathbf{x}_j)$ and $f = (1/n)\sum_{i=1}^n f_i$. Problem (5.3) can then be reformulated as follows:

$$\min_{\theta \in \mathbb{R}^d} R_n(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \psi(\boldsymbol{\theta}). \tag{2.16}$$

Note that the function f is L_f-Lipschitz, since all the f_i are L_f-Lipschitz.

2.4.2 Pairwise gossip dual averaging

In our methods, we use the dual averaging algorithm for local updates. The dual averaging algorithm (NESTEROV, 2009) maintains a sequence of primal iterates $(\boldsymbol{\theta}(t))_{t>0}$, and a sequence $(\mathbf{z}(t))_{t\geq 0}$ of dual variables which collects the sum of (sub-)gradients seen up to time t. At each step $t > 0$, the dual variable $\mathbf{z}$ is updated as follows:

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \nabla f(\boldsymbol{\theta}(t))$$

and the primal variable is generated with the following rule:

$$\boldsymbol{\theta}(t+1) = \pi_t(\mathbf{z}(t+1)) := \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ -\mathbf{z}^\top \boldsymbol{\theta} + \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t)} \right\},$$

for some step size sequence $(\gamma(t))_{t\geq 0}$. This choice is guided by the fact that the structure of the updates makes dual averaging much easier to analyze in the distributed setting than stochastic gradient descent when the problem is constrained or regularized. This is because dual averaging maintains a simple sum of sub-gradients, while the (non-linear) projection operator is applied separately — see Section 5.3 for details about dual averaging.

Our work builds upon the analysis introduced in DUCHI, AGARWAL, and WAINWRIGHT, 2012, where a distributed version of the dual averaging algorithm is proposed. It aims at optimizing an average of *univariate* functions $f(\cdot\,; \mathbf{x}_i)$ in which each node i computes *unbiased* estimates of $\nabla f(\cdot\,; \mathbf{x}_i)$ that are iteratively averaged over the network. However, in our setting we cannot compute unbiased estimates of $\nabla f(\cdot; \mathbf{x}_i, \mathbf{x}_j)$, even when using our gossip data propagation step, as in GOSTA; instead, we use the auxiliary observation to compute *biased* estimates. By relying on the key observation that the bias contribution decreases exponentially fast with the number of iterations, we show that the convergence of dual averaging is preserved. We first present and analyze our algorithm in the synchronous setting, then turn to the more intricate analysis of the asynchronous setting.

Synchronous setting

In the synchronous setting, every node has access to a global clock, so every node will perform a local update (a dual averaging step) at each iteration. The communication step combines an averaging of the selected nodes' dual variables and an observation swap similar to GOSTA. The procedure is detailed in Algorithm 7, and the following theorem establishes its convergence rate.

Algorithm 7 Gossip dual averaging for pairwise function in synchronous setting

Require: Step size $(\gamma(t))_{t\geq 1} > 0$.
Each node i initializes $\mathbf{y}_i = \mathbf{x}_i$, $\mathbf{z}_i = \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}}_i = 0$.
for $t = 1, \dots, T$ **do**
  Draw (i, j) uniformly at random from $\mathcal{E}$
  Set $\mathbf{z}_i, \mathbf{z}_j \leftarrow \frac{\mathbf{z}_i + \mathbf{z}_j}{2}$
  Swap auxiliary observations: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
  for $k = 1, \dots, n$ **do**
    Update $\mathbf{z}_k \leftarrow \mathbf{z}_k + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k; \mathbf{x}_k, \mathbf{y}_k)$
    Compute $\boldsymbol{\theta}_k \leftarrow \pi_t(\mathbf{z}_k)$
    Average $\bar{\boldsymbol{\theta}}_k \leftarrow \left(1 - \frac{1}{t}\right) \bar{\boldsymbol{\theta}}_k + \frac{1}{t} \boldsymbol{\theta}_k$
  end for
end for
return Each node k has $\bar{\boldsymbol{\theta}}_k$, for $k = 1, \dots, n$

Theorem 9. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non-bipartite graph, and let $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} R_n(\boldsymbol{\theta})$. Let $(\gamma(t))_{t\geq 1}$ be a non-increasing and non-negative sequence. For any $i \in [n]$ and any $t \geq 0$, let $\mathbf{z}_i(t) \in \mathbb{R}^d$ and $\bar{\boldsymbol{\theta}}_i(t) \in \mathbb{R}^d$ be generated according to Algorithm 7. Then for any $i \in [n]$ and $T > 1$, we have:*

$$\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i) - R_n(\boldsymbol{\theta}^*)] \leq C_1(T) + C_2(T) + C_3(T),$$

where

$$\begin{cases} C_1(T) = \dfrac{1}{2T\gamma(T)} \|\boldsymbol{\theta}^*\|^2 + \dfrac{L_f^2}{2T} \displaystyle\sum_{t=1}^{T-1} \gamma(t), \\ C_2(T) = \dfrac{3L_f^2}{T\left(1 - \sqrt{\lambda_2}\right)} \displaystyle\sum_{t=1}^{T-1} \gamma(t), \\ C_3(T) = \dfrac{1}{T} \displaystyle\sum_{t=1}^{T-1} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)], \end{cases}$$

and $1 - \lambda_2 = \beta_{n-1}/|\mathcal{E}| > 0$ and β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}^{\mathcal{G}}$.

The rate of convergence is divided into three parts: $C_1(T)$ is a *data dependent* term which corresponds to the rate of convergence of the centralized dual averaging, while $C_2(T)$ is a *network dependent* term depending on the spectral gap β_{n-1} of $\mathcal{G}$. $C_3(T)$ depends on the bias of the gradient estimates $\bar{\boldsymbol{\epsilon}}$ which we expect to vanish quickly: the propagation scheme is a random walk, so the auxiliary observations distribution tends towards a uniform at an exponential rate. As in the estimation case, the spectral gap of the network $\mathcal{G}$ is key for establishing the error bound of our algorithm.

The upper-bound established in Theorem 9 does not ensure the convergence of our method: the bias term, although bounded, is not guaranteed to vanish without further analysis. Extending the ergodic analysis of the mirror descent provided in DUCHI et al., 2012, we studied the dual averaging with biased gradient estimates to prove the convergence of this algorithm, with little impact in comparison to the unbiased case.

Asynchronous Setting

Algorithm 8 Gossip dual averaging for pairwise function in asynchronous setting

Require: Step size $(\gamma(t))_{t\geq 0} > 0$, probabilities $(p_k)_{k\in[n]}$.
Each node i initializes $\mathbf{y}_i = \mathbf{x}_i$, $\mathbf{z}_i = \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}}_i = 0$, $m_i = 0$.
for $t = 1, \dots, T$ **do**
  Draw (i, j) uniformly at random from E
  Swap auxiliary observations: $y_i \leftrightarrow y_j$
  for $k \in \{i, j\}$ **do**
    Set $\mathbf{z}_k \leftarrow \frac{\mathbf{z}_i + \mathbf{z}_j}{2}$
    Update $\mathbf{z}_k \leftarrow \frac{1}{p_k} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k; \mathbf{x}_k, \mathbf{y}_k)$
    Increment $m_k \leftarrow m_k + \frac{1}{p_k}$
    Compute $\boldsymbol{\theta}_k \leftarrow \pi_{m_k}(\mathbf{z}_k)$
    Average $\bar{\boldsymbol{\theta}}_k \leftarrow \left(1 - \frac{1}{m_k p_k}\right) \bar{\boldsymbol{\theta}}_k$
  end for
end for
return Each node k has $\bar{\boldsymbol{\theta}}_k$

Using the time estimator of Section 2.2.2, we can now adapt Algorithm 7 to the fully asynchronous case, as shown in Algorithm 8. Similarly to the estimation case, the updates need to be weighted according to the activation probabilities.The following result is the analogous of Theorem 9 for the asynchronous setting.

Theorem 10. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph. Let $(\gamma(t))_{t\geq 1}$ be defined as $\gamma(t) = c/t^{1/2+\alpha}$ for some constant $c > 0$ and $\alpha \in (0, 1/2)$. For $i \in [n]$, let $(\mathbf{d}_i(t))_{t\geq 1}$, $(\mathbf{g}_i(t))_{t\geq 1}$, $(\boldsymbol{\epsilon}_i(t))_{t\geq 1}$, $(\mathbf{z}_i(t))_{t\geq 1}$ and $(\boldsymbol{\theta}_i(t))_{t\geq 1}$ be generated as described in Algorithm 8. Then, there exists some constant $C < +\infty$ such that, for $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}' \in \mathbb{R}^d} R_n(\boldsymbol{\theta}')$, $i \in [n]$ and $T > 0$,*

$$R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*) \leq C \max(T^{-\alpha/2}, T^{\alpha-1/2}) + \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)].$$

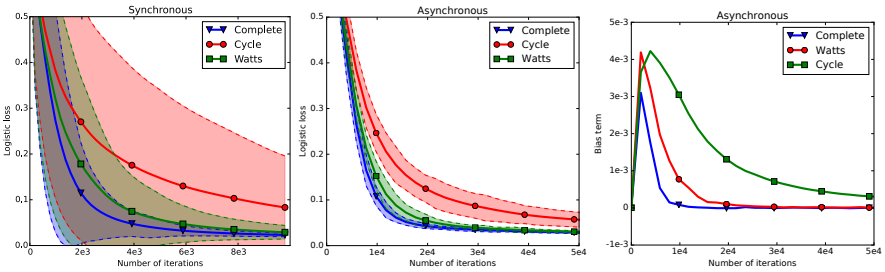
In the asynchronous setting, no convergence rate was known even for the distributed dual averaging algorithm of DUCHI, AGARWAL, and WAINWRIGHT (2012), which deals with the simpler problem of minimizing *univariate* functions. The arguments used to derive Theorem 10 can be adapted to derive a convergence rate (without the bias term) for an asynchronous version of their algorithm.

2.4.3 Numerical experiments

To study the influence of the network topology, we perform our simulations on three types of network : complete graph, Watts-Strogatz graph and cycle graph.

We initialized each $\boldsymbol{\theta}_i$ to 0 and for each network, we ran 50 times Algorithms 7 and 8 with $\gamma(t) = 1/\sqrt{t}$.[4] Figure 2.7a shows the evolution of the objective function and the associated standard deviation (across nodes) with the number of iterations in the synchronous setting. As expected, the

4. Even if this scaling sequence does not fulfill the hypothesis of Theorem 24 for the asynchronous setting, the convergence rate is acceptable in practice.



(A) Synchronous. (B) Asynchronous. (C) Bias term (asynchronous).

FIGURE 2.7: AUC maximization.

average convergence rate on the complete and the Watts-Strogatz networks is much better than on the poorly connected cycle network. The standard deviation of the node estimates also decreases with the connectivity of the network.

The results for the asynchronous setting are shown in Figure 2.7b. As expected, the convergence rate is slower in terms of number of iterations (roughly 5 times) than in the synchronous setting. Note however that much fewer dual averaging steps are performed: for instance, on the network generated with Watts-Strogatz method, reaching a 0.1 loss requires $210,000$ (partial) gradient computations in the synchronous setting and only $25,000$ in the asynchronous setting. Moreover, the standard deviation of the estimates is much lower than in the synchronous setting. This is because communication and local optimization are better balanced in the asynchronous setting (one optimization step for each gradient accumulator averaged) than in the synchronous setting (n optimization steps for 2 gradient accumulators averaged).

The good practical convergence of our algorithm comes from the fact that the bias term $\bar{\epsilon}(t)^\top \boldsymbol{\omega}(t)$ vanishes quite fast. Figure 2.7c shows that its average value quickly converges to 0 on all networks. Moreover, its order of magnitude is negligible compared to the objective function.

2.5 Conclusion

We provided theoretical guarantees for different sampling schemes for U-statistic-based ERM, as well as strong numerical results. We also provided methods for estimating and optimizing pairwise functions in a gossip setting, again with both theoretical guarantees and experimental results.

Chapter 3

Scaling-up Empirical Risk Minimization: Optimization of incomplete U-statistics

3.1 Introduction

In classification/regression, empirical risk estimates are sample mean statistics and the theory of *Empirical Risk Minimization* (ERM) has been originally developed in this context, see DEVROYE, GYÖRFI, and LUGOSI, 1996. The ERM theory essentially relies on the study of maximal deviations between these empirical averages and their expectations, under adequate complexity assumptions on the set of prediction rule candidates. The relevant tools are mainly concentration inequalities for empirical processes, see LEDOUX and TALAGRAND, 1991 for instance.

In a wide variety of problems that received a good deal of attention in the machine learning literature and ranging from clustering to image recognition through ranking or learning on graphs, natural estimates of the risk are not basic sample means but take the form of averages of d-tuples, usually referred to as U-statistics in Probability and Statistics, see LEE, 1990a. In CLÉMENÇON, LUGOSI, and VAYATIS, 2005 for instance, ranking is viewed as pairwise classification and the empirical ranking error of any given prediction rule is a U-statistic of order 2, just like the *within cluster point scatter* in cluster analysis (see CLÉMENÇON, 2014) or empirical performance measures in metric learning, refer to CAO, GUO, and YING, 2012 for instance. Because empirical functionals are computed by averaging over tuples of sampling observations, they exhibit a complex dependence structure, which appears as the price to be paid for low variance estimates. *Linearization techniques* (see HOEFFDING, 1948) are the main ingredient in studying the behavior of empirical risk minimizers in this setting, allowing to establish probabilistic upper bounds for the maximal deviation of collection of centered U-statistics under appropriate conditions by reducing the analysis to that of standard empirical processes. However, while the ERM theory based on minimization of U-statistics is now consolidated (see CLÉMENÇON, LUGOSI, and VAYATIS, 2008), putting this approach in practice generally leads to significant computational difficulties that are not sufficiently well documented in the machine learning literature. In many concrete cases, the mere computation of the risk involves a summation over an extremely high number of tuples and runs out of time or memory on most machines.

Whereas the availability of massive information in the Big Data era, which machine learning procedures could theoretically now rely on, has motivated the recent development of *parallelized / distributed* approaches in order to scale-up certain statistical learning algorithms, see BEKKERMAN, BILENKO, and LANGFORD, 2011 or BIANCHI et al., 2013 and the references therein, the present paper proposes to use *sampling techniques* as a remedy to the apparent intractability of learning from data sets of explosive size, in order to break the current computational barriers. More precisely, it is the major goal of this article to study how a simplistic sampling technique (*i.e.* drawing with replacement) applied to risk estimation, as originally proposed by BLOM, 1976 in the context of asymptotic pointwise estimation, may efficiently remedy this issue without damaging too much the "reduced variance" property of the estimates, while preserving the learning rates (including certain "fast-rate" situations). For this purpose, we investigate to which extent a U-process, that is a collection of U-statistics, can be accurately approximated by a Monte-Carlo version (which shall be referred to

as an *incomplete U-process* throughout the paper) involving much less terms, provided it is indexed by a class of kernels of controlled complexity (in a sense that will be explained later). A maximal deviation inequality connecting the accuracy of the approximation to the number of terms involved in the approximant is thus established. This result is the key to the analysis of the statistical performance of minimizers of risk estimates when they are in the form of an incomplete U-statistic. In particular, this allows us to show the advantage of using this specific sampling technique, compared to more naive approaches with exactly the same computational cost, consisting for instance in first drawing a subsample and then computing a risk estimate of the form of a (complete) U-statistic based on it. We also show how to incorporate this sampling strategy into iterative statistical learning techniques based on stochastic gradient descent (SGD), see BOTTOU, 1998. The variant of the SGD method we propose involves the computation of an incomplete U-statistic to estimate the gradient at each step. For the estimator thus produced, rate bounds describing its statistical performance are established under mild assumptions. Beyond theoretical results, we present illustrative numerical experiments on metric learning and clustering with synthetic and real-world data that support the relevance of our approach.

The rest of the chapter is organized as follows. In Section 3.2, we recall basic definitions and concepts pertaining to the theory of U-statistics and U-processes and present important examples in machine learning where natural estimates of the performance/risk measure are U-statistics. We then review the existing results for the empirical minimization of complete U-statistics. In Section 3.3, we recall the notion of incomplete U-statistic and we derive maximal deviation inequalities describing the error made when approximating a U-statistic by its incomplete counterpart uniformly over a class of kernels that fulfills appropriate complexity assumptions. This result is next applied to derive (possibly fast) learning rates for minimizers of the incomplete version of the empirical risk and to model selection. Extensions to incomplete U-statistics built by means of other sampling schemes than sampling with replacement are also investigated. In Section 3.4, estimation by means of incomplete U-statistics is applied to stochastic gradient descent for iterative ERM. Section 3.5 presents some numerical experiments. Finally, Section 3.6 collects some concluding remarks. Technical details are deferred to the Appendix.

3.2 Background and Preliminaries

As a first go, we briefly recall some key notions of the theory of U-statistics (Section 3.2.1) and provide several examples of statistical learning problems for which natural estimates of the performance/risk measure are in the form of U-statistics (Section 3.2.2). Finally, we review and extend the existing rate bound analysis for the empirical minimization of (complete) generalized U-statistics (Section 3.2.3). Here and throughout, $\mathbb{N}^*$ denotes the set of all strictly positive integers, $\mathbb{R}_+$ the set of nonnegative real numbers.

3.2.1 U-Statistics/Processes: Definitions and Properties

For clarity, we recall the definition of generalized U-statistics. An excellent account of properties and asymptotic theory of U-statistics can be found in LEE, 1990a.

Definition 1. (GENERALIZED U-STATISTIC) Let $K \geq 1$ and $(d_1, \ldots, d_K) \in \mathbb{N}^{*K}$. Let $\mathbf{X}_{\{1,\ldots,n_k\}} = (X_1^{(k)}, \ldots, X_{n_k}^{(k)})$, $1 \leq k \leq K$, be K independent samples of sizes $n_k \geq d_k$ and composed of i.i.d. random variables taking their values in some measurable space $\mathcal{X}_k$ with distribution $F_k(dx)$ respectively. Let $H : \mathcal{X}_1^{d_1} \times \cdots \times \mathcal{X}_K^{d_K} \to \mathbb{R}$ be a measurable function, square integrable with respect to the probability distribution $\mu = F_1^{\otimes d_1} \otimes \cdots \otimes F_K^{\otimes d_K}$. Assume in addition (without loss of generality) that $H(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)})$ is symmetric within each block of arguments $\mathbf{x}^{(k)}$ (valued in $\mathcal{X}_k^{d_k}$), $1 \leq k \leq K$. The generalized (or K-sample) U-statistic of degrees $(d_1, \ldots, d_K)$ with kernel H, is then defined as

$$U_{\mathbf{n}}(H) = \frac{1}{\prod_{k=1}^K \binom{n_k}{d_k}} \sum_{I_1} \cdots \sum_{I_K} H(\mathbf{X}_{I_1}^{(1)}, \mathbf{X}_{I_2}^{(2)}, \ldots, \mathbf{X}_{I_K}^{(K)}), \tag{3.1}$$

where the symbol $\sum_{I_k}$ refers to summation over all $\binom{n_k}{d_k}$ subsets $\mathbf{X}_{I_k}^{(k)} = (X_{i_1}^{(k)}, \ldots, X_{i_{d_k}}^{(k)})$ related to a set I_k of d_k indexes $1 \leq i_1 < \ldots < i_{d_k} \leq n_k$ and $\mathbf{n} = (n_1, \ldots, n_K)$.

The above definition generalizes standard sample mean statistics, which correspond to the case $K = 1 = d_1$. More generally when $K = 1$, $U_{\mathbf{n}}(H)$ is an average over all d_1-tuples of observations, while $K \geq 2$ corresponds to the multi-sample situation with a d_k-tuple for each sample $k \in \{1, \ldots, K\}$. A U-process is defined as a collection of U-statistics indexed by a set $\mathcal{H}$ of kernels. This concept generalizes the notion of empirical process.

Many statistics used for pointwise estimation or hypothesis testing are actually generalized U-statistics (*e.g.* the sample variance, the Gini mean difference, the Wilcoxon Mann-Whitney statistic, Kendall tau). Their popularity mainly arises from their "reduced variance" property: the statistic

$U_{\mathbf{n}}(H)$ has minimum variance among all unbiased estimators of the parameter

$$\begin{aligned}\mu(H) &= \mathbb{E}\left[H(X_1^{(1)},\ \ldots,\ X_{d_1}^{(1)},\ \ldots,\ X_1^{(K)},\ \ldots,\ X_{d_K}^{(K)})\right] \\ &= \int_{\mathbf{x}^{(1)}\in\mathcal{X}_1^{d_1}} \cdots \int_{\mathbf{x}^{(K)}\in\mathcal{X}_K^{d_K}} H(\mathbf{x}^{(1)},\ \ldots,\ \mathbf{x}^{(K)})\mathrm{d}\mu(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(K)}) \\ &= \mathbb{E}\left[U_{\mathbf{n}}(H)\right].\end{aligned} \tag{3.2}$$

Classically, the limit properties of these statistics (law of large numbers, central limit theorem, *etc.*) are investigated in an asymptotic framework stipulating that, as the size of the full pooled sample

$$n \overset{\text{def}}{=} n_1 + \ \ldots + n_K \tag{3.3}$$

tends to infinity, we have:

$$n_k/n \to \lambda_k > 0 \text{ for } k = 1,\ \ldots,\ K. \tag{3.4}$$

Asymptotic results and deviation/moment inequalities for K-sample U-statistics can be classically established by means of specific representations of this class of functionals, see (3.15) and (3.26) introduced in later sections. Significant progress in the analysis of U-statistics and U-processes has then recently been achieved by means of decoupling theory, see PEÑA and GINÉ, 1999. For completeness, we point out that the asymptotic behavior of (multisample) U-statistics has been investigated under weaker integrability assumptions than that stipulated in Definition 1, see LEE, 1990a.

3.2.2 Motivating Examples

In this section, we review important supervised and unsupervised statistical learning problems where the empirical performance/risk measure is of the form of a generalized U-statistics. They shall serve as running examples throughout the paper.

Clustering

Clustering refers to the unsupervised learning task that consists in partitioning a set of data points $X_1,\ \ldots,\ X_n$ in a feature space $\mathcal{X}$ into a finite collection of subgroups depending on their similarity (in a sense that must be specified): roughly, data points in the same subgroup should be more similar to each other than to those lying in other subgroups. One may refer to Chapter 14 in FRIEDMAN, HASTIE, and TIBSHIRANI, 2009 for an account of state-of-the-art clustering techniques. Formally, let $M \geq 2$ be the number of desired clusters and consider a symmetric function $D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ such that $D(x, x) = 0$ for any $x \in \mathcal{X}$. D measures the dissimilarity between pairs of observations $(x, x') \in \mathcal{X}^2$: the larger $D(x, x')$, the less similar x and x'. For instance, if $\mathcal{X} \subset \mathbb{R}^d$, D could take the form $D(x, x') = \Psi(\|x - x'\|_q)$, where $q \geq 1$, $||a||_q = (\sum_{i=1}^d |a_i|^q)^{1/q}$ for all $a \in \mathbb{R}^d$ and $\Psi : \mathbb{R}_+ \to \mathbb{R}_+$ is any borelian nondecreasing function such that $\Psi(0) = 0$. In this context, the goal of clustering methods is to find a partition $\mathcal{P}$ of the feature space $\mathcal{X}$ in a class Π of partition candidates that minimizes the following *empirical*

clustering risk:

$$\widehat{W}_n(\mathcal{P}) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} D(X_i, X_j) \cdot \Phi_{\mathcal{P}}(X_i, X_j), \tag{3.5}$$

where $\Phi_{\mathcal{P}}(x, x') = \sum_{\mathcal{C} \in \mathcal{P}} \mathbb{I}\{(x, x') \in \mathcal{C}^2\}$. Assuming that the data $X_1, \ldots, X_n$ are i.i.d. realizations of a generic random variable X drawn from an unknown probability distribution $F(dx)$ on $\mathcal{X}$, the quantity $\widehat{W}_n(\mathcal{P})$, also known as the *intra-cluster similarity* or *within cluster point scatter*, is a one sample U-statistic of degree two ($K = 1$ and $d_1 = 2$) with kernel given by:

$$\forall (x, x') \in \mathcal{X}^2, \quad H_{\mathcal{P}}(x, x') = D(x, x') \cdot \Phi_{\mathcal{P}}(x, x'), \tag{3.6}$$

according to Definition 1 provided that the following condition holds:

$$\int \int_{(x,x') \in \mathcal{X}^2} D^2(x, x') \cdot \Phi_{\mathcal{P}}(x, x') F(dx) F(dx') < +\infty.$$

The expectation of the empirical clustering risk $\widehat{W}_n(\mathcal{P})$ is given by

$$W(\mathcal{P}) = \mathbb{E}\left[D(X, X') \cdot \Phi_{\mathcal{P}}(X, X')\right], \tag{3.7}$$

where X' is an independent copy of the r.v. X, and is named the *clustering risk* of the partition $\mathcal{P}$. The statistical analysis of the clustering performance of minimizers $\widehat{\mathcal{P}}_n$ of the empirical risk (3.5) over a class Π of appropriate complexity can be found in CLÉMENÇON, 2014. Based on the theory of U-processes, it is shown in particular how to establish rate bounds for the excess of clustering risk of any empirical minimizer, $W(\widehat{\mathcal{P}}_n) - \inf_{\mathcal{P} \in \Pi} W(\mathcal{P})$ namely, under appropriate complexity assumptions on the cells forming the partition candidates.

Metric Learning

Many problems in machine learning, data mining and pattern recognition (such as the clustering problem described above) rely on a metric to measure the distance between data points. Choosing an appropriate metric for the problem at hand is crucial to the performance of these methods. Motivated by a variety of applications ranging from computer vision to information retrieval through bioinformatics, metric learning aims at adapting the metric to the data and has attracted a lot of interest in recent years (see for instance BELLET, HABRARD, and SEBBAN, 2013, for an account of metric learning and its applications). As an illustration, we consider the metric learning problem for supervised classification. In this setting, we observe independent copies $(X_1, Y_1), \ldots, (X_n, Y_n)$ of a random couple (X, Y), where the r.v. X takes values in some feature space $\mathcal{X}$ and Y in a finite set of labels, $\mathcal{Y} = \{1, \ldots, C\}$ with $C \geq 2$ say. Consider a set $\mathcal{D}$ of distance measures $D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. Roughly speaking, the goal of metric learning in this context is to find a metric under which pairs of points with the same label are close to each other and those with different labels are far away. The risk of a metric D can be expressed as:

$$R(D) = \mathbb{E}\left[\phi\left((1 - D(X, X') \cdot (2\mathbb{I}\{Y = Y'\} - 1)\right)\right], \tag{3.8}$$

where $\phi(u)$ is a convex loss function upper bounding the indicator function $\mathbb{I}\{u \geq 0\}$, such as the hinge loss $\phi(u) = \max(0, 1-u)$. The natural empirical estimator of this risk is

$$R_n(D) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \phi\left((D(X_i, X_j) - 1) \cdot (2\mathbb{I}\{Y_i = Y_j\} - 1)\right), \quad (3.9)$$

which is a one sample U-statistic of degree two with kernel given by:

$$H_D\left((x, y), (x', y)\right) = \phi\left((D(x, x') - 1) \cdot (2\mathbb{I}\{y = y'\} - 1)\right). \quad (3.10)$$

The convergence to (3.8) of a minimizer of (3.9) has been studied in the frameworks of algorithmic stability (JIN, WANG, and ZHOU, 2009), algorithmic robustness (BELLET and HABRARD, 2015) and based on the theory of U-processes under appropriate regularization (CAO, GUO, and YING, 2012).

Multipartite Ranking

Given objects described by a random vector of attributes/features $X \in \mathcal{X}$ and the (temporarily hidden) ordinal labels $Y \in \{1, \ldots, K\}$ assigned to it, the goal of *multipartite ranking* is to rank them in the same order as that induced by the labels, on the basis of a training set of labeled examples. This statistical learning problem finds many applications in a wide range of fields (*e.g.* medicine, finance, search engines, e-commerce). Rankings are generally defined by means of a scoring function $s : \mathcal{X} \to \mathbb{R}$, transporting the natural order on the real line onto the feature space and the gold standard for evaluating the ranking performance of $s(x)$ is the ROC manifold, or its usual summary the VUS criterion (VUS standing for *Volume Under the* ROC *Surface*), see CLÉMENÇON and ROBBIANO, 2014 and the references therein. In CLÉMENÇON, ROBBIANO, and VAYATIS, 2013, optimal scoring functions have been characterized as those that are optimal for all bipartite subproblems. In other words, they are increasing transforms of the likelihood ratio dF_{k+1}/dF_k, where F_k denotes the class-conditional distribution for the k-th class. When the set of optimal scoring functions is non-empty, the authors also showed that it corresponds to the functions which maximize the volume under the ROC surface

$$VUS(s) = \mathbb{P}\{s(X_1) < \ldots < s(X_K) | Y_1 = 1, \ldots, Y_K = K\}.$$

Given K independent samples $(X_1^{(k)}, \ldots, X_{n_k}^{(k)}) \overset{\text{i.i.d.}}{\sim} F_k(dx)$ for $1 \leq k \leq K$, the empirical counterpart of the VUS can be written in the following way:

$$\widehat{VUS}(s) = \frac{1}{\prod_{k=1}^K n_k} \sum_{i_1=1}^{n_1} \cdots \sum_{i_K=1}^{n_K} \mathbb{I}\{s(X_{i_1}^{(1)}) < \ldots < s(X_{i_K}^{(K)})\}. \quad (3.11)$$

The empirical VUS (3.11) is a K-sample U-statistic of degree $(1, \ldots, 1)$ with kernel given by:

$$H_s(x_1, \ldots, x_K) = \mathbb{I}\{s(x_1) < \ldots < s(x_K)\}. \quad (3.12)$$

3.2.3 Empirical Minimization of U-Statistics

As illustrated by the examples above, many learning problems can be formulated as finding a certain rule g in a class $\mathcal{G}$ in order to minimize a risk of the same form as (3.2), $\mu(H_g)$, with kernel $H = H_g$. Based on $K \geq 1$ independent i.i.d. samples

$$\mathbf{X}^{(k)}_{\{1,\,\ldots,\,n_k\}} = (X_1^{(k)},\,\ldots,\,X_{n_k}^{(k)}) \text{ with } 1 \leq k \leq K,$$

the ERM paradigm in statistical learning suggests to replace the risk by the U-statistic estimation $U_{\mathbf{n}}(H_g)$ in the minimization problem. The study of the performance of minimizers $\widehat{g}_{\mathbf{n}}$ of the empirical estimate $U_{\mathbf{n}}(H_g)$ over the class $\mathcal{G}$ of rule candidates naturally leads to analyze the fluctuations of the U-process

$$\{U_{\mathbf{n}}(H_g) - \mu(H_g) :\ g \in \mathcal{G}\}. \tag{3.13}$$

Given the bound

$$\mu(H_{\widehat{g}_{\mathbf{n}}}) - \inf_{g\in\mathcal{G}} \mu(H_g) \leq 2 \sup_{g\in\mathcal{G}} |U_{\mathbf{n}}(H_g) - \mu(H_g)|, \tag{3.14}$$

a probabilistic control of the maximal deviation $\sup_{g\in\mathcal{G}} |U_{\mathbf{n}}(H_g) - \mu(H_g)|$ naturally provides statistical guarantees for the generalization ability of the empirical minimizer $\widehat{g}_{\mathbf{n}}$. As shown at length in the case $K = 1$ and $d_1 = 2$ in CLÉMENÇON, LUGOSI, and VAYATIS, 2008 and in CLÉMENÇON, 2014 for specific problems, this can be achieved under adequate complexity assumptions of the class $\mathcal{H}_{\mathcal{G}} = \{H_g :\ g \in \mathcal{G}\}$. These results rely on the *Hoeffding's representation* of U-statistics, which we recall now for clarity in the general multisample U-statistics setting. Denote by $\mathfrak{S}_m$ the symmetric group of order m for any $m \geq 1$ and by $\sigma(i)$ the i-th coordinate of any permutation $\sigma \in \mathfrak{S}_m$ for $1 \leq i \leq m$. Let $\lfloor z \rfloor$ be the integer part of any real number z and set

$$N = \min\left\{\lfloor n_1/d_1 \rfloor,\,\ldots,\,\lfloor n_K/d_K \rfloor\right\}.$$

Observe that the K-sample U-statistic (3.1) can be expressed as

$$U_{\mathbf{n}}(H) = \frac{1}{\prod_{k=1}^K n_k!} \sum_{\sigma_1\in\mathfrak{S}_{n_1}} \cdots \sum_{\sigma_K\in\mathfrak{S}_{n_K}} V_H\left(X^{(1)}_{\sigma_1(1)},\,\ldots,\,X^{(K)}_{\sigma_K(n_K)}\right), \tag{3.15}$$

where

$$\begin{aligned}
&V_H\left(X_1^{(1)},\,\ldots,\,X_{n_1}^{(1)},\,\ldots,\,X_1^{(K)},\,\ldots,\,X_{n_K}^{(K)}\right) \\
&\quad= \frac{1}{N}\Big[H\left(X_1^{(1)},\,\ldots,\,X_{d_1}^{(1)},\,\ldots,\,X_1^{(K)},\,\ldots,\,X_{d_K}^{(K)}\right) \\
&\qquad + H\left(X_{d_1+1}^{(1)},\,\ldots,\,X_{2d_1}^{(1)},\,\ldots,\,X_{d_K+1}^{(K)},\,\ldots,\,X_{2d_K}^{(K)}\right) \\
&\qquad + \ldots \\
&\qquad + H\left(X_{(N-1)d_1+1}^{(1)},\,\ldots,\,X_{Nd_1}^{(1)},\,\ldots,\,X_{(N-1)d_K+1}^{(K)},\,\ldots,\,X_{Nd_K}^{(K)}\right)\Big].
\end{aligned}$$

This representation, sometimes referred to as the *first Hoeffding's decomposition* (see HOEFFDING, 1948), allows to reduce a first order analysis to the

case of sums of i.i.d. random variables. The following result extends Corollary 3 in CLÉMENÇON, LUGOSI, and VAYATIS, 2008 to the multisample situation.

Proposition 2. *Let $\mathcal{H}$ be a collection of bounded symmetric kernels on $\prod_{k=1}^{K} \mathcal{X}_k^{d_k}$ such that*

$$\mathcal{M}_{\mathcal{H}} \stackrel{def}{=} \sup_{(H,x)\in\mathcal{H}\times\mathcal{X}} |H(x)| < +\infty. \tag{3.16}$$

Suppose also that $\mathcal{H}$ is a Vapnik-Chervonenkis major class of functions with finite VC *dimension $V < +\infty$. For all $\delta \in (0,1)$, we have with probability at least $1-\delta$,*

$$\sup_{H\in\mathcal{H}} |U_{\mathbf{n}}(H) - \mu(H)| \leq \mathcal{M}_{\mathcal{H}} \left\{ 2\sqrt{\frac{2V\log(1+N)}{N}} + \sqrt{\frac{\log(1/\delta)}{N}} \right\}, \tag{3.17}$$

where $N = \min\{\lfloor n_1/d_1 \rfloor, \ldots, \lfloor n_K/d_K \rfloor\}$.

Observe that, in the usual asymptotic framework (3.4), the bound (3.17) shows that the learning rate is, as expected, of order $O_{\mathbb{P}}(\sqrt{\log n/n})$, where n denotes the size of the pooled sample.

Remark 1. (UNIFORM BOUNDEDNESS) We point out that condition (3.16) is clearly satisfied for the class of kernels considered in the multipartite ranking situation, whatever the class of scoring functions considered. In the case of the clustering example, it is fulfilled as soon as the essential supremum of $D(X,X') \cdot \Phi_{\mathcal{P}}(X,X')$ is uniformly bounded over $\mathcal{P} \in \Pi$, whereas in the metric learning example, it is satisfied when the essential supremum of the r.v. $\phi((D(X,X')-1)\cdot(2\mathbb{I}\{Y=Y'\}-1))$ is uniformly bounded over $D \in \mathcal{D}$. We underline that this simplifying condition can be easily relaxed and replaced by appropriate tail assumptions for the variables $H(X_1^{(1)}, \ldots, X_{d_K}^{(K)})$, $H \in \mathcal{H}$, combining the arguments of the subsequent analysis with the classical "truncation trick" originally introduced in FUK and NAGAEV, 1971.

Remark 2. (COMPLEXITY ASSUMPTIONS) Following in the footsteps of CLÉMENÇON, LUGOSI, and VAYATIS, 2008 which considered 1-sample U-statistics of degree 2, define the Rademacher average

$$\mathcal{R}_N = \sup_{H\in\mathcal{H}} \frac{1}{N} \left| \sum_{l=1}^{N} \epsilon_l H\left(X_{(l-1)d_1+1}^{(1)}, \ldots, X_{ld_1}^{(1)}, \ldots, X_{(l-1)d_K+1}^{(K)}, \ldots, X_{ld_K}^{(K)} \right) \right|, \tag{3.18}$$

where $\epsilon_1, \ldots, \epsilon_N$ are independent Rademacher random variables (random symmetric sign variables), independent from the $X_i^{(k)}$'s. As can be seen by simply examining the proof of Proposition 2 (Section 3.7.1), a control of the maximal deviations similar to (3.17) relying on this particular complexity measure can be obtained: the first term on the right hand side is then replaced by the expectation of the Rademacher average $\mathbb{E}[\mathcal{R}_N]$, up to a constant multiplicative factor. This expected value can be bounded by standard metric entropy techniques and in the case where $\mathcal{H}$ is a VC major class of functions of dimension V, we have:

$$\mathbb{E}[\mathcal{R}_N] \leq \mathcal{M}_{\mathcal{H}} \sqrt{\frac{2V\log(N+1)}{N}}.$$

See Section 3.7.1 for further details.

3.3 Empirical Minimization of Incomplete U-Statistics

We have seen in the last section that empirical minimization of U-statistics leads to a learning rate of $O_{\mathbb{P}}(\sqrt{\log n/n})$. However, the computational cost required to find the empirical minimizer in practice is generally prohibitive, as the number of terms to be summed up to compute the U-statistic (3.1) is equal to:

$$\binom{n_1}{d_1} \times \cdots \times \binom{n_K}{d_K}.$$

In the usual asymptotic framework (3.4), it is of order $O(n^{d_1+\ldots+d_K})$ as $n \to +\infty$. It is the major purpose of this section to show that, in the minimization problem, the U-statistic $U_{\mathbf{n}}(H_g)$ can be replaced by a Monte-Carlo estimation, referred to as an *incomplete U-statistic*, whose computation requires to average much less terms, without damaging the learning rate (Section 3.3.1). We further extend these results to model selection (Section 3.3.2), fast rates situations (Section 3.3.3) and alternative sampling strategies (Section 3.3.4).

3.3.1 Uniform Approximation of Generalized U-Statistics

As a remedy to the computational issue mentioned above, the concept of *incomplete generalized U-statistic* has been introduced in the seminal contribution of Blom, 1976. The calculation of such a functional involves a summation over low cardinality subsets of the $\binom{n_k}{d_k}$ d_k-tuples of indices, $1 \leq k \leq K$, solely. In the simplest formulation, the subsets of indices are obtained by *sampling independently with replacement*, leading to the following definition.

Definition 2. (Incomplete Generalized U-statistic) Let $B \geq 1$. The incomplete version of the U-statistic (3.1) based on B terms is defined by:

$$\widetilde{U}_B(H) = \frac{1}{B} \sum_{I=(I_1,\ldots,I_K)\in\mathcal{D}_B} H(\mathbf{X}_{I_1}^{(1)}, \ldots, \mathbf{X}_{I_K}^{(K)}) = \frac{1}{B} \sum_{I\in\mathcal{D}_B} H(\mathbf{X}_I), \quad (3.19)$$

where $\mathcal{D}_B$ is a set of cardinality B built by sampling with replacement in the set

$$\Lambda = \left\{ ((i_1^{(1)},\ldots,i_{d_1}^{(1)}),\ldots,(i_1^{(K)},\ldots,i_{d_K}^{(K)})) \;:\; \begin{array}{c} 1 \leq i_1^{(k)} < \ldots < i_{d_k}^{(k)} \leq n_k \\ 1 \leq k \leq K \end{array} \right\},$$

and $\mathbf{X}_I = (\mathbf{X}_{I_1}^{(1)}, \ldots, \mathbf{X}_{I_K}^{(K)})$ for all $I = (I_1, \ldots, I_K) \in \Lambda$.

We stress that the distribution of a complete U-statistic built from subsamples of reduced sizes n_k' drawn uniformly at random is quite different from that of an incomplete U-statistic based on $B = \prod_{k=1}^K \binom{n_k'}{d_k}$ terms sampled with replacement in Λ, although they involve the summation of the same number of terms, as depicted by Fig. 3.1.

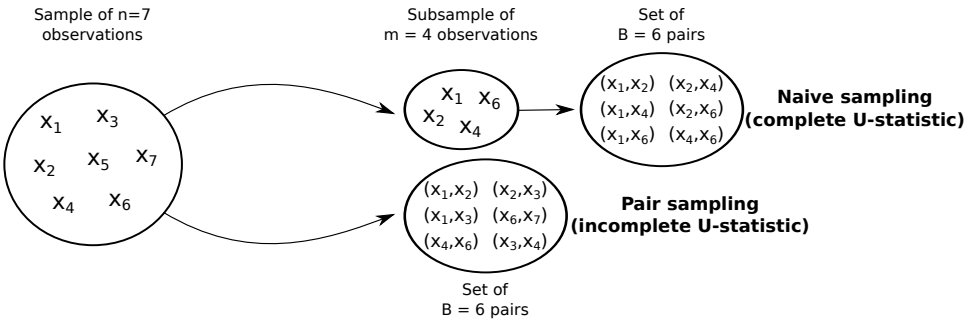


FIGURE 3.1: Illustration of the difference between an incomplete U-statistic and a complete U-statistic based on a subsample. For simplicity, we focus on the case $K = 1$ and $d_1 = 2$. In this simplistic example, a sample of $n = 7$ observations is considered. To construct a complete U-statistic of reduced complexity, we first sample a set of $m = 4$ observations and then form all possible pairs from this subsample, *i.e.* $B = m(m-1)/2 = 6$ pairs in total. In contrast, an incomplete U-statistic with the same number of terms is obtained by sampling B pairs directly from the set Λ of all possible pairs based on the original statistical population.

In practice, B should be chosen much smaller than the cardinality of Λ, namely $|\Lambda| = \prod_{k=1}^{K} \binom{n_k}{d_k}$, in order to overcome the computational issue previously mentioned. We emphasize the fact that the cost related to the computation of the value taken by the kernel H at a given point $(x_{I_1}^{(1)}, \ldots, x_{I_K}^{(K)})$ depending on the form of H is not considered here: the focus is on the number of terms involved in the summation solely. As an estimator of $\mu(H)$, the statistic (3.19) is still unbiased, *i.e.* $\mathbb{E}[\widetilde{U}_B(H)] = \mu(H)$, but its variance is naturally larger than that of the complete U-statistic $U_n(H)$. Precisely, writing the variance of the r.v. $\widetilde{U}_B(H)$ as the expectation of its conditional variance given $(\mathbf{X}_I)_{I\in\Lambda}$ plus the variance of its conditional expectation given $(\mathbf{X}_I)_{I\in\Lambda}$, we obtain

$$\mathrm{Var}(\widetilde{U}_B(H)) = \left(1 - \frac{1}{B}\right)\mathrm{Var}(U_{\mathbf{n}}(H)) + \frac{1}{B}\mathrm{Var}(H(X_1^{(1)}, \ldots, X_{d_K}^{(K)})). \quad (3.20)$$

One may easily check that $\mathrm{Var}(\widetilde{U}_B(H)) \geq \mathrm{Var}(U_{\mathbf{n}}(H))$, and the difference vanishes as B increases. Refer to LEE, 1990a for further details (see p. 193 therein). Incidentally, we underline that the empirical variance of (3.19) is not easy to compute either since it involves summing approximately $|\Lambda|$ terms and bootstrap techniques should be used for this purpose, as proposed in BERTAIL and TRESSOU, 2006. The asymptotic properties of incomplete U-statistics have been investigated in several articles, see JANSON, 1984; BROWN and KILDEA, 1978; ENQVIST, 1978. The angle embraced in the present paper is of very different nature: the key idea we promote here is to use incomplete versions of collections of U-statistics in learning problems such as that described in Section 3.2.2. The result stated below shows that this approach solves the numerical problem, while not damaging the learning rates under appropriate complexity assumptions on the collection $\mathcal{H}$ of (symmetric) kernels H considered, the complexity being described here

in terms of VC dimension for simplicity. In particular, it reveals that concentration results established for U-processes (*i.e.* collections of U-statistics) such as Proposition 2 may extend to their incomplete versions, as shown by the following theorem.

Theorem 11. (MAXIMAL DEVIATION) *Let $\mathcal{H}$ be a collection of bounded symmetric kernels on $\prod_{k=1}^{K} \mathcal{X}_k^{d_k}$ that fulfills the assumptions of Proposition 2. Then, the following assertions hold true.*

(i) For all $\delta \in (0,1)$, with probability at least $1-\delta$, we have: for all $B \geq 1$ and for all $\mathbf{n} = (n_1, \ldots, n_K) \in \mathbb{N}^{*K}$,

$$\sup_{H \in \mathcal{H}} \left| \widetilde{U}_B(H) - U_{\mathbf{n}}(H) \right| \leq \mathcal{M}_{\mathcal{H}} \times \sqrt{2 \frac{V \log(1+|\Lambda|) + \log(2/\delta)}{B}}$$

(ii) For all $\delta \in (0,1)$, with probability at least $1-\delta$, we have: $\forall \mathbf{n} \in \mathbb{N}^{*K}$, $\forall B \geq 1$,

$$\frac{1}{\mathcal{M}_{\mathcal{H}}} \sup_{H \in \mathcal{H}} \left| \widetilde{U}_B(H) - \mu(H) \right| \leq 2\sqrt{\frac{2V \log(1+N)}{N}} + \sqrt{\frac{\log(2/\delta)}{N}} + \sqrt{2 \frac{V \log(1+|\Lambda|) + \log(4/\delta)}{B}},$$

where $N = \min\{\lfloor n_1/d_1 \rfloor, \ldots, \lfloor n_K/d_K \rfloor\}$.

Remark 3. (COMPLEXITY ASSUMPTIONS CONTINUED) We point out that a bound of the same order as that stated above can be obtained under standard metric entropy conditions by means of classical chaining arguments, or under the assumption that the Rademacher average defined by

$$\widetilde{\mathcal{R}}_B = \sup_{H \in \mathcal{H}} \frac{1}{B} \left| \sum_{b=1}^{B} \epsilon_b \left\{ \sum_{I \in \Lambda} \zeta_b(I) H(\mathbf{X}_I) \right\} \right| \tag{3.21}$$

has an expectation of the order $O(1/\sqrt{B})$. The quantity $\zeta_b(I)$ indicates whether the subset of indexes I has been picked at the b-th draw ($\zeta_b(I) = +1$) or not ($\zeta_b(I) = 0$), see the calculation at the end of Appendix 3.7.3. Equipped with this notation, notice that the ζ_b's are i.i.d. multinomial random variables such that $\sum_{I \in \Lambda} \zeta_b(I) = +1$. This assumption can be easily shown to be fulfilled in the case where $\mathcal{H}$ is a VC major class of finite VC dimension (see the proof of Theorem 11 in Appendix 3.7.2). Notice however that although the variables $\sum_{I \in \Lambda} \zeta_b(I) H(\mathbf{X}_I)$, $1 \leq b \leq B$, are conditionally i.i.d. given $(\mathbf{X}_I)_{I \in \Lambda}$, they are not independent and the quantity (3.21) cannot be related to complexity measures of the type (3.18) mentioned in Remark 2.

Remark 4. We also underline that, whereas the supremum $\sup_{H \in \mathcal{H}} |U_{\mathbf{n}}(H) - \mu(H)|$ can be proved to be of order $O_{\mathbb{P}}(1/n)$ under adequate complexity assumptions in the specific situation where $\{U_{\mathbf{n}}(H) : H \in \mathcal{H}\}$ is a collection of degenerate U-statistics (see subsection 3.3.3), bound (i) in Theorem 11 cannot be improved in the degenerate case. Observe indeed that, conditioned upon the observations $X_l^{(k)}$, the deviations of the approximation (3.19) from its mean are of order $O_{\mathbb{P}}(1/\sqrt{B})$, as a basic average of B i.i.d. terms.

From the theorem stated above, one may straightforwardly deduce a bound on the excess risk of kernels $\widehat{H}_B$ minimizing the incomplete version of the empirical risk based on B terms, *i.e.* such that

$$\widetilde{U}_B\left(\widehat{H}_B\right) = \min_{H \in \mathcal{H}} \widetilde{U}_B(H). \tag{3.22}$$

Corollary 1. *Let $\mathcal{H}$ be a collection of symmetric kernels on $\prod_{k=1}^{K} \mathcal{X}_k^{d_k}$ that satisfies the conditions stipulated in Proposition 2. Let $\delta > 0$. For any minimizer $\widehat{H}_B$ of the statistical estimate of the risk (3.19), the following assertions hold true*

(i) We have with probability at least $1-\delta$: $\forall \mathbf{n} \in \mathbb{N}^{*K}$, $\forall B \geq 1$,

$$\mu(\widehat{H}_B) - \inf_{H \in \mathcal{H}} \mu(H) \leq 2\mathcal{M}_{\mathcal{H}} \times \left\{ 2\sqrt{\frac{2V\log(1+N)}{N}} + \sqrt{\frac{\log(2/\delta)}{N}} + \sqrt{2\frac{V\log(1+|\Lambda|)+\log(4/\delta)}{B}} \right\}.$$

(ii) We have: $\forall \mathbf{n} \in \mathbb{N}^{*K}$, $\forall B \geq 1$,

$$\mathbb{E}\left[\sup_{H \in \mathcal{H}} \left|\widetilde{U}_B(H) - \mu(H)\right|\right] \leq \mathcal{M}_{\mathcal{H}} \left(2\sqrt{\frac{2V\log(1+N)}{N}} + \sqrt{\frac{2(\log 2 + V\log(1+|\Lambda|))}{B}} \right).$$

The first assertion of Theorem 11 provides a control of the deviations between the U-statistic (3.1) and its incomplete counterpart (3.19) uniformly over the class $\mathcal{H}$. As the number of terms B increases, this deviation decreases at a rate of $O(1/\sqrt{B})$. The second assertion of Theorem 11 gives a maximal deviation result with respect to $\mu(H)$. Observe in particular that, with the asymptotic settings previously specified, $N = O(n)$ and $\log(|\Lambda|) = O(\log n)$ as $n \to +\infty$. The bounds stated above thus show that, for a number $B = B_n$ of terms tending to infinity at a rate $O(n)$ as $n \to +\infty$, the maximal deviation $\sup_{H \in \mathcal{H}} |\widetilde{U}_B(H) - \mu(H)|$ is asymptotically of the order $O_{\mathbb{P}}((\log(n)/n)^{1/2})$, just like $\sup_{H \in \mathcal{H}} |U_{\mathbf{n}}(H) - \mu(H)|$, see bound (3.17) in Proposition 2. In short, when considering an incomplete U-statistic (3.19) with $B = O(n)$ terms only, the learning rate for the corresponding minimizer is of the same order as that of the minimizer of the complete risk (3.1), whose computation requires to average $|\Lambda| = O(n^{d_1+\ldots+d_K})$ terms. Minimizing such incomplete U-statistics thus yields a significant gain in terms of computational cost while fully preserving the learning rate. In contrast, as implied by Proposition 2, the minimization of a complete U-statistic involving $O(n)$ terms, obtained by drawing subsamples of sizes $n_k' = O(n^{1/(d_1+\ldots+d_K)})$ uniformly at random, leads to a rate of convergence of $O(\sqrt{\log(n)/n^{1/(d_1+\ldots+d_K)}})$, which is much slower except in the trivial case where $K = 1$ and $d_1 = 1$. These striking results are summarized in Table 3.1.

The important practical consequence of the above is that when n is too large for the complete risk (3.1) to be used, one should instead use the incomplete risk (3.19) (setting the number of terms B as large as the computational budget allows).

Empirical risk criterion	Nb of terms	Rate bound
Complete U-statistic	$O(n^{d_1+\ldots+d_K})$	$O_{\mathbb{P}}(\sqrt{\log(n)/n})$
Complete U-statistic based on subsamples	$O(n)$	$O_{\mathbb{P}}\left(\sqrt{\log(n)/n^{\frac{1}{d_1+\ldots+d_K}}}\right)$
Incomplete U-statistic (our result)	$\boldsymbol{O(n)}$	$\boldsymbol{O_{\mathbb{P}}(\sqrt{\log(n)/n})}$

TABLE 3.1: Rate bound for the empirical minimizer of several empirical risk criteria *versus* the number of terms involved in the computation of the criterion. For a computational budget of $O(n)$ terms, the rate bound for the incomplete U-statistic criterion is of the same order as that of the complete U-statistic, which is a huge improvement over a complete U-statistic based on a subsample.

3.3.2 Model Selection Based on Incomplete U-Statistics

Automatic selection of the model complexity is a crucial issue in machine learning: it includes the number of clusters in cluster analysis (see CLÉMENÇON, 2014) or the choice of the number of possible values taken by a piecewise constant scoring function in multipartite ranking for instance (*cf.* CLÉMENÇON and VAYATIS, 2009). In the present situation, this boils down to choosing the adequate level of complexity of the class of kernels $\mathcal{H}$, measured through its (supposedly finite) VC dimension for simplicity, in order to minimize the (theoretical) risk of the empirical minimizer. It is the purpose of this subsection to show that the incomplete U-statistic (3.19) can be used to define a penalization method to select a prediction rule with nearly minimal risk, avoiding procedures based on data splitting/resampling and extending the celebrated *structural risk minimization* principle, see VAPNIK, 1999. Let $\mathcal{H}$ be the collection of all symmetric kernels on $\prod_{k=1}^{K} \mathcal{X}_k^{d_k}$ and set $\mu^* = \inf_{H\in\mathcal{H}} \mu(H)$. Let $\mathcal{H}_1, \mathcal{H}_2, \ldots$ be a sequence of uniformly bounded major subclasses of $\mathcal{H}$, of increasing complexity (VC dimension). For any $m \geq 1$, let V_m denote the VC dimension of the class $\mathcal{H}_m$ and set $\mathcal{M}_{\mathcal{H}_m} = \sup_{(H,x)\in\mathcal{H}_m\times\mathcal{X}} |H(x)| < +\infty$. We suppose that there exists $\mathcal{M} < +\infty$ such that $\sup_{m\geq 1} \mathcal{M}_{\mathcal{H}_m} \leq \mathcal{M}$. Given $1 \leq B \leq |\Lambda|$ and $m \geq 1$, the complexity penalized empirical risk of a solution $\widetilde{U}_{B,m}$ of the ERM problem (3.22) with $\mathcal{H} = \mathcal{H}_m$ is

$$\widetilde{U}_B(\widehat{H}_{B,m}) + \text{pen}(B, m), \tag{3.23}$$

where the quantity $\text{pen}(B, m)$ is a *distribution free* penalty given by:

$$\begin{aligned}\text{pen}(B, m) = &2\mathcal{M}_{\mathcal{H}_m}\left(\sqrt{\frac{2V_m \log(1+N)}{N}} + \sqrt{\frac{2(\log 2 + V_m \log(1+|\Lambda|))}{B}}\right)\\ &+ 2\mathcal{M}\sqrt{\frac{(B+n)\log m}{B^2}}.\end{aligned} \tag{3.24}$$

As shown in Assertion (ii) of Corollary 1, the quantity above is an upper bound for the expected maximal deviation $\mathbb{E}[\sup_{H\in\mathcal{H}_m} |\widetilde{U}_B(H) - \mu(H)|]$ and

is thus a natural penalty candidate to compensate the overfitting within class $\mathcal{H}_m$. We thus propose to select

$$\widehat{m}_B = \arg\min_{m \geq 1} \left\{ \widetilde{U}_B(\widehat{H}_{B,m}) + \text{pen}(B, m) \right\}. \tag{3.25}$$

As revealed by the theorem below, choosing $B = O(n)$, the prediction rule $\widehat{H}_{\widehat{m}_B}$ based on a penalized criterion involving the summation of $O(n)$ terms solely, achieves a nearly optimal trade-off between the bias and the distribution free upper bound (3.24) on the variance term.

Theorem 12. (ORACLE INEQUALITY) *Suppose that Theorem 11's assumptions are fulfilled for all $m \geq 1$ and that $\sup_{m \geq 1} \mathcal{M}_{\mathcal{H}_m} \leq \mathcal{M} < +\infty$. Then, we have: $\forall \mathbf{n} \in \mathbb{N}^{*K}$, $\forall B \in \{1, \ldots, |\Lambda|\}$,*

$$\mu(\widehat{H}_{B,\widehat{m}}) - \mu^* \leq \inf_{k \geq 1} \left\{ \inf_{H \in \mathcal{H}_m} \mu(H) - \mu^* + \text{pen}(B, m) \right\} + \mathcal{M} \frac{\sqrt{2\pi(B+n)}}{B}.$$

We point out that the argument used to obtain the above result can be straightforwardly extended to other (possibly data-dependent) complexity penalties (*cf.* MASSART, 2006), see the proof in Appendix 3.7.4.

3.3.3 Fast Rates for ERM of Incomplete U-Statistics

In CLÉMENÇON, LUGOSI, and VAYATIS, 2008, it has been proved that, under certain "low-noise" conditions, the minimum variance property of the U-statistics used to estimate the ranking risk (corresponding to the situation $K = 1$ and $d_1 = 2$) leads to learning rates faster than $O_{\mathbb{P}}(1/\sqrt{n})$. These results rely on the *Hajek projection*, a linearization technique originally introduced in HOEFFDING, 1948 for the case of one sample U-statistics and next extended to the analysis of a much larger class of functionals in HÁJEK, 1968. It consists in writing $U_{\mathbf{n}}(H)$ as the sum of the orthogonal projection

$$\widehat{U}_{\mathbf{n}}(H) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathbb{E}\left[U_{\mathbf{n}}(H) \mid X_i^{(k)} \right] - (n-1)\mu(H), \tag{3.26}$$

which is itself a sum of K independent basic sample means based on i.i.d. r.v.'s (of the order $O_{\mathbb{P}}(1/\sqrt{n})$ each, after recentering), plus a possible negligible term. This representation was used for instance by GRAMS and SERFLING, 1973 to refine the CLT in the multisample U-statistics framework. It should be noticed that the quantity $\widehat{U}_{\mathbf{n}}(H)$ cannot be considered as a statistic in general, since the conditional expectations involved in the summation are unknown in practice.

Although incomplete U-statistics do not share the minimum variance property (see Section 3.3.1), we will show that the same fast rate bounds for the excess risk as those reached by ERM of U-statistics (corresponding to the summation of $O(n^2)$ pairs of observations) can be attained by empirical ranking risk minimizers, when estimating the ranking risk by incomplete U-statistics involving the summation of $o(n^2)$ terms solely.

For clarity (and comparison purpose), we first recall the statistical learning framework considered in CLÉMENÇON, LUGOSI, and VAYATIS, 2008. Let (X, Y) be a pair of random variables defined on the same probability space,

where Y is a real-valued label and X models some input information taking its values in a measurable space $\mathcal{X}$ hopefully useful to predict Y. Denoting by (X', Y') an independent copy of the pair (X, Y). The goal pursued here is to learn how to rank the input observations X and X', by means of an antisymmetric *ranking rule* $r : \mathcal{X}^2 \to \{-1, +1\}$ (*i.e.* $r(x, x') = -r(x'x)$ for any $(x, x') \in \mathcal{X}^2$), so as to minimize the *ranking risk*

$$L(r) = \mathbb{P}\{(Y - Y') \cdot r(X, X') < 0\}. \tag{3.27}$$

The minimizer of the ranking risk is the ranking rule $r^*(X, X') = 2\mathbb{I}\{\mathbb{P}\{Y > Y' \mid (X, X')\} \geq \mathbb{P}\{Y < Y' \mid (X, X')\} - 1$ (see Proposition 1 in CLÉMENÇON, LUGOSI, and VAYATIS, 2008). The natural empirical counterpart of (3.27) based on a sample of independent copies $(X_1, Y_1), \ldots, (X_n, Y_n)$ of the pair (X, Y) is the 1-sample U-statistic $U_n(H_r)$ of degree two with kernel $H_r((x, y), (x', y')) = \mathbb{I}\{(y - y') \cdot r(x, x') < 0\}$ for all (x, y) and $(x', y'))$ in $\mathcal{X} \times \mathbb{R}$ given by:

$$L_n(r) = U_n(H_r) = \frac{2}{n(n-1)} \sum_{i<j} \mathbb{I}\{(Y_i - Y_j) \cdot r(X_i, X_j) < 0\}. \tag{3.28}$$

Equipped with these notations, a statistical version of the excess risk $\Lambda(r) = L(r) - L(r^*)$ is a U-statistic $\lambda_n(r)$ with kernel $q_r = H_r - H_{r^*}$. The key "noise-condition", which allows to exploit the Hoeffding/Hajek decomposition of $\Lambda_n(r)$, is stated below.

Assumption 1. There exist constants $c > 0$ and $\alpha \in [0, 1]$ such that:

$$\forall r \in \mathcal{R}, \quad \mathrm{Var}\left(h_r(X, Y)\right) \leq c\Lambda(r)^\alpha,$$

where we set $h_r(x, y) = \mathbb{E}[q_r((x, y), (X', Y')]$.

Recall incidentally that very general sufficient conditions guaranteeing that this assumption holds true have been exhibited, see Section 5 in CLÉMENÇON, LUGOSI, and VAYATIS, 2008 (notice that the condition is void for $\alpha = 0$). Since our goal is to explain the main ideas rather than achieving a high level of generality, we consider a very simple setting, stipulating that the cardinality of the class of ranking rule candidates $\mathcal{R}$ under study is finite, $|\mathcal{R}| = M < +\infty$, and that the optimal rule r^* belongs to $\mathcal{R}$. The following proposition is a simplified version of the fast rate result proved in CLÉMENÇON, LUGOSI, and VAYATIS, 2008 for the empirical minimizer $\widehat{r}_n = \arg\min_{r \in \mathcal{R}} L_n(r)$.

Proposition 3. *(CLÉMENÇON, LUGOSI, and VAYATIS, 2008, COROLLARY 6) Suppose that Assumption 1 is fulfilled. Then, there exists a universal constant $C > 0$ such that for all $\delta \in (0, 1)$, we have: $\forall n \geq 2$,*

$$L(\widehat{r}_n) - L(r^*) \leq C \left(\frac{\log(M/\delta)}{n} \right)^{\frac{1}{2-\alpha}}. \tag{3.29}$$

Consider now the minimizer $\widetilde{r}_B$ of the incomplete U-statistic risk estimate

$$\widetilde{U}_B(H_r) = \frac{1}{B} \sum_{k=1}^{B} \sum_{(i,j):1 \leq i < j \leq n} \epsilon_k((i, j)) \mathbb{I}\{(Y_i - Y_j) \cdot r(X_i, X_j) < 0\} \tag{3.30}$$

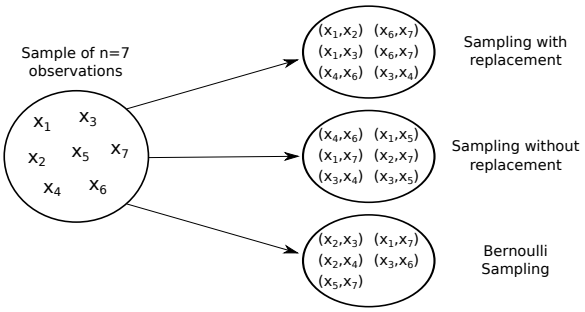


FIGURE 3.2: Illustration of different sampling schemes for approximating a U-statistic. For simplicity, consider again the case $K = 1$ and $d_1 = 2$. Here $n = 7$ and the expected number of terms is $B = 6$. Sampling with or without replacement results in exactly B terms, with possible repetitions when sampling with replacement, *e.g.* (x_6, x_7) in this example. In contrast, Bernoulli sampling with $\pi_I = B/|\Lambda|$ results in B terms only in expectation, with individual realizations that may exhibit more or fewer terms.

over $\mathcal{R}$, where $\epsilon_k((i,j))$ indicates whether the pair (i,j) has been picked at the k-th draw ($\epsilon_k((i,j)) = 1$ in this case, which occurs with probability $1/\binom{n}{2}$) or not (then, we set $\epsilon_k((i,j)) = 0$). Observe that $\widetilde{r}_B$ also minimizes the empirical estimate of the excess risk $\widetilde{\Lambda}_B(r) = \widetilde{U}_B(q_r)$ over $\mathcal{R}$.

Theorem 13. *Let $\alpha \in [0,1]$ and suppose that Assumption 1 is fulfilled. If we set $B = O(n^{2/(2-\alpha)})$, there exists some constant $C < +\infty$ such that, for all $\delta \in (0,1)$, we have with probability at least $1-\delta$: $\forall n \geq 2$,*

$$L(\widetilde{r}_B) - L(r^*) \leq C \left(\frac{\log(M/\delta)}{n} \right)^{\frac{1}{2-\alpha}}.$$

As soon as $\alpha < 1$, this result shows that the same fast rate of convergence as that reached by $\widehat{r}_n$ can be attained by the ranking rule $\widetilde{r}_B$, which minimizes an empirical version of the ranking risk involving the summation of $O(n^{2/(2-\alpha)})$ terms solely. For comparison purpose, minimization of the criterion (3.27) computed with a number of terms of the same order leads to a rate bound of order $O_{\mathbb{P}}(n^{1/(2-\alpha)^2})$.

Finally, we point out that fast rates for the clustering problem have been also investigated in CLÉMENÇON, 2014, see Section 5.2 therein. The present analysis can be extended to the clustering framework by means of the same arguments.

3.3.4 Alternative Sampling Schemes

Sampling with replacement is not the sole way of approximating generalized U-statistics with a controlled computational cost. As proposed in JANSON, 1984, other sampling schemes can be considered, Bernoulli sampling or sampling without replacement in particular (see Figure 3.2 for an illustration). We now explain how the results of this paper can be extended to these situations. The population of interest is the set Λ and a *survey sample* of (possibly random) size $b \leq n$ is any subset s of cardinality $b = b(s)$ less than $|\Lambda|$ in the power set $\mathcal{P}(\Lambda)$. Here, a general *survey scheme without replacement*

is any conditional probability distribution R on the set of all possible samples $s \in \mathcal{P}(\Lambda)$ given $(\mathbf{X}_I)_{I\in\Lambda}$. For any $I \in \Lambda$, the first order *inclusion probability* $\pi_I(R) = \mathbb{P}_R\{I \in S\}$, is the probability that the unit I belongs to a random sample S drawn from distribution R. We set $\boldsymbol{\pi}(R) = (\pi_I(R))_{I\in\Lambda}$. The second order inclusion probabilities are denoted by $\pi_{I,J}(R) = \mathbb{P}_R\{(I,J) \in S^2\}$ for any $I \neq J$ in Λ. When no confusion is possible, we omit to mention the dependence in R when writing the first/second order probabilities of inclusion. The information related to the observed sample $S \subset \Lambda$ is fully enclosed in the random vector $\boldsymbol{\Delta} = (\Delta(I))_{I\in\Lambda}$, where $\Delta(I) = \mathbb{I}\{I \in S\}$ for all $I \in \Lambda$. The 1-d marginal distributions of the sampling scheme $\boldsymbol{\Delta}_n$ are the Bernoulli distributions with parameters π_I, $I \in \Lambda$, and the covariance matrix of the r.v. $\boldsymbol{\Delta}_\mathbf{n}$ is given by $\Gamma = \{\pi_{I,J} - \pi_I\pi_J\}_{I,J}$ with the convention $\pi_{I,I} = \pi_I$ for all $I \in \Lambda$. Observe that, equipped with the notations above, $\sum_{I\in\Lambda} \Delta(I) = b(S)$.

One of the simplest survey plans is the Poisson scheme (without replacement), for which the $\Delta(I)$'s are independent Bernoulli random variables with parameters π_I, $I \in \Lambda$, in $(0,1)$. The first order inclusion probabilities fully characterize such a plan. Observe in addition that the size $b(\mathcal{S})$ of a sample generated this way is random with expectation $B = \mathbb{E}[b(S) \mid (\mathbf{X}_I)_{I\in\Lambda}] = \sum_{I\in\Lambda} \pi_I$. The situation where the π_I's are all equal corresponds to the Bernoulli sampling scheme: $\forall I \in \Lambda$, $\pi_I = B/|\Lambda|$. The Poisson survey scheme plays a crucial role in sampling theory, inso far as a wide range of survey schemes can be viewed as conditional Poisson schemes, see HÁJEK, 1964. For instance, one may refer to COCHRAN, 1977 or DEVILLE, 1987 for accounts of survey sampling techniques.

Following in the footsteps of the seminal contribution of HORVITZ and THOMPSON, 1951, an estimate of (3.1) based on a sample drawn from a survey scheme R with first order inclusion probabilities $(\pi_I)_{I\in\Lambda}$ is given by:

$$\bar{U}_{HT}(H) = \frac{1}{|\Lambda|} \sum_{I\in\Lambda} \frac{\Delta(I)}{\pi_I} H(\mathbf{X}_I), \tag{3.31}$$

with the convention that $0/0 = 0$. Notice that it is an unbiased estimate of (3.1):

$$\mathbb{E}[\bar{U}_{HT}(H) \mid (\mathbf{X}_I)_{I\in\Lambda}] = U_\mathbf{n}(H).$$

In the case where the sample size is deterministic, its conditional variance is given by:

$$Var(\bar{U}_{HT}(H) \mid (\mathbf{X}_I)_{I\in\Lambda}) = \frac{1}{2} \sum_{I\neq J} \left(\frac{H(\mathbf{X}_I)}{\pi_I} - \frac{H(\mathbf{X}_J)}{\pi_J} \right)^2 (\pi_{I,J} - \pi_I\pi_J).$$

We point out that the computation of (3.31) involves summing over a possibly random number of terms, equal to $B = \mathbb{E}[b(S)] = \sum_{I\in\Lambda} \pi_I$ in average and whose variance is equal to $Var(b(S)) = \sum_{I\in\Lambda} \pi_I(1-\pi_I) + \sum_{I\neq J}\{\pi_{I,J} - \pi_I\pi_J\}$.

Here, we are interested in the situation where the $\Delta(I)$'s are independent from $(X_I)_{I\in\Lambda}$, and either a sample of size $B \leq |\Lambda|$ fixed in advance is chosen uniformly at random among the $\binom{|\Lambda|}{B}$ possible choices (this survey scheme is sometimes referred to as *rejective sampling* with equal first order inclusion probabilities), or else it is picked by means of a Bernoulli sampling

with parameter $B/|\Lambda|$. Observe that, in both cases, we have $\pi_I = B/|\Lambda|$ for all $I \in \Lambda$. The following theorem shows that in both cases, similar results as those obtained for *sampling with replacement* can be derived for minimizers of the Horvitz-Thompson risk estimate (3.31).

Theorem 14. *Let $\mathcal{H}$ be a collection of bounded symmetric kernels on $\prod_{k=1}^{K} \mathcal{X}_k^{d_k}$ that fulfills the assumptions involved in Proposition 2. Let $B \in \{1, \ldots, |\Lambda|\}$. Suppose that, for any $H \in \mathcal{H}$, $\bar{U}_{HT}(H)$ is the incomplete U-statistic based on either a Bernoulli sampling scheme with parameter $B/|\Lambda|$ or else a sampling without replacement scheme of size B. For all $\delta \in (0,1)$, we have with probability at least $1-\delta$: $\forall \mathbf{n} \in \mathbb{N}^{*K}$, $\forall B \in \{1, \ldots, |\Lambda|\}$,*

$$\sup_{H \in \mathcal{H}} \left| \bar{U}_{HT}(H) - U_{\mathbf{n}}(H) \right| \leq 2\mathcal{M}_{\mathcal{H}} \sqrt{\frac{\log(2(1+|\Lambda|)^V/\delta)}{B}} + \frac{2\log(2(1+|\Lambda|)^V/\delta)\mathcal{M}_{\mathcal{H}}}{3B},$$

in the case of the Bernoulli sampling design, and

$$\sup_{H \in \mathcal{H}} \left| \bar{U}_{HT}(H) - U_{\mathbf{n}}(H) \right| \leq \sqrt{2}\mathcal{M}_{\mathcal{H}} \sqrt{\frac{\log(2(1+|\Lambda|)^V/\delta)}{B}},$$

in the case of the sampling without replacement design.

We highlight the fact that, from a computational perspective, sampling with replacement is undoubtedly much more advantageous than Bernoulli sampling or sampling without replacement. Indeed, although its expected value is equal to B, the size of a Bernoulli sample is stochastic and the related sampling algorithm requires a loop through the elements I of Λ and the practical implementation of sampling without replacement is generally based on multiple iterations of sampling with replacement, see TILLÉ, 2006.

3.4 Application to Stochastic Gradient Descent

The theoretical analysis carried out in the preceding sections focused on the properties of empirical risk minimizers but ignored the issue of finding such a minimizer. In this section, we show that the sampling technique introduced in Section 3.3 also provides practical means of scaling up iterative statistical learning techniques. Indeed, large-scale training of many machine learning models, such as SVM, DEEP NEURAL NETWORKS or SOFT K-MEANS among others, is based on stochastic gradient descent (SGD in abbreviated form), see BOTTOU, 1998. When the risk is of the form (3.2), we now investigate the benefit of using, at each iterative step, a gradient estimate of the form of an incomplete U-statistic, instead of an estimate of the form of a complete U-statistic with exactly the same number of terms based on subsamples drawn uniformly at random.

Let $\Theta \subset \mathbb{R}^q$ with $q \geq 1$ be some parameter space and $H : \prod_{k=1}^K \mathcal{X}_k^{d_k} \times \Theta \to \mathbb{R}$ be a loss function which is convex and differentiable in its last argument. Let $(X_1^{(k)}, \ldots, X_{d_k}^{(k)})$, $1 \leq k \leq K$, be K independent random vectors with distribution $F_k^{\otimes d_k}(dx)$ on $\mathcal{X}_k^{d_k}$ respectively such that the random vector $H(X_1^{(1)}, \ldots, X_{d_1}^{(1)}, \ldots, X_1^{(K)}, \ldots, X_{d_K}^{(K)}; \theta)$ is square integrable for any $\theta \in \Theta$. For all $\theta \in \Theta$, set

$$L(\theta) = \mathbb{E}[H(X_1^{(1)}, \ldots, X_{d_1}^{(1)}, \ldots, X_1^{(K)}, \ldots, X_{d_K}^{(K)}; \theta)] = \mu(H(\cdot; \theta))$$

and consider the *risk minimization* problem $\min_{\theta \in \Theta} L(\theta)$. Based on K independent i.i.d. samples $X_1^{(k)}, \ldots, X_{n_k}^{(k)}$ with $1 \leq k \leq K$, the empirical version of the risk function is $\theta \in \Theta \mapsto \widehat{L}_{\mathbf{n}}(\theta) \stackrel{def}{=} U_{\mathbf{n}}(H(\cdot; \theta))$. Here and throughout, we denote by ∇_θ the gradient operator w.r.t. θ.

Gradient descent Many practical machine learning algorithms use variants of the standard gradient descent method, following the iterations:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \widehat{L}_{\mathbf{n}}(\theta_t), \tag{3.32}$$

with an arbitrary initial value $\theta_0 \in \Theta$ and a learning rate (step size) $\eta_t \geq 0$ such that $\sum_{t=1}^{+\infty} \eta_t = +\infty$ and $\sum_{t=1}^{+\infty} \eta_t^2 < +\infty$.

Here we place ourselves in a large-scale setting, where the sample sizes $n_1, \ldots, n_K$ of the training data sets are so large that computing the gradient of $\widehat{L}_{\mathbf{n}}$

$$\widehat{g}_{\mathbf{n}}(\theta) = \frac{1}{\prod_{k=1}^K \binom{n_k}{d_k}} \sum_{I_1} \cdots \sum_{I_K} \nabla_\theta H(\mathbf{X}_{I_1}^{(1)}; \mathbf{X}_{I_2}^{(2)}; \ldots; \mathbf{X}_{I_K}^{(K)}; \theta) \tag{3.33}$$

at each iteration (3.32) is computationally too expensive. Instead, Stochastic Gradient Descent uses an unbiased estimate $\widetilde{g}(\theta)$ of the gradient (3.33) that is cheap to compute. A natural approach consists in replacing (3.33) by a complete U-statistic constructed from subsamples of reduced sizes $n_k' << n_k$ drawn uniformly at random, leading to the following gradient estimate:

$$\widetilde{g}_{\mathbf{n}'}(\theta) = \frac{1}{\prod_{k=1}^K \binom{n_k'}{d_k}} \sum_{I_1} \cdots \sum_{I_K} \nabla_\theta H(\mathbf{X}_{I_1}^{(1)}; \mathbf{X}_{I_2}^{(2)}; \ldots; \mathbf{X}_{I_K}^{(K)}; \theta), \tag{3.34}$$

where the symbol $\sum_{I_k}$ refers to summation over all $\binom{n'_k}{d_k}$ subsets $\mathbf{X}_{I_k}^{(k)} = (X_{i_1}^{(k)}, \ldots, X_{i_{d_k}}^{(k)})$ related to a set I_k of d_k indexes $1 \leq i_1 < \ldots < i_{d_k} \leq n'_k$ and $\mathbf{n}' = (n'_1, \ldots, n'_K)$.

We propose an alternative strategy based on the sampling scheme described in Section 3.3, *i.e.* a gradient estimate in the form of an *incomplete* U-statistic:

$$\widetilde{g}_B(\theta) = \frac{1}{B} \sum_{(I_1, \ldots, I_K) \in \mathcal{D}_B} \nabla_\theta H(\mathbf{X}_{I_1}^{(1)}, \ldots, \mathbf{X}_{I_K}^{(K)}; \theta), \tag{3.35}$$

where $\mathcal{D}_B$ is built by sampling with replacement in the set Λ.

It is well-known that the variance of the gradient estimate negatively impacts on the convergence of SGD. Consider for instance the case where the loss function H is $(1/\gamma)$-smooth in its last argument, *i.e.*, $\forall \theta_1, \theta_2 \in \Theta$:

$$\|\nabla_\theta H(\cdot; \theta_1) - \nabla_\theta H(\cdot; \theta_2)\| \leq \frac{1}{\gamma} \|\theta_1 - \theta_2\|.$$

Then one can show that if $\widetilde{g}$ is the gradient estimate:

$$\begin{aligned}
\mathbb{E}[\widehat{L}_{\mathbf{n}}(\theta_{t+1})] &= \mathbb{E}[\widehat{L}_{\mathbf{n}}(\theta_t - \eta_t \widetilde{g}(\theta_t))] \\
&\leq \mathbb{E}[\widehat{L}_{\mathbf{n}}(\theta_t)] - \eta_t \|\mathbb{E}[\widehat{g}_{\mathbf{n}}(\theta_t)]\|^2 + \frac{\eta_t^2}{2\gamma} \mathbb{E}[\|\widetilde{g}(\theta_t)\|^2] \\
&\leq \mathbb{E}[\widehat{L}_{\mathbf{n}}(\theta_t)] - \eta_t \left(1 - \frac{\eta_t}{2\gamma}\right) \mathbb{E}[\|\widehat{g}_{\mathbf{n}}(\theta_t)\|^2] + \frac{\eta_t^2}{2\gamma} \mathrm{Var}[\widetilde{g}(\theta_t)].
\end{aligned}$$

In other words, the smaller the variance of the gradient estimate, the larger the expected reduction in objective value. Some recent work has focused on variance-reduction strategies for SGD when the risk estimates are basic sample means (see for instance Le Roux, Schmidt, and Bach, 2012; Johnson and Zhang, 2013).

In our setting where the risk estimates are of the form of a U-statistic, we are interested in comparing the variance of $\widetilde{g}_{\mathbf{n}'}(\theta)$ and $\widetilde{g}_B(\theta)$ when $B = \prod_{k=1}^K \binom{n'_k}{d_k}$ so that their computation requires to average over the same number of terms and thus have similar computational cost. [1] Our result is summarized in the following proposition.

Proposition 4. *Let $B = \prod_{k=1}^K \binom{n'_k}{d_k}$ for $n'_k \ll n_k$, $k = 1, \ldots, K$. In the asymptotic framework (3.4), we have:*

$$\mathrm{Var}[\widetilde{g}_{\mathbf{n}'}(\theta)] = O\left(\frac{1}{\sum_{k=1}^K n'_k}\right), \quad \mathrm{Var}[\widetilde{g}_B(\theta)] = O\left(\frac{1}{\prod_{k=1}^K \binom{n'_k}{d_k}}\right),$$

as $n' = n'_1 + \ldots + n'_K \to +\infty$.

Proposition 4 shows that the convergence rate of $\mathrm{Var}[\widetilde{g}_B(\theta)]$ is faster than that of $\mathrm{Var}[\widetilde{g}_{\mathbf{n}'}(\theta)]$ except when $K = 1$ and $d_1 = 1$. Thus the expected improvement in objective function at each SGD step is larger when using

1. Note that sampling B sets from Λ to obtain (3.35) is potentially more efficient than sampling n'_k points from $\mathbf{X}_{\{1,\ldots,n_k\}}$ for each $k = 1, \ldots, K$ and then forming all combinations to obtain (3.34).

a gradient estimate in the form of (3.35) instead of (3.34), although both strategies require to average over the same number of terms. This is also supported by the experimental results reported in the next section.

In PAPA, CLÉMENÇON, and BELLET, 2015, this analysis is further developed and a bound on the mean excess risk is explicited. They show that for a class $\mathcal{H}$ of VC dimension $V < +\infty$ such that

$$\mathcal{M}_\Theta := \sup_{\theta\in\Theta,(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(K)}))\in\prod_{k=1}^K \mathcal{X}_k^{d_k}} |H(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(K)};\theta)| < +\infty,$$

and for a sequence $(\theta_t)_{t\geq 0}$ generated by incomplete stochastic gradient algorithm with step size $\eta_t \propto t^{-\beta}$ and $1/2 < \beta < 1$, the following bound holds:

$$|L(\theta_t) - L(\theta^*)| \leq \frac{C}{Bt^\beta} + 4\mathcal{M}_\Theta\sqrt{\frac{2V\log(1+\kappa)}{\kappa}}, \tag{3.36}$$

where $C > 0$ is a constant depending on the problem conditioning. With a similar reasoning to Propostion 4, one can show that the corresponding bound for a complete U-statistic estimator from a subsample is greatly increased, as the factor $B = \prod_{k=1}^K \binom{n'_k}{d_k}$ turns into $\sum_{k=1}^K n'_k$ when considered complete statistic. This furthermore evidences the interest of the incomplete sampling scheme for estimating gradients.

3.5 Numerical Experiments

We show the benefits of the sampling approach promoted in this paper on two applications: metric learning for classification, and model selection in clustering.

3.5.1 Metric Learning

In this section, we focus on the metric learning problem (see Section 3.2.2). As done in much of the metric learning literature, we restrict our attention to the family of pseudo-distance functions $D_{\boldsymbol{M}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ defined as

$$D_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} - \boldsymbol{x}')\boldsymbol{M}(\boldsymbol{x} - \boldsymbol{x}')^T,$$

where $\boldsymbol{M} \in \mathbb{S}_+^d$, and $\mathbb{S}_+^d$ is the cone of $d \times d$ symmetric positive-semidefinite (PSD) matrices.

Given a training sample $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, \dots, C\}$, let $y_{ij} = 1$ if $y_i = y_j$ and 0 otherwise for any pair of samples. Given a threshold $b \geq 0$, we define the empirical risk as follows:

$$R_n(D_{\boldsymbol{M}}) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \left[y_{ij}(b - D_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)) \right]_+, \tag{3.37}$$

where $[u]_+ = \max(0, 1 - u)$ is the hinge loss. This risk estimate is convex and was used for instance by JIN, WANG, and ZHOU (2009) and CAO, GUO, and YING (2012). Our goal is the find the empirical risk minimizer among our family of distance functions, i.e.:

$$\widehat{\boldsymbol{M}} = \operatorname*{arg\,min}_{\boldsymbol{M} \in \mathbb{S}_+^d} R_n(D_{\boldsymbol{M}}). \tag{3.38}$$

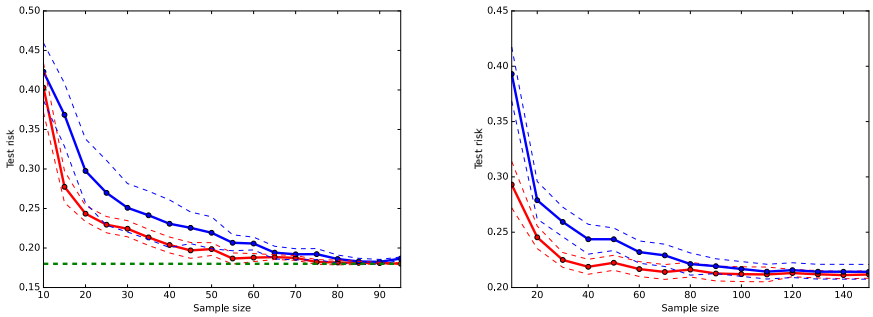
In our experiments, we use the following two data sets:

— **Synthetic data set**: some synthetic data that we generated for illustration. X is a mixture of 10 gaussians in $\mathbb{R}^{40}$ – each one corresponding to a class – such that all gaussian means are contained in an subspace of dimension 15 and their shared covariance matrix is proportional to identity with a variance factor such that some overlap is observed. That is, the solution to the metric learning problem should be proportional to the linear projection over the subspace containing the gaussians means. Training and testing sets contain respectively 50,000 and 10,000 observations.

— **MNIST data set**: a handwritten digit classification data set which has 10 classes and consists of 60,000 training images and 10,000 test images.[2] This data set has been used extensively to benchmark metric learning (WEINBERGER and SAUL, 2009). As done by previous authors, we reduce the dimension from 784 to 164 using PCA so as to retain 95% of the variance, and normalize each sample to unit norm.

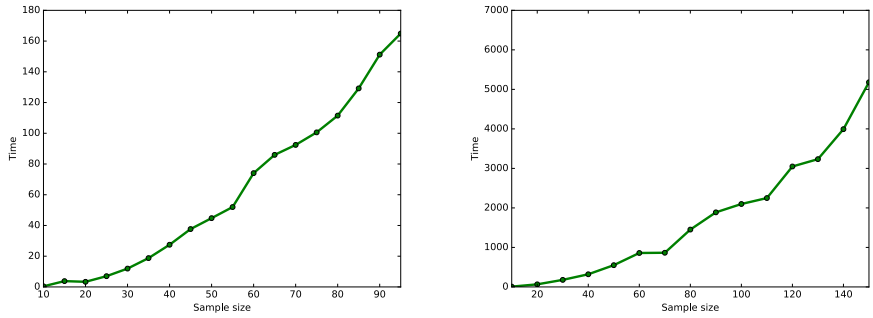
Note that for both datasets, merely computing the empirical risk (3.37) for a given $\boldsymbol{M}$ involves averaging over more than 10^9 pairs.

2. http://yann.lecun.com/exdb/mnist/



(A) Synthetic dataset. (B) MNIST data set.

FIGURE 3.3: Test risk with respect to the sample size p of the ERM when the risk is approximated using complete (blue) or incomplete (red) U-statistics. Solid lines represent means and dashed ones represent standard deviation. For the synthetic data set, the green dotted line represent the performance of the true risk minimizer.



(A) Synthetic data set. (B) MNIST data set.

FIGURE 3.4: Average training time (in seconds) with respect to the sample size p.

We conduct two types of experiment. In Section 3.5.1, we subsample the data before learning and evaluate the performance of the ERM on the subsample. In Section 3.5.1, we use Stochastic Gradient Descent to find the ERM on the original sample, using subsamples at each iteration to estimate the gradient.

One-Time Sampling

We compare two sampling schemes to approximate the empirical risk:
— Complete U-statistic: p indices are uniformly picked at random in $\{1, \dots, n\}$. The empirical risk is approximated using any possible pair formed by the p indices, that is $\frac{p(p-1)}{2}$ pairs.
— Incomplete U-statistic: the empirical risk is approximated using $\frac{p(p-1)}{2}$ pairs picked uniformly at random in $\{1, \dots, n\}^2$.

For each strategy, we use a projected gradient descent method in order to solve (3.38), using several values of p and averaging the results over 50 random trials. As the testing sets are large, we evaluate the test risk on 100,000 randomly picked pairs.

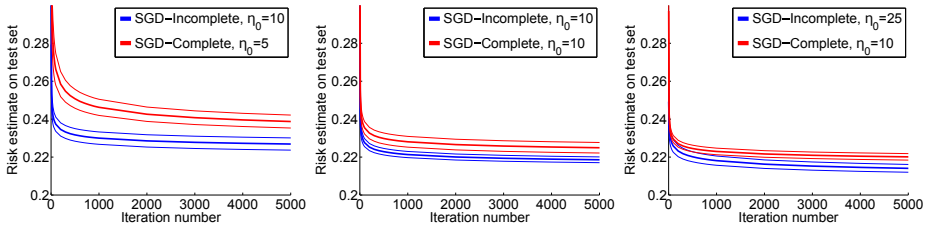


FIGURE 3.5: SGD results on the MNIST data set for various mini-batch size m. Solid and thin lines respectively shows the means and standard deviations over 50 runs.

Figure 3.3a shows the test risk of the ERM with respect to the sample size p for both sampling strategies on the synthetic data set. As predicted by our theoretical analysis, the incomplete U-statistic strategy achieves a significantly smaller risk on average. For instance, it gets within 5% error of the true risk minimizer for $p = 50$, while the complete U-statistic needs $p > 80$ to reach the same performance. This represents twice more computational time, as shown in Figure 3.4a (as expected, the runtime increases roughly quadratically with p). The incomplete U-statistic strategy also has the advantage of having a much smaller variance between the runs, which makes it more reliable. The same conclusions hold for the MNIST data set, as can be seen in Figure 3.3b and Figure 3.4b.

Stochastic Gradient Descent

In this section, we focus on solving the ERM problem (3.38) using Stochastic Gradient Descent and compare two approaches (analyzed in Section 3.4) to construct a mini-batch at each iteration. The first strategy, SGD-Complete, is to randomly draw (with replacement) a subsample and use the complete U-statistic associated with the subsample as the gradient estimate. The second strategy, SGD-Incomplete (the one we promote in this paper), consists in sampling an incomplete U-statistic with the same number of terms as in SGD-Complete.

For this experiment, we use the MNIST data set. We set the threshold in (3.37) to $b = 2$ and the learning rate of SGD at iteration t to $\eta_t = 1/(\eta_0 t)$ where $\eta_0 \in \{1, 2.5, 5, 10, 25, 50\}$. To reduce computational cost, we only project our solution onto the PSD cone at the end of the algorithm, following the "one projection" principle used by CHECHIK et al. (2010). We try several values m for the mini-batch size, namely $m \in \{10, 28, 55, 105, 253\}$. [3] For each mini-batch size, we run SGD for 10,000 iterations and select the learning rate parameter η_0 that achieves the minimum risk on 100,000 pairs randomly sampled from the training set. We then estimate the generalization risk using 100,000 pairs randomly sampled from the test set.

For all mini-batch sizes, SGD-Incomplete achieves significantly better test risk than SGD-Complete. Detailed results are shown in Figure 3.5 for three mini-batch sizes, where we plot the evolution of the test risk with

3. For each m, we can construct a complete U-statistic from n' samples with $n'(n'-1)/2 = m$ terms.

respect to the iteration number.[4] We make several comments. First, notice that the best learning rate is often larger for SGD-Incomplete than for SGD-Complete ($m = 10$ and $m = 253$). This confirms that gradient estimates from the former strategy are generally more reliable. This is further supported by the fact that even though larger learning rates increase the variance of SGD, in these two cases SGD-Complete and SGD-Incomplete have similar variance. On the other hand, for $m = 55$ the learning rate is the same for both strategies. SGD-Incomplete again performs significantly better on average and also has smaller variance. Lastly, as one should expect, the gap between SGD-Complete and SGD-Incomplete reduces as the size of the mini-batch increases. Note however that in practical implementations, the relatively small mini-batch sizes (in the order of a few tens or hundreds) are generally those which achieve the best error/time trade-off.

3.5.2 Model Selection in Clustering

In this section, we are interested in the clustering problem described in Section 3.2.2. Specifically, let $X_1, \ldots, X_n \in \mathbb{R}^d$ be the set of points to be clustered. Let the clustering risk associated with a partition $\mathcal{P}$ into M groups $\mathcal{C}_1, \ldots, \mathcal{C}_M$ be:

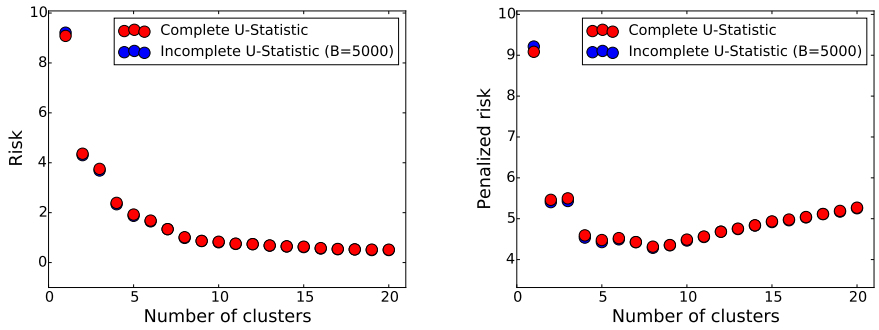
$$\widehat{W}_n(\mathcal{P}) = \frac{2}{n(n-1)} \sum_{m=1}^{M} \sum_{1 \leq i < j \leq n} D(X_i, X_j) \cdot \mathbb{I}\{(X_i, X_j) \in \mathcal{C}_m^2\}. \tag{3.39}$$

In this experiment, given a set of candidate partitions, we want to perform model selection by picking the partition which minimizes the risk (3.39) plus some term penalizing the complexity of the partition. When the number of points n is large, the complete risk is very expensive to compute. Our strategy is to replace it with an incomplete approximation with much fewer terms. Like in the approach theoretically investigated in Section 3.3.2, the goal pursued is to show that using the incomplete approximation as goodness-of-fit measure in a comlexity penalized criterion instead of the complete version does not damage the selection, while reducing the computational cost (notice incidentally that the complexity penalty used in this example is not of the same type as the structural VC dimension-based penalty considered in Theorem 12).

The experimental setup is as follows. We used the forest cover type data set,[5] which is popular to benchmark clustering algorithms (see for instance KANUNGO et al., 2004). To be able to evaluate the complete risk, we work with $n = 5,000$ points subsampled at random from the entire data set of 581,012 points in dimension 54. We then generated a hierarchical clustering of these points using agglomerative clustering with Ward's criterion (WARD, 1963) as implemented in the `scikit-learn` Python library (PEDREGOSA et al., 2011). This defines n partitions $\mathcal{P}_1, \ldots, \mathcal{P}_n$ where $\mathcal{P}_m$ consists of m clusters ($\mathcal{P}_1$ corresponds to a single cluster containing all points, while in $\mathcal{P}_n$ each point has its own cluster).

4. We point out that the figures look the same if we plot the runtime instead of the iteration number. Indeed, the time spent on computing the gradients (which is the same for both variants) largely dominates the time spent on the random draws.

5. https://archive.ics.uci.edu/ml/datasets/Covertype



(A) Risk. (B) Penalized risk.

FIGURE 3.6: Clustering model selection results on the forest cover type data set. Figure 3.6a shows the risk (complete and incomplete with $B = 5,000$ terms) for the first 20 partitions, while Figure 3.6b shows the penalized risk for $c = 1.1$.

For each partition size, we first compare the value of the complete risk (3.39) with $n(n-1) = 24,995,000$ terms with that of an incomplete version with only $B = n = 5,000$ pairs drawn at random. As shown in Figure 3.6a, the incomplete U-statistic is a very accurate approximation of the complete one, despite consisting of 5000 times less terms. It will thus lead to similar results in model selection. To illustrate, we use a simple penalty term of the form $\text{pen}(\mathcal{P}_m) = c \cdot \log(m)$ where c is a scaling constant. Figure 3.6b shows that both selection criteria choose the same model $\mathcal{P}_8$. Performing this model selection over $\mathcal{P}_1, \ldots, \mathcal{P}_{20}$ took about 66 seconds for the complete U-statistic, compared to only 0.1 seconds for the incomplete version. [6]

Finally, we generated 100 incomplete U-statistics with different random seeds ; all of them correctly identified $\mathcal{P}_8$ as the best model. Using $B = 5,000$ pairs is thus sufficient to obtain reliable results with an incomplete U-statistic for this data set. In contrast, the complete U-statistics based on a subsample (leading to the same number of pairs) selected the correct model in only 57% of cases.

6. The $n \times n$ distance matrix was precomputed before running the agglomerative clustering algorithm. The associated runtime is thus not taken into account in these timing results.

3.6 Conclusion

In a wide variety of statistical learning problems, U-statistics are natural estimates of the risk measure one seeks to optimize. As the sizes of the samples increase, the computation of such functionals involves summing a rapidly exploding number of terms and becomes numerically unfeasible. In this paper, we argue that for such problems, *Empirical Risk Minimization* can be implemented using statistical counterparts of the risk based on much less terms (picked randomly by means of sampling with replacement), referred to as *incomplete U-statistics*. Using a novel deviation inequality, we have shown that this approximation scheme does not deteriorate the learning rates, even preserving fast rates in certain situations where they are proved to occur. Furthermore, we have extended these results to U-statistics based on different sampling schemes (Bernoulli sampling, sampling without replacement) and shown how such functionals can be used for the purpose of model selection and for implementing ERM iterative procedures based on stochastic gradient descent. Beyond theoretical rate bounds, the efficiency of the approach we promote is illustrated by several numerical experiments.

In the next chapter, we focus on decentralized estimation of a U-statistic. In decentralized setting, the U-statistic itself is not directly computable — or at a prohibitive cost — so usual methods have to be adapted in order to keep an acceptable convergence rate.

3.7 Proofs

3.7.1 Proof of Proposition 2

Set $N = \min\{\lfloor n_1/d_1 \rfloor, \ldots, \lfloor n_K/d_K \rfloor\}$ and let

$$
\begin{aligned}
&V_H\left(X_1^{(1)}, \ldots, X_{n_1}^{(1)}, \ldots, X_1^{(K)}, \ldots, X_{n_K}^{(K)}\right) \\
&\quad = \frac{1}{N}\Big[H\left(X_1^{(1)}, \ldots, X_{d_1}^{(1)}, \ldots, X_1^{(K)}, \ldots, X_{d_K}^{(K)}\right) \\
&\qquad + H\left(X_{d_1+1}^{(1)}, \ldots, X_{2d_1}^{(1)}, \ldots, X_{d_K+1}^{(K)}, \ldots, X_{2d_K}^{(K)}\right) \\
&\qquad + \ldots \\
&\qquad + H\left(X_{(N-1)d_1+1}^{(1)}, \ldots, X_{Nd_1}^{(1)}, \ldots, X_{(N-1)d_K+1}^{(K)}, \ldots, X_{Nd_K}^{(K)}\right)\Big],
\end{aligned}
$$

for any $H \in \mathcal{H}$. Recall that the K-sample U-statistic $U_{\mathbf{n}}(H)$ can be expressed as

$$
U_{\mathbf{n}}(H) = \frac{1}{n_1! \cdots n_K!} \sum_{\sigma_1 \in \mathfrak{S}_{n_1}, \ldots, \sigma_K \in \mathfrak{S}_{n_K}} V_H\left(X_{\sigma_1(1)}^{(1)}, \ldots, X_{\sigma_K(n_K)}^{(K)}\right), \quad (3.40)
$$

where $\mathfrak{S}_m$ denotes the symmetric group of order m for any $m \geq 1$. This representation as an average of sums of N independent terms is known as the (first) Hoeffding's decomposition, see HOEFFDING, 1948. Then, using Jensen's inequality in particular, one may easily show that, for any nondecreasing convex function $\psi : \mathbb{R}_+ \to \mathbb{R}$, we have:

$$
\mathbb{E}\left[\psi\left(\sup_{H \in \mathcal{H}} \left|U_{\mathbf{n}}(\bar{H})\right|\right)\right] \leq \mathbb{E}\left[\psi\left(\sup_{H \in \mathcal{H}} \left|V_{\bar{H}}(X_1^{(1)}, \ldots, X_{n_K}^{(K)})\right|\right)\right], \quad (3.41)
$$

where we set $\bar{H} = H - \mu(H)$ for all $H \in \mathcal{H}$. Now, using standard symmetrization and randomization arguments (see GINÉ and ZINN, 1984 for instance) and (3.41), we obtain that

$$
\mathbb{E}\left[\psi\left(\sup_{H \in \mathcal{H}} \left|U_{\mathbf{n}}(\bar{H})\right|\right)\right] \leq \mathbb{E}\left[\psi\left(2\mathcal{R}_N\right)\right], \quad (3.42)
$$

where

$$
\mathcal{R}_N = \sup_{H \in \mathcal{H}} \frac{1}{N} \sum_{l=1}^{N} \epsilon_l H\left(X_{(l-1)d_1+1}^{(1)}, \ldots, X_{ld_1}^{(1)}, \ldots, X_{(l-1)d_K+1}^{(K)}, \ldots, X_{ld_K}^{(K)}\right),
$$

is a Rademacher average based on the Rademacher chaos $\epsilon_1, \ldots, \epsilon_N$ (independent random symmetric sign variables), independent from the $X_i^{(k)}$'s. We now apply the bounded difference inequality (see MCDIARMID, 1989) to the functional $\mathcal{R}_N$, seen as a function of the i.i.d. random variables $(\epsilon_l, X_{(l-1)d_1+1}^{(1)}, \ldots, X_{ld_1}^{(1)}, \ldots, X_{(l-1)d_K+1}^{(K)}, \ldots, X_{ld_K}^{(K)})$, $1 \leq l \leq N$: changing any of these random variables change the value of $\mathcal{R}_N$ by at most $\mathcal{M}_{\mathcal{H}}/N$. One thus obtains from (3.42) with $\psi(x) = \exp(\lambda x)$, where $\lambda > 0$ is a parameter which shall be chosen later, that:

$$
\mathbb{E}\left[\exp\left(\lambda \sup_{H \in \mathcal{H}} \left|U_{\mathbf{n}}(\bar{H})\right|\right)\right] \leq \exp\left(2\lambda \mathbb{E}[\mathcal{R}_N] + \frac{\mathcal{M}_{\mathcal{H}}^2 \lambda^2}{4N}\right). \quad (3.43)
$$

Applying Chernoff's method, one then gets:

$$\mathbb{P}\left\{\sup_{H\in\mathcal{H}}\left|U_{\mathbf{n}}(\bar{H})\right|>\eta\right\}\leq\exp\left(-\lambda\eta+2\lambda\mathbb{E}[\mathcal{R}_N]+\frac{\mathcal{M}_{\mathcal{H}}^2\lambda^2}{4N}\right). \tag{3.44}$$

Using the bound (see Eq. (6) in BOUCHERON, BOUSQUET, and LUGOSI, 2005 for instance)

$$\mathbb{E}[\mathcal{R}_N]\leq\mathcal{M}_{\mathcal{H}}\sqrt{\frac{2V\log(1+N)}{N}}$$

and taking $\lambda=2N(\eta-2\mathbb{E}[\mathcal{R}_N])/\mathcal{M}_{\mathcal{H}}^2$ in (3.44), one finally establishes the desired result.

3.7.2 Proof of Theorem 11

For convenience, we introduce the random sequence $\zeta=((\zeta_k(I))_{I\in\Lambda})_{1\leq k\leq B}$, where $\zeta_k(I)$ is equal to 1 if the tuple $I=(I_1,\ \ldots,\ I_K)$ has been selected at the k-th draw and to 0 otherwise: the ζ_k's are i.i.d. random vectors and, for all $(k,I)\in\{1,\ \ldots,\ B\}\times\Lambda$, the r.v. $\zeta_k(I)$ has a Bernoulli distribution with parameter $1/|\Lambda|$. We also set $\mathbf{X}_I=(\mathbf{X}_{I_1}^{(1)},\ \ldots,\ \mathbf{X}_{I_K}^{(K)})$ for any I in Λ. Equipped with these notations, observe first that one may write: $\forall B\geq 1, \forall\mathbf{n}\in\mathbb{N}^{*K}$,

$$\widetilde{U}_B(H)-U_{\mathbf{n}}(H)=\frac{1}{B}\sum_{k=1}^{B}\mathcal{Z}_k(H),$$

where $\mathcal{Z}_k(H)=\sum_{I\in\Lambda}(\zeta_k(I)-1/|\Lambda|)H(\mathbf{X}_I)$ for any $(k,I)\in\{1,\ \ldots,\ B\}\times\Lambda$. It follows from the independence between the $\mathbf{X}_I$'s and the $\zeta(I)$'s that, for all $H\in\mathcal{H}$, conditioned upon the $\mathbf{X}_I$'s, the variables $\mathcal{Z}_1(H),\ \ldots,\ \mathcal{Z}_B(H)$ are independent, centered and almost-surely bounded by $2\mathcal{M}_{\mathcal{H}}$ (notice that $\sum_{I\in\Lambda}\zeta_k(I)=1$ for all $k\geq 1$). By virtue of Sauer's lemma, since $\mathcal{H}$ is a VC major class with finite VC dimension V, we have, for fixed $\mathbf{X}_I$'s:

$$|\{(H(\mathbf{X}_I))_{I\in\Lambda}:\ H\in\mathcal{H}\}|\leq(1+|\Lambda|)^V.$$

Hence, conditioned upon the $\mathbf{X}_I$'s, using the union bound and next Hoeffding's inequality applied to the independent sequence $\mathcal{Z}_1(H),\ \ldots,\ \mathcal{Z}_B(H)$, for all $\eta>0$, we obtain that:

$$\begin{aligned}\mathbb{P}\left(\sup_{H\in\mathcal{H}}\left|\widetilde{U}_B(H)-U_{\mathbf{n}}(H)\right|>\eta\ \mid\ (\mathbf{X}_I)_{I\in\Lambda}\right)&\\ \leq\mathbb{P}\left(\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\mathcal{Z}_k(H)\right|>\eta\mid(\mathbf{X}_I)_{I\in\Lambda}\right)&\\ \leq 2(1+|\Lambda|)^V e^{-B\eta^2/(2\mathcal{M}_{\mathcal{H}}^2)}.&\end{aligned}$$

Taking the expectation, this proves the first assertion of the theorem. Notice that this can be formulated: for any $\delta\in(0,1)$, we have with probability at least $1-\delta$:

$$\sup_{H\in\mathcal{H}}\left|\widetilde{U}_B(H)-U_{\mathbf{n}}(H)\right|\leq\mathcal{M}_{\mathcal{H}}\times\sqrt{2\frac{V\log(1+|\Lambda|)+\log(2/\delta)}{B}}.$$

Turning to the second part of the theorem, it straightforwardly results from the first part combined with Proposition 2.

3.7.3 Proof of Corollary 1

Assertion (i) is a direct application of Assertion (ii) in Theorem 11 combined with the bound $\mu(\widehat{H}_B) - \inf_{H\in\mathcal{H}} \mu(H) \leq 2\sup_{H\in\mathcal{H}} |\widetilde{U}_B(H) - \mu(H)|$.

Turning next to Assertion (ii), observe that by triangle inequality we have:

$$\mathbb{E}\left[\sup_{H\in\mathcal{H}_m} |\widetilde{U}_B(H) - \mu(H)|\right] \leq \mathbb{E}\left[\sup_{H\in\mathcal{H}_m} |\widetilde{U}_B(H) - U_{\mathbf{n}}(H)|\right] + \mathbb{E}\left[\sup_{H\in\mathcal{H}_m} |U_{\mathbf{n}}(H) - \mu(H)|\right]. \tag{3.45}$$

The same argument as that used in Theorem 11 (with $\psi(u) = u$ for any $u \geq 0$) yields a bound for the second term on the right hand side of Eq. (3.45):

$$\mathbb{E}\left[\sup_{H\in\mathcal{H}} |U_{\mathbf{n}}(H) - \mu(H)|\right] \leq 2\mathcal{M}_{\mathcal{H}}\sqrt{\frac{2V\log(1+N)}{N}}. \tag{3.46}$$

The first term can be controlled by means of the following lemma, whose proof can be found for instance in Lugosi (2002, Lemmas 1.2 and 1.3).

Lemma 1. The following assertions hold true.

(i) Hoeffding's lemma. Let Z be an integrable r.v. with mean zero such that $a \leq Z \leq b$ almost-surely. Then, we have: $\forall s > 0$

$$\mathbb{E}[\exp(sZ)] \leq \exp\left(s^2(b-a)^2/8\right).$$

(ii) Let $M \geq 1$ and $Z_1, \ldots, Z_M$ be real valued random variables. Suppose that there exists $\sigma > 0$ such that $\forall s \in \mathbb{R}$: $\mathbb{E}[\exp(sZ_i)] \leq e^{s^2\sigma^2/2}$ for all $i \in \{1, \ldots, M\}$. Then, we have:

$$\mathbb{E}\left[\max_{1\leq i\leq M} |Z_i|\right] \leq \sigma\sqrt{2\log(2M)}. \tag{3.47}$$

Assertion (i) shows that, since $-\mathcal{M}_{\mathcal{H}} \leq \mathcal{Z}_k(H) \leq \mathcal{M}_{\mathcal{H}}$ almost surely,

$$\mathbb{E}\left[\exp(s\sum_{k=1}^{B} \mathcal{Z}_k(H)) \mid (\mathbf{X}_I)_{I\in\Lambda}\right] \leq e^{\frac{1}{2}Bs^2\mathcal{M}_{\mathcal{H}}^2}.$$

With $\sigma = \mathcal{M}_{\mathcal{H}}\sqrt{B}$ and $M = |\{H(\mathbf{X}_I) : H \in \mathcal{H}\}| \leq (1+|\Lambda|)^V$, conditioning upon $(\mathbf{X}_I)_{I\in\Lambda}$, this result yields:

$$\mathbb{E}\left[\sup_{H\in\mathcal{H}} \left|\frac{1}{B}\sum_{k=1}^{B} \mathcal{Z}_k(H)\right| \mid (\mathbf{X}_I)_{I\in\Lambda}\right] \leq \mathcal{M}_{\mathcal{H}}\sqrt{\frac{2(\log 2 + V\log(1+|\Lambda|))}{B}}. \tag{3.48}$$

Integrating next over $(\mathbf{X}_I)_{I\in\Lambda}$ and combining the resulting bound with (3.45) and (3.46) leads to the inequality stated in (ii).

A bound for the expected value. For completeness, we point out that the expected value of $\sup_{H\in\mathcal{H}} |(1/B)\sum_{k=1}^{B} \mathcal{Z}_k(H)|$ can also be bounded by

means of classical symmetrization and randomization devices. Considering a "ghost" i.i.d. sample $\zeta'_1, \ldots, \zeta'_B$ independent from $((\mathbf{X}_I)_{I\in\Lambda}, \zeta)$, distributed as ζ, Jensen's inequality yields:

$$\begin{aligned}
&\mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\mathcal{Z}_k(H)\right|\right] \\
&\quad= \mathbb{E}\left[\sup_{H\in\mathcal{H}}\left\{\mathbb{E}\left[\left|\frac{1}{B}\sum_{k=1}^{B}\sum_{I\in\Lambda}H(\mathbf{X}_I)\left(\zeta_k(I)-\zeta'_k(I)\right)\right| \mid (\mathbf{X}_I)_{I\in\Lambda}\right]\right\}\right] \\
&\quad\leq \mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\sum_{I\in\Lambda}H(\mathbf{X}_I)\left(\zeta_k(I)-\zeta'_k(I)\right)\right|\right].
\end{aligned}$$

Introducing next independent Rademacher variables $\epsilon_1, \ldots, \epsilon_B$, independent from $((\mathbf{X}_I)_{I\in\Lambda}, \zeta, \zeta')$, we have:

$$\begin{aligned}
&\mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\sum_{I\in\Lambda}H(\mathbf{X}_I)\left(\zeta_k(I)-\zeta'_k(I)\right)\right| \mid (\mathbf{X}_I)_{I\in\Lambda}\right] \\
&\qquad= \mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\epsilon_k\sum_{I\in\Lambda}H(\mathbf{X}_I)\left(\zeta_k(I)-\zeta'_k(I)\right)\right| \mid (\mathbf{X}_I)_{I\in\Lambda}\right] \\
&\qquad\leq 2\mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\epsilon_k\sum_{I\in\Lambda}H(\mathbf{X}_I)\zeta_k(I)\right| \mid (\mathbf{X}_I)_{I\in\Lambda}\right].
\end{aligned}$$

We thus obtained:

$$\mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\mathcal{Z}_k(H)\right|\right] \leq 2\mathbb{E}\left[\sup_{H\in\mathcal{H}}\left|\frac{1}{B}\sum_{k=1}^{B}\epsilon_k\sum_{I\in\Lambda}H(\mathbf{X}_I)\zeta_k(I)\right|\right].$$

3.7.4 Proof of Theorem 12

We start with proving the intermediary result, stated below.

Lemma 2. Under the assumptions stipulated in Theorem 12, we have: $\forall m \geq 1, \forall \epsilon > 0$,

$$\mathbb{P}\left\{\sup_{H\in\mathcal{H}_m}|\mu(H)-\widetilde{U}_B(H)| > \alpha+\epsilon\right\} \leq \exp\left(-B^2\epsilon^2/\left(2(B+n)\mathcal{M}^2_{\mathcal{H}_m}\right)\right).$$

where

$$\alpha = 2\mathcal{M}_{\mathcal{H}_m}\left(\sqrt{\frac{2V_m\log(1+N)}{N}}+\sqrt{\frac{2(\log 2+V_m\log(1+|\Lambda|))}{B}}\right).$$

Proof. This is a direct application of the bounded difference inequality (see McDiarmid, 1989) applied to the quantity $\sup_{H\in\mathcal{H}_m}|\mu(H)-\widetilde{U}_B(H)|$, which is viewed here as a function of the $(B+n)$ independent random variables $(X_1^{(1)}, X_{n_K}^{(K)}, \epsilon_1, \ldots, \epsilon_B)$ (jumps being bounded by $2\mathcal{M}_H/B$), combined with Assertion (ii) of Corollary 1. □

Let $m \geq 1$ and decompose the expected excess of risk of the rule picked by means of the complexity regularized incomplete U-statistic criterion as follows:

$$\mathbb{E}\left[\mu(\widehat{H}_{B,\widehat{m}}) - \mu^*_m\right] = \mathbb{E}\left[\mu(\widehat{H}_{B,\widehat{m}}) - \widetilde{U}_B(\widehat{H}_{B,\widehat{m}}) - \text{pen}(B,\widehat{m})\right] + \mathbb{E}\left[\inf_{j\geq 1}\left\{\widetilde{U}_B(\widehat{H}_{B,j}) + \text{pen}(B,j)\right\} - \mu^*_m\right],$$

where we set $\mu^*_m = \inf_{H\in\mathcal{H}_m}\mu(H)$. In order to bound the first term on the right hand side of the equation above, observe that we have: $\forall \epsilon > 0$,

$$\begin{aligned}
&\mathbb{P}\left\{\mu(\widehat{H}_{B,\widehat{m}}) - \widetilde{U}_B(\widehat{H}_{B,\widehat{m}}) - \text{pen}(B,\widehat{m}) > \epsilon\right\} \\
&\quad\leq \mathbb{P}\left\{\sup_{j\geq 1}\left\{\mu(\widehat{H}_{B,j}) - \widetilde{U}_B(\widehat{H}_{B,j}) - \text{pen}(B,j)\right\} > \epsilon\right\} \\
&\quad\leq \sum_{j\geq 1}\mathbb{P}\left\{\mu(\widehat{H}_{B,j}) - \widetilde{U}_B(\widehat{H}_{B,j}) - \text{pen}(B,j) > \epsilon\right\} \\
&\quad\leq \sum_{j\geq 1}\mathbb{P}\left\{\sup_{H\in\mathcal{H}_j}|\mu(\widehat{H}) - \widetilde{U}_B(H)| - \text{pen}(B,j) > \epsilon\right\} \\
&\leq \sum_{j\geq 1}\exp\left(-\frac{-B^2}{2(B+n)\mathcal{M}^2}\left(\epsilon + 2\mathcal{M}\sqrt{\frac{(B+n)\log j}{B^2}}\right)^2\right) \\
&\quad\leq \exp\left(-\frac{B^2\epsilon^2}{2(B+n)\mathcal{M}^2}\right)\sum_{j\geq 1}1/j^2 \leq 2\exp\left(-\frac{B^2\epsilon^2}{2(B+n)\mathcal{M}^2}\right),
\end{aligned}$$

using successively the union bound and Lemma 2. Integrating over $[0,+\infty)$, we obtain that:

$$\mathbb{E}\left[\mu(\widehat{H}_{B,\widehat{m}}) - \widetilde{U}_B(\widehat{H}_{B,\widehat{m}}) - \text{pen}(B,\widehat{m})\right] \leq \mathcal{M}\frac{\sqrt{2\pi(B+n)}}{B}. \tag{3.49}$$

Considering now the second term, notice that

$$\begin{aligned}
\mathbb{E}\left[\inf_{j\geq 1}\left\{\widetilde{U}_B(\widehat{H}_{B,j}) + \text{pen}(B,j)\right\} - \mu^*_m\right] &\leq \mathbb{E}\left[\widetilde{U}_B(\widehat{H}_{B,m}) + \text{pen}(B,m) - \mu^*_m\right] \\
&\leq \text{pen}(B,m).
\end{aligned}$$

Combining the bounds, we obtain that: $\forall m \geq 1$,

$$\mathbb{E}\left[\mu(\widehat{H}_{B,\widehat{m}})\right] \leq \mu^*_m + \text{pen}(B,m) + \mathcal{M}\frac{\sqrt{2\pi(B+n)}}{B}.$$

The oracle inequality is thus proved.

3.7.5 Proof of Theorem 13

We start with proving the following intermediary result, based on the U-statistic version of the Bernstein exponential inequality.

Lemma 3. Suppose that the assumptions of Theorem 13 are fulfilled. Then, for all $\delta \in (0,1)$, we have with probability larger than $1-\delta$: $\forall r \in \mathcal{R}$, $\forall n \geq 2$,

$$0 \leq \Lambda_n(r) - \Lambda(r) + \sqrt{\frac{2c\Lambda(r)^\alpha \log(\#\mathcal{R}/\delta)}{n}} + \frac{4\log(\#\mathcal{R}/\delta)}{3n}.$$

Proof. The proof is a straightforward application of Theorem A on p. 201 in SERFLING, 1980, combined with the union bound and Assumption 1. □

The same argument as that used to prove Assertion (i) in Theorem 11 (namely, freezing the $\mathbf{X}_I$'s, applying Hoeffding inequality and the union bound) shows that, for all $\delta \in (0,1)$, we have with probability at least $1-\delta$: $\forall r \in \mathcal{R}$,

$$0 \leq \widetilde{U}_B(q_r) - U_{\mathbf{n}}(q_r) + \sqrt{\frac{M + \log(M/\delta)}{B}}$$

for all $n \geq 2$ and $B \geq 1$ (observe that $\mathcal{M}_{\mathcal{H}} \leq 1$ in this case). Now, combining this bound with the previous one and using the union bound, one gets that, for all $\delta \in (0,1)$, we have with probability larger than $1-\delta$: $\forall r \in \mathcal{R}$, $\forall n \geq 2$, $\forall B \geq 1$,

$$0 \leq \widetilde{U}_B(q_r) - \Lambda(r) + \sqrt{\frac{2c\Lambda(r)^\alpha \log(2M/\delta)}{n}} + \frac{4\log(2M/\delta)}{3n} + \sqrt{\frac{M + \log(2M/\delta)}{B}}.$$

Observing that, $\widetilde{U}_B(q_{\widetilde{r}_B}) \leq 0$ by definition, we thus have with probability at least $1-\delta$:

$$\Lambda(\widetilde{r}_B) \leq \sqrt{\frac{2c\Lambda(\widetilde{r}_B)^\alpha \log(2M/\delta)}{n}} + \frac{4\log(2M/\delta)}{3n} + \sqrt{\frac{M + \log(2M/\delta)}{B}}.$$

Choosing finally $B = O(n^{2/(2-\alpha)})$, the desired result is obtained by solving the inequality above for $\Lambda(\widetilde{r}_B)$.

3.7.6 Proof of Theorem 14

As shown by the following lemma, which is a slight modification of Lemma 1 in JANSON, 1984, the deviation between the incomplete U-statistic and its complete version is of order $O_{\mathbb{P}})(1/\sqrt{B})$ for both sampling schemes.

Lemma 4. Suppose that the assumptions of 14 are fulfilled. Then, we have: $\forall H \in \mathcal{H}$,

$$\mathbb{E}\left[\left(\bar{U}_{HT}(H) - U_{\mathbf{n}}(H)\right)^2 \mid (\mathbf{X}_I)_{I\in\Lambda}\right] \leq 2\mathcal{M}_{\mathcal{H}}^2/B.$$

Proof. Observe first that, in both cases (sampling without replacement and Bernoulli sampling), we have: $\forall I \neq J$ in Λ,

$$\mathbb{E}\left[\left(\Delta(I) - \frac{B}{|\Lambda|}\right)^2\right] \leq \frac{B}{|\Lambda|} \text{ and } \mathbb{E}\left[\left(\Delta(I) - \frac{B}{|\Lambda|}\right)\left(\Delta(J) - \frac{B}{|\Lambda|}\right)\right] \leq \frac{1}{|\Lambda|}\cdot\frac{B}{|\Lambda|}.$$

Hence, as $(\Delta(I))_{I\in\Lambda}$ and $(\mathbf{X}_I)_{I\in\Lambda}$ are independent by assumption, we have:

$$
\begin{aligned}
B^2\mathbb{E}\Big[\big(\bar{U}_{HT}(H) - U_{\mathbf{n}(H)}\big)^2 \mid (\mathbf{X}_I)_{I\in\Lambda}\Big] & \\
&= \mathbb{E}\left[\left(\sum_{I\in\Lambda}\left(\Delta(I) - \frac{B}{|\Lambda|}\right)H(\mathbf{X}_I)\right)^2 \mid (\mathbf{X}_I)_{I\in\Lambda}\right] \\
&\leq \mathcal{M}_{\mathcal{H}}^2 \sum_{I\in\Lambda}\mathbb{E}\left[\left(\Delta(I) - \frac{B}{|\Lambda|}\right)^2\right] \\
&\quad + \mathcal{M}_{\mathcal{H}}^2 \sum_{I\neq J}\mathbb{E}\left[\left(\Delta(I) - \frac{B}{|\Lambda|}\right)\left(\Delta(J) - \frac{B}{|\Lambda|}\right)\right] \\
&\leq 2B\mathcal{M}_{\mathcal{H}}^2.
\end{aligned}
$$

□

Consider first the case of Bernoulli sampling. By virtue of Bernstein inequality applied to the independent variables $(\Delta(I) - B/|\Lambda|)H(\mathbf{X}_I)$ conditioned upon $(\mathbf{X}_I)_{I\in\Lambda}$, we have: $\forall H \in \mathcal{H}$, $\forall t > 0$,

$$
\mathbb{P}\left\{\left|\sum_{I\in\Lambda}(\Delta(I) - B/|\Lambda|)H(\mathbf{X}_I)\right| > t \mid (\mathbf{X}_I)_{I\in\Lambda}\right\} \leq 2e^{-\frac{t^2}{4B\mathcal{M}_{\mathcal{H}}^2 + 2\mathcal{M}_{\mathcal{H}}t/3}}.
$$

Hence, combining this bound and the union bound, we obtain that: $\forall t > 0$,

$$
\mathbb{P}\left\{\sup_{H\in\mathcal{H}}\left|\bar{U}_{HT}(H) - U_{\mathbf{n}(H)}\right| > t \mid (\mathbf{X}_I)_{I\in\Lambda}\right\} \leq 2(1+|\Lambda|)^V e^{-\frac{Bt^2}{4\mathcal{M}_{\mathcal{H}}^2 + 2\mathcal{M}_{\mathcal{H}}t/3}}.
$$

Solving

$$
\delta = 2(1+|\Lambda|)^V \exp\left(-\frac{Bt^2}{4\mathcal{M}_{\mathcal{H}}^2 + 2\mathcal{M}_{\mathcal{H}}t/3}\right)
$$

yields the desired bound.

Consider next the case of the sampling without replacement scheme. Using the exponential inequality tailored to this situation proved in SERFLING, 1974 (see Corollary 1.1 therein), we obtain: $\forall H \in \mathcal{H}$, $\forall t > 0$,

$$
\mathbb{P}\left\{\frac{1}{B}\left|\sum_{I\in\Lambda}(\Delta(I) - B/|\Lambda|)H(\mathbf{X}_I)\right| > t \mid (\mathbf{X}_I)_{I\in\Lambda}\right\} \leq 2\exp\left(-\frac{Bt^2}{2\mathcal{M}_{\mathcal{H}}^2}\right).
$$

The proof can be then ended using the union bound, just like above.

3.7.7 Proof of Proposition 4

For simplicity, we focus on one sample U-statistics of degree two ($K = 1$, $d_1 = 2$) since the argument easily extends to the general case. Let $U_n(H)$ be a non-degenerate U-statistic of degree two:

$$
U_n(H) = \frac{2}{n(n-1)}\sum_{i<j} H(x_i, x_j).
$$

In order to express the variance of $U_n(H)$ based on its second Hoeffding decomposition (see Section 3.2.1), we first introduce more notations: $\forall (x,x') \in \mathcal{X}_1^2$,

$$H_1(x) \stackrel{def}{=} \mathbb{E}\left[H(x,X)\right] - \mu(H)$$

and

$$H_2(x,x') \stackrel{def}{=} H(x,x') - \mu(H) - H_1(x) - H_1(x').$$

Equipped with these notations, the (orthogonal) Hoeffding/Hajek decomposition of $U_n(H)$ can be written as

$$U_n(H) = \mu(H) + 2T_n(H) + W_n(H),$$

involving centered and decorrelated random variables given by

$$\begin{aligned} T_n(H) &= \frac{1}{n}\sum_{i=1}^{n} H_1(x_i), \\ W_n(H) &= \frac{2}{n(n-1)}\sum_{i<j} H_2(x_i,x_j). \end{aligned}$$

Recall that the U-statistic $W_n(H)$ is said to be degenerate, since $\mathbb{E}[H_2(x,X)] = 0$ for all $x \in \mathcal{X}_1$. Based on this representation and setting $\sigma_1^2 = \mathrm{Var}[H_1(X)]$ and $\sigma_2^2 = \mathrm{Var}[H_2(X,X')]$, the variance of $U_n(H)$ is given by

$$\mathrm{Var}[U_n(H)] = \frac{4\sigma_1^2}{n} + \frac{2\sigma_2^2}{n(n-1)}. \tag{3.50}$$

As already pointed out in Section 3.3.1, the variance of the incomplete U-statistic built by sampling with replacement is

$$\begin{aligned} \mathrm{Var}[\widetilde{U}_B(H)] &= \mathrm{Var}[U_n(H)] + \frac{1}{B}\left(1 - \frac{2}{n(n-1)}\right)\mathrm{Var}[H(X,X')] \\ &= \mathrm{Var}[U_n(H)] + \frac{1}{B}\left(1 - \frac{2}{n(n-1)}\right)(2\sigma_1^2 + \sigma_2^2). \end{aligned} \tag{3.51}$$

Take $B = n'(n'-1)$ for $n' \ll n$. It follows from (3.50) and (3.51) that in the asymptotic framework (3.4), the quantities $\mathrm{Var}[U_{n'}(H)]$ and $\mathrm{Var}[\widetilde{U}_B(H)]$ are of the order $O(1/n')$ and $O(1/n'^2)$ respectively as $n' \to +\infty$. Hence these convergence rates hold for $\widetilde{g}_{\mathbf{n}'}(\theta)$ and $\widetilde{g}_B(\theta)$ respectively.

Chapter 4

Extending Gossip Algorithms to Estimation of U-statistics

4.1 Introduction

Decentralized computation and estimation have many applications in sensor and peer-to-peer networks as well as for extracting knowledge from massive information graphs such as interlinked Web documents and online social media. Algorithms running on such networks must often operate under tight constraints: the nodes forming the network cannot rely on a centralized entity for communication and synchronization, without being aware of the global network topology and/or have limited resources (computational power, memory, energy). Gossip algorithms TSITSIKLIS, 1984; SHAH, 2009; DIMAKIS et al., 2010, where each node exchanges information with at most one of its neighbors at a time, have emerged as a simple yet powerful technique for distributed computation in such settings. Given a data observation on each node, gossip algorithms can be used to compute averages or sums of functions of the data that are *separable across observations* (see for example KEMPE, DOBRA, and GEHRKE, 2003; BOYD et al., 2006; MOSK-AOYAMA and SHAH, 2008; KOWALCZYK and VLASSIS, 2004; KARP et al., 2000 and references therein). Unfortunately, these algorithms cannot be used to efficiently compute quantities that take the form of an average over *pairs of observations*, also known as U-statistics LEE, 1990b. Among classical U-statistics used in machine learning and data mining, one can mention, among others: the sample variance, the Area Under the Curve (AUC) of a classifier on distributed data, the Gini mean difference, the Kendall tau rank correlation coefficient, the within-cluster point scatter and several statistical hypothesis test statistics such as Wilcoxon Mann-Whitney MANN and WHITNEY, 1947.

In this chapter, we propose randomized synchronous and asynchronous gossip algorithms to efficiently compute a U-statistic, in which each node maintains a local estimate of the quantity of interest throughout the execution of the algorithm. Our methods rely on two types of iterative information exchange in the network: propagation of local observations across the network, and averaging of local estimates. We show that the local estimates generated by our approach converge in expectation to the value of the U-statistic at rates of $O(1/t)$ and $O(\log t/t)$ for the synchronous and asynchronous versions respectively, where t is the number of iterations. These convergence bounds feature data-dependent terms that reflect the hardness of the estimation problem, and network-dependent terms related to the spectral gap of the network graph CHUNG, 1997, showing that our algorithms are faster on well-connected networks. The proofs rely on an original reformulation of the problem using "phantom nodes", *i.e.,* on additional nodes that account for data propagation in the network. Our results largely improve upon those presented in PELCKMANS and SUYKENS, 2009: in particular, we achieve faster convergence together with lower memory and communication costs. Experiments conducted on AUC and within-cluster point scatter estimation using real data confirm the superiority of our approach.

The rest of this chapter is organized as follows. Section 4.2 introduces the problem of interest as well as relevant notation. Section 4.3 provides a brief review of the related work in gossip algorithms. We then describe our approach along with the convergence analysis in Section 4.4, both in synchronous and asynchronous settings. Section 4.5 presents numerical results.

Finally, concluding remarks are presented in Section 4.6.

4.2 Background

4.2.1 Definitions and Notations

For any integer $p > 0$, we denote by $[p]$ the set $\{1, \dots, p\}$ and by $|\mathcal{C}|$ the cardinality of any finite set $\mathcal{C}$. We represent a network of size $n > 0$ as an undirected graph $\mathcal{G} = ([n], \mathcal{E})$, where $[n]$ is the set of vertices and $\mathcal{E} \subseteq [n] \times [n]$ the set of edges. We denote by $\mathbf{A}^{\mathcal{G}}$ the adjacency matrix related to the graph $\mathcal{G}$, that is for all $(i, j) \in [n]^2$, $[\mathbf{A}^{\mathcal{G}}]_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$. For any node $i \in [n]$, we denote its degree by $d_i = |\{j : (i, j) \in \mathcal{E}\}|$. We denote by $\mathbf{L}^{\mathcal{G}}$ the graph Laplacian of $\mathcal{G}$, defined by $\mathbf{L}^{\mathcal{G}} = \mathbf{D}^{\mathcal{G}} - \mathbf{A}^{\mathcal{G}}$ where $\mathbf{D}^{\mathcal{G}} = \mathrm{diag}(d_1, \dots, d_n)$ is the matrix of degrees. When it is clear from context, we will drop the $\mathcal{G}$ exponent. A graph $\mathcal{G} = ([n], \mathcal{E})$ is said to be connected if for all $(i, j) \in [n]^2$ there exists a path connecting i and j; it is bipartite if there exist $\mathcal{S}, \mathcal{T} \subset [n]$ such that $\mathcal{S} \cup \mathcal{T} = [n]$, $\mathcal{S} \cap \mathcal{T} = \emptyset$ and $\mathcal{E} \subseteq (\mathcal{S} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S})$.

A matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is nonnegative (resp. positive) if and only if for all $(i, j) \in [n]^2$, $[\mathbf{M}]_{ij} \geq 0$, (resp. $[\mathbf{M}]_{ij} > 0$). We write $\mathbf{M} \geq 0$ (resp. $\mathbf{M} > 0$) when this holds. The transpose of $\mathbf{M}$ is denoted by $\mathbf{M}^\top$. A matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ is stochastic if and only if $\mathbf{P} \geq 0$ and $\mathbf{P}\mathbf{1}_n = \mathbf{1}_n$, where $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$. The matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ is bi-stochastic if and only if $\mathbf{P}$ and $\mathbf{P}^\top$ are stochastic. We denote by $\mathbf{I}_n$ the identity matrix in $\mathbb{R}^{n \times n}$, $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ the standard basis in $\mathbb{R}^n$, $\mathbb{1}_{\{\mathcal{X}\}}$ the indicator function of an event $\mathcal{X}$ and $\|\cdot\|$ the usual ℓ_2 norm.

4.2.2 Problem Statement

Let $\mathcal{X}$ be an input space and $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{X}^n$ a sample of $n \geq 2$ points in that space. We assume $\mathcal{X} \subseteq \mathbb{R}^d$ for some $d > 0$ throughout the paper, but our results straightforwardly extend to the more general setting. We denote as $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$ the design matrix. Let $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a measurable function, symmetric in its two arguments and with $h(\mathbf{x}, \mathbf{x}) = 0$, $\forall \mathbf{x} \in \mathcal{X}$. We consider the problem of estimating the following quantity, known as a degree two U-statistic LEE, 1990b:

$$\hat{U}_n(h) = \frac{1}{n^2} \sum_{i,j=1}^{n} h(\mathbf{x}_i, \mathbf{x}_j). \tag{4.1}$$

Notice that this is slightly abusive insofar as one generally defines the U-statistic as the unbiased estimator of the parameter $\mathbb{E}[h(X_1, X_2)]$, differing from (4.1) by a factor of $n/(n-1)$ (and their difference is of order $O_{\mathbb{P}}(n^{-3/2})$ provided that $h(X_1, X_2)$ is square integrable). When clear from context, we will use the notation $\hat{U}_n$. We define $\mathbf{H} \in \mathbb{R}^{n \times n}$ such that, for any $1 \leq k, l \leq n$,

$$[\mathbf{H}]_{kl} := h(\mathbf{x}_k, \mathbf{x}_l). \tag{4.2}$$

Finally, we define $\overline{\mathbf{h}} = \mathbf{H}\mathbf{1}_n/n$ the vector of partial sums.

In this chapter, we will illustrate the interest of U-statistics on two applications, among many others. The first one is the within-cluster point scatter CLÉMENÇON, 2011, which measures the clustering quality of a partition $\mathcal{P}$ of $\mathcal{X}$ as the average distance between points in each cell $\mathcal{C} \in \mathcal{P}$. It is of the

form (4.1) with

$$h_{\mathcal{P}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| \cdot \sum_{\mathcal{C} \in \mathcal{P}} \mathbb{1}_{\{(\mathbf{x}, \mathbf{x}') \in \mathcal{C}^2\}}. \tag{4.3}$$

This criterion measures the average distance between points of each cell. Since the goal of clustering is to group close points together, the lower the within-cluster point scatter, the better the partition. We also study the AUC measure HANLEY and MCNEIL, 1982. For a given sample $(\mathbf{x}_1, \ell_1), \ldots, (\mathbf{x}_n, \ell_n)$ on $\mathcal{X} \times \{-1, +1\}$, the AUC measure of a linear classifier $\boldsymbol{\theta} \in \mathbb{R}^{d-1}$ is given by:

$$\mathrm{AUC}(\boldsymbol{\theta}) = \frac{\displaystyle\sum_{1 \leq i,j \leq n} (1 - \ell_i \ell_j) \mathbb{1}_{\{\ell_i (\boldsymbol{\theta}^\top \mathbf{x}_i) > -\ell_j (\boldsymbol{\theta}^\top \mathbf{x}_j)\}}}{4 \left(\displaystyle\sum_{1 \leq i \leq n} \mathbb{1}_{\{\ell_i = 1\}} \right) \left(\displaystyle\sum_{1 \leq i \leq n} \mathbb{1}_{\{\ell_i = -1\}} \right)}. \tag{4.4}$$

This score is the probability for a classifier to rank a positive observation higher than a negative one.

We focus here on the *decentralized setting*, where the data sample is partitioned across a set of nodes in a network. We are interested in estimating (4.1) efficiently using a gossip algorithm.

4.3 Related Work

4.3.1 Sample mean estimation

Gossip algorithms have been extensively studied in the context of decentralized averaging in networks, where the goal is to compute the average of n real vectors ($\mathcal{X} = \mathbb{R}^d$):

$$\bar{\mathbf{x}}_n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i = \frac{1}{n}\mathbf{X}^\top \mathbf{1}_n. \tag{4.5}$$

One of the earliest work on this canonical problem is due to TSITSIKLIS, 1984, but more efficient algorithms have recently been proposed, see for instance KEMPE, DOBRA, and GEHRKE, 2003; BOYD et al., 2006. Of particular interest to us is the work of BOYD et al., 2006, which introduces a randomized gossip algorithm for computing the empirical mean (4.5) in a context where nodes wake up asynchronously and simply average their local estimate with that of a randomly chosen neighbor. The communication probabilities are given by a stochastic matrix $\mathbf{P}$, where $[\mathbf{P}]_{ij}$ is the probability that a node i selects neighbor j at a given iteration. As long as the network graph is connected and non-bipartite, the local estimates converge to (4.5) at a rate $O(e^{-ct})$ where the constant c can be tied to the spectral gap of the network graph CHUNG, 1997, showing faster convergence for well-connected networks. [1] Such algorithms can be extended to compute other functions such as maxima and minima, or sums of the form $\sum_{i=1}^{n} g(\mathbf{x}_i)$ for some function $g : \mathcal{X} \to \mathbb{R}$ (as done for instance in MOSK-AOYAMA and SHAH, 2008). Some work has also gone into developing faster gossip algorithms for poorly connected networks, assuming that nodes know their (partial) geographic location DIMAKIS, SARWATE, and WAINWRIGHT, 2008; LI, DAI, and ZHANG, 2010. For a detailed account of the literature on gossip algorithms, we refer the reader to SHAH, 2009; DIMAKIS et al., 2010.

4.3.2 U-statistics estimation

Existing gossip algorithms for estimating sample mean cannot be extended to efficiently compute (4.1) as it depends on *pairs* of observations. To the best of our knowledge, this problem has only been investigated in PELCKMANS and SUYKENS, 2009. In this paper, the authors tackle two distinct problems. First, their algorithm U1-GOSSIP, described in Algorithm 9, allows each node $i \in [n]$ to estimate the partial U-statistic:

$$\hat{U}_n^{(i)} := \frac{1}{n}\sum_{j=1}^{n} h(\mathbf{x}_i, \mathbf{x}_j). \tag{4.6}$$

The spirit of the algorithm is a bit different from the standard gossip case. For each node $k \in [n]$, an estimator $z_k(t)$ is initialized to zero and an auxiliary observation $\mathbf{y}_k(t)$ is initialized to $\mathbf{x}_k$. At each iteration t, an edge (i, j) is picked uniformly at random over $\mathcal{E}$ [2] and the corresponding nodes swap

1. For the sake of completeness, we provide an analysis of this algorithm in the last section of this chapter.

2. This accounts for modelling the uncertainty in communication capabilities over the network. See Appendix for details on clock modelling.

Algorithm 9 U1-GOSSIP algorithm for computing (4.6)

Require: Each node k holds observation $\mathbf{x}_k$
1: Each node initializes its auxiliary observation $\mathbf{y}_k \leftarrow \mathbf{x}_k$
2: Each node initializes its estimator $z_k \leftarrow 0$
3: **for** $t = 1, 2, \ldots$ **do**
4: Draw (i, j) uniformly at random from $\mathcal{E}$
5: Nodes i and j swap their auxiliary observations: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
6: **for** $k = 1, \ldots, n$ **do**
7: $z_k \leftarrow \frac{t-1}{t} z_k + \frac{1}{t} h(\mathbf{x}_k, \mathbf{y}_k)$
8: **end for**
9: **end for**
10: **return** Each node k has z_k

their auxiliary observations:

$$\begin{cases} \mathbf{y}_i(t) &= \mathbf{y}_j(t-1), \\ \mathbf{y}_j(t) &= \mathbf{y}_i(t-1). \end{cases}$$

Then, *every* node updates its estimator using its pair of observations:

$$z_k(t) = \frac{t-1}{t} z_k(t-1) + \frac{1}{t} h(\mathbf{x}_k, \mathbf{y}_k(t)), \forall k \in [n].$$

This method can be shown to converge at a $O(1/t)$ rate, as stated by the theorem below.

Theorem 15. *Let us assume that $\mathcal{G} = ([n], \mathcal{E})$ is connected and non bipartite. Then, for $\mathbf{z}(t) = (z_1(t), \ldots, z_n(t))^\top$ defined in Algorithm 9, we have that for all $k \in [n]$:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n} \sum_{l=1}^{n} h(\mathbf{x}_k, \mathbf{x}_l) = \hat{U}_n^{(k)}$$

Moreover, for any $t > 0$ and any $k \in [n]$,

$$\left| \mathbb{E}[z_k(t)] - \hat{U}_n^{(k)} \right| \leq \frac{1}{ct} \|\mathbf{H}\mathbf{e}_k\|,$$

where $c = \frac{2\beta_{n-1}}{|\mathcal{E}|} > 0$ and β_{n-1} is the second smallest eigenvalue of $\mathbf{L}$.

Proof. See Section 4.7. □

PELCKMANS and SUYKENS, 2009 also tackles the complete U-statistic estimation. Their algorithm, coined U2-GOSSIP, also achieves $O(1/t)$ convergence rate. Each node stores two auxiliary observations that are propagated using independent random walks. These two auxiliary observations will then be used for estimating the U-statistic – see Algorithm 10 for details.

The following theorem states an explicit upper bound on the expected estimate error.

Algorithm 10 U2-GOSSIP PELCKMANS and SUYKENS, 2009

Require: Each node k holds observation $\mathbf{x}_k$

Each node initializes $\mathbf{y}_k^{(1)} \leftarrow \mathbf{x}_k$, $\mathbf{y}_k^{(2)} \leftarrow \mathbf{x}_k$, $z_k \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
  for $p = 1, \ldots, n$ **do**
    $z_p \leftarrow \frac{t-1}{t} z_p + \frac{1}{t} h(\mathbf{y}_p^{(1)}, \mathbf{y}_p^{(2)})$
  end for
  Draw (i, j) uniformly at random from $\mathcal{E}$
  Nodes i and j swap their first auxiliary observations: $\mathbf{y}_i^{(1)} \leftrightarrow \mathbf{y}_j^{(1)}$
  Draw (k, l) uniformly at random from $\mathcal{E}$
  Nodes k and l swap their second auxiliary observations: $\mathbf{y}_k^{(2)} \leftrightarrow \mathbf{y}_l^{(2)}$
end for
return Each node k has z_k

Theorem 16. *Let us assume that $\mathcal{G}$ is connected and non bipartite. Then, for $\mathbf{z}(t) = (z_1(t), \ldots, z_n(t))$ defined in Algorithm 10, we have that for all $k \in [n]$:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n$$

Moreover, for any $t > 0$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n \mathbf{1}_n \right\| \leq \frac{\sqrt{n}}{t} \left(\frac{1}{c} \cdot \left\| \overline{\mathbf{h}} - \hat{U}_n \mathbf{1}_n \right\| + \frac{1}{c'} \cdot \left\| \mathbf{H} - \overline{\mathbf{h}} \mathbf{1}_n^\top \right\| \right),$$

where $c = \frac{\beta_{n-1}}{|\mathcal{E}|}$, $c' = 4c \left(1 - \frac{\beta_{n-1}}{|\mathcal{E}|}\right)$ and β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$.

Proof. See Section 4.7. □

The $O(1/t)$ convergence rate is satisfying in the sense that it matches the partial estimation rate. This algorithm however has several drawbacks. First, each node must store two auxiliary observations, and two pairs of nodes must exchange an observation at each iteration. For high-dimensional problems (large d), this leads to a significant memory and communication load. Second, the algorithm is not asynchronous as every node must update its estimate at each iteration. Consequently, nodes must have access to a global clock, which is often unrealistic in practice.

In the next section, we introduce new synchronous and asynchronous algorithms with faster convergence as well as smaller memory and communication cost per iteration.

Algorithm 11 GoSta-sync: a synchronous gossip algorithm for computing a U-statistic

Require: Each node k holds observation $\mathbf{x}_k$

Each node k initializes its auxiliary observation $\mathbf{y}_k = \mathbf{x}_k$ and its estimate $z_k = 0$

for $t = 1, 2, \ldots$ **do**
  for $p = 1, \ldots, n$ **do**
    Set $z_p \leftarrow \frac{t-1}{t} z_p + \frac{1}{t} H(\mathbf{x}_p, \mathbf{y}_p)$
  end for
  Draw (i, j) uniformly at random from $\mathcal{E}$
  Set $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
  Swap auxiliary observations of nodes i and j: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
end for
return Each node k has z_k

4.4 GoSta Algorithms

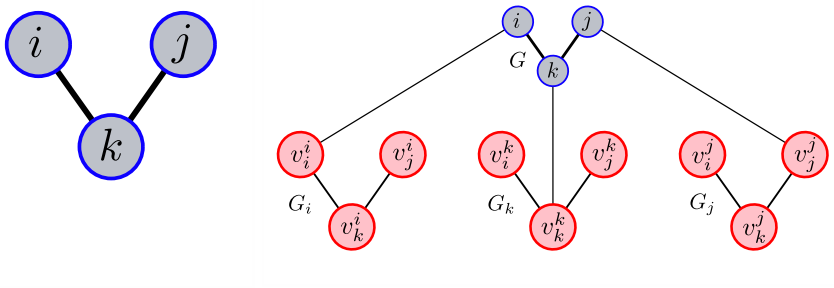
In this section, we introduce gossip algorithms for computing (4.1). Our approach is based on the observation that $\hat{U}_n = n^{-1} \sum_{i=1}^n \hat{U}_n^{(i)}$, with $\hat{U}_n^{(i)}$ defined in (4.6).The goal is thus similar to the usual distributed averaging problem (4.5), with the key difference that each local value $\hat{U}_n^{(i)}$ is itself an average depending on the entire data sample. Consequently, our algorithms will combine two steps at each iteration: a data propagation step to allow each node i to estimate $\hat{U}_n^{(i)}$, and an averaging step to ensure convergence to the desired value $\hat{U}_n$. We first present the algorithm and its analysis for the (simpler) synchronous setting in Section 4.4.1, before introducing an asynchronous version (Section 4.4.2).

4.4.1 Synchronous Setting

In the synchronous setting, we assume that the nodes have access to a global clock so that they can all update their estimate at each time instance. We stress that the nodes need not to be aware of the global network topology as they will only interact with their direct neighbors in the graph.

Let us denote by $z_k(t)$ the (local) estimate of $\hat{U}_n$ by node k at iteration t. In order to propagate data across the network, each node k maintains an auxiliary observation $\mathbf{y}_k$, initialized to $\mathbf{x}_k$. Our algorithm, coined GoSta, goes as follows. At each iteration, each node k updates its local estimate by taking the running average of $\mathbf{z}_k(t)$ and $h(\mathbf{x}_k, \mathbf{y}_k)$. Then, an edge of the network is drawn uniformly at random, and the corresponding pair of nodes average their local estimates and swap their auxiliary observations. The observations are thus each performing a random walk (albeit coupled) on the network graph. The full procedure is described in Algorithm 5.

In order to prove the convergence of Algorithm 5, we consider an equivalent reformulation of the problem which allows us to model the data propagation and the averaging steps separately. Specifically, for each $k \in [n]$, we define a phantom $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ of the original network $\mathcal{G}$, with $\mathcal{V}_k = \{v_i^{(k)}, 1 \leq i \leq n\}$ and $\mathcal{E}_k = \{(v_i^{(k)}, v_j^{(k)}); (i, j) \in \mathcal{E}\}$. We then create a



(A) Original graph $\mathcal{G}$. (B) New graph $\tilde{\mathcal{G}}$.

FIGURE 4.1: Comparison of original network and "phantom network".

new graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ where each node $k \in [n]$ is connected to its counterpart $v_k^{(k)} \in \mathcal{V}_k$:

$$\begin{cases} \tilde{\mathcal{V}} &= [n] \cup (\cup_{k=1}^n \mathcal{V}_k) \\ \tilde{\mathcal{E}} &= \mathcal{E} \cup (\cup_{k=1}^n \mathcal{E}_k) \cup \{(k, v_k^{(k)}); k \in [n]\} \end{cases}$$

The construction of $\tilde{\mathcal{G}}$ is illustrated in Figure 4.1. In this new graph, the nodes $1, \dots, n$ from the original network will respectively hold the estimates $z_1(t), \dots, z_n(t)$ as described above. The role of each $\mathcal{G}_k$ is to simulate the data propagation in the original graph $\mathcal{G}$. For $1 \leq k, l \leq n$, $v_l^{(k)} \in \mathcal{V}^k$ initially holds the value $h(\mathbf{x}_k, \mathbf{x}_l)$. At each iteration, we draw a random edge (i, j) of $\mathcal{G}$ and nodes $v_i^{(k)}$ and $v_j^{(k)}$ swap their value for all $k \in [n]$. To update its estimate, each node k will use the current value at $v_k^{(k)}$.

We can now represent the system state at iteration t by a vector $\mathbf{s}(t) = (\mathbf{s}_1(t)^\top, \mathbf{s}_2(t)^\top)^\top \in \mathbb{R}^{n+n^2}$. The first n coefficients, $\mathbf{s}_1(t)$, are associated with nodes in $[n]$ and correspond to the estimate vector $\mathbf{z}(t) = (z_1(t), \dots, z_n(t))^\top$. The last n^2 coefficients, $\mathbf{s}_2(t)$, are associated with nodes in $(\mathcal{V}_k)_{1 \leq k \leq n}$ and represent the data propagation in the network. Their initial value is set to $\mathbf{s}_2(0) = (\mathbf{e}_1^\top \mathbf{H}, \dots, \mathbf{e}_n^\top \mathbf{H})^\top$ so that for any $(k, l) \in [n]^2$, node $v_l^{(k)}$ initially stores the value $h(\mathbf{x}_k, \mathbf{x}_l)$.

Remark 5. The "phantom network" $\tilde{\mathcal{G}}$ is of size $O(n^2)$, but we stress the fact that it is used solely as a tool for the convergence analysis: Algorithm 5 operates on the original graph $\mathcal{G}$.

The transition matrix of this system accounts for three events: the *averaging step* (the action of $\mathcal{G}$ on itself), the *data propagation* (the action of $\mathcal{G}_k$ on itself for all $k \in [n]$) and the *estimate update* (the action of $\mathcal{G}_k$ on node k for all $k \in [n]$). At a given step $t > 0$, we are interested in characterizing the transition matrix $\mathbf{M}(t)$ such that $\mathbb{E}[\mathbf{s}(t+1)] = \mathbf{M}(t)\mathbb{E}[\mathbf{s}(t)]$. For the sake of clarity, we write $\mathbf{M}(t)$ as an upper block-triangular $(n + n^2) \times (n + n^2)$ matrix:

$$\mathbf{M}(t) = \begin{pmatrix} \mathbf{M}_1(t) & \mathbf{M}_2(t) \\ 0 & \mathbf{M}_3(t) \end{pmatrix}, \tag{4.7}$$

with $\mathbf{M}_1(t) \in \mathbb{R}^{n\times n}$, $\mathbf{M}_2(t) \in \mathbb{R}^{n\times n^2}$ and $\mathbf{M}_3(t) \in \mathbb{R}^{n^2\times n^2}$. The bottom left part is necessarily 0, because $\mathcal{G}$ does not influence any $\mathcal{G}_k$. The upper left $\mathbf{M}_1(t)$ block corresponds to the averaging step; therefore, for any $t > 0$, we have:

$$\mathbf{M}_1(t) = \frac{t-1}{t} \cdot \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \left(\mathbf{I}_n - \frac{1}{2}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \right).$$

Using the definition of a graph Laplacian, one has:

$$\frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \left(\mathbf{I}_n - \frac{1}{2}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \right) = \mathbf{I}_n - \frac{\mathbf{L}}{|\mathcal{E}|}. \tag{4.8}$$

Furthermore, $\mathbf{M}_2(t)$ is a block diagonal matrix corresponding to the observations being propagated, and is defined as follows:

$$\mathbf{M}_2(t) = \frac{1}{t} \begin{pmatrix} \mathbf{e}_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{e}_n^\top \end{pmatrix}.$$

Since $t\cdot\mathbf{M}_2(t)$ does not depend on t, we define $\mathbf{B} := \mathbf{M}_2(1)$, so for any $t \geq 1$, $\mathbf{M}_2(t) = \mathbf{B}/t$. Finally, $\mathbf{M}_3(t)$ represents the estimate update for each node k and can be written for any $t \geq 1$ as follows:

$$\mathbf{M}_3(t) = \left(\frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \left(\mathbf{I}_n - (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \right) \right) \otimes \mathbf{I}_n = \left(\mathbf{I}_n - \frac{2\mathbf{L}}{|\mathcal{E}|} \right) \otimes \mathbf{I}_n,$$

where $\otimes$ is the Kronecker product. Since $\mathbf{M}_3(t)$ does not depend on t, we define $\mathbf{C} := \mathbf{M}_3(1)$.

We can now describe the expected state evolution. At iteration $t = 1$, one has:

$$\mathbb{E}[\mathbf{s}(1)] = \mathbf{M}(1)\mathbb{E}[\mathbf{s}(0)] = \mathbf{M}(1)\mathbf{s}(0) = \begin{pmatrix} 0 & \mathbf{B} \\ 0 & \mathbf{C} \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{s}_2(0) \end{pmatrix} = \begin{pmatrix} \mathbf{B}\mathbf{s}_2(0) \\ \mathbf{C}\mathbf{s}_2(0) \end{pmatrix}. \tag{4.9}$$

Using recursion, we can write:

$$\begin{aligned} \mathbb{E}[\mathbf{s}(t)] &= \mathbf{M}(t:0)\mathbf{s}(0) \\ &= \begin{pmatrix} (1/t)\sum_{s=1}^{t} (\mathbf{I}_n - \mathbf{L}/|\mathcal{E}|)^{t-s}\mathbf{B}\mathbf{C}^{s-1}\mathbf{s}_2(0) \\ \mathbf{C}^t\mathbf{s}_2(0) \end{pmatrix}. \end{aligned}$$

Therefore, in order to prove the convergence of Algorithm 5, one needs to show that

$$\lim_{t\to+\infty} \frac{1}{t} \sum_{s=1}^{t} (\mathbf{I}_n - \mathbf{L}/|\mathcal{E}|)^{t-s}\mathbf{B}\mathbf{C}^{s-1}\mathbf{s}_2(0) = \hat{U}_n\mathbf{1}_n.$$

We state this precisely in the next theorem.

Theorem 17. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non-bipartite graph,*

$(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ a data sample and $(\mathbf{z}(t))$ the sequence of estimates generated by Algorithm 5. For all $k \in [n]$, we have:

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n. \tag{4.10}$$

Moreover, for any $t > 0$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n \mathbf{1}_n \right\| \leq \frac{1}{ct} \left\| \overline{\mathbf{h}} - \hat{U}_n \mathbf{1}_n \right\| + \left(\frac{2}{ct} + e^{-ct} \right) \left\| \mathbf{H} - \overline{\mathbf{h}} \mathbf{1}_n^\top \right\|,$$

where $c = c(\mathcal{G}) := \beta_{n-1}/|\mathcal{E}|$ and β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$.

In order to prove this theorem, our main goal is to characterize the behavior of $\mathbf{s}_1(t)$, as it corresponds to the estimates $\mathbf{z}(t)$. As for the standard gossip averaging, our proof relies on the study of eigenvalues and eigenvectors of the transition matrix $\mathbf{M}(t)$. This can be done using the following lemma.

Lemma 5. Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph. Then for any $\alpha \geq 1$,

$$1 = \lambda_1(\alpha) > \lambda_2(\alpha) = 1 - \frac{2\beta_{n-1}}{\alpha |\mathcal{E}|},$$

where β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$ and $\lambda_1(\alpha), \lambda_2(\alpha)$ are respectively the largest and the second largest eigenvalues of the following symmetric matrix

$$\mathbf{I}_n - \frac{2\mathbf{L}}{\alpha |\mathcal{E}|}.$$

Proof. See Section 4.7.1

□

We are now ready to show the results of Theorem 17.

Proof of Theorem 17. For $\alpha \geq 1$, let us define $\mathbf{D}_\alpha := \mathrm{diag}(\lambda_1(\alpha), \dots, \lambda_n(\alpha))$ where $(\lambda_i(\alpha))_{1 \leq i \leq n}$ are the eigenvalues — sorted in decreasing order — of $\mathbf{I}_n - 2\mathbf{L}/(\alpha|\mathcal{E}|)$. For any $t \geq 1$, $\mathbf{M}_1(t)$ is a real-valued, symmetric matrix, therefore there exists an orthogonal matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{M}_1(t) = \frac{t-1}{t} \left(\mathbf{I}_n - \frac{\mathbf{L}}{|\mathcal{E}|} \right) = \frac{t-1}{t} \mathbf{P} \mathbf{D}_2 \mathbf{P}^\top, \tag{4.11}$$

and each column $\mathbf{p}_i$ of $\mathbf{P}$ is an eigenvector of $\mathbf{M}_1(t)$, associated to the eigenvalue $\lambda_i(2)$. Similarly, using the fact that $\mathbf{C} = (\mathbf{I}_n - 2\mathbf{L}/|\mathcal{E}|) \otimes \mathbf{I}_n$, one can write

$$\mathbf{C} = \mathbf{P}_1 \mathbf{D}_1 \mathbf{P}_1^\top,$$

where $\mathbf{P}_1 = \mathbf{P} \otimes \mathbf{I}_n$. The expected value of $\mathbf{s}_1(t)$ can then be rewritten:

$$\begin{aligned}\mathbb{E}[\mathbf{s}_1(t)] &= \frac{1}{t}\sum_{s=1}^{t}\left(\mathbf{I}_n - \frac{\mathbf{L}}{|\mathcal{E}|}\right)^{t-s}\mathbf{B}\mathbf{C}^{s-1}\mathbf{s}_2(0) \\ &= \frac{1}{t}\mathbf{P}\left(\sum_{s=1}^{t}\mathbf{D}_2^{t-s}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{D}_1^{s-1}\right)\mathbf{P}_1^\top\mathbf{s}_2(0).\end{aligned} \tag{4.12}$$

Our objective is to extract the value $\hat{U}_n$ from the expression (4.12) by separating $\lambda_1(1)$ and $\lambda_1(2)$ from other eigenvalues. Let $\mathbf{Q} = (\mathbf{p}_1, 0, \ldots, 0)$ be the part of $\mathbf{P}$ associated to $\lambda_1(2)$ and let $\mathbf{R} = \mathbf{P} - \mathbf{Q}$ be its counterpart. We can decompose $\mathbb{E}[\mathbf{s}_1(t)]$ as follows

$$\mathbb{E}[\mathbf{s}_1(t)] = \mathbf{L}_1(t) + \mathbf{L}_2(t) + \mathbf{L}_3(t) + \mathbf{L}_4(t),$$

where:

$$\begin{cases} \mathbf{L}_1(t) &= \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{Q}^{t-s}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{Q}_1\mathbf{P}_1^\top\mathbf{s}_2(0), \\ \mathbf{L}_2(t) &= \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{R}^{t-s}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{Q}_1\mathbf{P}_1^\top\mathbf{s}_2(0), \\ \mathbf{L}_3(t) &= \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{Q}^{t-s}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{R}_1^{s-1}\mathbf{P}_1^\top\mathbf{s}_2(0), \\ \mathbf{L}_4(t) &= \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{R}^{t-s}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{R}_1^{s-1}\mathbf{P}_1^\top\mathbf{s}_2(0), \end{cases}$$

and $\mathbf{Q}_1 = \mathbf{Q} \otimes \mathbf{I}_n$, $\mathbf{R}_1 = \mathbf{R} \otimes \mathbf{I}_n$. We will now show that for any $t > 0$, $\mathbf{L}_1(t)$ is actually $\hat{U}_n$. We have:

$$\mathbf{P}\mathbf{Q}\mathbf{P}^\top = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top.$$

Similarly, we have:

$$\mathbf{P}_1\mathbf{Q}_1\mathbf{P}_1^\top = (\mathbf{P}\mathbf{Q}\mathbf{P}^\top) \otimes \mathbf{I}_n = \frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top) \otimes \mathbf{I}_n.$$

Finally, we can write:

$$\begin{aligned}\mathbf{L}_1(t) &= \mathbf{P}\mathbf{Q}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{Q}_1\mathbf{P}_1^\top\mathbf{s}_2(0) \\ &= \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_n^\top\mathbf{B}\left((\mathbf{1}_n\mathbf{1}_n^\top) \otimes \mathbf{I}_n\right)\mathbf{s}_2(0) \\ &= \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_n^\top(\mathbf{1}_n^\top \otimes \mathbf{I}_n)\mathbf{s}_2(0) \\ &= \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_{n^2}^\top\mathbf{s}_2(0) \\ &= \hat{U}_n\mathbf{1}_n.\end{aligned}$$

Let us now focus on the other terms. For $t > 0$, we have:

$$\begin{aligned}\|\mathbf{L}_2(t)\| &\leq \frac{1}{t}\sum_{s=1}^{t}\left\|\mathbf{P}\mathbf{R}^{t-s}\mathbf{P}^\top\mathbf{B}\mathbf{P}_1\mathbf{Q}_1\mathbf{P}_1^\top\mathbf{s}_2(0)\right\| \\ &= \frac{1}{t}\sum_{s=1}^{t}\left\|\mathbf{P}\mathbf{R}^{t-s}\mathbf{P}^\top\mathbf{B}\left(\frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top) \otimes \mathbf{I}_n\right)\mathbf{s}_2(0)\right\| \\ &= \frac{1}{t}\sum_{s=1}^{t}\left\|\mathbf{P}\mathbf{R}^{t-s}\mathbf{P}^\top\left(\frac{1}{n}\mathbf{1}_n^\top \otimes \mathbf{I}_n\right)\mathbf{s}_2(0)\right\|.\end{aligned}$$

One has:

$$\left\| \left(\frac{1}{n} \mathbf{1}_n^\top \otimes \mathbf{I}_n \right) \mathbf{s}_2(0) \right\|^2 = \sum_{i=1}^{n} \left(\frac{1}{n} \mathbf{1}_n^\top \mathbf{H} \mathbf{e}_i \right)^2 = \left\| \frac{1}{n} \mathbf{H} \mathbf{1}_n \right\|^2 = \|\overline{\mathbf{h}}\|^2.$$

Therefore, we obtain:

$$\|\mathbf{L}_2(t)\| \leq \frac{1}{t} \sum_{s=1}^{t} \left\| \mathbf{P} \mathbf{R}^{t-s} \mathbf{P}^\top \overline{\mathbf{h}} \right\|.$$

By definition, for any $t \geq s > 0$, $\mathbf{P}\mathbf{R}^{t-s}\mathbf{P}^\top \mathbf{1}_n = 0$. Therefore, one has:

$$\begin{aligned} \|\mathbf{L}_2(t)\| &\leq \frac{1}{t} \sum_{s=1}^{t} \left\| \mathbf{P} \mathbf{R}^{t-s} \mathbf{P}^\top \overline{\mathbf{h}} \right\| \\ &\leq \frac{1}{t} \sum_{s=1}^{t} (\lambda_2(2))^{t-s} \|\overline{\mathbf{h}} - \hat{U}_n \mathbf{1}_n\| \\ &\leq \frac{1}{t} \cdot \frac{1}{1-\lambda_2(2)} \|\overline{\mathbf{h}} - U_n \mathbf{1}_n\|, \end{aligned}$$

since $\mathbf{1}_n^\top \overline{\mathbf{h}} = \hat{U}_n$. Similarly, one has:

$$\|\mathbf{L}_3(t)\| \leq \frac{1}{t} \cdot \frac{1}{1-\lambda_2(1)} \|\mathbf{H} - \overline{\mathbf{h}} \mathbf{1}_n^\top\|,$$

using $\mathbf{P}\mathbf{R}_1\mathbf{P}^\top \mathbf{1}_{n^2} = 0$. Let us now focus on the final term in the decomposition:

$$\begin{aligned} \|\mathbf{L}_4(t)\| &\leq \frac{1}{t} \sum_{s=1}^{t} \left\| \mathbf{P} \mathbf{R}^{t-s} \mathbf{P}^\top \mathbf{B} \mathbf{P}_1 \mathbf{R}_1^s \mathbf{P}_1^\top \mathbf{s}_2(0) \right\| \\ &\leq \frac{1}{t} \sum_{s=1}^{t} \left\| \mathbf{P} \mathbf{R}^{t-s} \mathbf{P}^\top \mathbf{B} \mathbf{P}_1 \mathbf{R}_1^s \mathbf{P}_1^\top \left(\mathbf{s}_2(0) - \frac{1}{n} \mathbf{1}_n^\top \mathbf{s}_2(0) \right) \right\| \\ &\leq \frac{1}{t} \left(\sum_{s=1}^{t} (\lambda_2(2))^{t-s} \lambda_2(1)^s \right) \|\mathbf{H} - \overline{\mathbf{h}} \mathbf{1}_n^\top\|. \end{aligned}$$

Lemma 5 indicates that $\lambda_2(2) > \lambda_2(1)$, so

$$\mathbf{L}_4(t) \leq (\lambda_2(2))^t \|\mathbf{H} - \overline{\mathbf{h}} \mathbf{1}_n^\top\|.$$

Using Lemma 5 and above inequalities, one can finally write:

$$\begin{aligned} \left\| \mathbf{s}_1(t) - \hat{U}_n \mathbf{1}_n \right\| &\leq \|\mathbf{L}_2(t)\| + \|\mathbf{L}_3(t)\| + \|\mathbf{L}_4(t)\| \\ &\leq \frac{c}{t} \|\overline{\mathbf{h}} - \hat{U}_n \mathbf{1}_n\| + \left(\frac{2}{ct} + e^{-ct} \right) \|\mathbf{H} - \overline{\mathbf{h}} \mathbf{1}_n^\top\|, \end{aligned}$$

with $c = 1 - \lambda_2(2)$, which concludes the proof. □

Theorem 7 shows that the local estimates generated by Algorithm 5 converge to $\hat{U}_n$ at a rate $O(1/t)$. Furthermore, the constants reveal the rate dependency on the particular problem instance. Indeed, the two norm terms

are *data-dependent* and quantify the difficulty of the estimation problem itself through a dispersion measure. In contrast, $c(\mathcal{G})$ is a *network-dependent* term since $1 - \lambda_2(2) = \beta_{n-1}/|\mathcal{E}|$, where β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$. The value β_{n-1} is also known as the spectral gap of $\mathcal{G}$ and graphs with a larger spectral gap typically have better connectivity Chung, 1997. This will be illustrated in Section 4.5.

Comparison to U2-gossip**.** To estimate $\hat{U}_n$, U2-gossip — introduced in Pelckmans and Suykens, 2009 — does not use averaging. Instead, each node k requires two auxiliary observations $\mathbf{y}_k^{(1)}$ and $\mathbf{y}_k^{(2)}$ which are both initialized to $\mathbf{x}_k$, as details in Section 4.3. U2-gossip has several drawbacks compared to GoSta: it requires initiating communication between two pairs of nodes at each iteration, and the amount of communication and memory required is higher (especially when data is high-dimensional). Furthermore, applying our convergence analysis to U2-gossip, we obtain the following refined rate:

$$\left\|\mathbb{E}[\mathbf{z}(t)] - \hat{U}_n \mathbf{1}_n\right\| \leq \frac{\sqrt{n}}{t}\left(\frac{2}{1-\tilde{\lambda}}\left\|\overline{\mathbf{h}} - \hat{U}_n \mathbf{1}_n\right\| + \frac{1}{1-\tilde{\lambda}^2}\left\|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\right\|\right), \tag{4.13}$$

where $1 - \tilde{\lambda} := 2c(\mathcal{G})$. The advantage of propagating two observations in U2-gossip is seen in the $1/(1-\tilde{\lambda}^2)$ term, however the absence of averaging leads to an overall $\sqrt{n}$ factor. Intuitively, this is because nodes do not benefit from each other's estimates. In practice, $\tilde{\lambda}$ is close to 1 for reasonably-sized networks (for instance, $\tilde{\lambda} = 1 - 2/n$ for the complete graph), so the square term does not provide much gain and the $\sqrt{n}$ factor dominates in (4.13). We thus expect U2-gossip to converge slower than GoSta, which is confirmed by the numerical results presented in Section 4.5.

4.4.2 Asynchronous Setting

In practical settings, nodes may not have access to a global clock to synchronize the updates. In this section, we remove the global clock assumption and propose a fully asynchronous algorithm where each node has a local clock, ticking at a rate 1 Poisson process. Yet, local clocks are i.i.d. so one can use an equivalent model with a global clock ticking at a rate n Poisson process and a random edge draw at each iteration, as in synchronous setting (one may refer to Boyd et al., 2006 for more details on clock modeling). However, at a given iteration, the estimate update step now only involves the selected pair of nodes. Therefore, the nodes need to maintain an estimate of the current iteration number to ensure convergence to an unbiased estimate of $\hat{U}_n(H)$. Hence for all $k \in [n]$, let $p_k \in [0,1]$ denote the probability of node k being picked at any iteration. With our assumption that nodes activate with a uniform distribution over $\mathcal{E}$, $p_k = 2d_k/|\mathcal{E}|$. Moreover, the number of times a node k has been selected at a given iteration $t > 0$ follows a binomial distribution with parameters t and p_k. Let us define $m_k(t)$ such that $m_k(0) = 0$ and for $t > 0$:

$$m_k(t) = \begin{cases} m_k(t-1) + \frac{1}{p_k} & \text{if } k \text{ is picked at iteration t,} \\ m_k(t-1) & \text{otherwise.} \end{cases} \tag{4.14}$$

Algorithm 12 GoSta-async: an asynchronous gossip algorithm for computing a U-statistic

Require: Each node k holds observation $\mathbf{x}_k$ and $w_k = 2d_k/|\mathcal{E}|$
Each node k initializes $\mathbf{y}_k = \mathbf{x}_k$, $z_k = 0$ and $m_k = 0$
for $t = 1, 2, \dots$ **do**
  Draw (i, j) uniformly at random from $\mathcal{E}$
  Set $m_i \leftarrow m_i + 1/w_i$ and $m_j \leftarrow m_j + 1/w_j$
  Set $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
  Set $z_i \leftarrow (1 - \frac{1}{w_i m_i})z_i + \frac{1}{w_i m_i} h(\mathbf{x}_i, \mathbf{y}_i)$
  Set $z_j \leftarrow (1 - \frac{1}{w_j m_j})z_j + \frac{1}{w_j m_j} h(\mathbf{x}_j, \mathbf{y}_j)$
  Swap auxiliary observations of nodes i and j: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
end for
return $\mathbf{z} = (z_1, \dots, z_n)$

For any $k \in [n]$ and any $t > 0$, one has $\mathbb{E}[m_k(t)] = t \times p_k \times 1/p_k = t$. Therefore, given that every node knows its degree and the total number of edges in the network, the iteration estimates are unbiased. We can now give an asynchronous version of GoSta, as stated in Algorithm 12.

To show that local estimates converge to $\hat{U}_n$, we use a similar model as in the synchronous setting. The time dependency of the transition matrix is more complex ; so is the upper bound.

Theorem 18. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph, $\mathbf{X} \in \mathbb{R}^{n \times d}$ a design matrix and $(\mathbf{z}(t))$ the sequence of estimates generated by Algorithm 12. For all $k \in [n]$, we have:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} h(\mathbf{x}_i, \mathbf{x}_j) = \hat{U}_n.$$

Moreover, there exists a constant $c'(G) > 0$ such that, for any $t > 1$,

$$\left\| \mathbb{E}[\mathbf{z}(t)] - \hat{U}_n \mathbf{1}_n \right\| \leq c'(\mathcal{G}) \cdot \frac{\log t}{t} \|\mathbf{H}\|.$$

Sketch of proof. Since updates are weighted differently, the expected asynchronous transition matrix is different from the synchronous one. More specifically, the propagation part $\mathbf{M}_3(t)$ is unaltered but $\mathbf{M}_1(t)$ and $\mathbf{M}_2(t)$ must but analyzed more carefully. Now, only the selected nodes update their estimators from their associated phantom graph. Therefore, we have:

$$\mathbf{M}_2(t) = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \mathbb{E} \begin{pmatrix} \mathbb{1}_{\{1 \in (i,j)\}} \frac{1}{m_1(t) p_1} \mathbf{e}_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{1}_{\{n \in (i,j)\}} \frac{1}{m_n(t) p_n} \mathbf{e}_n^\top \end{pmatrix}.$$

Since for any $i \in [n]$, $m_i(t)$ is an unbiased estimator of t, this can be rewritten as follows:

$$\begin{aligned}\mathbf{M}_2(t) &= \frac{1}{t}\begin{pmatrix}\mathbf{e}_1^\top & 0 & \cdots & 0\\ 0 & \ddots & & \vdots\\ \vdots & & \ddots & 0\\ 0 & \cdots & 0 & \mathbf{e}_n^\top\end{pmatrix}\\ &= \frac{\mathbf{B}}{t},\end{aligned}$$

Similarly for $\mathbf{M}_1(t)$:

$$\mathbf{M}_1(t) = \mathbf{I}_n - \frac{\mathbf{L}}{|\mathcal{E}|} - \frac{1}{2t}\left(\mathbf{I}_n + \mathbf{D}^{-1}\mathbf{A}\right).$$

Using this transition matrix, we can now write the expected values of the vector $\mathbf{s}(t)$ defined previously. For any $t > 0$, one has:

$$\mathbb{E}[\mathbf{s}(t)] = \mathbb{E}\begin{bmatrix}\mathbf{s}_1(t)\\ \mathbf{s}_2(t)\end{bmatrix} = \begin{pmatrix}\sum_{s=1}^{t}\left(\mathbf{M}_1(t)\dots\mathbf{M}_1(s+1)\right)\frac{\mathbf{B}}{s}\mathbf{C}^{s-1}\mathbf{s}_2(0)\\ \mathbf{C}^t\mathbf{s}_2(0)\end{pmatrix}.$$

As in the synchronous setting, our proof rely on the eigenvalues of $\mathbf{M}_1(t)$. However, the analysis is harder in the asynchronous case, the second largest eigenvalue of $\mathbf{M}_1(t)$ now depending on t. A quick analysis of $\mathbf{M}_1(t)$ shows that the largest eigenvalue can be greater than $(1 - \frac{1}{t})$ for iterates $t < t_c$, where t_c is the expected number of iterations needed for every node to have been picked at least once.

For $t \geq t_c$, one can use the matrix of eigenvectors $\mathbf{P}$ defined in 4.11 and write $\mathbf{M}_1(t)\mathbf{P} = \mathbf{P}\mathbf{K}(t)$, for $\mathbf{K}(t)$ defined as follows:

$$\mathbf{K}(t) = \left(1 - \frac{1}{t}\right)\mathbf{I}_n - \frac{1}{|\mathcal{E}|}\left(\mathbf{I}_n - \frac{|\mathcal{E}|}{2t}\mathbf{P}^\top\mathbf{D}^{-1}\mathbf{P}\right)\operatorname{diag}(\beta_n, \dots, \beta_1).$$

Let $P_1 = \big(\phi_1, 0, \dots, 0\big)$. The matrix $\mathbf{K}(t)$ can be rewritten as follows:

$$\mathbf{K}(t) = \left(1 - \frac{1}{t}\right)\mathbf{Q} + \frac{1}{t}\mathbf{U} + \mathbf{R}(t),$$

where $\mathbf{Q}$, $\mathbf{U}$ and $\mathbf{R}(t)$ are defined by:

$$\begin{cases}\mathbf{Q} &= \operatorname{diag}(1, 0, \dots, 0),\\ \mathbf{U} &= \frac{1}{2}\mathbf{P}_1^\top\mathbf{D}^{-1}\mathbf{P}\operatorname{diag}(\beta_n, \dots, \beta_1),\\ \mathbf{R}(t) &= \mathbf{K}(t) - \big(1 - \frac{1}{t}\big)\mathbf{Q} - \frac{1}{t}\mathbf{U}, \text{ for all } t > 0.\end{cases}$$

Using the fact that $\beta_n = 0$, one can show that $\mathbf{U}$ has the form

$$\begin{pmatrix}0 & * & \cdots & *\\ 0 & & & \\ \vdots & & 0 & \\ 0 & & & \end{pmatrix}.$$

Since $\mathbf{M}_1(t)\mathbf{1}_n = \left(1 - \frac{1}{t}\right)\mathbf{1}_n$, we can also show that, for $t > 0$, $\mathbf{R}(t)\mathbf{e}_1 = 0$ and $\mathbf{e}_1^\top \mathbf{R}(t) = 0$; recursively, we obtain, for $t > s > 0$:

$$\mathbf{M}_1(t:s) = \mathbf{M}_1(t)\dots\mathbf{M}_1(s+1) = \mathbf{P}\left(\frac{s}{t}\mathbf{Q} + \frac{1}{t}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s) + \mathbf{R}(t:s)\right)\mathbf{P}^\top,$$

where we use the convention $\mathbf{R}(t:t) = \mathbf{I}_n$. Let us now write the expected value of the estimates:

$$\begin{aligned}
\mathbb{E}[\mathbf{s}_1(t)] &= \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{C}^{s-1}\mathbf{s}_2(0) \\
&= \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_1\mathbf{Q}_1\mathbf{P}_1^\top\mathbf{S}_2(0) + \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_1\mathbf{R}_1^{s-1}\mathbf{P}_1^\top\mathbf{s}_2(0) \\
&= \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_1\mathbf{R}_1^{s-1}\mathbf{P}_1^\top\mathbf{s}_2(0).
\end{aligned}$$

Splitting the sum at t_c and then using a similar analysis to Theorem 18, one obtains several terms that are in $O(\log t/t)$, which concludes the proof — see Section 4.7 for a detailed proof. □

Remark 6. Our methods can be extended to the situation where nodes contain multiple observations: when drawn, a node will pick a random auxiliary observation to swap. Similar convergence results are achieved by splitting each node into a set of nodes, each containing only one observation and new edges weighted judiciously.

Dataset	Complete graph	Watts-Strogatz	2d-grid graph
Wine Quality ($n = 1599$)	$6.26 \cdot 10^{-4}$	$2.72 \cdot 10^{-5}$	$3.66 \cdot 10^{-6}$
SVMguide3 ($n = 1260$)	$7.94 \cdot 10^{-4}$	$5.49 \cdot 10^{-5}$	$6.03 \cdot 10^{-6}$

TABLE 4.1: Value of $\beta_{n-1}/|\mathcal{E}|$ for each network.

4.5 Experiments

4.5.1 Comparison to U2-GOSSIP

In this section, we present two applications on real datasets: the decentralized estimation of the Area Under the ROC Curve (AUC) and of the within-cluster point scatter. We compare the performance of our algorithms to that of U2-GOSSIP. We perform our simulations on the three types of network described below (corresponding values of $\beta_{n-1}/|\mathcal{E}|$ are shown in Table 4.1).

***Complete graph*:** This is the case where all nodes are connected to each other. It is the ideal situation in our framework, since any pair of nodes can communicate directly. For a complete graph $\mathcal{G}$ of size $n > 0$, $\beta_{n-1}/|\mathcal{E}| = 1/n$, see BOLLOBÁS, 1998, Ch.9 or CHUNG, 1997, Ch.1 for details.

Two-dimensional grid: Here, nodes are located on a 2d grid, and each node is connected to its four neighbors on the grid. This network offers a regular graph with isotropic communication, but its diameter ($\sqrt{n}$) is quite high, especially in comparison to usual scale-free networks.

Watts-Strogatz: This random network generation technique is introduced in WATTS and STROGATZ, 1998 and allows us to create networks with various communication properties. It relies on two parameters: the average degree of the network k and a rewiring probability p. In expectation, the higher the rewiring probability, the better the connectivity of the network. Here, we use $k = 5$ and $p = 0.3$ to achieve a connectivity compromise between the complete graph and the two-dimensional grid.

AUC measure

We first focus on the AUC measure of a linear classifier θ as defined in (4.4). We use the SMVGUIDE3 binary classification dataset which contains $n = 1260$ points in $d = 23$ dimensions.[3] We set θ to the difference between the class means. For each generated network, we perform 50 runs of GOSTA-SYNC (Algorithm 5) and U2-GOSSIP. The top row of Figure 4.2 shows the evolution over time of the average relative error and the associated standard deviation *across nodes* for both algorithms on each type of network. On average, GOSTA-SYNC outperforms U2-GOSSIP on every network. The variance of the estimates across nodes is also lower due to the

3. This dataset is available at http://mldata.org/repository/data/viewslug/svmguide3/

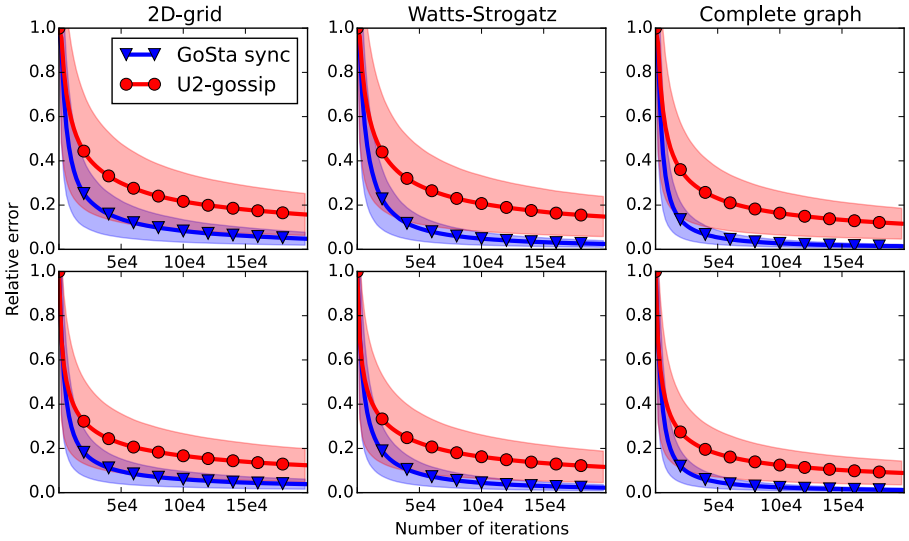
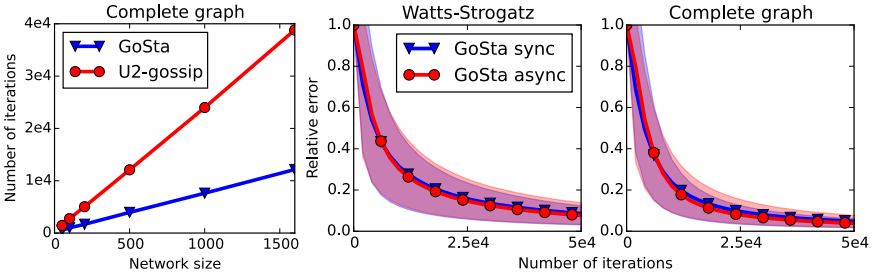


FIGURE 4.2: Evolution of the average relative error (solid line) and its standard deviation (filled area) with the number of iterations for U2-GOSSIP (red) and Algorithm 5 (blue) on the SVMGUIDE3 dataset (top row) and the WINE QUALITY dataset (bottom row).



(A) 20% error reaching time.

(B) Average relative error.

FIGURE 4.3: Panel (a) shows the average number of iterations needed to reach an relative error below 0.2, for several network sizes $n \in [50, 1599]$. Panel (b) compares the relative error (solid line) and its standard deviation (filled area) of synchronous (blue) and asynchronous (red) versions of GOSTA.

averaging step. Interestingly, the performance gap between the two algorithms is greatly increasing early on, presumably because the exponential term in the convergence bound of GOSTA-SYNC is significant in the first steps.

Within-cluster point scatter

We then turn to the within-cluster point scatter defined in (4.3). We use the Wine Quality dataset which contains $n = 1599$ points in $d = 12$ dimensions, with a total of $K = 11$ classes.[4] We focus on the partition $\mathcal{P}$ associated to class centroids and run the aforementioned methods 50 times.

4. This dataset is available at https://archive.ics.uci.edu/ml/datasets/Wine

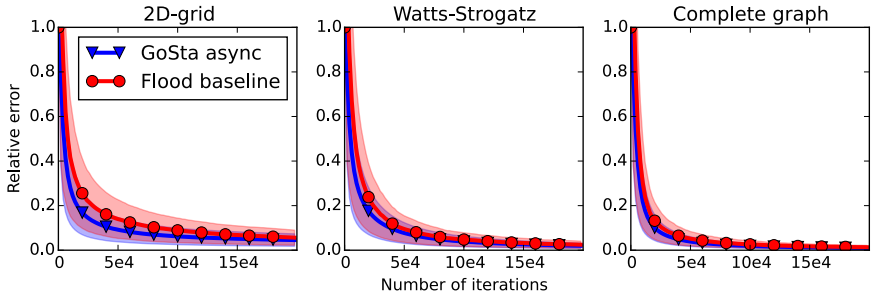


FIGURE 4.4: Comparison to the gossip-flooding baseline.

The results are shown in the bottom row of Figure 4.2. As in the case of AUC, GOSTA-SYNC achieves better perfomance on all types of networks, both in terms of average error and variance. In Figure 4.3a, we show the average time needed to reach a 0.2 relative error on a complete graph ranging from $n = 50$ to $n = 1599$. As predicted by our analysis, the performance gap widens in favor of GOSTA as the size of the graph increases. Finally, we compare the performance of GOSTA-SYNC and GOSTA-ASYNC (Algorithm 12) in Figure 4.3b. Despite the slightly worse theoretical convergence rate for GOSTA-ASYNC, both algorithms have comparable performance in practice.

4.5.2 Comparison to Baseline Methods

In this section, we use the within-cluster point scatter problem studied in Section 4.5.1 to compare our algorithms to two — more naive — baseline methods, described below.

Gossip-flooding baseline. This baseline uses the same communication scheme than GOSTA-ASYNC (Algorithm 12) to flood observations across the network, but we assume that each node has enough memory to store all the observations it receives. At each iteration, each selected node picks a random observation among those it currently holds and send it to the other (tagged with the node which initially possessed it, to avoid storing duplicates). The local estimates are computed using the subset of observations available at each node (the averaging step is removed).

Figure 4.4 shows the evolution over time of the average relative error and the associated standard deviation *across nodes* for this baseline and GOSTA-ASYNC on the networks introduced in Section 4.5. On average, GOSTA-ASYNC slightly outperforms gossip-flooding, and this difference gets larger as the network connectivity decreases. The variance of the estimates across nodes is also lower for GOSTA-ASYNC. This confirms the interest of averaging the estimates, and shows that assuming large memory at each node is not necessary to achieve good performance. Finally, note that updating the local estimate of a node is computationally much cheaper in GOSTA-ASYNC (only one function evaluation) than in gossip-flooding (as many function evaluations as there are observations on the node).

Master-node baseline. This baseline has access to a master node $\mathcal{M}$ which is connected to every other node in the network. Initially, at $t = 0$, each node $i \in [n]$ sends its observation $\mathbf{x}_i$ to $\mathcal{M}$. Then, at each iteration

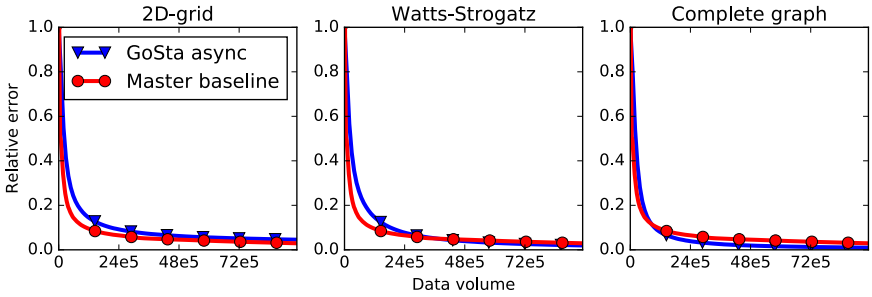


FIGURE 4.5: Comparison to the master-node baseline. One unit of data corresponds to one observation coordinate.

$t \in [n]$, $\mathcal{M}$ sends observation $\mathbf{x}_t$ to every node of the network. As in gossip-flooding, the estimates are computed using the subset of observations available at each node. The performance of this baseline does not depend on the original network, since communication goes through the master-node $\mathcal{M}$. This allows us to compare our approach to the ideal scenario of a star network, where a central node can efficiently broadcast information to the entire network.

For a fair comparison with GOSTA-ASYNC, we evaluate the methods with respect to the communication cost instead of the number of iterations. Figure 4.5 shows the evolution of the average relative error for this baseline and GOSTA-ASYNC. We can see that the Master-node baseline performs better early on, but GOSTA-ASYNC quickly catches up (the better the connectivity, the sooner). This shows that our data propagation and averaging mechanisms compensate well for the lack of central node.

4.6 Conclusion

We have introduced new synchronous and asynchronous randomized gossip algorithms to compute statistics that depend on pairs of observations (U-statistics). We have proved the convergence rate in both settings, and numerical experiments confirm the practical interest of the proposed algorithms. In future work, we plan to investigate whether adaptive communication schemes (such as those of DIMAKIS, SARWATE, and WAINWRIGHT, 2008; LI, DAI, and ZHANG, 2010) can be used to speed-up our algorithms.

We will use our contribution as a building block for decentralized *optimization* of U-statistics, extending for instance the approaches of DUCHI, AGARWAL, and WAINWRIGHT, 2012; NEDIC and OZDAGLAR, 2009. Therefore, we will now focus on the optimization of objectives that are separable in pairs of observations. Again, our work will tackle the decentralized setting, since the centralized setting has been studied in Chapter 3

4.7 Proofs

4.7.1 Preliminary Results

Here, we state preliminary results on the matrices $\mathbf{W}_\alpha(\mathcal{G})$ that will be useful for deriving convergence proofs and compare the algorithms.

First, we characterize the eigenvalues of $\mathbf{W}_\alpha(\mathcal{G})$ in terms of those of the graph Laplacian.

Lemma 6. Let $\mathcal{G} = ([n], \mathcal{E})$ be an undirected graph and let $(\beta_i)_{1\leq i\leq n}$ be the eigenvalues of $\mathbf{L}^{\mathcal{G}}$, sorted in decreasing order. For any $\alpha \geq 1$, we denote as $(\lambda_i(\alpha))_{1\leq i\leq n}$ the eigenvalues of $\mathbf{W}_\alpha(\mathcal{G})$, sorted in decreasing order. Then, for any $1 \leq i \leq n$,

$$\lambda_i(\alpha) = 1 - \frac{2\beta_{n-i+1}}{\alpha|\mathcal{E}|}. \tag{4.15}$$

Proof. Let $\alpha \geq 1$. The matrix $\mathbf{W}_\alpha(\mathcal{G})$ can be rewritten as follow:

$$\mathbf{W}_\alpha(\mathcal{G}) = \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in E}\left(\mathbf{I}_n - \frac{1}{\alpha}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top\right) \tag{4.16}$$

$$= \mathbf{I}_n - \frac{1}{\alpha|\mathcal{E}|}\sum_{(i,j)\in E}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top = \mathbf{I}_n - \frac{2}{\alpha|\mathcal{E}|}\mathbf{L}^{\mathcal{G}}. \tag{4.17}$$

Let $\phi_i \in \mathbb{R}^n$ be an eigenvector of $\mathbf{L}^{\mathcal{G}}$ corresponding to an eigenvalue β_i, then we have:

$$\mathbf{W}_\alpha(\mathcal{G})\phi_i = \left(\mathbf{I}_n - \frac{2}{\alpha|\mathcal{E}|}\mathbf{L}^{\mathcal{G}}\right)\phi_i = \left(1 - \frac{2}{\alpha|\mathcal{E}|}\beta_i\right)\phi_i.$$

Thus, ϕ_i is also an eigenvector of $\mathbf{W}_\alpha(\mathcal{G})$ for the eigenvalue $1 - \frac{2}{\alpha|\mathcal{E}|}\beta_i$ and the result holds. □

The following lemmata provide essential properties on $\mathbf{W}_\alpha(\mathcal{G})$ eigenvalues.

Lemma 7. Let $n > 0$ and let $\mathcal{G} = ([n], \mathcal{E})$ be an undirected graph. If $\mathcal{G}$ is connected and non-bipartite, then for any $\alpha \geq 1$, $\mathbf{W}_\alpha(\mathcal{G})$ is primitive, *i.e.*, there exists $k > 0$ such that $\mathbf{W}_\alpha(\mathcal{G})^k > 0$.

Proof. Let $\alpha \geq 1$. For every $(i, j) \in \mathcal{E}$, $\mathbf{I}_n - \frac{1}{\alpha}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top$ is nonnegative. Therefore $\mathbf{W}_\alpha(\mathcal{G})$ is also nonnegative. For any $1 \leq k < l \leq n$, by definition of $\mathbf{W}_\alpha(\mathcal{G})$, one has the following equivalence:

$$\left([\mathbf{A}^{\mathcal{G}}]_{kl} > 0\right) \Leftrightarrow \left([\mathbf{W}_\alpha(\mathcal{G})]_{kl} > 0\right).$$

By hypothesis, $\mathcal{G}$ is connected. Therefore, for any pair of nodes $(k, l) \in V^2$ there exists an integer $s_{kl} > 0$ such that $\left[(\mathbf{A}^{\mathcal{G}})^{s_{kl}}\right]_{kl} > 0$ so $\mathbf{W}_\alpha(\mathcal{G})$ is irreducible. Also, $\mathcal{G}$ is non bipartite so similar reasoning can be used to show that $\mathbf{W}_\alpha(\mathcal{G})$ is aperiodic.

By the Lattice Theorem (see BRÉMAUD, 1999, Th. 4.3, p.75), for any $1 \leq k, l \leq n$ there exists an integer m_{kl} such that, for any $m \geq m_{kl}$:

$$[\mathbf{W}_\alpha(\mathcal{G})^m]_{kl} > 0.$$

Finally, we can define $\bar{m} = \sup_{k,l} m_{kl}$ and observe that $\mathbf{W}_\alpha(\mathcal{G})^{\bar{m}} > 0$. □

Lemma 8. Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph. Then for any $\alpha \geq 1$,

$$1 = \lambda_1(\alpha) > \lambda_2(\alpha),$$

where $\lambda_1(\alpha)$ and $\lambda_2(\alpha)$ are respectively the largest and the second largest eigenvalue of $\mathbf{W}_\alpha(\mathcal{G})$.

Proof. Let $\alpha \geq 1$. The matrix $\mathbf{W}_\alpha(\mathcal{G})$ is bistochastic, so $\lambda_1(\alpha) = 1$. By Lemma 7, $\mathbf{W}_\alpha(\mathcal{G})$ is primitive. Therefore, by the Perron-Frobenius Theorem (see BRÉMAUD, 1999, Th. 1.1, p.197), we can conclude that $\lambda_1(\alpha) > \lambda_2(\alpha)$. □

4.7.2 Convergence Proofs for GoSta

Proof of Theorem 18 (Asynchronous Setting)

For $t > 0$, let us denote as $\mathbf{M}(t)$ the expected transition matrix at iteration t. With the notation introduced in the synchronous setting, it yields

$$\begin{pmatrix} \mathbf{M}_1(t) & \mathbf{M}_2(t) \\ 0 & \mathbf{C} \end{pmatrix}.$$

The propagation step is unaltered w.r.t. the synchronous case, thus the bottom right block is unmodified. On the other hand, both the transmission step and the averaging step differ: only the selected nodes update their estimators from their associated phantom graph. Therefore, we have:

$$\mathbf{M}_2(t) = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \mathbb{E} \begin{pmatrix} \mathbb{1}_{\{1\in(i,j)\}} \frac{1}{m_1(t)p_1} \mathbf{e}_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{1}_{\{n\in(i,j)\}} \frac{1}{m_n(t)p_n} \mathbf{e}_n^\top \end{pmatrix}.$$

For any $k \in [n]$ and $t > 0$, $m_k(t)$ is an unbiased estimator of t. Moreover, $\sum_{(i,j)\in E} \mathbb{1}_{\{k\in(i,j)\}} = 2d_k$. Therefore, we can write:

$$\mathbf{M}_2(t) = \frac{1}{t|\mathcal{E}|} \begin{pmatrix} \frac{2d_1}{p_1}\mathbf{e}_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{2d_n}{p_n}\mathbf{e}_n^\top \end{pmatrix} = \frac{1}{t} \begin{pmatrix} \mathbf{e}_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{e}_n^\top \end{pmatrix} = \frac{\mathbf{B}}{t}.$$

Similarly for $\mathbf{M}_1(t)$:

$$\mathbf{M}_1(t) = \mathbf{W}_2(\mathcal{G}) - \frac{1}{2t|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \left(\frac{1}{p_i}\mathbf{e}_i(\mathbf{e}_i+\mathbf{e}_j)^\top + \frac{1}{p_j}\mathbf{e}_j(\mathbf{e}_i+\mathbf{e}_j)^\top \right).$$

Using the definition of $(p_k)_{k\in[n]}$ yields:

$$\mathbf{M}_1(t) = \mathbf{W}_2(\mathcal{G}) - \frac{1}{2t}\left(\mathbf{I}_n + (\mathbf{D}^{\mathcal{G}})^{-1}\mathbf{A}^{\mathcal{G}}\right).$$

We can now write the expected value of the state vector $\mathbf{s}(t)$ similarly to the synchronous setting:

$$\mathbb{E}[\mathbf{s}(t)] = \mathbb{E}\begin{bmatrix} \mathbf{s}_1(t) \\ \mathbf{s}_2(t) \end{bmatrix} = \begin{pmatrix} \sum_{s=1}^{t} \left(\mathbf{M}_1(t)\dots\mathbf{M}_1(s+1)\right) \frac{\mathbf{B}}{s} \mathbf{C}^{s-1}\mathbf{s}_2(0) \\ \mathbf{C}^t\mathbf{s}_2(0) \end{pmatrix}.$$

As in the synchronous setting, our proof rely on the eigenvalues of $\mathbf{M}(t)$.

Proof. For $t > 0$, we have:

$$\mathbf{M}_1(t) = \mathbf{W}_2(\mathcal{G}) - \frac{1}{2t}\left(\mathbf{I}_n + \mathbf{D}^{-1}\mathbf{A}\right) = \mathbf{W}_2(\mathcal{G}) - \frac{1}{t}\mathbf{I}_n + \frac{1}{2t}\mathbf{D}^{-1}\mathbf{L}.$$

Since $\mathbf{M}_1(t)\mathbf{1}_n = \left(1 - \frac{1}{t}\right)\mathbf{1}_n$, we have $|||\mathbf{M}_1(t)||| \geq 1 - \frac{1}{t}$. Let us denote $\mathrm{Sp}(\mathbf{L}) = \left\{\beta \in \mathbb{R}, \exists \phi \in \mathbb{R}^n, \mathbf{L}\phi = \beta\phi\right\}$. Let $\beta \in \mathrm{Sp}(\mathbf{L})$ and $\phi \in \mathbb{R}^n$ a corresponding eigenvector. One can write:

$$\begin{aligned}\mathbf{M}_1(t)\phi &= \left(\mathbf{W}_2\left(\mathcal{G}\right) - \frac{1}{t}\mathbf{I}_n + \frac{1}{2t}\mathbf{D}\mathbf{L}^{-1}\right)\phi \\ &= \left(1 - \frac{\beta}{|\mathcal{E}|} - \frac{1}{t}\right)\phi + \frac{\beta}{2t}\mathbf{D}^{-1}\phi \\ &= \left(\left(1 - \frac{1}{t}\right)\mathbf{I}_n - \frac{\beta}{|\mathcal{E}|}\left(\mathbf{I}_n - \frac{|\mathcal{E}|\mathbf{D}^{-1}}{2t}\right)\right)\phi.\end{aligned}$$

The above matrix is diagonal, therefore we can write:

$$\begin{aligned}\|\mathbf{M}_1(t)\phi\| &\leq \max_i \left(1 - \frac{1}{t} - \frac{\beta}{|\mathcal{E}|}\left(1 - \frac{|\mathcal{E}|}{2d_i t}\right)\right)\|\phi\| \\ &= \left(1 - \frac{1}{t} - \frac{\beta}{|\mathcal{E}|}\left(1 - \frac{1}{\overline{p}t}\right)\right)\|\phi\|,\end{aligned}$$

where

$$\overline{p} := \min_i \frac{2d_i}{|\mathcal{E}|}$$

is the minimum probability of a node being picked at any iteration. Thus, we can see that if $\beta > 0$, one has

$$\|\mathbf{M}_1(t)\phi\| < \left(1 - \frac{1}{t}\right)\|\phi\|$$

if $t < t_c = \overline{p}^{-1}$. Consequently, if $t \geq t_c$ then $|||\mathbf{M}_1(t)||| = 1 - 1/t$. Here, t_c represents the minimum number of iteration needed for every node to have been picked at least once, in expectation.

Let $(\beta_1, \ldots, \beta_n) \in \mathbb{R}$ and $\mathbf{P} = (\phi_1, \ldots, \phi_n) \in \mathbb{R}^{n\times n}$ be respectively the eigenvalues and eigenvectors of $\mathbf{L}^{\mathcal{G}}$ (sorted in decreasing order), such that $\mathbf{P}$ is the same matrix than the one introduced in Section 4.4.1. We have:

$$\mathbf{M}_1(t)\mathbf{P} = \mathbf{P}\mathbf{K}(t) = \mathbf{P}\left(\left(1 - \frac{1}{t}\right)\mathbf{I}_n - \frac{1}{|\mathcal{E}|}\left(\mathbf{I}_n - \frac{|\mathcal{E}|}{2t}\mathbf{P}^\top\mathbf{D}^{-1}\mathbf{P}\right)\mathbf{D}_\mathbf{L}\right),$$

where $\mathbf{D}_\mathbf{L} = \mathrm{diag}(\beta_n, \ldots, \beta_1)$. Let $\mathbf{P}_1 = \left(\phi_1, 0, \ldots, 0\right)$. The matrix $\mathbf{K}(t)$ can be rewritten as follows:

$$\mathbf{K}(t) = \left(1 - \frac{1}{t}\right)\mathbf{I}_n - \frac{1}{|\mathcal{E}|}\left(\mathbf{I}_n - \frac{|\mathcal{E}|}{2t}\mathbf{P}^\top\mathbf{D}^{-1}\mathbf{P}\right)\mathbf{D}_\mathbf{L} = \left(1 - \frac{1}{t}\right)\mathbf{Q} + \frac{1}{t}\mathbf{U} + \mathbf{R}(t),$$

where $\mathbf{Q}$, $\mathbf{U}$ and $\mathbf{R}(t)$ are defined by:

$$\left\{\begin{array}{rcl}\mathbf{Q} &=& \mathrm{diag}(1, 0, \ldots, 0), \\ \mathbf{U} &=& \frac{1}{2}\mathbf{P}_1^\top\mathbf{D}^{-1}\mathbf{P}\mathbf{D}_\mathbf{L}, \\ \mathbf{R}(t) &=& \mathbf{K}(t) - \left(1 - \frac{1}{t}\right)\mathbf{Q} - \frac{1}{t}\mathbf{U}, \text{ for all } t > 0.\end{array}\right.$$

Using the fact that $\beta_n = 0$, one can show that $\mathbf{U}$ has the form

$$\begin{pmatrix} 0 & * & \cdots & * \\ 0 & & & \\ \vdots & & 0 & \\ 0 & & & \end{pmatrix}.$$

Since $\mathbf{M}_1(t)\mathbf{1}_n = \left(1 - \frac{1}{t}\right)\mathbf{1}_n$, we can also show that, for $t > 0$, $\mathbf{R}(t)\mathbf{e}_1 = 0$ and $\mathbf{e}_1^\top \mathbf{R}(t) = 0$. Let $t > 0$. We can write:

$$\begin{aligned}
&\mathbf{M}_1(t+1)\mathbf{M}_1(t) \\
&\qquad = \mathbf{P}\mathbf{K}(t+1)\mathbf{K}(t)\mathbf{P}^\top \\
&\qquad = \mathbf{P}\left(\frac{t}{t+1}\mathbf{Q} + \frac{1}{t+1}\mathbf{U} + \mathbf{R}(t+1)\right)\left(\frac{t-1}{t}\mathbf{Q} + \frac{1}{t}\mathbf{U} + \mathbf{R}(t)\right)\mathbf{P}^\top \\
&\qquad = \mathbf{P}\left(\frac{t-1}{t+1}\mathbf{Q} + \frac{1}{t+1}\mathbf{U}\big(\mathbf{I}_n + \mathbf{R}(t)\big) + \mathbf{R}(t+1)\mathbf{R}(t)\right)\mathbf{P}^\top.
\end{aligned}$$

Recursively, we obtain, for $t > s > 0$:

$$\mathbf{M}_1(t:s) = \mathbf{M}_1(t)\dots\mathbf{M}_1(s+1) = \mathbf{P}\left(\frac{s}{t}\mathbf{Q} + \frac{1}{t}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s) + \mathbf{R}(t:s)\right)\mathbf{P}^\top,$$

where we use the convention $\mathbf{R}(t-1:t-1) = \mathbf{I}_n$.

Let us now write the expected value of the estimates:

$$\begin{aligned}
\mathbb{E}[\mathbf{s}_1(t)] &= \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}C^{s-1}\mathbf{s}_2(0) \\
&= \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{Q}_c\mathbf{P}_c^\top\mathbf{s}_2(0) + \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{R}_c^{s-1}\mathbf{P}_c^\top\mathbf{s}_2(0) \\
&= \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{R}_c^{s-1}\mathbf{P}_c^\top\mathbf{s}_2(0).
\end{aligned}$$

The first term can be rewritten as:

$$\begin{aligned}
&\sum_{s=1}^{t}\mathbf{M}_1(t:s)\frac{\overline{\mathbf{h}}}{s} \\
&\quad = \sum_{s=1}^{t}\mathbf{P}\left(\frac{s}{t}\mathbf{Q} + \frac{\mathbf{U}}{t}\sum_{r=s}^{t-1}\mathbf{R}(r:s) + \mathbf{R}(t:s)\right)\mathbf{P}^\top\frac{\overline{\mathbf{h}}}{s} \\
&\quad = \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{Q}\mathbf{P}^\top\overline{\mathbf{h}} + \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s)\mathbf{P}^\top\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t}\mathbf{P}\mathbf{R}(t:s)\mathbf{P}^\top\frac{\overline{\mathbf{h}}}{s} \\
&\quad = \hat{U}_n(h)\mathbf{1}_n + \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s)\mathbf{P}^\top\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t}\mathbf{P}\mathbf{R}(t:s)\mathbf{P}^\top\frac{\overline{\mathbf{h}}}{s} \\
&\quad = \hat{U}_n(h)\mathbf{1}_n + \mathbf{L}_1(t) + \mathbf{L}_2(t).
\end{aligned}$$

The second term of the expected estimates can be rewritten as:

$$\begin{aligned}
&\sum_{s=1}^{t} \mathbf{M}_1(t:s)\frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{R}_c^{s-1}\mathbf{P}_c^\top \mathbf{s}_2(0) \\
&\qquad = \sum_{s=1}^{t} \mathbf{P}\left(\frac{s}{t}\mathbf{Q} + \frac{\mathbf{U}}{t}\sum_{r=s}^{t-1}\mathbf{R}(r:s) + \mathbf{R}(t:s)\right)\mathbf{P}^\top \frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{R}_c^{s-1}\mathbf{P}_c^\top \mathbf{s}_2(0) \\
&\qquad = \mathbf{L}_3(t) + \mathbf{L}_4(t) + \mathbf{L}_5(t).
\end{aligned}$$

Now, we need to upper bound $\|\mathbf{L}_i(t)\|$ for $1 \leq i \leq 5$. One has:

$$\begin{aligned}
\|\mathbf{L}_1(t)\| &= \left\| \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s)\mathbf{P}^\top\frac{\overline{\mathbf{h}}}{s}\right\| \\
&= \left\| \frac{1}{t}\sum_{s=1}^{t}\mathbf{P}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s)\mathbf{P}^\top\frac{\left(\overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n\right)}{s}\right\| \\
&\leq \frac{1}{t}\sum_{s=1}^{t}\left\|\mathbf{P}\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s)\mathbf{P}^\top\frac{\left(\overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n\right)}{s}\right\| \\
&\leq \frac{|||\mathbf{U}|||}{t}\left(\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}|||\mathbf{R}(r:s)|||\right)\|\overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n\|.
\end{aligned}$$

The norm of $\mathbf{U}$ can be developed:

$$|||\mathbf{U}||| \leq \frac{1}{2}\left|||\mathbf{D}^{-1}|||\right| |||\mathbf{D}_{\mathbf{L}}||| = \frac{\beta_1}{|\mathcal{E}|\overline{p}}.$$

Moreover, for $2 \leq i \leq n$, one has:

$$\begin{aligned}
\|\mathbf{R}(t)\mathbf{e}_i\| &= \left\|\left(1-\frac{1}{t}\right)\mathbf{e}_i - \frac{\beta_{n-i+1}}{|\mathcal{E}|}\mathbf{e}_i + \frac{\beta_{n-i+1}}{2t}\mathbf{P}_{2:}^\top\mathbf{D}^{-1}\phi_i\right\| \\
&\leq \left(\left(1-\frac{1}{t}\right) - \frac{\beta_{n-i+1}}{|\mathcal{E}|}\right)\|\mathbf{e}_i\| + \left\|\frac{\beta_{n-i+1}}{2t}\mathbf{P}_{2:}^\top\mathbf{D}^{-1}\phi_i\right\| \\
&\leq \left(\left(1-\frac{1}{t}\right) - \frac{\beta_{n-i+1}}{|\mathcal{E}|}\left(1-\frac{1}{\overline{p}t}\right)\right)\|\mathbf{e}_i\|.
\end{aligned}$$

For $t > 0$, let us define $\mu_{\mathbf{R}}(t)$ by:

$$\mu_{\mathbf{R}}(t) = \left(1-\frac{1}{t}\right) - \frac{\beta_{n-1}}{|\mathcal{E}|}\left(1-\frac{1}{\overline{p}t}\right).$$

We then have, for any $t > 0$, $|||\mathbf{R}(t)||| < \mu_{\mathbf{R}}(t)$. Thus,

$$\|\mathbf{L}_1(t)\| \leq \frac{\beta_1}{|\mathcal{E}|\overline{p}t}\left(\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_{\mathbf{R}}(r:s)\right)\|\overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n\|.$$

Also,

$$\|\mathbf{L}_2(t)\| = \left\|\sum_{s=1}^{t}\mathbf{P}\mathbf{R}(t:s)\mathbf{P}^\top\frac{\overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n}{s}\right\| \leq \sum_{s=1}^{t}\frac{\mu_{\mathbf{R}}(t:s)}{s}\|\overline{\mathbf{h}} - \hat{U}_n(h)\mathbf{1}_n\|$$

A reasoning similar to the synchronous setting can be applied to $\mathbf{L}_3(t)$:

$$\|\mathbf{L}_3(t)\| = \frac{1}{t}\left\|\sum_{s=1}^{t}\mathbf{P}\mathbf{Q}\mathbf{P}^\top\frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{R}_c^{s-1}\mathbf{P}_c^\top\mathbf{h}\right\| \leq \frac{1}{t}\cdot\frac{1}{1-\lambda_2(1)}\|\mathbf{H}-\overline{\mathbf{h}}\mathbf{1}_n^\top\|.$$

Regarding $\mathbf{L}_4(t)$, one can write:

$$\begin{aligned}\|\mathbf{L}_4(t)\| &= \frac{1}{t}\left\|\sum_{s=1}^{t}\left(\mathbf{U}\sum_{r=s}^{t-1}\mathbf{R}(r:s)\right)\frac{\mathbf{B}}{s}\mathbf{P}_c\mathbf{R}_c^{s-1}\mathbf{P}_c^\top\mathbf{s}_2(0)\right\| \\ &\leq \frac{\beta_1}{|\mathcal{E}|\overline{p}t}\sum_{s=1}^{t}\frac{1}{s}\left(\sum_{r=s}^{t-1}\mu_{\mathbf{R}}(r:s)\right)(\lambda_2(1))^{s-1}\|\mathbf{H}-\overline{\mathbf{h}}\mathbf{1}_n^\top\|,\end{aligned}$$

where $\lambda_2(\alpha)$ is defined in Section 4.7.1. Similarly, one has:

$$\|\mathbf{L}_5(t)\| \leq \|\mathbf{H}-\overline{\mathbf{h}}\mathbf{1}_n^\top\|\sum_{s=1}^{t}\frac{\mu_{\mathbf{R}}(t:s)}{s}(\lambda_2(1))^{s-1}.$$

Now, for $t > s > 1$, one only need to find appropriate rates on the quantities

$$\sum_{s=1}^{t}\frac{1}{s}\mu_{\mathbf{R}}(t:s)$$

and

$$\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_{\mathbf{R}}(r:s)$$

to conclude. Here, for $t > 1$, $\mu_{\mathbf{R}}(t)$ can be rewritten as follow:

$$\mu_{\mathbf{R}}(t) = \left(\frac{t-1}{t}\right)\lambda_2(1)\left(1+(1-\lambda_2(1))\frac{c}{t}\right),$$

with $c = \frac{1}{\lambda_2(1)\overline{p}} - 1$. If $c < 1$, one cas use a reasoning similar to the synchronous setting and conclude. However, c is often greater than 1. In this case, one has:

$$\mu_{\mathbf{R}}(t) \leq \left(\frac{t-1}{t}\right)\lambda_2(1)\left(1+\frac{c}{t}\right).$$

For $t > s > 0$, the product $\mu_{\mathbf{R}}(t:s)$ can then be bounded as follows:

$$\mu_{\mathbf{R}}(t:s) \leq \frac{s}{t}\lambda_2(1)^{t-s}\left(1+\frac{c}{t-1}\right)\ldots\left(1+\frac{c}{s}\right).$$

Using the definition of t_c, it is clear that, for $t \geq t_c$, one has:

$$\lambda_2(1)(1+\frac{c}{t-1}) < 1.$$

We can use this result to upper bound $\sum_{s=1}^{t} \mu_{\mathbf{R}}(t:s)/s$ with a geometric series:

$$\begin{aligned}
\sum_{s=1}^{t} \frac{1}{s}\mu_{\mathbf{R}}(t:s) &\leq \frac{1}{t}\sum_{s=1}^{t} \lambda_2(1)^{t-s}\left(1+\frac{c}{t-1}\right)\dots\left(1+\frac{c}{s}\right) \\
&\leq \frac{1}{t}\sum_{s=t_c+1}^{t} \lambda_2(1)^{t-s}\left(1+\frac{c}{t_c}\right)^{t-s} \\
&\quad + \frac{1}{t}\sum_{s=1}^{t_c} \lambda_2(1)^{t-s}\left(1+\frac{c}{t-1}\right)\dots\left(1+\frac{c}{s}\right) \\
&\leq \frac{1}{t}\cdot\frac{1}{1-\lambda_c} + \frac{t_c}{t}(1+c)^{t_c}e^{-(1-\lambda_c)(t-t_c)},
\end{aligned}$$

where $\lambda_c := \lambda_2(1)\left(1+\frac{c}{t_c}\right)$. Therefore, we have that $\sum_{s=1}^{t} \mu_{\mathbf{R}}(t:s)/s = O(1/t)$. Let us now focus on the second bound. For $t > t_c$ and $1 < s < t$, one has:

$$\begin{aligned}
\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_{\mathbf{R}}(r:s) &\leq \sum_{s=1}^{t_c}\frac{1}{s}\sum_{r=s}^{t_c}\mu_{\mathbf{R}}(r:s) + \sum_{s=1}^{t_c}\frac{1}{s}\sum_{r=t_c+1}^{t-1}\mu_{\mathbf{R}}(r:s) \\
&\quad + \sum_{s=t_c+1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_{\mathbf{R}}(r:s) \\
&\leq \sum_{s=1}^{t_c}\sum_{r=s}^{t_c}\frac{1}{r}\lambda_2(2)^{r-s}(1+c)^{r-s} + \sum_{s=1}^{t_c}\sum_{r=t_c+1}^{t-1}\frac{\lambda_c^{r-s}}{r} \\
&\quad + \sum_{s=t_c+1}^{t}\sum_{r=s}^{t-1}\frac{\lambda_c^{r-s}}{r} \\
&\leq t_c\sum_{r=1}^{t_c}\frac{1}{r}\lambda_2(2)^r(1+c)^r + t_c\lambda_c^{-t_c}\sum_{r=t_c+1}^{t-1}\frac{\lambda_c^r}{r} + \sum_{s=1}^{t}\frac{1}{s}\sum_{r=0}^{s-1}\lambda_c^r \\
&\leq t_c(1+c)^{t_c} - t_c\lambda_c^{-t_c}\log(1-\lambda_c) + \frac{1}{1-\lambda_c}\sum_{s=1}^{t}\frac{1}{s} \\
&\leq t_c(1+c)^{t_c} + \frac{t_c\lambda_c^{t-c}}{1-\lambda_c} + \frac{1}{1-\lambda_c}\log(t+1).
\end{aligned}$$

Thus, $\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_{\mathbf{R}}(r:s) = O(\log t)$.

Using these results and the previous expressions of $\mathbf{L}_1(t), \dots, \mathbf{L}_5(t)$, one can conclude that, for $t > 1$, $\|\mathbb{E}[\mathbf{z}(t)] - \hat{U}_n(h)\mathbf{1}_n\| = O(\log t/t)$. □

4.7.3 U2-gossip Algorithm

U2-gossip PELCKMANS and SUYKENS, 2009 is an alternative approach for computing U-statistics. In this algorithm, each node stores two auxiliary observations that are propagated using independent random walks. These two auxiliary observations will be used for estimating the U-statistic – see Algorithm 10 for details. This algorithm has an $O(1/t)$ convergence rate, as stated in Theorem 16.

Let $k \in [n]$. At iteration $t = 1$, the auxiliary observations have not been swapped yet, so the expected estimator $\mathbb{E}[z_k(1)]$ is simply updated as follow:

$$\mathbb{E}[z_k(1)] = \mathbb{E}[z_k(0)] + \mathbf{e}_k^\top \mathbf{H} \mathbf{e}_k.$$

Then, at the end of the iteration, some auxiliary observations are randomly swapped. Therefore, one has:

$$\mathbb{E}[z_k(2)] = \frac{1}{2}\mathbb{E}[z_k(1)] + \frac{1}{2}\left(\mathbf{W}_1(\mathcal{G})\,\mathbf{e}_k^\top\right)\mathbf{H}\mathbf{W}_1(\mathcal{G})\,\mathbf{e}_k,$$

where $\mathbf{W}_\alpha(\mathcal{G})$ is defined in Section 4.7.1. Using recursion, we can write, for any $t > 0$ and any $k \in [n]$:

$$\mathbb{E}[z_k(t)] = \sum_{s=0}^{t-1} \mathbf{e}_k^\top \mathbf{W}_1(\mathcal{G})^s \mathbf{H}\mathbf{W}_1(\mathcal{G})^s \mathbf{e}_k. \tag{4.18}$$

Proof of Theorem 16. Let $k \in [n]$ and $t > 0$. Using the expression of $\mathbb{E}[z_k(t)]$ established in (4.18), one has:

$$\mathbb{E}[z_k(t)] = \frac{1}{t}\sum_{s=0}^{t} \mathbf{e}_k^\top \mathbf{W}_1(\mathcal{G})^s \mathbf{H}\mathbf{W}_1(\mathcal{G})^s \mathbf{e}_k = \frac{1}{t}\sum_{s=0}^{t} \mathbf{e}_k^\top \mathbf{P}^\top \mathbf{D}_1^s \mathbf{P}\mathbf{H}\mathbf{P}\mathbf{D}_1^s \mathbf{P}^\top \mathbf{e}_k,$$

where $\mathbf{P}$ is the eigenvectors matrix introduced in Section 4.4.1 and $\mathbf{D}_1 = \operatorname{diag}(\lambda_1(1), \ldots, \lambda_n(1))$. Similarly to previous proofs, we split $\mathbf{D}_1 = \mathbf{Q}_1 + \mathbf{R}_1$ where $\mathbf{Q}_1 = \operatorname{diag}(1, 0, \ldots, 0)$ and $\mathbf{R}_1 = \operatorname{diag}(0, \lambda_2(1), \ldots, \lambda_n(1))$. Now, we can write $\mathbb{E}[z_k(t)] = L_1(t) + L_2(t) + L_3(t) + L_4(t)$ with $L_1(t)$, $L_2(t)$, $L_3(t)$ and $L_4(t)$ defined as follows:

$$\begin{cases} L_1(t) &= \frac{1}{t}\sum_{s=1}^{t} \mathbf{e}_k^\top \mathbf{P}^\top \mathbf{Q}_1^s \mathbf{P}\mathbf{H}\mathbf{P}\mathbf{Q}_1^s \mathbf{P}^\top \mathbf{e}_k \\ L_2(t) &= \frac{1}{t}\sum_{s=1}^{t} \mathbf{e}_k^\top \mathbf{P}^\top \mathbf{R}_1^s \mathbf{P}\mathbf{H}\mathbf{P}\mathbf{Q}_1^s \mathbf{P}^\top \mathbf{e}_k \\ L_3(t) &= \frac{1}{t}\sum_{s=1}^{t} \mathbf{e}_k^\top \mathbf{P}^\top \mathbf{Q}_1^s \mathbf{P}\mathbf{H}\mathbf{P}\mathbf{R}_1^s \mathbf{P}^\top \mathbf{e}_k \\ L_4(t) &= \frac{1}{t}\sum_{s=1}^{t} \mathbf{e}_k^\top \mathbf{P}^\top \mathbf{R}_1^s \mathbf{P}\mathbf{H}\mathbf{P}\mathbf{R}_1^s \mathbf{P}^\top \mathbf{e}_k \end{cases}.$$

The first term can be rewritten:

$$L_1(t) = \mathbf{e}_k^\top \mathbf{P}^\top \mathbf{Q}_1 \mathbf{P}\mathbf{H}\mathbf{P}\mathbf{Q}_1 \mathbf{P}^\top \mathbf{e}_k = \frac{1}{n^2}\mathbf{1}_n^\top \mathbf{H}\mathbf{1}_n = \hat{U}_n(h).$$

Then, one has:

$$\begin{aligned}
|L_2(t)| &\leq \frac{1}{t}\sum_{s=0}^{t}\|\mathbf{e}_k^\top \mathbf{P}\mathbf{R}_1^s\mathbf{P}^\top\mathbf{H}\mathbf{P}\mathbf{Q}_1\mathbf{P}^\top\mathbf{e}_k\| \\
&\leq \frac{1}{t}\sum_{s=0}^{t}\|\mathbf{P}\mathbf{R}_1^s\mathbf{P}^\top\overline{\mathbf{h}}\| \\
&\leq \frac{1}{t}\sum_{s=0}^{t}(\lambda_2(1))^s\|\overline{\mathbf{h}}-\hat{U}_n(h)\mathbf{1}_n\| \\
&\leq \frac{1}{t}\cdot\frac{1}{1-\lambda_2(1)}\|\overline{\mathbf{h}}-\hat{U}_n(h)\mathbf{1}_n\|,
\end{aligned}$$

since $\lambda_2(1) < 1$. Similarly, we have $|L_3(t)| \leq \frac{1}{t}\cdot\frac{\lambda_2(1)}{1-\lambda_2(1)}\|\overline{\mathbf{h}}-\hat{U}_n(h)\mathbf{1}_n\|$. The final term $L_4(t)$ can be bounded as follow:

$$\begin{aligned}
L_4(t) &\leq \frac{1}{t}\sum_{s=0}^{t}\left|\mathbf{e}_k^\top \mathbf{P}\mathbf{R}_1^s\mathbf{P}^\top\mathbf{H}\mathbf{P}\mathbf{Q}_1\mathbf{P}^\top\mathbf{e}_k\right| \\
&= \frac{1}{t}\sum_{s=0}^{t}\left|\mathbf{e}_k^\top \mathbf{P}\mathbf{R}_1^s\mathbf{P}^\top\left(\mathbf{H}-\mathbf{1}_n\overline{\mathbf{h}}^\top\right)\mathbf{P}\mathbf{Q}_1\mathbf{P}^\top\mathbf{e}_k\right| \\
&\leq \frac{1}{t}\sum_{s=0}^{t}(\lambda_2(1))^{2s}\left\|\mathbf{H}-\mathbf{1}_n\overline{\mathbf{h}}^\top\right\| \\
&\leq \frac{1}{t}\cdot\frac{1}{1-(\lambda_2(1))^2}\left\|\mathbf{H}-\mathbf{1}_n\overline{\mathbf{h}}^\top\right\|.
\end{aligned}$$

With above relations, the expected difference can be bounded as follow:

$$\begin{aligned}
\left|\mathbb{E}[z_k(t)]-\hat{U}_n(h)\right| &\leq |L_2(t)|+|L_3(t)|+|L_4(t)| \\
&\leq \frac{1}{t}\cdot\frac{2}{1-\lambda_2(1)}\left\|\overline{\mathbf{h}}-\hat{U}_n(h)\mathbf{1}_n\right\| \\
&\quad+\frac{1}{t}\cdot\frac{1}{1-(\lambda_2(1))^2}\left\|\mathbf{H}-\mathbf{1}_n\overline{\mathbf{h}}^\top\right\|.
\end{aligned}$$

Finally, we can conclude:

$$\begin{aligned}
\left\|\mathbb{E}[\mathbf{Z}(t)]-\hat{U}_n(h)\right\| &\leq \sqrt{n}\max_{k\in[n]}\left|\mathbb{E}[z_k(t)]-\hat{U}_n(h)\right| \\
&\leq \frac{\sqrt{n}}{t}\cdot\frac{2}{1-\lambda_2(1)}\left\|\overline{\mathbf{h}}-\hat{U}_n(h)\mathbf{1}_n\right\| \\
&\quad+\frac{\sqrt{n}}{t}\cdot\frac{1}{1-(\lambda_2(1))^2}\left\|\mathbf{H}-\mathbf{1}_n\overline{\mathbf{h}}^\top\right\|.
\end{aligned}$$

□

Chapter 5

Gossip Dual Averaging for Decentralized Optimization of Pairwise Functions

5.1 Introduction

The increasing popularity of large-scale and fully decentralized computational architectures, fueled for instance by the advent of the "Internet of Things", motivates the development of efficient optimization algorithms adapted to this setting. An important application is machine learning in wired and wireless networks of agents (sensors, connected objects, mobile phones, *etc.*), where the agents seek to minimize a global learning objective which depends of the data collected locally by each agent. In such networks, it is typically impossible to efficiently centralize data or to globally aggregate intermediate results: agents can only communicate with their immediate neighbors (*e.g.,* agents within a small distance), often in a completely asynchronous fashion. Standard distributed optimization and machine learning algorithms (implemented for instance using MapReduce or Spark) require a coordinator node and/or to maintain synchrony, and are thus unsuitable for use in decentralized networks.

In contrast, *gossip algorithms* (TSITSIKLIS, 1984; BOYD et al., 2006; KEMPE, DOBRA, and GEHRKE, 2003; SHAH, 2009) are tailored to this setting because they only rely on simple peer-to-peer communication: each agent only exchanges information with one neighbor at a time. Various gossip algorithms have been proposed to solve the flagship problem of decentralized optimization, namely to find a parameter vector $\boldsymbol{\theta}$ which minimizes an average of convex functions:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{\theta}; \mathbf{x}_i), \tag{5.1}$$

where the data $\mathbf{x}_i$ is only known to agent i. The most popular algorithms are based on (sub)gradient descent (JOHANSSON, RABI, and JOHANSSON, 2010; NEDIĆ and OZDAGLAR, 2009; RAM, NEDIĆ, and VEERAVALLI, 2010; BIANCHI and JAKUBOWICZ, 2013), ADMM (WEI and OZDAGLAR, 2012; WEI and OZDAGLAR, 2013; IUTZELER et al., 2013) or dual averaging (DUCHI, AGARWAL, and WAINWRIGHT, 2012; YUAN et al., 2012; LEE, NEDIĆ, and RAGINSKY, 2015; TSIANOS, LAWLOR, and RABBAT, 2015), some of which can also accommodate constraints or regularization on $\boldsymbol{\theta}$. The main idea underlying these methods is that each agent seeks to minimize its local function by applying local updates (*e.g.,* gradient steps) while exchanging information with neighbors to ensure a global convergence to the consensus value.

In this chapter, we tackle the problem of minimizing an average of *pairwise* functions of the agents' data:

$$\min_{\theta} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j). \tag{5.2}$$

This problem finds numerous applications in statistics and machine learning, *e.g.,* Area Under the ROC Curve (AUC) maximization (ZHAO et al., 2011), distance or similarity learning (BELLET, HABRARD, and SEBBAN, 2015), ranking (CLÉMENÇON, LUGOSI, and VAYATIS, 2008), supervised graph inference (BIAU and BLEAKLEY, 2006) and multiple kernel learning (KUMAR et al., 2012), to name a few. As a motivating example, consider a mobile phone application which locally collects information about its users.

The provider could be interested in learning pairwise similarity functions between users in order to group them into clusters or to recommend them content without having to centralize data on a server or to synchronize phones.

The main difficulty in Problem (5.2) comes from the fact that each term of the sum depends on two agents i and j, making the local update schemes of previous approaches impossible to apply unless data is exchanged between nodes. Although gossip algorithms have recently been introduced to evaluate such pairwise functions for a *fixed* $\boldsymbol{\theta}$ (see PELCKMANS and SUYKENS, 2009 and Chapter 4 in this thesis), to the best of our knowledge, efficiently finding the *optimal solution* $\boldsymbol{\theta}$ in a decentralized way remains an open challenge. Our contributions towards this objective are as follows. We propose new gossip algorithms based on dual averaging (NESTEROV, 2009; XIAO, 2010) to efficiently solve Problem (5.2) and its constrained or regularized variants. Central to our methods is a light data propagation scheme which allows the nodes to compute *biased* estimates of the gradients of functions in (5.2). We then propose a theoretical analysis of our algorithms both in synchronous and asynchronous settings establishing their convergence under an additional hypothesis that the bias term decreases fast enough over the iterations (and we have observed such a fast decrease in all our experiments). Finally, we present some numerical simulations on Area Under the ROC Curve (AUC) maximization and metric learning problems. These experiments illustrate the practical performance of the proposed algorithms and the influence of network topology, and show that in practice the influence of the bias term is negligible as it decreases very fast with the number of iterations.

The chapter is organized as follows. Section 5.2 formally introduces the problem of interest. Section 5.3 then introduces the dual averaging algorithm — which is at the root of our method — and provides a theoretical analysis of this method. Section 5.4 presents the decentralized dual averaging method for solving Problem (5.1). Section 5.5 introduces the proposed gossip algorithms and their convergence analysis. Section 5.6 extends our results to multiple observations per node. Section 5.7 displays our numerical simulations. Finally, concluding remarks are collected in Section 5.8.

5.2 Notations and problem statement

5.2.1 Definitions and Notation

For any integer $p > 0$, we denote by $[p]$ the set $\{1, \dots, p\}$ and by $|\mathcal{C}|$ the cardinality of any finite set $\mathcal{C}$. We denote an undirected graph by $\mathcal{G} = ([n], \mathcal{E})$, where $[n]$ is the set of vertices and $\mathcal{E} \subseteq [n] \times [n]$ is the set of edges. We denote by $\mathbf{A}^{\mathcal{G}}$ the adjacency matrix related to the graph $\mathcal{G}$, that is for all $(i, j) \in [n]^2$, $[\mathbf{A}^{\mathcal{G}}]_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$. For any node $i \in [n]$, we denote its degree by $d_i = |\{j : (i, j) \in \mathcal{E}\}|$. We denote by $\mathbf{L}^{\mathcal{G}}$ the graph Laplacian of $\mathcal{G}$, defined by $\mathbf{L}^{\mathcal{G}} = \mathbf{D}^{\mathcal{G}} - \mathbf{A}^{\mathcal{G}}$ where $\mathbf{D}^{\mathcal{G}} = \operatorname{diag}(d_1, \dots, d_n)$ is the matrix of degrees. When it is clear from context, we will drop the $\mathcal{G}$ exponent. A graph $\mathcal{G} = ([n], \mathcal{E})$ is said to be connected if for all $(i, j) \in [n]^2$ there exists a path connecting i and j; it is bipartite if there exist $\mathcal{S}, \mathcal{T} \subset [n]$ such that $\mathcal{S} \cup \mathcal{T} = [n]$, $\mathcal{S} \cap \mathcal{T} = \emptyset$ and $\mathcal{E} \subseteq (\mathcal{S} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S})$.

The transpose of a matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is denoted by $\mathbf{M}^\top$. A matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ is termed stochastic whenever $\mathbf{P} \geq 0$ and $\mathbf{P}\mathbf{1}_n = \mathbf{1}_n$, where $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$, and bi-stochastic whenever both $\mathbf{P}$ and $\mathbf{P}^\top$ are stochastic. We denote by $\mathbf{I}_n$ the identity matrix in $\mathbb{R}^{n \times n}$, by $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ the canonical basis of $\mathbb{R}^n$, by $\mathbb{1}_{\{\mathcal{X}\}}$ the indicator function of any event $\mathcal{X}$ and by $\|\cdot\|$ the usual ℓ_2-norm. For $\boldsymbol{\theta} \in \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}$, we denote by $\nabla g(\boldsymbol{\theta})$ the gradient of g at $\boldsymbol{\theta}$. Finally, given a collection of vectors $\mathbf{u}_1, \dots, \mathbf{u}_n$, we denote by $\bar{\mathbf{u}} = (1/n) \sum_{i=1}^n \mathbf{u}_i$ its empirical mean.

5.2.2 Problem Statement

We represent a network of n agents as an undirected graph $\mathcal{G} = ([n], \mathcal{E})$, where each node $i \in [n]$ corresponds to an agent and $(i, j) \in \mathcal{E}$ if nodes i and j can exchange information directly (*i.e.*, they are neighbors). For ease of exposition, we assume that each node $i \in [n]$ holds a single data point $\mathbf{x}_i \in \mathcal{X}$. Though restrictive in practice, this assumption can easily be relaxed, but it would lead to more technical details to handle the storage size, without changing the overall analysis (see Section 5.6 for details).

Given $d > 0$, let $f : \mathbb{R}^d \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a differentiable and convex function with respect to the first variable. We assume that for any $(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2$, there exists $L_f > 0$ such that $f(\cdot; \mathbf{x}, \mathbf{x}')$ is L_f-Lipschitz (with respect to the ℓ_2-norm). Let $\psi : \mathbb{R}^d \to \mathbb{R}^+$ be a non-negative, convex, possibly non-smooth, function such that, for simplicity, $\psi(0) = 0$. We aim at solving the following optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) + \psi(\boldsymbol{\theta}). \tag{5.3}$$

In a typical machine learning scenario, Problem (5.3) is a (regularized) empirical risk minimization problem and θ corresponds to the model parameters to be learned. The quantity $f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j)$ is a pairwise loss measuring the performance of the model $\boldsymbol{\theta}$ on the data pair $(\mathbf{x}_i, \mathbf{x}_j)$, while $\psi(\boldsymbol{\theta})$ represents a regularization term penalizing the complexity of $\boldsymbol{\theta}$. Common examples of regularization terms include indicator functions of a closed convex set to model explicit convex constraints, or norms enforcing specific properties such as sparsity (a canonical example being the ℓ_1-norm).

Many machine learning problems can be cast as Problem (5.3). For instance, in AUC maximization (ZHAO et al., 2011), binary labels $(\ell_1, \dots, \ell_n) \in \{-1, 1\}^n$ are assigned to the data points and we want to learn a (linear) scoring rule $\mathbf{x} \mapsto \mathbf{x}^\top \boldsymbol{\theta}$ which hopefully gives larger scores to positive data points than to negative ones. One may use the logistic loss

$$f(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j) = \mathbb{1}_{\{\ell_i > \ell_j\}} \log\left(1 + \exp((\mathbf{x}_j - \mathbf{x}_i)^\top \boldsymbol{\theta})\right),$$

and the regularization term $\psi(\boldsymbol{\theta})$ can be the square ℓ_2-norm of $\boldsymbol{\theta}$ (or the ℓ_1-norm when a sparse model is desired). Other popular instances of Problem (5.3) include metric learning (BELLET, HABRARD, and SEBBAN, 2015), ranking (CLÉMENÇON, LUGOSI, and VAYATIS, 2008), supervised graph inference (BIAU and BLEAKLEY, 2006) and multiple kernel learning (KUMAR et al., 2012).

For notational convenience, we denote by f_i the partial sum function $(1/n) \sum_{j=1}^n f(\cdot; \mathbf{x}_i, \mathbf{x}_j)$ for $i \in [n]$ and by $f = (1/n) \sum_{i=1}^n f_i$. Problem (5.3) can then be recast as:

$$\min_{\theta \in \mathbb{R}^d} R_n(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \psi(\boldsymbol{\theta}). \tag{5.4}$$

Note that the function f is L_f-Lipschitz, since all the f_i are L_f-Lipschitz.

Throughout the chapter, we assume that the function f is differentiable, but we expect all our results to hold even when f is non-smooth, for instance in ℓ_1-regression problems or when using the hinge loss. In this case, one simply needs to replace gradients by subgradients in our algorithms, and a similar analysis could be performed.

We now study the dual averaging method and some of its extensions in the centralized case, as several results will be needed for the analysis of the decentralized setting.

5.3 Centralized Dual Averaging

5.3.1 Deterministic Setting

In this section, we review the dual averaging optimization algorithm (NESTEROV, 2009; XIAO, 2010) to solve Problem (5.3) in the centralized setting (where all data lie on the same machine). This method is at the root of our gossip algorithms, for reasons that will be made clear in Section 5.5. To explain the main idea behind dual averaging, let us first consider the iterations of Stochastic Gradient Descent (SGD), assuming $\psi \equiv 0$ for simplicity:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \gamma(t)\mathbf{g}(t),$$

where $\mathbb{E}[\mathbf{g}(t)|\boldsymbol{\theta}(t)] = \nabla f(\boldsymbol{\theta}(t))$, and $(\gamma(t))_{t\geq 0}$ is a non-negative and non-increasing step size sequence. This update rule can be rewritten equivalently as follows:

$$\boldsymbol{\theta}(t+1) = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{ f(\boldsymbol{\theta}(t)) + (\boldsymbol{\theta} - \boldsymbol{\theta}(t))^\top \mathbf{g}(t) + \frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}(t)\|^2}{2\gamma(t)} \right\},$$

meaning that $\boldsymbol{\theta}(t+1)$ is the minimizer of some quadratic approximation of f around $\boldsymbol{\theta}(t)$. Recursively and assuming that $\boldsymbol{\theta}(0) = 0$, one can obtain:

$$\boldsymbol{\theta}(t+1) = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{ \boldsymbol{\theta}^\top \left(\sum_{s=0}^{t} \gamma(s)\mathbf{g}(s) \right) + \frac{\|\boldsymbol{\theta}\|}{2} \right\}. \tag{5.5}$$

For SGD to converge to an optimal solution, the step size sequence must satisfy $\gamma(t) \underset{t\to+\infty}{\longrightarrow} 0$ and $\sum_{t=0}^{\infty} \gamma(t) = \infty$. As noticed by NESTEROV (2009), an undesirable consequence is that new gradient estimates are given smaller weights than old ones in (5.5). Dual averaging aims at integrating all gradient estimates with the same weight.

Let $(\gamma(t))_{t\geq 0}$ be a positive and non-increasing step size sequence. The dual averaging algorithm maintains a sequence of primal iterates $(\boldsymbol{\theta}(t))_{t>0}$, and a sequence $(\mathbf{z}(t))_{t\geq 0}$ of dual variables which collects the sum of the unbiased gradient estimates seen up to time t. We initialize to $\boldsymbol{\theta}(1) = \mathbf{z}(0) = 0$. At each step $t > 0$, we compute an unbiased estimate $\mathbf{g}(t)$ of $\nabla f(\boldsymbol{\theta}(t))$. The most common choice is to take $\mathbf{g}(t) = \nabla f(\boldsymbol{\theta}; \mathbf{x}_{I_t}, \mathbf{x}_{J_t})$ where I_t and J_t are drawn uniformly at random from $[n]$. We then set $\mathbf{z}(t+1) = \mathbf{z}(t) + \mathbf{g}(t)$ and generate the next iterate with the following rule:

$$\begin{cases} \boldsymbol{\theta}(t+1) = \pi_t^{\psi}(\mathbf{z}(t+1)), \\ \pi_t^{\psi}(\mathbf{z}) := \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{ -\mathbf{z}^\top\boldsymbol{\theta} + \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t)} + t\psi(\boldsymbol{\theta}) \right\}. \end{cases}$$

This particular formulation was introduced in (XIAO, 2009; XIAO, 2010), extending the method introduced by NESTEROV, 2009 in the specific case of indicator functions. In this work, we borrow the notation from XIAO, 2010. When it is clear from the context, we will drop the dependence in ψ and simply write $\pi_t(\mathbf{z}) = \pi_t^{\psi}(\mathbf{z})$.

Note that $\pi_t(\cdot)$ is related to the proximal operator of a function $\phi : \mathbb{R}^d \to \mathbb{R}$ defined by

$$\text{prox}_{\phi}(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathbb{R}^d} \left\{ \frac{\|\mathbf{z}-\mathbf{x}\|^2}{2} + \phi(\mathbf{x}) \right\}.$$

Indeed, one can write:

$$\pi_t(\mathbf{z}) = \text{prox}_{t\gamma(t)\psi}\left(\gamma(t)\mathbf{z}\right).$$

For many functions ψ of practical interest, $\pi_t(\cdot)$ has a closed form solution. For instance, when $\psi = \|\cdot\|^2$, $\pi_t(\cdot)$ corresponds to a simple scaling, and when $\psi = \|\cdot\|_1$ it is a soft-thresholding operator. If ψ is the indicator function of a closed convex set $\mathcal{C}$, then $\pi_t(\cdot)$ is the projection operator onto $\mathcal{C}$.

The dual averaging method is summarized in Algorithm 13. In order to perform a theoretical analysis of this algorithm, we introduce the following function. Let us define, for $t \geq 0$

$$V_t(\mathbf{z}) := \max_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{ \mathbf{z}^\top\boldsymbol{\theta} - \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t)} - t\psi(\boldsymbol{\theta}) \right\}.$$

Remark that with the assumption that $\psi(0) = 0$, then $V_t(0) = 0$. Strong convexity in $\boldsymbol{\theta}$ of the objective function, ensures that the solution of the optimization problem is unique. The following lemma links the function V_t and the algorithm update and is a simple application of the results from (XIAO, 2009, Lemma 10):

Lemma 9. For any $\mathbf{z} \in \mathbb{R}^d$, one has:

$$\pi_t(\mathbf{z}) = \nabla V_t(\mathbf{z}), \tag{5.6}$$

and the following statements hold true: for any $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^d$

$$\|\pi_t(\mathbf{z}_1) - \pi_t(\mathbf{z}_2)\| \leq \gamma(t)\|\mathbf{z}_1 - \mathbf{z}_2\|, \tag{5.7}$$

and for any $\mathbf{g}, \mathbf{z} \in \mathbb{R}^d$,

$$V_t(\mathbf{z}+\mathbf{g}) \leq V_t(\mathbf{z}) + \mathbf{g}^\top\nabla V_t(\mathbf{z}) + \frac{\gamma(t)}{2}\|\mathbf{g}\|^2. \tag{5.8}$$

Moreover, adapting (XIAO, 2009, Lemma 11) we can state:

Lemma 10. For any $t \geq 1$ and any non-increasing sequence $(\gamma(t))_{t\geq 1}$, we have

$$V_t\left(-\mathbf{z}(t+1)\right) + \psi(\boldsymbol{\theta}(t+1)) \leq V_{t-1}\left(-\mathbf{z}(t+1)\right). \tag{5.9}$$

We also need a last technical result that we will use several times in the following:

Lemma 11. Let $\boldsymbol{\theta}(t) = \pi_t(\sum_{s=1}^{t-1}\mathbf{g}(s))$, and let $(\gamma(t))_{t\geq 1}$ be a non-increasing and non-negative sequence sequence (with the convention $\gamma(0) = 0$), then for any $\boldsymbol{\theta} \in \mathbb{R}^d$:

Algorithm 13 Centralized dual averaging

Require: Step size $(\gamma(t))_{t\geq 1} > 0$.
1: Initialization $\boldsymbol{\theta} = 0$, $\bar{\boldsymbol{\theta}} = 0$, $z = 0$.
2: **for** $t = 1, \dots, T$ **do**
3: Update $\mathbf{z} \leftarrow \mathbf{z} + \nabla f(\boldsymbol{\theta})$
4: Update $\boldsymbol{\theta} \leftarrow \pi_t(\mathbf{z})$
5: Update $\bar{\boldsymbol{\theta}} \leftarrow \left(1 - \frac{1}{t}\right)\bar{\boldsymbol{\theta}} + \frac{1}{t}\boldsymbol{\theta}$
6: **end for**
7: **return** $\bar{\theta}$

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{g}(t)^\top(\boldsymbol{\theta}(t)-\boldsymbol{\theta}) + \frac{1}{T}\sum_{t=1}^{T}(\psi(\boldsymbol{\theta}(t))-\psi(\boldsymbol{\theta})) \leq \frac{1}{T}\sum_{t=1}^{T}\frac{\gamma(t-1)}{2}\|\mathbf{g}(t)\|^2 + \frac{\|\boldsymbol{\theta}\|^2}{2T\gamma(T)}. \tag{5.10}$$

Proof. Using the definition of V_T, one can get the following upper bound:

$$\begin{aligned}\sum_{t=1}^{T}\Big(\mathbf{g}(t)^\top\boldsymbol{\theta} + \psi(\boldsymbol{\theta})\Big) &= \mathbf{z}(T+1)^\top\boldsymbol{\theta} + T\psi(\boldsymbol{\theta}) \\ &= \mathbf{z}(T+1)^\top\boldsymbol{\theta} + T\psi(\boldsymbol{\theta}) + \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(T)} - \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(T)} \\ &\leq \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(T)} + V_T(-\mathbf{z}(T+1)).\end{aligned} \tag{5.11}$$

Then one can check that with (5.8) and Lemma 10 that, for any $1 \leq t \leq T$:

$$\begin{aligned}V_t(-\mathbf{z}(t+1)) + \psi(\boldsymbol{\theta}(t+1)) &\leq V_{t-1}(-\mathbf{z}(t+1)) \\ &= V_{t-1}(-\mathbf{z}(t) - \mathbf{g}(t)) \\ &\leq V_{t-1}(-\mathbf{z}(t)) - \mathbf{g}(t)^\top\boldsymbol{\theta}(t) + \frac{\gamma(t-1)}{2}\|\mathbf{g}(t)\|^2.\end{aligned}$$

From the last display, the following holds:

$$\mathbf{g}(t)^\top\boldsymbol{\theta}(t) + \psi(\boldsymbol{\theta}(t+1)) \leq V_{t-1}(-\mathbf{z}(t)) - V_t(-\mathbf{z}(t+1)) + \frac{\gamma(t-1)}{2}\|\mathbf{g}(t)\|^2.$$

Summing the former for $t = 1, \dots, T$ yields

$$\sum_{t=1}^{T}\mathbf{g}(t)^\top\boldsymbol{\theta}(t) + \psi(\boldsymbol{\theta}(t+1)) \leq V_0(-\mathbf{z}_0) - V_T(-\mathbf{z}_T) + \sum_{t=1}^{T}\frac{\gamma(t-1)}{2}\|\mathbf{g}_t\|^2.$$

Remark that $V_0(0) = 0$ and $\psi(\boldsymbol{\theta}(1)) - \psi(\boldsymbol{\theta}(T+1)) = -\psi(\boldsymbol{\theta}(T+1)) \leq 0$, so the previous display can be reduced to:

$$\sum_{t=1}^{T}\mathbf{g}(t)^\top\boldsymbol{\theta}(t) + \psi(\boldsymbol{\theta}(t)) + V_T(-\mathbf{z}(T+1)) \leq \sum_{t=1}^{T}\frac{\gamma(t-1)}{2}\|\mathbf{g}(t)\|^2. \tag{5.12}$$

Combining with (5.11), the lemma holds true. □

Bounding the error of the dual averaging is provided in the next theorem, where we remind that $R_n = f + \psi$:

Theorem 19 (Dual averaging). *Let $(\gamma(t))_{t\geq 1}$ be a non increasing sequence. Let $(\mathbf{z}(t))_{t\geq 1}$, $(\boldsymbol{\theta}(t))_{t\geq 1}$, $(\bar{\boldsymbol{\theta}}(t))_{t\geq 1}$ and $(\mathbf{g}(t))_{t\geq 1}$ be generated according to Algorithm 13. Assume that the function f is L_f-Lipschitz and let $\boldsymbol{\theta}^* \in \mathbb{R}^d$ be a minimizer of R_n,* i.e., *$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}' \in \mathbb{R}^d} R_n(\boldsymbol{\theta}')$. Then for any $T \geq 2$, one has:*

$$R_n(\bar{\boldsymbol{\theta}}(T)) - R_n(\boldsymbol{\theta}^*) \leq \frac{\|\boldsymbol{\theta}^*\|^2}{2T\gamma(T)} + \frac{L_f^2}{2T}\sum_{t=1}^{T-1}\gamma(t). \tag{5.13}$$

Moreover, if one knows $D > 0$ such that $\|\boldsymbol{\theta}^*\| \leq D$, then for the choice $\gamma(t) = \frac{D}{L_f\sqrt{2t}}$, one has:

$$R_n(\bar{\boldsymbol{\theta}}(T)) - R_n(\boldsymbol{\theta}^*) \leq \frac{\sqrt{2}DL_f}{\sqrt{T}}.$$

Proof. Let $T \geq 2$. Using the convexity of f and ψ, we can get:

$$\begin{aligned}
R_n(\bar{\boldsymbol{\theta}}(T)) - R_n(\boldsymbol{\theta}^*) &\leq \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*) + \psi(\bar{\boldsymbol{\theta}}) - \psi(\boldsymbol{\theta}^*) \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \mathbf{g}(t)^\top(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*) + \frac{1}{T}\sum_{t=1}^{T}(\psi(\boldsymbol{\theta}(t)) - \psi(\boldsymbol{\theta}^*)) \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \frac{\gamma(t-1)}{2}\|\mathbf{g}(t)\|^2 + \frac{\|\boldsymbol{\theta}\|^2}{2T\gamma(T)}\,.
\end{aligned}$$

where the second inequality holds since $\mathbf{g}(t) = \nabla f(\boldsymbol{\theta}(t))$, and the third one is from an application of Lemma 11 with the choice $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. We can conclude the proof provided that $\|\mathbf{g}(t)\| \leq L_f$, which is true whenever f is L_f-Lipschitz. □

5.3.2 Stochastic Dual Averaging

Similarly to sub-gradient descent algorithms, one can adapt dual averaging algorithm to a stochastic setting; this was studied extensively by Xiao (2009). Instead of updating the dual variable $\mathbf{z}(t)$ with the (full) gradient of f at $\boldsymbol{\theta}(t)$, one now only requires the *expected* value of the update to be the gradient, that is:

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \mathbf{g}(t),$$

with $\mathbb{E}[\mathbf{g}(t)|\boldsymbol{\theta}(t)] = \nabla f(\boldsymbol{\theta}(t))$. As in the gradient descent case, convergence results still hold in expectation, as stated in Theorem 20.

Theorem 20 (Stochastic dual averaging). *Let $(\gamma(t))_{t\geq 1}$ be a non increasing sequence. Let $(\mathbf{z}(t))_{t\geq 1}$, $(\boldsymbol{\theta}(t))_{t\geq 1}$ and $(\mathbf{g}(t))_{t\geq 1}$ be generated according to stochastic dual averaging rules. Assume that the function f is L_f-Lipschitz and that $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}' \in \mathbb{R}^d} R_n(\boldsymbol{\theta}')$, then for any $T \geq 2$, one has:*

$$\mathbb{E}_T\Big[R_n(\bar{\boldsymbol{\theta}}(T)) - R_n(\boldsymbol{\theta}^*)\Big] \leq \frac{\|\boldsymbol{\theta}^*\|^2}{2T\gamma(T)} + \frac{L_f^2}{2T}\sum_{t=1}^{T-1}\gamma(t), \tag{5.14}$$

where $\mathbb{E}_T$ is the expectation over all possible sequence $(\mathbf{g}(t))_{1\leq t\leq T}$.

Moreover, if one knows that $D > 0$ such that $\|\boldsymbol{\theta}^*\| \leq D$, then for $\gamma(t) = \frac{D}{L_f\sqrt{2t}}$, one has:

$$\mathbb{E}_T\big[R_n(\bar{\boldsymbol{\theta}}(T)) - R_n(\boldsymbol{\theta}^*)\big] \leq \frac{\sqrt{2}DL_f}{\sqrt{T}}.$$

Proof. One only has to prove that the convexity inequality in Lemma 11 holds in expectation. The rest of the proof can be directly adapted from Theorem 19.

Let $T \geq 2$; using the convexity of f, one obtains:

$$\mathbb{E}_T[f(\bar{\boldsymbol{\theta}}(T)) - f(\boldsymbol{\theta}^*)] \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_T[f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*)].$$

For any $0 < t \leq T$, $\mathbb{E}[\boldsymbol{\theta}(t)|\mathbf{g}(0),\dots,\mathbf{g}(t-1)] = \boldsymbol{\theta}(t)$. Therefore, we have:

$$\mathbb{E}_T[f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*)] = \mathbb{E}_{t-1}[f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*)].$$

The vector $\mathbb{E}_t[\mathbf{g}(t)|\boldsymbol{\theta}(t)]$ is the gradient of f at $\boldsymbol{\theta}(t)$, we can then use f convexity to write:

$$\mathbb{E}_{t-1}[f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*)] \leq \mathbb{E}_{t-1}\Big[(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)^\top\mathbb{E}_t[\mathbf{g}(t)|\boldsymbol{\theta}(t)]\Big].$$

Using properties of conditional expectation, we obtain:

$$\begin{aligned}\mathbb{E}_{t-1}\Big[(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)^\top\mathbb{E}_t[\mathbf{g}(t)|\boldsymbol{\theta}(t)]\Big] &= \mathbb{E}_{t-1}\Big[\mathbb{E}_t[(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)^\top\mathbf{g}(t)|\boldsymbol{\theta}(t)]\Big] \\ &= \mathbb{E}_t[(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)^\top\mathbf{g}(t)].\end{aligned}$$

Finally, we can write:

$$\begin{aligned}\mathbb{E}_T[f(\bar{\boldsymbol{\theta}}(T) - f(\boldsymbol{\theta}^*)] &\leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_t[(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)^\top\mathbf{g}(t)] \\ &= \mathbb{E}_T\left[\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*)^\top\mathbf{g}(t)\right]. \qquad (5.15)\end{aligned}$$

Therefore, the convexity inequality holds in expectation and one can adapt the proof of Theorem 19 to conclude. □

5.3.3 Ergodic dual averaging

The previous analysis is sufficient for providing convergence rate of a decentralized optimization when the objective is separable in the observations. For pairwise objectives however, an additional look at the dual averaging is needed. Indeed, one key insight to the method we describe later on is that biased estimates of gradients are computed in opposition to unbiased estimates of the stochastic dual averaging. However, estimate bias decreases exponentially fast, so it should not penalize heavily the convergence rate. We thus study the bias influence using an ergodic analysis.

Problem setting

We define $F : \mathcal{X} \times \Delta_n$ as follows:

$$F : \left\{ \begin{array}{ccc} \mathcal{X} \times \Delta_n & \to & \mathbb{R} \\ (\boldsymbol{\theta}, \boldsymbol{\xi}) & \mapsto & \sum_{i=1}^n \xi_i f_i(\boldsymbol{\theta}) \end{array} \right. ,$$

where Δ_n is the simplex in $\mathbb{R}^n$, *i.e.*,

$$\Delta_n = \{\boldsymbol{\xi} \in \mathbb{R}_+^n, \|\boldsymbol{\xi}\|_1 = 1\}.$$

Our goal is to solve the following optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathcal{X}} f(\boldsymbol{\theta}) = F\left(\boldsymbol{\theta}, \frac{\mathbf{1}_n}{n}\right) = \frac{1}{n} \sum_{i=1}^n f_i(\boldsymbol{\theta}). \tag{5.16}$$

Throughout this section, we make the assumption that there exists $D > 0$, such that if $\boldsymbol{\theta} \in \mathcal{X}$ then $\|\boldsymbol{\theta}\| \leq D$. Using the dual averaging approach, one aims at finding an algorithm for solving problem (5.16) with "noisy" information, in a way to be defined later. In the dual averaging method with true gradient information, variables are updated as follows:

$$\left\{ \begin{array}{ccc} \mathbf{z}(t+1) & = & \mathbf{z}(t) + \nabla f(\boldsymbol{\theta}(t)) \\ \boldsymbol{\theta}(t+1) & = & \pi_t(\mathbf{z}(t+1)) \end{array} \right. . \tag{5.17}$$

As mentioned previously, we focus here on a noisy setting, similar to ergodic mirror descent introduced in DUCHI et al., 2012. Let $(\boldsymbol{\xi}(t))_{t \geq 0}$ be a sequence of — non necessarily independent — random variables over Δ_n. For $t \geq 0$, we denote as $P(t)$ the distribution of $\boldsymbol{\xi}(t)$ and we assume that there exists P^∞ such that $\lim_{t \to +\infty} \|P(t) - P^\infty\|_{TV} = 0$ and

$$\mathbb{E}_{P^\infty}\left[F(\cdot, \boldsymbol{\xi})\right] = f. \tag{5.18}$$

We make the additional assumption that one may not access the true value of $\nabla_{\boldsymbol{\theta}} F\left(., \mathbf{1}_n/n\right)$. Instead, at iteration t, one can only compute an estimate $\nabla_{\boldsymbol{\theta}} F\left(., \boldsymbol{\xi}(t)\right)$. The iterative process described in (5.17) can then be reformulated:

$$\left\{ \begin{array}{ccc} \mathbf{z}(t+1) & = & \mathbf{z}(t) + \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t)) \\ \boldsymbol{\theta}(t+1) & = & \pi_t(\mathbf{z}(t+1)) \end{array} \right. . \tag{5.19}$$

We aim at finding a condition on the random process $(\boldsymbol{\xi}(t))_{t \geq 0}$ for this method to converge to the solution of the original problem (5.16). For this purpose we introduce, for any $t > 0$, the mixing time of the distribution $P(t)$ towards its limit P^∞:

$$\tau(t, \cdot) : \epsilon \mapsto \inf\left\{s \geq 0, ||P(t+s|t) - P^\infty||_{TV} \leq \epsilon\right\}, \tag{5.20}$$

where $||\cdot||_{TV}$ is the total variation distance between two distributions and $P(t+s|t)$ is the distribution of $\boldsymbol{\xi}(t+s)$ conditioned on the natural filtration $\mathcal{F}_t := \sigma(\boldsymbol{\xi}(1), \ldots, \boldsymbol{\xi}(t))$.

Convergence analysis

The convergence analysis of such a setting relies on one key observation, described in DUCHI et al., 2012: for any $0 \leq \tau < T$ and any $\boldsymbol{\theta}^* \in \mathcal{X}$, the regret can be decomposed as follows

$$\sum_{t=1}^{T}(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*)) =$$

$$\sum_{t=1}^{T-\tau}(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*) + F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t+\tau)) - F(\boldsymbol{\theta}^*, \boldsymbol{\xi}(t+\tau))) \quad (5.21)$$

$$+ \sum_{t=1}^{T-\tau}(F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t+\tau)) - F(\boldsymbol{\theta}(t+\tau), \boldsymbol{\xi}(t+\tau))) \quad (5.22)$$

$$+ \sum_{t=\tau+1}^{T}(F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t)) - F(\boldsymbol{\theta}^*, \boldsymbol{\xi}(t))) \quad (5.23)$$

$$+ \sum_{t=T-\tau+1}^{T}(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*)). \quad (5.24)$$

The term (5.21) represents the difference between evaluating error on the true function and on a noisy function after τ steps of mixing. Next, the term (5.22) corresponds to the gap between noisy objectives of τ consecutive iterates. Term (5.23) matches the usual optimization error obtained in a noiseless setting. Finally, the last term (5.24) is a residual term and will be negligible when T goes to infinity. This decomposition is particularly helpful for convergence analysis. Indeed, the structure of dependence between $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ can be very complex and one could struggle to adapt standard optimization analysis to this setting. In each of the four terms of the regret reformulation, we focus on variations of either $\boldsymbol{\theta}$ or $\boldsymbol{\xi}$ but never both simultaneously, thus removing some dependence issues.

Following the reasoning of DUCHI et al., 2012, we will provide a bound on each term of the decomposition — some bounds actually being expected bounds (see Section 5.9 for detailed proofs).

Lemma 12 (Error after mixing)**.** Let $\boldsymbol{\theta}$ be a $\mathcal{F}_t$-mesurable variable. Then for any $\boldsymbol{\theta}^* \in \mathcal{X}$ and any $\tau > 0$, one has:

$$\mathbb{E}[f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}^*) + F(\boldsymbol{\theta}, \boldsymbol{\xi}(t+\tau)) - F(\boldsymbol{\theta}^*, \boldsymbol{\xi}(t+\tau))|\mathcal{F}_t] \leq 2LD\|P(t+\tau|t) - P^{\infty}\|_{TV}. \quad (5.25)$$

Lemma 13 (Consecutive iterates bound)**.** Let $(\boldsymbol{\theta}(t))_{t\geq 0}$ be generated according to (5.19). Then, for any $t \geq 0$:

$$\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \leq 3L_f\left(1 + \frac{1}{2t+1}\right)\big(\Gamma(t+1) - \Gamma(t)\big), \quad (5.26)$$

where for $t \geq 0$, $\Gamma(t) = t\gamma(t)$. In addition, if $\gamma(t) \propto t^{\alpha}$ for some $\alpha \in (-1, 0)$, then for any $t \geq 0$:

$$\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \leq 3L_f\left(1 + \frac{1}{2t+1}\right)(\alpha+1)\gamma(t) \leq 6L_f\gamma(t). \quad (5.27)$$

When $\alpha = 1/2$, the bound is equivalent to $\frac{3}{2}L_f\gamma(t)$ when t goes to infinity. This is quite similar to the $L_f\gamma(t)$ bound of other first order methods (gradient descent, mirror descent, *etc.*).

Lemma 13 provides a bound over the distance of two consecutive iterates. We now use this result to control term (5.22) in the regret decomposition.

Lemma 14 (Gap with noisy objectives)**.** Let $\tau \geq 0$. If $(\boldsymbol{\theta}(t))_{t\geq 0}$ is generated according to (5.19), then for any $t \geq 0$:

$$F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t+\tau)) - F(\boldsymbol{\theta}(t+\tau), \boldsymbol{\xi}(t+\tau)) \leq 3L^2 \left(1 + \frac{1}{2t+1}\right) \big(\Gamma(t+\tau) - \Gamma(t)\big). \tag{5.28}$$

Moreover, if $\gamma(t) \propto t^\alpha$ for some $\alpha \in (-1, 0)$, one has:

$$\begin{aligned} F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t+\tau)) - F(\boldsymbol{\theta}(t+\tau), \boldsymbol{\xi}(t+\tau)) &\leq 3L^2\tau \left(1 + \frac{1}{2t+1}\right)(1+\alpha)\gamma(t) \\ &\leq 6L^2\tau\gamma(t). \end{aligned} \tag{5.29}$$

Finally, we bound the term (5.23), corresponding to the optimization regret. This bound is a quite straightforward adpatation from the regular dual averaging algorithm bound in NESTEROV, 2009.

Lemma 15 (Optimization error)**.** For any $\boldsymbol{\theta}^* \in \mathcal{X}$, one has:

$$\sum_{t=\tau+1}^{T} F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t)) - F(\boldsymbol{\theta}^*, \boldsymbol{\xi}(t)) \leq \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)} + \frac{L^2}{2} \sum_{t=\tau+1}^{T} \gamma(t). \tag{5.30}$$

Proof. For any $\boldsymbol{\xi} \in \Delta_n$, $F(\cdot, \boldsymbol{\xi})$ is convex. Therefore, one has for any $\boldsymbol{\theta}^* \in \mathcal{X}$:

$$\sum_{t=\tau+1}^{T} F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t)) - F(\boldsymbol{\theta}^*, \boldsymbol{\xi}(t)) \leq \sum_{t=\tau+1}^{T} \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t))^\top (\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*). \tag{5.31}$$

One can then conclude using the definition of $(\boldsymbol{\theta}(t))_{t\geq 0}$ and the proof of dual averaging convergence in a standard setting – see NESTEROV, 2009, Theorem 1 for instance. □

From now on, we assume that there exists $\alpha \in (-1, 0)$ such that $\gamma(t) \propto \alpha$. This allows for easier convergence analysis but a more general analysis can still be performed using the bound provided in Lemma 14. We can now

apply previous results to the expected regret; for any $\tau \geq 0$, one has:

$$\sum_{t=1}^{t} \mathbb{E}[(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*))] \leq 2LD \sum_{t=1}^{T-\tau} \|P(t+\tau|t) - P^{\infty}\|_{TV} \tag{5.32}$$

$$+ 3L^2 \sum_{t=1}^{T-\tau} \left(1 + \frac{1}{2t+1}\right)(1+\alpha)\gamma(t) \tag{5.33}$$

$$+ \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)} + \frac{L^2}{2} \sum_{t=\tau+1}^{T} \gamma(t) \tag{5.34}$$

$$+ \sum_{t=T-\tau+1}^{T} \mathbb{E}[(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*))]. \tag{5.35}$$

We made the assumption $\boldsymbol{\theta} \leq D$, so $f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*) \leq 2LD$, and one has the following bound:

$$\sum_{t=1}^{t} \mathbb{E}[(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*))] \leq 2LD \sum_{t=1}^{T-\tau} \|P(t+\tau|t) - P^{\infty}\|_{TV} \tag{5.36}$$

$$+ 6L^2 \sum_{t=1}^{T-\tau} \gamma(t) + \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)} + \frac{L^2}{2} \sum_{t=\tau+1}^{T} \gamma(t) + \tau LD.$$

Let us assume that the mixing times are uniform, that is for any $\epsilon > 0$, there exists $\tau(\epsilon)$ such that:

$$\forall t \geq 0, \tau(t, \epsilon) \leq \tau(\epsilon). \tag{5.37}$$

Thus, deriving the bound (5.36), one has for any $\epsilon > 0$:

$$\sum_{t=1}^{t} \mathbb{E}[(f(\boldsymbol{\theta}(t)) - f(\boldsymbol{\theta}^*))] \leq 2LD(T\epsilon + \tau(\epsilon)) + \frac{L^2}{2}\big(1 + 12\tau(\epsilon)\big) \sum_{t=1}^{T} \gamma(t) + \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)}, \tag{5.38}$$

and we can write the following theorem.

Theorem 21 (Ergodic dual averaging)**.** *Let $(\boldsymbol{\theta}(t))_{t\geq 0}$ be generated according to (5.19) and let $\boldsymbol{\theta}^* \in \mathcal{X}$ be a minimizer of the optimization problem (5.16). We make the following assumptions:*

1. *There exists $\alpha \in (-1, 0)$ such that $\gamma(t) \propto t^{\alpha}$.*
2. *There exists $D > 0$ such that for any $\boldsymbol{\theta} \in \mathcal{X}$, $\|\boldsymbol{\theta}\| \leq D$.*
3. *For any $\epsilon > 0$, there exists $\tau(\epsilon)$ such that for any $t \geq 0$, $\tau(t, \epsilon) \leq \tau(\epsilon)$.*

Then, for any $\epsilon > 0$:

$$\mathbb{E}[(f(\bar{\boldsymbol{\theta}}(T))] - f(\boldsymbol{\theta}^*)] \leq 2LD\left(\epsilon + \frac{\tau(\epsilon)}{T}\right) + \frac{L^2}{2T}\big(1 + 12\tau(\epsilon)\big) \sum_{t=1}^{T} \gamma(t) + \frac{\|\boldsymbol{\theta}^*\|^2}{2T\gamma(T)}, \tag{5.39}$$

where $\bar{\boldsymbol{\theta}}(T) = (1/T)\sum_{t=1}^{T} \boldsymbol{\theta}(t)$ is the iterates average at time T.

Note that if one is able to compute $\nabla F(\cdot, \mathbf{1}_n/n)$ at every iteration, then $\tau(\epsilon) = 0$ for any $\epsilon > 0$ and one can recover the dual averaging convergence rate when $\epsilon \to 0$ in (5.39). This upper-bound evidences the need to compare

the mixing time to the optimization rate: if $\tau(\epsilon) \ll \sqrt{T}$ then similar bounds are preserved.

Algorithm 14 Distributed dual averaging algorithm in standard setting.

Require: Step size $(\gamma(t))_{t\geq 1} > 0$, weight matrix $\mathbf{W}$.
1: Each node i initializes $\boldsymbol{\theta}_i(0) = 0$, $\mathbf{z}_i(0) = 0$.
2: **for** $t = 0, \dots, T-1$ **do**
3: Update $\mathbf{Z}(t+1) = \mathbf{W}\mathbf{Z}(t) + \mathbf{G}(t)$, where $\mathbf{G}(t) = [\nabla f_1(\boldsymbol{\theta}_1(t)), \dots, \nabla f_n(\boldsymbol{\theta}_n(t))]$
4: **for** $i = 1, \dots, n$ **do**
5: Update $\boldsymbol{\theta}_i(t+1) = \pi_t(\mathbf{z}_i(t+1))$
6: **end for**
7: **end for**
8: **return** $(\bar{\boldsymbol{\theta}}_i(T))_{1\leq i\leq n}$

5.4 Decentralized Dual Averaging

We now focus on a decentralized setting, where each node $i \in [n]$ holds one observation $\mathbf{x}_i$ for simplicity. Section 5.6 provides more detailed analysis of the multiple observations per node setting.

The distributed dual averaging algorithm for solving (5.1) was first introduced by AGARWAL, WAINWRIGHT, and DUCHI, 2010 and consists in the following: each node $i \in [n]$ stores its own primal and dual sequences $(\boldsymbol{\theta}_i(t), \mathbf{z}_i(t))_{1\leq i\leq n}$. We denote as $\mathbf{Z}(t)$ the matrix of dual variables $\mathbf{Z}(t) = (\mathbf{z}_1(t), \dots, \mathbf{z}_n(t))^\top$. At iteration $t+1$, a node i will perform the following update:

$$\begin{cases} \mathbf{z}_i(t+1) &= \mathbf{g}_i(t) + \sum_{j=1}^n \mathbf{W}_{ij}\mathbf{z}_j(t) \\ \boldsymbol{\theta}_i(t+1) &= \pi_t(\mathbf{z}_i(t+1)), \end{cases} \tag{5.40}$$

where $\mathbf{W}$ is a doubly stochastic matrix such that

$$(i,j) \notin \mathcal{E} \Rightarrow \mathbf{W}_{ij} = 0. \tag{5.41}$$

Update (5.40) only differs in the dual update: gradients are now added to an average of neighbors dual variables. Let us point out that the dual update can be reformulated as follows:

$$\mathbf{Z}(t+1) = \mathbf{G}(t) + \mathbf{W}\mathbf{Z}(t), \tag{5.42}$$

where $\mathbf{G}(t) = (\mathbf{g}_1(t), \dots, \mathbf{g}_n(t))^\top$. This is detailed in Algorithm 14. In order to prove the convergence of this algorithm, we need to introduce two quantities. First, for $t > 0$, let us denote $\bar{\mathbf{z}}(t) := (1/n)\sum_{i=1}^n \mathbf{z}_i(t)$ the average of all dual variables at iteration t. One can see from (5.42) that $\bar{\mathbf{z}}$ is updated as follows:

$$\bar{\mathbf{z}}(t+1) = \bar{\mathbf{z}}(t) + \bar{\mathbf{g}}(t), \tag{5.43}$$

where $\bar{\mathbf{g}} := (1/n)\sum_{i=1}^n \mathbf{g}_i(t)$. This is very similar to the standard setting; however in this case, $\bar{\mathbf{g}}(t)$ is not necessarily a gradient of f since every $\mathbf{g}_i(t)$ can be a gradient taken at different points. Also, we define $\boldsymbol{\omega}(t) := \pi_t(\bar{\mathbf{z}}(t))$, the primal variable associated to $\bar{\mathbf{z}}(t)$.

With this introduced notation, one can establish the convergence of the distributed dual averaging method:

Theorem 22. *Let $(\gamma(t))_{t\geq 1}$ be a non increasing and non-negative sequence. For $i \in [n]$, let $(\mathbf{g}_i(t))_{t\geq 1}$, $(\mathbf{z}_i(t))_{t\geq 1}$ and $(\boldsymbol{\theta}_i(t))_{t\geq 1}$ be generated according to Algorithm 14. Assume that the function f is L-Lipschitz. Then for any $i \in [n]$ and $T \geq 2$, one has:*

$$\begin{aligned}\mathbb{E}[R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*) \leq \frac{1}{2T\gamma(T)}\|\boldsymbol{\theta}^*\|^2 + \frac{L_f^2}{2T}\sum_{t=2}^{T}\gamma(t-1) \\ + \frac{1}{T}\sum_{t=2}^{T}\gamma(t-1)\left(2L_f\|\bar{\mathbf{z}}(t) - \mathbf{z}_i(t)\| + \frac{\|\bar{\mathbf{z}}(t) - \mathbf{z}_i(t)\|^2}{2(t-1)}\right) \\ + \frac{L_f}{nT}\sum_{t=2}^{T}\gamma(t-1)\sum_{j=1}^{n}\Big(\|\mathbf{z}_i(t) - \mathbf{z}_j(t)\| + \|\bar{\mathbf{z}}(t) - \mathbf{z}_j(t)\|\Big),\end{aligned}$$

where $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}'\in\mathbb{R}^d} R_n(\boldsymbol{\theta}')$.

Proof. Let $T \geq 2$ and $i \in [n]$. Since f and ψ are convex, one has

$$R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*) \leq \frac{1}{T}\sum_{t=1}^{T}\Big(R_n(\boldsymbol{\theta}_i(t)) - R_n(\boldsymbol{\theta}^*)\Big).$$

Now remark that $g_i(t)$ is a gradient of f_i at $\boldsymbol{\theta}_i(t)$ but here we would need a gradient of f. However, by definition of f, one has for any $t \in [T]$:

$$f(\boldsymbol{\theta}_i(t)) - f(\boldsymbol{\theta}^*) = \frac{1}{n}\sum_{j=1}^{n} f_j(\boldsymbol{\theta}_i(t)) - f_j(\boldsymbol{\theta}^*).$$

Now, one can use subgradient inequality by inserting $\boldsymbol{\theta}_j(t)$ into the equation:

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[f_j(\boldsymbol{\theta}_i(t))] - f_j(\boldsymbol{\theta}^*) &= \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[f_j(\boldsymbol{\theta}_i(t)) - f_j(\boldsymbol{\theta}_j(t))] \\ &\quad + \mathbb{E}[f_j(\boldsymbol{\theta}_j(t))] - f_j(\boldsymbol{\theta}^*) \\ &\leq \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[f_j(\boldsymbol{\theta}_i(t)) - f_j(\boldsymbol{\theta}_j(t))] \\ &\quad + \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[\big(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}^*\big)^\top \mathbf{g}_j(t)].\end{aligned}$$

Using that both f and $\pi_{t-1}(\cdot)$ are Lipschitz, the first term in the right hand side above can be bounded as follows

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[f_j(\boldsymbol{\theta}_i(t))] - f_j(\boldsymbol{\theta}^*) \leq \frac{L\gamma(t-1)}{n}\sum_{j=1}^{n}\mathbb{E}\|\mathbf{z}_i(t)) - \mathbf{z}_j(t)\| \\ + \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[\big(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}^*\big)^\top \mathbf{g}_j(t)],\end{aligned}$$

which leads to:

$$\begin{aligned}\mathbb{E}[R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*) \leq & \frac{L_f}{nT}\sum_{t=1}^{T}\left(\gamma(t-1)\sum_{j=1}^{n}\mathbb{E}\|\mathbf{z}_i(t)) - \mathbf{z}_j(t)\|\right) \\ & + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\left(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}^*\right)^\top \mathbf{g}_j(t)\right]\right) \\ & + \frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}[\psi(\boldsymbol{\theta}_i(t))] - \psi(\boldsymbol{\theta}^*)\right).\end{aligned} \tag{5.44}$$

The term in (5.44) needs to be altered again in order to use the same reasoning than in the centralized setting. We write

$$\begin{aligned}&\frac{1}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}\left(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}^*\right)^\top \mathbf{g}_j(t) \\ &\qquad = \frac{1}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}\left(\boldsymbol{\theta}_j(t) - \boldsymbol{\omega}(t) + \boldsymbol{\omega}(t) - \boldsymbol{\theta}^*\right)^\top \mathbf{g}_j(t) \\ &\qquad = \frac{1}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}\left(\boldsymbol{\theta}_j(t) - \boldsymbol{\omega}(t)\right)^\top \mathbf{g}_j(t) + \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*\right)^\top \bar{\mathbf{g}}(t) \\ &\qquad \leq \frac{L_f}{nT}\sum_{t=1}^{T}\gamma(t-1)\sum_{j=1}^{n}\|\mathbf{z}_j(t) - \bar{\mathbf{z}}(t)\| + \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*\right)^\top \bar{\mathbf{g}}(t),\end{aligned} \tag{5.45}$$

since $\|\mathbf{g}_j(t)\| \leq L_f$ for any $t \geq 2$ and any $j \in [n]$. Now we can use Lemma 11 with $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, $\mathbf{g} = \bar{\mathbf{g}}$, $\boldsymbol{\theta}(t) = \boldsymbol{\omega}(t)$ to write:

$$\sum_{t=2}^{T}\left(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*\right)^\top \bar{\mathbf{g}}(t) + \sum_{t=2}^{T}\left(\psi(\boldsymbol{\omega}(t)) - \psi(\boldsymbol{\theta}^*)\right) \leq \frac{L_f^2}{2}\sum_{t=1}^{T-1}\gamma(t) + \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)},$$

which can be reformulated as follows:

$$\begin{aligned}\sum_{t=2}^{T}\left(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*\right)^\top \bar{\mathbf{g}}(t) + & \sum_{t=2}^{T}\left(\psi(\boldsymbol{\theta}_i(t)) - \psi(\boldsymbol{\theta}^*)\right) \\ & \leq \frac{L_f^2}{2}\sum_{t=1}^{T-1}\gamma(t) + \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)} + \sum_{t=2}^{T}\left(\psi(\boldsymbol{\theta}_i(t)) - \psi(\boldsymbol{\omega}(t))\right).\end{aligned}$$

Now, one only needs to provide an upper bound on the sum of differences $\sum_{t=2}^{T}\left(\psi(\boldsymbol{\theta}_i(t)) - \psi(\boldsymbol{\omega}(t))\right)$ to conclude. By definition, for $t \geq 2$:

$$\boldsymbol{\theta}_i(t) = \operatorname*{arg\,max}_{\boldsymbol{\theta}\in\mathbb{R}^d}\left\{\mathbf{z}_i^\top\boldsymbol{\theta} - \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t-1)} - (t-1)\psi(\boldsymbol{\theta})\right\},$$

so one has for any $\boldsymbol{\theta}' \in \mathbb{R}^d$:

$$\psi(\boldsymbol{\theta}_i(t)) \leq \frac{\mathbf{z}_i(t)^\top(\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}')}{t-1} + \frac{\|\boldsymbol{\theta}'\|^2 - \|\boldsymbol{\theta}_i(t)\|^2}{2(t-1)\gamma(t-1)} + \psi(\boldsymbol{\theta}').$$

Therefore, one has in particular for $\boldsymbol{\theta}' = \boldsymbol{\omega}(t)$

$$\begin{aligned}\psi(\boldsymbol{\theta}_i(t)) \leq & \frac{\mathbf{z}_i(t)^\top(\boldsymbol{\theta}_i(t) - \boldsymbol{\omega}(t))}{t-1} + \frac{\|\boldsymbol{\omega}(t)\|^2 - \|\boldsymbol{\theta}_i(t)\|^2}{2(t-1)\gamma(t-1)} + \psi(\boldsymbol{\omega}(t)) \\ \leq & \frac{\big(\boldsymbol{\theta}_i(t) - \gamma(t-1)\mathbf{z}_i(t)\big)^\top(\boldsymbol{\omega}(t) - \boldsymbol{\theta}_i(t))}{(t-1)\gamma(t-1)} + \frac{\|\boldsymbol{\omega}(t) - \boldsymbol{\theta}_i(t)\|^2}{2(t-1)\gamma(t-1)} \\ & + \psi(\boldsymbol{\omega}(t)).\end{aligned}$$

Using that π_t is $\gamma(t)$-Lipschitz, one can write:

$$\begin{aligned}\psi(\boldsymbol{\theta}_i(t)) \leq & \left(\frac{\|\mathbf{z}_i(t)\|}{t-1} + \frac{\|\boldsymbol{\theta}_i(t)\|}{(t-1)\gamma(t-1)}\right)\|\boldsymbol{\omega}(t) - \boldsymbol{\theta}_i(t)\| + \frac{\|\boldsymbol{\omega}(t) - \boldsymbol{\theta}_i(t)\|^2}{2(t-1)\gamma(t-1)} \\ & + \psi(\boldsymbol{\omega}(t)) \\ \leq & 2\gamma(t-1)\frac{\|\mathbf{z}_i(t)\|}{t-1}\|\bar{\mathbf{z}}(t) - \mathbf{z}_i(t)\| + \frac{\gamma(t-1)\|\bar{\mathbf{z}}(t) - \mathbf{z}_i(t)\|^2}{2(t-1)} \\ & + \psi(\boldsymbol{\omega}(t)).\end{aligned}$$

Using the fact that $\|g_j(t)\| \leq L_f$ for any $j \in [n]$ and any $t \geq 1$, one can easily see that $\|\mathbf{z}_i(t)\| \leq (t-1)L_f$. Finally, we obtain:

$$\begin{aligned}&\frac{1}{T}\sum_{t=2}^{T}\psi(\boldsymbol{\theta}_i(t)) - \psi(\boldsymbol{\omega}(t)) \\ &\qquad\qquad \leq \frac{1}{T}\sum_{t=2}^{T}\gamma(t-1)\left(2L_f\|\bar{\mathbf{z}}(t) - \mathbf{z}_i(t)\| + \frac{\|\bar{\mathbf{z}}(t) - \mathbf{z}_i(t)\|^2}{2(t-1)}\right),\end{aligned}$$

and the result holds. □

Algorithm 15 Gossip dual averaging for pairwise function in synchronous setting

Require: Step size $(\gamma(t))_{t\geq 1} > 0$.
Each node i initializes $\mathbf{y}_i = \mathbf{x}_i$, $\mathbf{z}_i = \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}}_i = 0$.
for $t = 1, \dots, T$ **do**
  Draw (i, j) uniformly at random from $\mathcal{E}$
  Set $\mathbf{z}_i, \mathbf{z}_j \leftarrow \frac{\mathbf{z}_i + \mathbf{z}_j}{2}$
  Swap auxiliary observations: $\mathbf{y}_i \leftrightarrow \mathbf{y}_j$
  for $k = 1, \dots, n$ **do**
    Update $\mathbf{z}_k \leftarrow \mathbf{z}_k + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k; \mathbf{x}_k, \mathbf{y}_k)$
    Compute $\boldsymbol{\theta}_k \leftarrow \pi_t(\mathbf{z}_k)$
    Average $\bar{\boldsymbol{\theta}}_k \leftarrow \left(1 - \frac{1}{t}\right) \bar{\boldsymbol{\theta}}_k + \frac{1}{t} \boldsymbol{\theta}_k$
  end for
end for
return Each node k has $\bar{\boldsymbol{\theta}}_k$

5.5 Pairwise Gossip Dual Averaging

We now turn to our main goal, namely to develop efficient gossip algorithms for solving Problem (5.3) in the decentralized setting. The methods we propose rely on dual averaging (see Section 5.3). This choice is guided by the fact that the structure of the updates makes dual averaging much easier to analyze in the distributed setting than sub-gradient descent when the problem is constrained or regularized. This is because dual averaging maintains a simple sum of sub-gradients, while the (non-linear) smoothing operator π_t is applied separately.

Our work builds upon the analysis of DUCHI, AGARWAL, and WAINWRIGHT (2012), who proposed a distributed dual averaging algorithm to optimize an average of *univariate* functions $f(\cdot; \mathbf{x}_i)$. In their algorithm, each node i computes *unbiased* estimates of its local function $\nabla f(\cdot; \mathbf{x}_i)$ that are iteratively averaged over the network – see Section 5.4 for details. Unfortunately, in our setting, the node i cannot compute unbiased estimates of $\nabla f_i(\cdot) = \nabla(1/n) \sum_{j=1}^n f(\cdot; \mathbf{x}_i, \mathbf{x}_j)$: the latter depends on all data points while each node $i \in [n]$ only holds $\mathbf{x}_i$. To go around this problem, we rely on a gossip data propagation step similar to the one introduced in Chapter 4 so that the nodes are able to compute *biased* estimates of $\nabla f_i(\cdot)$ while keeping the communication and memory overhead to a small level for each node.

We present and analyze our algorithm in the synchronous setting in Section 5.5.1. We then turn to the more intricate analysis of the asynchronous setting in Section 5.5.2.

5.5.1 Synchronous Setting

In the synchronous setting, we assume that each node has access to a global clock such that every node can update simultaneously at each tick of the clock. Although not very realistic, this setting allows for simpler analysis. We assume that the scaling sequence $(\gamma(t))_{t\geq 0}$ is the same for every node. At any time, each node i has the following quantities in its local

memory register: a variable $\mathbf{z}_i$ (the gradient accumulator), its original observation $\mathbf{x}_i$, and an *auxiliary observation* $\mathbf{y}_i$, which is initialized at $\mathbf{x}_i$ but will change throughout the algorithm as a result of data propagation.

The algorithm goes as follows. At each iteration, an edge $(i, j) \in \mathcal{E}$ of the graph is drawn uniformly at random. Then, nodes i and j average their gradient accumulators $\mathbf{z}_i$ and $\mathbf{z}_j$, and swap their auxiliary observations $\mathbf{y}_i$ and $\mathbf{y}_j$. Finally, every node of the network performs a dual averaging step, using their original observation and their current auxiliary one to estimate the partial gradient. The procedure is detailed in Algorithm 15, and the following proposition adapts the convergence rate of centralized dual averaging under the hypothesis that the contribution of the bias term decreases fast enough over the iterations.

Proposition 5. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non-bipartite graph, and let $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} R_n(\boldsymbol{\theta})$. Let $(\gamma(t))_{t \geq 1}$ be a non-increasing and non-negative sequence. For any $i \in [n]$ and any $t \geq 0$, let $\mathbf{z}_i(t) \in \mathbb{R}^d$ and $\bar{\boldsymbol{\theta}}_i(t) \in \mathbb{R}^d$ be generated according to Algorithm 15. Then for any $i \in [n]$ and $T > 1$, we have:*

$$\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i) - R_n(\boldsymbol{\theta}^*)] \leq C_1(T) + C_2(T) + C_3(T),$$

where

$$\begin{cases} C_1(T) = \dfrac{1}{2T\gamma(T)} \|\boldsymbol{\theta}^*\|^2 + \dfrac{L_f^2}{2T} \displaystyle\sum_{t=1}^{T-1} \gamma(t), \\ C_2(T) = \dfrac{3L_f^2}{T\left(1 - \sqrt{\lambda_2}\right)} \displaystyle\sum_{t=1}^{T-1} \gamma(t), \\ C_3(T) = \dfrac{1}{T} \displaystyle\sum_{t=1}^{T-1} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)], \end{cases}$$

and $1 - \lambda_2 = \beta_{n-1}/|\mathcal{E}| > 0$ and β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$.

Sketch of proof. First notice that at a given (outer) iteration $t+1$, $\bar{\mathbf{z}}$ is updated as follows:

$$\bar{\mathbf{z}}(t+1) = \bar{\mathbf{z}}(t) + \frac{1}{n} \sum_{k=1}^{n} \mathbf{d}_k(t), \tag{5.46}$$

where for $k \in [n]$, $\mathbf{d}_k(t) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k(t); \mathbf{x}_k, \mathbf{y}_k(t+1))$ is a biased estimate of $\nabla f_k(\boldsymbol{\theta}_k(t))$. Let $\boldsymbol{\epsilon}_k(t) = \mathbf{d}_k(t) - \mathbf{g}_k(t)$ be the bias, so $\mathbf{g}_k(t)$ is an unbiased gradient estimate: $\mathbb{E}[\mathbf{g}_k(t) | \boldsymbol{\theta}_k(t)] = \nabla f_k(\boldsymbol{\theta}_k(t))$.

Let us define $\boldsymbol{\omega}(t) = \pi_t(\bar{\mathbf{z}}(t))$. Using convexity of R_n, the gradient's definition and the fact that the functions f and π_t are both L_f-Lipschitz, we

obtain: for $T \geq 2$ and $i \in [n]$,

$$\begin{aligned}
&\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*)] \\
&\leq \frac{L_f}{nT} \sum_{t=2}^{T} \gamma(t-1) \sum_{j=1}^{n} \mathbb{E}_t\Big[\|\mathbf{z}_i(t) - \mathbf{z}_j(t)\|\Big] && (5.47)\\
&+ \frac{L_f}{nT} \sum_{t=2}^{T} \gamma(t-1) \sum_{j=1}^{n} \mathbb{E}_t\Big[\|\bar{\mathbf{z}}(t) - \mathbf{z}_j(t)\|\Big] && (5.48)\\
&+ \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{g}}(t)]. && (5.49)
\end{aligned}$$

Using Lemma 16 (see Section 5.9), the terms (5.47)-(5.48) can be bounded by $C_2(T)$. The term (5.49) requires a specific analysis because the updates are performed using biased estimates. We decompose it as follows:

$$\begin{aligned}
\frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t\Big[\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{g}}(t)\Big] &= \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t\Big[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top (\bar{\mathbf{d}}(t) - \bar{\boldsymbol{\epsilon}}(t))\Big] \\
&\leq \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t\Big[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{d}}(t)\Big] && (5.50)\\
&\quad + \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}_t\Big[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)\Big].
\end{aligned}$$

The term (5.50) can be bounded by $C_1(T)$ (see XIAO, 2010, Lemma 9). We refer the reader to the Section 5.9 for the detailed proof. □

The rate of convergence in Proposition 5 is divided into three parts: $C_1(T)$ is a *data dependent* term which corresponds to the rate of convergence of the centralized dual averaging, while $C_2(T)$ and $C_3(T)$ are *network dependent* terms since $1 - \lambda_2 = \beta_{n-1}/|\mathcal{E}|$, where β_{n-1} is the second smallest eigenvalue of the graph Laplacian $\mathbf{L}$, also known as the spectral gap of $\mathcal{G}$. The convergence rate of our algorithm thus improves when the spectral gap is large, which is typically the case for well-connected graphs (CHUNG, 1997). Note that $C_2(T)$ corresponds to the network dependence for the distributed dual averaging algorithm of DUCHI, AGARWAL, and WAINWRIGHT (2012) while the term $C_3(T)$ comes from the bias of our partial gradient estimates. In practice, $C_3(T)$ vanishes quickly and has a small impact on the rate of convergence, as shown in Section 5.7.

Theorem 23 (Pairwise ergodic dual averaging). *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non-bipartite graph, and let $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} R_n(\boldsymbol{\theta})$. Let $(\gamma(t))_{t \geq 1}$ be a non-increasing and non-negative sequence. For any $i \in [n]$ and any $t \geq 0$, let $\mathbf{z}_i(t) \in \mathbb{R}^d$ and $\bar{\boldsymbol{\theta}}_i(t) \in \mathbb{R}^d$ be generated according to Algorithm 15. We make the following assumptions:*

1. *There exists $\alpha \in (-1, 0)$ such that $\gamma(t) \propto t^\alpha$.*
2. *There exists D such that for all $\boldsymbol{\theta} \in \mathcal{X}$, $\|\theta\| \leq D$.*

Then, for any $\epsilon > 0$:

$$\begin{aligned}\mathbb{E}[(R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*)] \leq & \frac{L^2}{2T}\big(1 + 12\tau(\epsilon)\big)\sum_{t=1}^{T}\gamma(t) + \frac{\|\boldsymbol{\theta}^*\|^2}{2T\gamma(T)} \\ & + 2LD\left(\epsilon + \frac{\tau(\epsilon)}{T}\right) \\ & + \frac{3L_f^2}{T\left(1-\sqrt{\lambda_2}\right)}\sum_{t=1}^{T-1}\gamma(t),\end{aligned}$$

where for any $\epsilon > 0$,

$$\tau(\epsilon) = \max\left(\frac{\log(\epsilon) - \log(c(\mathcal{G}))}{\log(c'(\mathcal{G}))}, 1\right),$$

with $c(\mathcal{G}) > 1$ and $c'(\mathcal{G}) \in (0,1)$ only depending on the graph connectivity.

Algorithm 16 Gossip dual averaging for pairwise function in asynchronous setting

Require: Step size $(\gamma(t))_{t\geq 0} > 0$, probabilities $(p_k)_{k\in[n]}$.
Each node i initializes $\mathbf{y}_i = \mathbf{x}_i$, $\mathbf{z}_i = \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}}_i = 0$, $m_i = 0$.
for $t = 1, \dots, T$ **do**
  Draw (i, j) uniformly at random from E
  Swap auxiliary observations: $y_i \leftrightarrow y_j$
  for $k \in \{i, j\}$ **do**
    Set $\mathbf{z}_k \leftarrow \frac{\mathbf{z}_i + \mathbf{z}_j}{2}$
    Update $\mathbf{z}_k \leftarrow \frac{1}{p_k} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_k; \mathbf{x}_k, \mathbf{y}_k)$
    Increment $m_k \leftarrow m_k + \frac{1}{p_k}$
    Compute $\boldsymbol{\theta}_k \leftarrow \pi_{m_k}(\mathbf{z}_k)$
    Average $\bar{\boldsymbol{\theta}}_k \leftarrow \left(1 - \frac{1}{m_k p_k}\right) \bar{\boldsymbol{\theta}}_k$
  end for
end for
return Each node k has $\bar{\boldsymbol{\theta}}_k$

5.5.2 Asynchronous Setting

For any variant of gradient descent over a network with a decreasing step size, there is a need for a common time scale to perform the suitable decrease. In the synchronous setting, this time scale information can be shared easily among nodes by assuming the availability of a global clock. This is convenient for theoretical considerations, but is unrealistic in practical (asynchronous) scenarios. In this section, we place ourselves in a fully asynchronous setting where each node has a local clock, ticking at a Poisson rate of 1, independently from the others. This is equivalent to a global clock ticking at a rate n Poisson process which wakes up an edge of the network uniformly at random (see BOYD et al., 2006, for details on clock modeling).

With this in mind, Algorithm 15 needs to be adapted to this setting. First, one cannot perform a full dual averaging update over the network since only two nodes wake up at each iteration. Also, as mentioned earlier, each node needs to maintain an estimate of the current iteration number in order for the scaling factor γ to be consistent across the network. For $k \in [n]$, let p_k denote the probability for the node k to be picked at any iteration. If the edges are picked uniformly at random, then one has $p_k = 2d_k/|\mathcal{E}|$. For simplicity, we focus only on this case, although our analysis holds in a more general setting.

Let us define an activation variable $(\delta_k(t))_{t\geq 1}$ such that for any $t \geq 1$,

$$\delta_k(t) = \begin{cases} 1 & \text{if node } k \text{ is picked at iteration } t, \\ 0 & \text{otherwise.} \end{cases}$$

One can immediately see that $(\delta_k(t))_{t\geq 1}$ are i.i.d. random variables, Bernoulli distributed with parameter p_k. Let us define $(m_k(t)) \geq 0$ such that $m_k(0) = 0$ and for $t \geq 0$, $m_k(t+1) = m_k(t) + \frac{\delta_k(t+1)}{p_k}$. Since $(\delta_k(t))_{t\geq 1}$ are Bernoulli random variables, $m_k(t)$ is an unbiased estimate of the time t.

Using this estimator, we can now adapt Algorithm 15 to the fully asynchronous case, as shown in Algorithm 16. The update step slightly differs from the synchronous case: the partial gradient has a weight $1/p_k$ instead of 1 so that all partial functions asymptotically count in equal way in every

gradient accumulator. In contrast, uniform weights would penalize partial gradients from low degree nodes since the probability of being drawn is proportional to the degree. This weighting scheme is essential to ensure the convergence to the global solution. The model averaging step also needs to be altered: in absence of any global clock, the weight $1/t$ cannot be used and is replaced by $1/(m_k p_k)$, where $m_k p_k$ corresponds to the average number of times that node k has been selected so far.

The following result is the analogous of Proposition 5 for the asynchronous setting.

Theorem 24. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph. Let $(\gamma(t))_{t\geq 1}$ be defined as $\gamma(t) = c/t^{1/2+\alpha}$ for some constant $c > 0$ and $\alpha \in (0, 1/2)$. For $i \in [n]$, let $(\mathbf{d}_i(t))_{t\geq 1}$, $(\mathbf{g}_i(t))_{t\geq 1}$, $(\boldsymbol{\epsilon}_i(t))_{t\geq 1}$, $(\mathbf{z}_i(t))_{t\geq 1}$ and $(\boldsymbol{\theta}_i(t))_{t\geq 1}$ be generated as described in Algorithm 16. Then, there exists some constant $C < +\infty$ such that, for $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}' \in \mathbb{R}^d} R_n(\boldsymbol{\theta}')$, $i \in [n]$ and $T > 0$,*

$$R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*) \leq C \max(T^{-\alpha/2}, T^{\alpha-1/2}) + \frac{1}{T}\sum_{t=2}^{T} \mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\boldsymbol{\epsilon}}(t)].$$

The proof is given in the supplementary material.

In the asynchronous setting, no convergence rate was known even for the distributed dual averaging algorithm of DUCHI, AGARWAL, and WAINWRIGHT (2012), which deals with the simpler problem of minimizing *univariate* functions. The arguments used to derive Theorem 24 can be adapted to derive a convergence rate (without the bias term) for an asynchronous version of their algorithm.

Remark 7. We have focused on the setting where all pairs of observations are involved in the objective. In practice, the objective may depend only on a subset of all pairs. To efficiently apply our algorithm to this case, one should take advantage of the potential structure of the subset of interest: for instance, one could attach some additional concise information to each observation so that a node can easily identify whether a pair contributes to the objective, and if not set the loss to be zero. This is essentially the case in the AUC optimization problem studied in Section 5.7, where pairs of similarly labeled observations do not contribute to the objective. If the subset of pairs cannot be expressed in such a compact form, then one would need to provide each node with an index list of active pairs, which could be memory-intensive when n is large.

5.6 Extension to Multiple Points per Node

For ease of presentation, we have assumed throughout the paper that each node i holds a single data point $\mathbf{x}_i$. In this section, we discuss simple extensions of our results to the case where each node holds the same number of points $k \geq 2$. First, it is easy to see that our results still hold if nodes swap their entire set of k points (essentially viewing the set of k points as a single one). However, depending on the network bandwidth, this solution may be undesirable.

We thus propose another strategy where only two data points are exchanged at each iteration, as in the algorithms proposed in the main text. The idea is to view each "physical" node $i \in [n]$ as a set of k "virtual" nodes, each holding a single observation. These k nodes are all connected to each other as well as to the neighbors of i in the initial graph $\mathcal{G}$ and their virtual nodes. Formally, this new graph $\mathcal{G}^{\otimes} = ([n]^{\otimes}, \mathcal{E}^{\otimes})$ is given by $\mathcal{G} \times \mathbb{K}_k$, the tensor product between G and the k-node complete graph $\mathbb{K}_k$. It is easy to see that $|\mathcal{V}^{\otimes}| = kn$ and $|\mathcal{E}^{\otimes}| = k^2|\mathcal{E}|$. We can then run our algorithms on $\mathcal{G}^{\otimes}$ (each physical node $i \in [n]$ simulating the behavior of its corresponding k virtual nodes) and the convergence results hold, replacing $1 - \lambda_2^{\mathcal{G}}$ by $1 - \lambda_2^{\mathcal{G}^{\otimes}}$ in the bounds. The following result gives the relationship between these two quantities.

Proposition 6. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected, non-bipartite and non-complete graph. Let $k \geq 2$ and let $\mathcal{G}^{\otimes}$ be the tensor product graph of $\mathcal{G}$ and $\mathbb{K}_k$. Let $1 - \lambda_2^{\mathcal{G}} = \beta_{n-1}^{\mathcal{G}}/|\mathcal{E}|$ and $1 - \lambda_2^{\mathcal{G}^{\otimes}} = \beta_{kn-1}^{\mathcal{G}^{\otimes}}/|\mathcal{E}^{\otimes}|$, where β_{n-1} and $\beta_{kn-1}^{\mathcal{G}^{\otimes}}$ are the second smallest eigenvalues of $\mathbf{L}^{\mathcal{G}}$ and $\mathbf{L}^{\mathcal{G}^{\otimes}}$ respectively. We have that*

$$1 - \lambda_2^{\mathcal{G}^{\otimes}} = \frac{1 - \lambda_2^{\mathcal{G}}}{k}.$$

Proof. Denoting the Kronecker product by $\otimes$, we can write:

$$\begin{aligned} \mathbf{A}^{\mathcal{G}^{\otimes}} &= \mathbf{1}_k \mathbf{1}_k^T \otimes \mathbf{A}^{\mathcal{G}}, \\ \mathbf{D}^{\mathcal{G}^{\otimes}} &= k\mathbf{I}_k \otimes \mathbf{D}^{\mathcal{G}}. \end{aligned}$$

Recall that $\mathbf{L}^{\mathcal{G}} = \mathbf{D}^{\mathcal{G}} - \mathbf{A}^{\mathcal{G}}$ and $\mathbf{L}^{\mathcal{G}^{\otimes}} = \mathbf{D}^{\mathcal{G}^{\otimes}} - \mathbf{A}^{\mathcal{G}^{\otimes}}$. Let $(\boldsymbol{\phi}, \beta) \in \mathbb{R}^{nk} \times \mathbb{R}$ be an eigenpair of $\mathbf{L}^{\mathcal{G}^{\otimes}}$, *i.e.*, $(\mathbf{D}^{\mathcal{G}^{\otimes}} - \mathbf{A}^{\mathcal{G}^{\otimes}})\boldsymbol{\phi} = \beta\boldsymbol{\phi}$ and $\boldsymbol{\phi} \neq 0$. Let us write $\boldsymbol{\phi} = [\boldsymbol{\phi}_1^\top \ldots \boldsymbol{\phi}_k^\top]^\top$ where $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_k \in \mathbb{R}^n$. Exploiting the structure of $\mathbf{A}^{\mathcal{G}^{\otimes}}$ and $\mathbf{D}^{\mathcal{G}^{\otimes}}$, we have:

$$k\mathbf{D}^{\mathcal{G}}\boldsymbol{\phi}_i - \sum_{j=1}^{k} \mathbf{A}^{\mathcal{G}}\boldsymbol{\phi}_j = \beta\boldsymbol{\phi}_i, \quad \text{for all } i \in \{1, \ldots, k\}. \tag{5.51}$$

Summing up (5.51) over all $i \in \{1, \ldots, k\}$ gives

$$\mathbf{D}^{\mathcal{G}} \sum_{i=1}^{k} \boldsymbol{\phi}_i - \mathbf{A}^{\mathcal{G}} \sum_{i=1}^{k} \boldsymbol{\phi}_i = \frac{\beta}{k} \sum_{i=1}^{k} \boldsymbol{\phi}_i,$$

which shows that if $(\boldsymbol{\phi}, \beta)$ is an eigenpair of $\mathbf{L}^{\mathcal{G}^{\otimes}}$ with $\sum_{i=1}^{k} \boldsymbol{\phi}_i \neq 0$, then $(\sum_{i=1}^{k} \boldsymbol{\phi}_i, \beta/k)$ is an eigenpair of $\mathbf{L}^{\mathcal{G}}$. In the case where $\sum_{i=1}^{k} \boldsymbol{\phi}_i = 0$, then there exists an index $j \in \{1, \ldots, k\}$ such that $\boldsymbol{\phi}_j = -\sum_{i \neq j} \boldsymbol{\phi}_j \neq 0$. Hence

(5.51) gives

$$\mathbf{D}^{\mathcal{G}}\boldsymbol{\phi}_j = \frac{\beta}{k}\boldsymbol{\phi}_j,$$

which shows that $(\boldsymbol{\phi}_j, \beta/k)$ is an eigenpair of $\mathbf{L}^{\mathcal{G}}$. Observe that $\beta = kd_i$ for some $i \in \{1, \dots, n\}$.

We have thus shown that any eigenvalue $\beta^{\mathcal{G}^\otimes}$ of $\mathbf{L}^{\mathcal{G}^\otimes}$ is either of the form $\beta^{\mathcal{G}^\otimes} = k\beta^{\mathcal{G}}$, where $\beta^{\mathcal{G}}$ is an eigenvalue of $\mathbf{L}^{\mathcal{G}}$, or of the form $\beta^{\mathcal{G}^\otimes} = kd_i$ for some $i \in \{1, \dots, n\}$.

Since $\mathbf{L}^{\mathcal{G}^\otimes}$ is a Laplacian matrix, its smallest eigenvalue is 0. Let $\beta^{\mathcal{G}^\otimes}_{nk-1}$ be the second smallest eigenvalue of $\mathbf{L}^{\mathcal{G}^\otimes}$. Note that $\mathcal{G}^\otimes$ is not a complete graph since $\mathcal{G}$ is not complete. Therefore, $\beta^{\mathcal{G}^\otimes}_{nk-1}$ is bounded above by the vertex connectivity of $\mathcal{G}^\otimes$ (FIEDLER, 1973), which is itself trivially bounded above by the minimum degree $d^\otimes_{min} = \min_{i\in[kn]}[\mathbf{D}^{\mathcal{G}^\otimes}]_{ii}$ of $\mathcal{G}^\otimes$. This implies that $\beta^{\mathcal{G}^\otimes}_{nk-1} = k\beta^{\mathcal{G}}_{n-1}$, and hence

$$1 - \lambda_2^{\mathcal{G}^\otimes} = \frac{\beta^{\mathcal{G}^\otimes}_{kn-1}}{|\mathcal{E}^\otimes|} = \frac{k\beta^{\mathcal{G}}_{n-1}}{k^2|E|} = \frac{1-\lambda_2^{\mathcal{G}}}{k}.$$

□

Proposition 6 shows that the network-dependent term in our convergence bounds is only affected by a factor k. Furthermore, note that iterations involving two virtual nodes corresponding to the same physical node will not require actual network communication, which somewhat attenuates this effect in practice.

Dataset	Complete graph	Watts-Strogatz	Cycle graph
Breast Cancer ($n = 699$)	$1.43 \cdot 10^{-3}$	$8.71 \cdot 10^{-5}$	$5.78 \cdot 10^{-8}$
Synthetic ($n = 1000$)	$1.00 \cdot 10^{-3}$	$6.23 \cdot 10^{-5}$	$1.97 \cdot 10^{-8}$

TABLE 5.1: Spectral gap values $1 - \lambda_2^{\mathcal{G}}$ for each network.

5.7 Numerical Simulations

In this section, we present numerical experiments on two popular machine learning problems involving pairwise functions: Area Under the ROC Curve (AUC) maximization and metric learning. Our results show that our algorithms converge and that the bias term vanishes very quickly with the number of iterations.

To study the influence of the network topology, we perform our simulations on three types of network (see Table 5.1 for the corresponding spectral gap values):

— *Complete graph:* All nodes are connected to each other. It is the ideal situation in our framework, since any pair of nodes can communicate directly. In this setting, the bias of gradient estimates should be very small, as one has for any $k \in [n]$ and any $t \geq 1$, $\mathbb{E}_t[d_k(t)|\theta_k(t)] = 1/(n-1)\sum_{y' \neq y_k(t)} \nabla_\theta f(\theta_k(t); \mathbf{x}_k, y')$. For a network size n, the complete graph achieves the highest spectral gap: $1 - \lambda_2^{\mathcal{G}} = 1/n$, see BOLLOBÁS (1998, Ch.9) or CHUNG (1997, Ch.1) for details.

— *Cycle graph:* This is the worst case in terms of connectivity: each node only has two neighbors. This network has a spectral gap of order $1/n^3$, and gives a lower bound in terms of convergence rate.

— *Watts-Strogatz:* This random network generation technique (WATTS and STROGATZ, 1998) relies on two parameters: the average degree of the network k and a rewiring probability p. In expectation, the higher the rewiring probability, the better the connectivity of the network. Here, we use $k = 5$ and $p = 0.3$ to achieve a compromise between the connectivities of the complete graph and the cycle graph.

AUC Maximization We first present an application of our algorithms to AUC maximization on a real dataset. Given a set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ with associated binary labels $\ell_1, \ldots, \ell_n \in \{-1, 1\}$, the goal is to learn a linear scoring rule $\mathbf{x} \mapsto \mathbf{x}^\top \boldsymbol{\theta}$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$ which maximizes:

$$\mathrm{AUC}(\boldsymbol{\theta}) = \frac{\sum_{1 \leq i,j \leq n} \mathbb{1}_{\{\ell_i > \ell_j\}} \mathbb{1}_{\{\mathbf{x}_i^\top \boldsymbol{\theta} > \mathbf{x}_j^\top \boldsymbol{\theta}\}}}{\sum_{1 \leq i,j \leq n} \mathbb{1}_{\{\ell_i > \ell_j\}}}.$$

It corresponds to the probability that the scoring rule associated with $\boldsymbol{\theta}$ outputs a higher score on a positively labeled sample than on a negatively labeled one. This formulation leads to a non-smooth optimization problem; therefore, one typically minimizes a convex surrogate such as the logistic loss:

$$R_n(\boldsymbol{\theta}) = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} \mathbb{1}_{\{\ell_i > \ell_j\}} \log\left(1 + \exp((\mathbf{x}_j - \mathbf{x}_i)^\top \boldsymbol{\theta})\right).$$

We do not apply any regularization (*i.e.*, $\psi \equiv 0$), and use the Breast Cancer Wisconsin dataset, [1] which consists of $n = 699$ points in $d = 11$ dimensions.

We initialize each $\boldsymbol{\theta}_i$ to 0 and for each network, we run 50 times Algorithms 15 and 16 with $\gamma(t) = 1/\sqrt{t}$. [2] Figure 5.1a shows the evolution of the objective function and the associated standard deviation (across nodes) with the number of iterations in the synchronous setting. As expected, the average convergence rate on the complete and the Watts-Strogatz networks is much better than on the poorly connected cycle network. The standard deviation of the node estimates also decreases with the connectivity of the network.

The results for the asynchronous setting are shown in Figure 5.1b. As expected, the convergence rate is slower in terms of number of iterations (roughly 5 times) than in the synchronous setting. Note however that much fewer dual averaging steps are performed in this case: for instance, on the Watts-Strogatz network, reaching a 0.1 loss requires $210,000$ (partial) gradient computations in the synchronous setting and only $25,000$ in the asynchronous setting. Moreover, the standard deviation of the estimates is much lower than in the synchronous setting. This is because communication and local optimization are better balanced in the asynchronous setting (one optimization step for each gradient accumulator averaged) than in the synchronous setting (n optimization steps for 2 gradient accumulators averaged).

The good practical convergence of our algorithm comes from the fact that the bias term $\bar{\boldsymbol{\epsilon}}(t)^\top \boldsymbol{\omega}(t)$ vanishes quite fast. Figure 5.1c shows that its average value quickly converges to 0 on all networks. Moreover, its order of magnitude is negligible compared to the objective function. In order to fully estimate the impact of this bias term on the performance, we also compare our algorithm to the ideal but unrealistic situation where each node is given an unbiased estimate of its partial gradient: instead of adding $\nabla f(\boldsymbol{\theta}_i(t); \mathbf{x}_i, \mathbf{y}_i(t))$ to $\mathbf{z}_i(t)$, a node i will add $\nabla f(\boldsymbol{\theta}_i(t); \mathbf{x}_i, \mathbf{x}_j)$ where $j \in [n]$ is picked uniformly at random. As shown in Figure 5.2, the performance of both methods are very similar on well-connected networks.

Metric Learning We now turn to a metric learning application. We consider the family of Mahalanobis distances $D_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \boldsymbol{\theta}(\mathbf{x}_i - \mathbf{x}_j)$ parameterized by $\boldsymbol{\theta} \in \mathbb{S}_+^d$, where $\mathbb{S}_+^d$ is the cone of $d \times d$ positive semi-definite real-valued matrices. Given a set of data points $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ with associated labels $\ell_1, \dots, \ell_n \in \{-1, 1\}$, the goal is to find $\boldsymbol{\theta} \in \mathbb{S}_+^d$ which minimizes the following criterion (JIN, WANG, and ZHOU, 2009):

$$R_n(\boldsymbol{\theta}) = \frac{1}{n^2} \sum_{1 \le i,j \le n} \left[\ell_i \ell_j (b - D_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j))\right]_+ + \psi(\boldsymbol{\theta}),$$

where $[u]_+ = \max(0, 1 - u)$, $b > 0$, and $\psi(\boldsymbol{\theta}) = \infty$ if $\boldsymbol{\theta} \notin \mathbb{S}_+^d$ and 0 otherwise. We use a synthetic dataset of $n = 1,000$ points generated as follows: each point is drawn from a mixture of 10 Gaussians in $\mathbb{R}^{40}$ (each corresponding to a class) with all Gaussian means contained in a 5d subspace and their

1. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

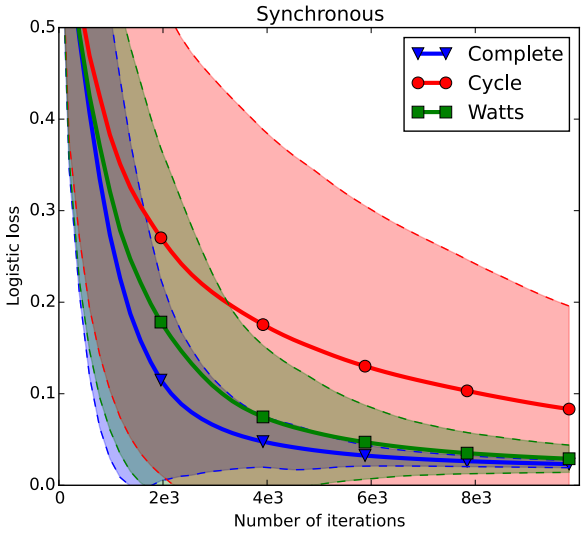
2. Even if this scaling sequence does not fulfill the hypothesis of Theorem 24 for the asynchronous setting, the convergence rate is acceptable in practice.

shared covariance matrix proportional to the identity with a variance factor such that some overlap is observed.

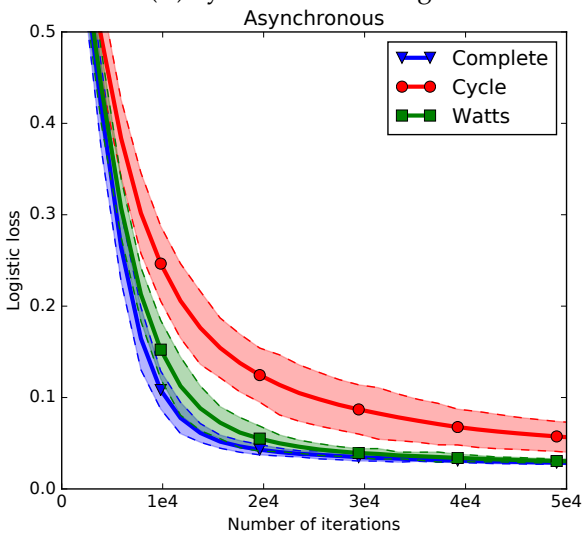
Figure 5.3a shows the evolution of the objective function and its standard deviation for the asynchronous setting. As in the case of AUC maximization, the algorithm converges much faster on the well-connected networks than on the cycle network. Again, we can see in Figure 5.3b that the bias vanishes very quickly with the number of iterations.

We also compare the logistic loss associated to our algorithm's iterates to the loss associated to the following baseline: instead of adding the biased estimate $\nabla f(\boldsymbol{\theta}_i(t); \mathbf{x}_i, \mathbf{y}_i(t))$ to its dual variable $\mathbf{z}_i(t)$, a node $i \in [n]$ receives a vector drawn uniformly at random from the set of gradients $\{\nabla f(\boldsymbol{\theta}_i(t); \mathbf{x}_i, \mathbf{x}_1), \ldots, \nabla f(\boldsymbol{\theta}_i(t); \mathbf{x}_i, \mathbf{x}_n)\}$. The bias introduced by the random walk procedure is already shown to be very small in comparison to the objective function on Figure 5.3b. Here, Figure 5.4 evidences the fact that this small bias has close to no influence on the optimization process for well-connected networks.

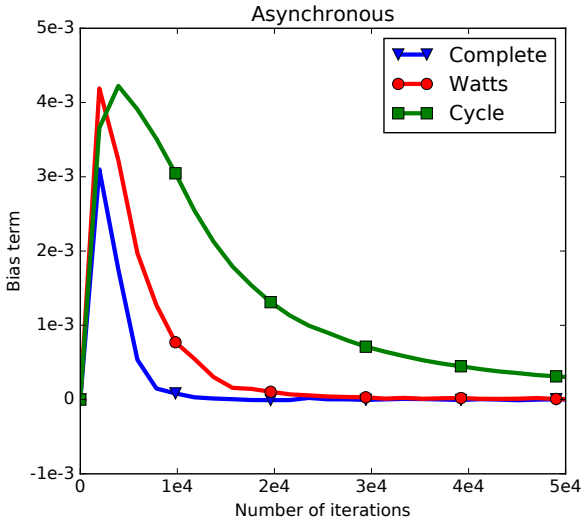
Finally, we focus on decentralized metric learning on the Breast Cancer Wisconsin Dataset. Figure 5.5a shows the evolution of the metric learning criterion with the number of iterations, averaged over 50 runs. As in previous experiments, there is almost no difference between the convergence rate of the Watts-Strogatz network and the complete network. Moreover, the bias term is again largely negligible when compared to the metric learning criterion, as shown on Figure 5.5b.



(A) Synchronous setting.

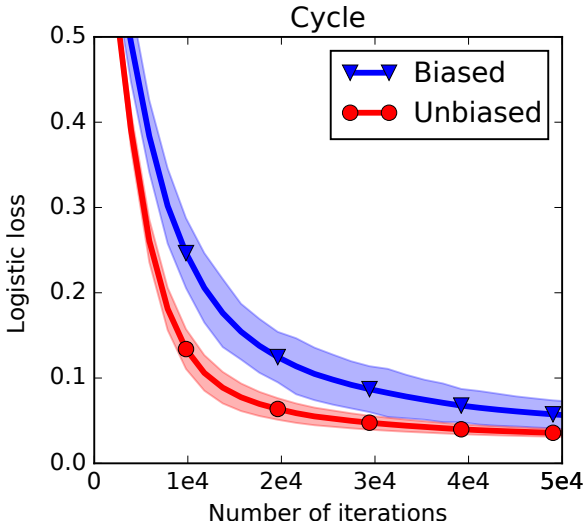


(B) Asynchronous setting.

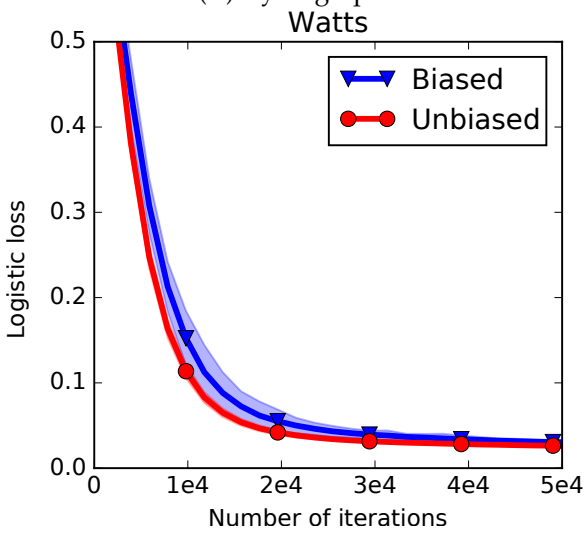


(C) Bias term (asynchronous setting).

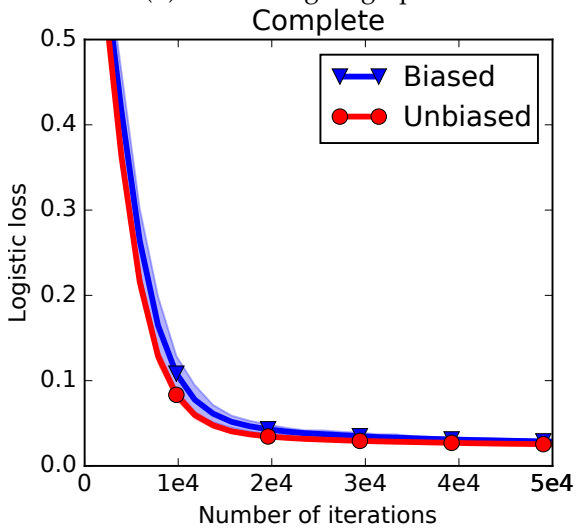
FIGURE 5.1: AUC maximization. Solid lines are averages and filled area are standard deviations.



(A) Cycle graph.

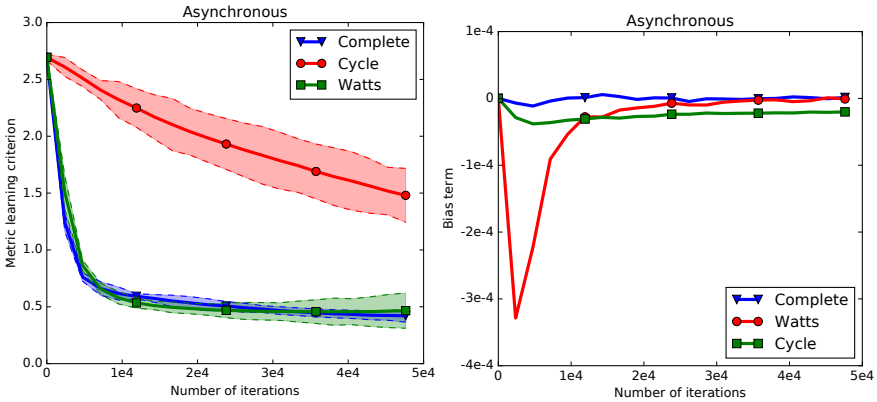


(B) Watts Strogatz graph.



(C) Complete graph.

FIGURE 5.2: AUC maximization: comparison between our algorithm and an unbiased version.



(A) Evolution of the objective function and its standard deviation (asynchronous setting)

(B) Evolution of the bias term

FIGURE 5.3: Metric learning experiments.

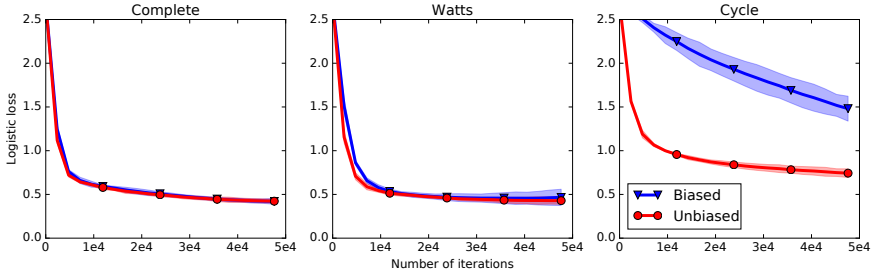
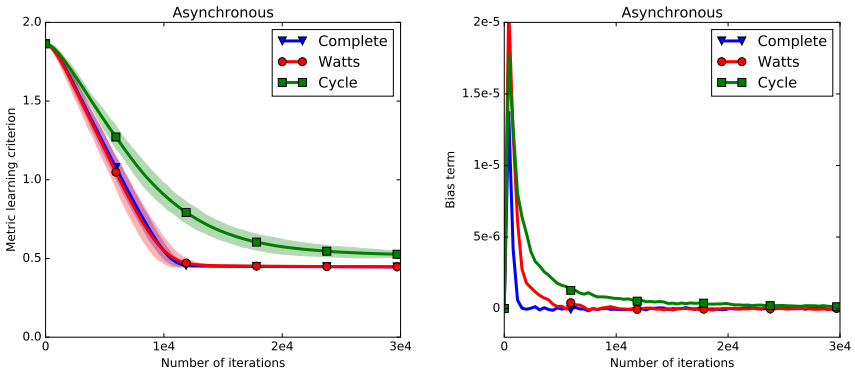


FIGURE 5.4: Metric learning: comparison between our algorithm and an unbiased version



(A) Evolution of the objective function and its standard deviation (asynchronous case).

(B) Evolution of the bias term.

FIGURE 5.5: Metric learning experiments on a real dataset.

5.8 Conclusion

In this work, we have introduced new synchronous and asynchronous gossip algorithms to optimize functions depending on pairs of data points distributed over a network. The proposed methods are based on dual averaging and can readily accommodate various popular regularization terms. We provided an analysis showing that they behave similarly to the centralized dual averaging algorithm, with additional terms reflecting the network connectivity and the gradient bias. Finally, we proposed some numerical experiments on AUC maximization and metric learning which illustrate the performance of the proposed algorithms, as well as the influence of network topology. A challenging line of future research consists in designing and analyzing novel adaptive gossip schemes, where the communication scheme is dynamic and depends on the network connectivity properties and on the local information carried by each node.

5.9 Proofs

This section is organized as follows. First, we establish proofs in Section 5.9.1 for each lemma involved in the ergodic dual averaging analysis. Then, we perform the analysis of the pairwise decentralized dual averaging in the synchronous case. Finally, we tackle the proof of convergence in the fully asynchronous setting.

5.9.1 Ergodic dual averaging

Error after mixing (Lemma 12)

Proof. Let $\boldsymbol{\theta}$ be a $\mathcal{F}_t$-mesurable variable, $\boldsymbol{\theta}^* \in \mathcal{X}$ and $\tau \geq 0$. By definition of $P(t+\tau|t)$ and P^∞, the LHS in (5.25) can be rewritten as follows:

$$\int_{\boldsymbol{\xi}\in\Delta_n} \big(F(\boldsymbol{\theta},\boldsymbol{\xi}) - F(\boldsymbol{\theta}^*,\boldsymbol{\xi})\big)\mathrm{d}P^\infty(\boldsymbol{\xi}) - \int_{\boldsymbol{\xi}\in\Delta_n} \big(F(\boldsymbol{\theta},\boldsymbol{\xi}) - F(\boldsymbol{\theta}^*,\boldsymbol{\xi})\big)\mathrm{d}P(t+\tau|t)(\boldsymbol{\xi})$$

Both expectations in (5.52) only differ from the probability measures involved; the above quantity can thus be bounded by:

$$\int_{\boldsymbol{\xi}\in\Delta_n} \big(F(\boldsymbol{\theta},\boldsymbol{\xi}) - F(\boldsymbol{\theta}^*,\boldsymbol{\xi})\big)|\mathrm{d}P^\infty(\boldsymbol{\xi}) - \mathrm{d}P(t+\tau|t)(\boldsymbol{\xi})|$$

Using the fact that for any $\boldsymbol{\xi} \in \Delta_n$, $F(\cdot,\boldsymbol{\xi})$ is L_f-Lipschitz, one has, for any $\boldsymbol{\theta},\boldsymbol{\theta}^* \in \mathcal{X}$:

$$|F(\boldsymbol{\theta},\boldsymbol{\xi}) - F(\boldsymbol{\theta}^*,\boldsymbol{\xi})| \leq L_f\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \leq L_f D,$$

the last inequality deriving from the definition of D. Then the result holds using the definition of the total variation norm. □

Consecutive iterates bound (Lemma 13)

Proof. Let $(\mathbf{z}(t),\boldsymbol{\theta}(t))_{t\geq 0}$ be generated according to (5.19) for some positive, non-increasing sequence $(\gamma(t))_{t\geq 0}$. For any $t \geq 0$, we aim at bounding $\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\|$. Let $s(t) \in \partial\psi(\boldsymbol{\theta}(t))$ and $s(t+1) \in \partial\psi(\boldsymbol{\theta}(t+1))$. The respective optimality conditions on $\boldsymbol{\theta}(t)$ and $\boldsymbol{\theta}(t+1)$ lead to the following inequalities:

$$\begin{cases} \big(\gamma(t)\mathbf{z}(t) - \boldsymbol{\theta}(t) - \Gamma(t)s(t)\big)^\top\big(\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\big) & \leq & 0 \\ \big(\gamma(t+1)\mathbf{z}(t+1) - \boldsymbol{\theta}(t+1) - \Gamma(t+1)s(t+1)\big)^\top\big(\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t+1)\big) & \leq & 0 \end{cases}. \tag{5.52}$$

Then, using convexity of ψ and the property of the subgradient leads to:

$$\begin{cases} s(t+1)^\top(\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)) & \geq & \psi(\boldsymbol{\theta}(t+1)) - \psi(\boldsymbol{\theta}(t)) \\ s(t)^\top(\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t+1)) & \geq & \psi(\boldsymbol{\theta}(t)) - \psi(\boldsymbol{\theta}(t+1)) \end{cases} \tag{5.53}$$

Summing both inequalities in (5.52) and using (5.53), one obtains:

$$\begin{aligned}\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\|^2 \leq& \big(\gamma(t+1)\mathbf{z}(t+1) - \gamma(t)\mathbf{z}(t)\big)^\top\big(\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\big) \\ &+ \big(\Gamma(t+1) - \Gamma(t)\big)\big(\psi(\boldsymbol{\theta}(t+1)) - \psi(\boldsymbol{\theta}(t))\big).\end{aligned} \tag{5.54}$$

The optimality of $\boldsymbol{\theta}(t+1)$ ensures the following relation:

$$\boldsymbol{\theta}(t+1)^\top \frac{\mathbf{z}(t+1)}{t+1} - \frac{\|\boldsymbol{\theta}(t+1)\|^2}{2\Gamma(t+1)} - \psi(\boldsymbol{\theta}(t+1)) \geq \boldsymbol{\theta}(t)^\top \frac{\mathbf{z}(t+1)}{t+1} - \frac{\|\boldsymbol{\theta}(t)\|^2}{2\Gamma(t+1)} - \psi(\boldsymbol{\theta}(t)).$$

We can reformulate this last inequality in order to provide an upper bound on $\psi(\boldsymbol{\theta}(t+1)) - \psi(\boldsymbol{\theta}(t))$:

$$\begin{aligned}\psi(\boldsymbol{\theta}(t+1)) - \psi(\boldsymbol{\theta}(t)) \leq & (\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t))^\top \left(\frac{\mathbf{z}(t+1)}{t+1} + \frac{\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t+1)}{2\Gamma(t+1)} \right) \\ & + (\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t))^\top \frac{\boldsymbol{\theta}(t+1)}{\Gamma(t+1)} \\ \leq & \|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \left(\frac{\|\mathbf{z}(t+1)\|}{t+1} + \frac{\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t+1)\|}{2\Gamma(t+1)} \right) \\ & + \|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \frac{\|\boldsymbol{\theta}(t+1)\|}{\Gamma(t+1)},\end{aligned}$$

where the last inequality is derived from Cauchy-Schwarz relation. Since π_{t+1} is $\gamma(t+1)$-Lipschitz and $\pi_{t+1}(0) = 0$, one has $\|\boldsymbol{\theta}(t+1)\| \leq \gamma(t+1)\|\mathbf{z}(t+1)\|$. Moreover, by definition of $\mathbf{z}(t+1)$ and since all f_i are L-Lipschitz, one has $\|\mathbf{z}(t+1)\| \leq (t+1)L$. These two last results lead the following bound:

$$\psi(\boldsymbol{\theta}(t+1)) - \psi(\boldsymbol{\theta}(t)) \leq 2L\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| + \frac{\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t+1)\|^2}{2\Gamma(t+1)}.$$

Now, we can use this bound in inequality (5.54):

$$\left(1 - \frac{\Gamma(t+1) - \Gamma(t)}{2\Gamma(t+1)}\right) \|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \leq \|\gamma(t+1)\mathbf{z}(t+1) - \gamma(t)\mathbf{z}(t)\| + 2(\Gamma(t+1) - \Gamma(t))L.$$

The first term in the RHS can be simply bounded as follows:

$$\begin{aligned}\|\gamma(t+1)\mathbf{z}(t+1) - \gamma(t)\mathbf{z}(t)\| \leq & \gamma(t+1)\|\mathbf{z}(t+1) - \mathbf{z}(t)\| \\ & + (\gamma(t+1) - \gamma(t))\|\mathbf{z}(t)\| \\ \leq & (\Gamma(t+1) - \Gamma(t))L.\end{aligned}$$

Since $\Gamma(t+1)$ and $\Gamma(t)$ are both positive, one has:

$$1 - \frac{\Gamma(t+1) - \Gamma(t)}{2\Gamma(t+1)} = \frac{\Gamma(t+1) + \Gamma(t)}{2\Gamma(t+1)} \geq 0,$$

which finally leads to:

$$\begin{aligned}\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t)\| \leq & 3L\big(\Gamma(t+1) - \Gamma(t)\big) \frac{2\Gamma(t+1)}{\Gamma(t+1) + \Gamma(t)} \\ \leq & 3L\big(\Gamma(t+1) - \Gamma(t)\big) \left(1 + \frac{1}{2t+1}\right). \qquad (5.55)\end{aligned}$$

We make the additional assumption that $\gamma(t) \propto t^\alpha$ for some $\alpha \in (-1, 0)$. This is not a particularly restrictive assumption since the dual averaging

algorithm imposes that:

1. $\lim_{t\to\infty} \gamma(t) = 0$, hence $\alpha < 0$.
2. $\lim_{t\to\infty} t\gamma(t) = +\infty$, hence $\alpha + 1 > 0$.

With this assumption and Taylor-Lagrange formula yields:

$$\Gamma(t+1) - \Gamma(t) \leq (\alpha+1)\gamma(t),$$

and the final result holds. □

Gap with noisy objectives (Lemma 14)

Proof. Let $\tau, t \geq 0$. One has:

$$\begin{aligned} F(\boldsymbol{\theta}(t), \boldsymbol{\xi}(t+\tau)) - F(\boldsymbol{\theta}(t+\tau), \boldsymbol{\xi}(t+\tau)) &\leq L\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t+\tau)\| \\ &\leq L\sum_{s=0}^{\tau-1} \|\boldsymbol{\theta}(t+s) - \boldsymbol{\theta}(t+s+1)\|. \end{aligned} \tag{5.56}$$

Using Lemma 13, one has for any $0 \leq s \leq \tau - 1$:

$$\begin{aligned} \|\boldsymbol{\theta}(t+s) - \boldsymbol{\theta}(t+s+1)\| &\leq 3L\left(1 + \frac{1}{2(t+s)+1}\right)\big(\Gamma(t+s+1) - \Gamma(t+s)\big) \\ &\leq 3L\left(1 + \frac{1}{2t+1}\right)\big(\Gamma(t+s+1) - \Gamma(t+s)\big). \end{aligned} \tag{5.57}$$

Summing over s leads to:

$$\sum_{s=0}^{\tau-1} \|\boldsymbol{\theta}(t+s) - \boldsymbol{\theta}(t+s+1)\| \leq 3L\left(1 + \frac{1}{2t+1}\right)\big(\Gamma(t+\tau) - \Gamma(t)\big), \tag{5.58}$$

and (5.28) holds.

We now make the assumption that $\gamma(t) \propto t^\alpha$, with $\alpha \in (-1, 0)$. As denoted in the proof of Lemma 13, one has:

$$\Gamma(t+\tau) - \Gamma(t) \leq \tau(1+\alpha)\gamma(t), \tag{5.59}$$

so (5.29) also holds. □

5.9.2 Synchronous Pairwise Gossip Dual Averaging

In this section, we focus on the synchronous setting. First, we establish a result on the expected dispersion of the dual variables over the network. We then use this result to detail the rate of the decentralized dual averaging, both for separable and pairwise objectives. Finally, we use the ergodic dual averaging to provide an explicit rate of convergence.

In DUCHI, AGARWAL, and WAINWRIGHT, 2012, the following convergence rate for distributed dual averaging is established:

$$\mathbb{E}[R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*) \leq \frac{1}{2T\gamma(T)}\|\boldsymbol{\theta}^*\|^2 + \frac{L_f^2}{2T}\sum_{t=2}^{T}\gamma(t-1) + \frac{L_f}{nT}\sum_{t=2}^{T}\gamma(t-1)\sum_{j=1}^{n}\Big(\|\mathbf{z}_i(t) - \mathbf{z}_j(t)\| + \|\bar{\mathbf{z}}(t) - \mathbf{z}_j(t)\|\Big).$$

The first part is an optimization term, which is exactly the same as in the centralized setting. Then, the second part is a network-dependent term which depends on the global variation of the dual variables; the following lemma provides an explicit dependence between this term and the topology of the network.

Lemma 16. Let $\mathbf{W}(\mathcal{G}) = \mathbf{I}_n - \frac{\mathbf{L}^{\mathcal{G}}}{|\mathcal{E}|}$ and let $(\mathbf{G}(t))_{t\geq 1}$ and $(\mathbf{Z}(t))_{t\geq 1}$ respectively be the gradients and the gradients cummulative sum of the distributed dual averaging algorithm. If $\mathcal{G}$ is connected and non bipartite, then one has for $t \geq 1$:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{z}_i(t) - \overline{\mathbf{z}}(t)\| \leq \frac{L_f}{1-\sqrt{\lambda_2^{\mathcal{G}}}},$$

where $\lambda_2^{\mathcal{G}}$ is the second largest eigenvalue of $\mathbf{W}(\mathcal{G})$.

Proof. For $t \geq 1$, let $\mathbf{W}(t)$ be the random matrix such that if $(i,j) \in \mathcal{E}$ is picked at t, then

$$\mathbf{W}(t) = \mathbf{I}_n - \frac{1}{2}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top.$$

As denoted in DUCHI, AGARWAL, and WAINWRIGHT, 2012, the update rule for $\mathbf{Z}$ can be expressed as follows:

$$\mathbf{Z}(t+1) = \mathbf{G}(t) + \mathbf{W}(t)\mathbf{Z}(t),$$

for any $t \geq 1$, reminding that $\mathbf{G}(0) = 0, \mathbf{Z}(1) = 0$. Therefore, one can obtain recursively

$$\mathbf{Z}(t) = \sum_{s=0}^{t}\mathbf{W}(t:s)\mathbf{G}(s),$$

where $\mathbf{W}(t:s) = \mathbf{W}(t)\dots\mathbf{W}(s+1)$, with the convention $\mathbf{W}(t:t) = \mathbf{I}_n$. For any $t \geq 1$, let $\mathbf{W}'(t)$ be defined as follows:

$$\mathbf{W}'(t) := \mathbf{W}(t) - \frac{\mathbf{1}_n\mathbf{1}_n^\top}{n}.$$

One can notice that for any $0 \leq s \leq t$, $\mathbf{W}'(t:s) = \mathbf{W}(t:s) - \frac{\mathbf{1}_n\mathbf{1}_n^\top}{n}$ and write:

$$\mathbf{Z}(t) - \mathbf{1}_n\overline{\mathbf{z}}(t)^\top = \sum_{s=0}^{t}\mathbf{W}'(t:s)\mathbf{G}(s).$$

We now take the expected value of the Frobenius norm:

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{Z}(t)-\mathbf{1}_n\overline{\mathbf{z}}(t)^\top\right\|_F\right] &\leq \sum_{s=0}^{t}\mathbb{E}\left[\|\mathbf{W}(t:s)\mathbf{G}(s)\|_F\right]\\ &\leq \sum_{s=0}^{t}\sqrt{\mathbb{E}\left[\|\mathbf{W}(t:s)\mathbf{G}(s)\|_F^2\right]}\\ &= \sum_{i=1}^{n}\sum_{s=0}^{t}\sqrt{\mathbb{E}\left[\mathbf{g}^{(i)}(s)^\top\mathbf{W}'(t:s)^\top\mathbf{W}'(t:s)\mathbf{g}^{(i)}(s)\right]},\end{aligned}$$

where $\mathbf{g}^{(i)}(s)$ is the i-th column of matrix $\mathbf{G}(s)$. Conditioning over $\mathcal{F}_{t-1}$ leads to:

$$\mathbb{E}\left[\mathbf{g}^{(i)}(s)^\top\mathbf{W}'(t:s)^\top\mathbf{W}'(t:s)\mathbf{g}^{(i)}(s)\right] \leq \lambda_2^{\mathcal{G}}\mathbb{E}\left[\mathbf{g}^{(i)}(s)^\top\mathbf{W}'(t-1:s)\mathbf{g}^{(i)}(s)\right],$$

and $\lambda_2^{\mathcal{G}}$ is the second largest eigenvalue of $\mathbf{W}(\mathcal{G}) := \mathbb{E}[\mathbf{W}(1)] = \mathbf{I}_n - \mathbf{L}/|\mathcal{E}|$ (see Section 4.7.1 for details about $\mathbf{W}(\mathcal{G})$). Using the fact that for any $s \geq 0$, $\|\mathbf{G}(s)\|_F^2 \leq nL_f^2$, one has:

$$\mathbb{E}\left[\left\|\mathbf{Z}(t)-\mathbf{1}_n\overline{\mathbf{z}}(t)^\top\right\|_F\right] \leq \sqrt{n}L_f\sum_{s=0}^{t}\left(\lambda_2^{\mathcal{G}}\right)^{\frac{t-s}{2}} \leq \frac{\sqrt{n}L_f}{1-\sqrt{\lambda_2^{\mathcal{G}}}}.$$

Finally, using the bounds between ℓ_1 and ℓ_2-norms yields:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{z}_i(t)-\overline{\mathbf{z}}(t)\| \leq \frac{1}{\sqrt{n}}\mathbb{E}\left\|\mathbf{Z}(t)-\mathbf{1}_n\overline{\mathbf{z}}(t)^\top\right\|_F \leq \frac{L_f}{1-\sqrt{\lambda_2^{\mathcal{G}}}}.$$

□

With this bound on the dual variables, one can reformulate the convergence rate as stated below.

Corollary 2. *Let $\mathcal{G} = ([n], \mathcal{E})$ be a connected and non bipartite graph. Let $(\gamma(t))_{t\geq 1}$ be a non-increasing and non-negative sequence. For $i \in [n]$, let $(\mathbf{g}_i(t))_{t\geq 1}$, $(\mathbf{z}_i(t))_{t\geq 1}$ and $(\boldsymbol{\theta}_i(t))_{t\geq 1}$ be generated according to the distributed dual averaging algorithm. For $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}'\in\mathbb{R}^d} R_n(\boldsymbol{\theta}')$, $i \in [n]$ and $T \geq 2$, one has:*

$$\begin{aligned}\mathbb{E}[R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*) \leq &\frac{1}{2T\gamma(T)}\|\boldsymbol{\theta}^*\|^2 + \frac{L_f^2}{2T}\sum_{t=1}^{T-1}\gamma(t)\\ &+ \frac{3L_f^2}{T\left(1-\sqrt{\lambda_2^{\mathcal{G}}}\right)}\sum_{t=1}^{T-1}\gamma(t),\end{aligned}$$

where $\lambda_2^{\mathcal{G}} < 1$ is the second largest eigenvalue of $\mathbf{W}(\mathcal{G})$.

We now focus on gossip dual averaging for pairwise functions, as shown in Algorithm 15. The key observation is that, at each iteration, the descent direction is stochastic but also a *biased* estimate of the gradient. That is, instead of updating a dual variable $\mathbf{z}_i(t)$ with $\mathbf{g}_i(t)$ such that

$\mathbb{E}[\mathbf{g}_i(t)|\boldsymbol{\theta}_i(t)] = \nabla f_i(\boldsymbol{\theta}_i(t))$, we perform some update $\mathbf{d}_i(t)$, and we denote by $\boldsymbol{\epsilon}_i(t)$ the quantity such that $\mathbb{E}[\mathbf{d}_i(t) - \boldsymbol{\epsilon}_i(t)|\boldsymbol{\theta}_i(t)] = \mathbb{E}[\mathbf{g}_i(t)|\boldsymbol{\theta}_i(t)] = \nabla f_i(\boldsymbol{\theta}_i(t))$. We will now prove Proposition 5, which allows to upper-bound the error with an additional bias-dependent term.

Proof. We can apply the same arguments as in the proofs of centralized and distributed dual averaging, so for $T > 0$ and $i \in [n]$:

$$\begin{aligned}
\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*) &\leq \frac{L}{nT}\sum_{t=2}^{T}\gamma(t-1)\sum_{j=1}^{n}\mathbb{E}\Big[\|\mathbf{z}_i(t) - \mathbf{z}_j(t)\| \\
&\quad + \frac{L}{nT}\sum_{t=2}^{T}\gamma(t-1)\sum_{j=1}^{n}\|\bar{\mathbf{z}}(t) - \mathbf{z}_j(t)\|\Big] \\
&\quad + \frac{1}{T}\sum_{t=2}^{T}\mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{g}}(t)] \\
&\leq \frac{3L}{nT}\sum_{t=2}^{T}\gamma(t-1)\sum_{j=1}^{n}\|\bar{\mathbf{z}}(t) - \mathbf{z}_j(t)\|\Big] \\
&\quad + \frac{1}{T}\sum_{t=2}^{T}\mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{g}}(t)].
\end{aligned}$$

The first term can be bound using Lemma 16. The second term however can no longer be bound using Lemma 11, since the updates are performed with $\mathbf{d}_j(t)$ and not $\mathbf{g}_j(t) = \mathbf{d}_j(t) - \boldsymbol{\epsilon}_j(t)$. With the definition of $\mathbf{d}_j(t)$, the former yields:

$$\frac{1}{T}\sum_{t=2}^{T}\mathbb{E}_t[\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{g}}(t)] = \frac{1}{T}\sum_{t=2}^{T}\mathbb{E}_t[(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top(\bar{\mathbf{d}}(t) - \bar{\boldsymbol{\epsilon}}(t))].$$

Now Lemma 11 can be applied to the first term in the right hand side and the result holds. □

We now focus on the proof of 23. This results is based both on Proposition 5 and ergodic dual averaging presented in Section 5.3.3.

Proof. Throughout this proof, we assume $\psi \equiv 0$ for simplicity; similar results can however be obtained in the case $\psi \not\equiv 0$. Let $i \in [n]$ and $T \geq 1$. Using a reasoning similar to Theorem 22, one has:

$$\begin{aligned}
R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*) &\leq \frac{1}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}L_f\gamma(t)\|\mathbf{z}_i(t) - \mathbf{z}_j(t)\| \\
&\quad + \frac{1}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}\big(f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*)\big) \\
&\leq \frac{2L_f}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}\gamma(t)\|\mathbf{z}_i(t) - \bar{\mathbf{z}}(t)\| \\
&\quad + \frac{1}{nT}\sum_{t=1}^{T}\sum_{j=1}^{n}\big(f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*)\big).
\end{aligned}$$

The first term in the right hand side can be bounded using Lemma 16, so we only need to handle the last term. To do so, we adapt the proof of convergence for the ergodic dual averaging to the partial objectives. For $j \in [n]$, let us define $F_j : \mathbb{R}^d \times \Delta_n \to \mathbb{R}$ such that for any $(\boldsymbol{\theta}, \boldsymbol{\xi}) \in \mathbb{R}^d \times \Delta_n$:

$$F_j(\boldsymbol{\theta}, \boldsymbol{\xi}) = \sum_{k=1}^{n} \xi_k f(\boldsymbol{\theta}; \mathbf{x}_j, \mathbf{x}_k).$$

In addition, for $t \geq 1$, we define $\boldsymbol{\xi}_j(t) \in \{0,1\}^n \cap \Delta_n$ such that for $k \in [n]$, $\boldsymbol{\xi}_j(t)^\top \mathbf{e}_k = 1$ if and only if $\mathbf{y}_j(t) = \mathbf{x}_k$. Deriving the decomposition of the ergodic dual averaging proof yields:

$$\sum_{t=1}^{T} (f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*)) =$$

$$\sum_{t=1}^{T-\tau} (f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*) + F_j(\boldsymbol{\theta}_j(t), \boldsymbol{\xi}_j(t+\tau)) - F_j(\boldsymbol{\theta}^*, \boldsymbol{\xi}_j(t+\tau))) \tag{5.60}$$

$$+ \sum_{t=1}^{T-\tau} (F_j(\boldsymbol{\theta}_j(t), \boldsymbol{\xi}_j(t+\tau)) - F_j(\boldsymbol{\theta}_j(t+\tau), \boldsymbol{\xi}_j(t+\tau))) \tag{5.61}$$

$$+ \sum_{t=\tau+1}^{T} (F_j(\boldsymbol{\theta}_j(t), \boldsymbol{\xi}_j(t)) - F_j(\boldsymbol{\theta}^*, \boldsymbol{\xi}_j(t))) \tag{5.62}$$

$$+ \sum_{t=T-\tau+1}^{T} (f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*)). \tag{5.63}$$

Bounding the term (5.60) requires the knowledge of the total variation distance between the random walk associated to j after τ algorithm steps and the uniform. CHUNG, 1997, Theorem 1.18 states that such norm for one random walk is upper-bounded as follows:

$$\|P(t+\tau|t) - P_\infty\|_{TV} \leq \frac{|\mathcal{E}|(\tilde{\lambda}_2^{\mathcal{G}})^\tau}{2 \min_{k \in [n]} d_k},$$

where $\tilde{\lambda}_2^{\mathcal{G}}$ is such that

$$\tilde{\lambda}_2^{\mathcal{G}} \leq 1 - \frac{\beta_{n-1}}{\max_{k \in [n]} d_k}.$$

However, in this case, the random walk associated to the j-th auxiliary observation will not necessarily be propagated τ times during τ algorithm steps. Since we are interested in expected bound, we bound the expected

total variation norm as follows:

$$\begin{aligned}\mathbb{E}\|P_j(t+\tau|t) - P_\infty\|_{TV} &\leq \sum_{s=0}^{\tau} \mathbb{P}\left(\sum_{r=t}^{t+\tau} \delta_j(r) = s\right) \frac{|\mathcal{E}|(\tilde{\lambda}_2^{\mathcal{G}})^s}{2\min_{k\in[n]} d_k} \\ &\leq \frac{|\mathcal{E}|}{2\min_{k\in[n]} d_k} \sum_{s=0}^{\tau} \binom{\tau}{s} p_j^s (1-p_j)^{\tau-s} (\tilde{\lambda}_2^{\mathcal{G}})^s \\ &= \frac{|\mathcal{E}|}{2\min_{k\in[n]} d_k} \left((1-p_j) + p_j\tilde{\lambda}_2^{\mathcal{G}}\right)^\tau \\ &\leq c(\mathcal{G}) \cdot c'(\mathcal{G})^\tau,\end{aligned}$$

where

$$c(\mathcal{G}) := \frac{|\mathcal{E}|}{2\min_{k\in[n]} d_k}$$

and

$$c'(\mathcal{G}) := \max_{k\in[n]} \left\{(1-p_k) + p_k\tilde{\lambda}_2^{\mathcal{G}}\right\}.$$

The term (5.61) provides an upper-bound similar to the centralized ergodic case, as it only depends on the Lipschitz constant L_f and the step size sequence. When averaging over all $j \in [n]$, the term (5.62) can be upper-bounded as follows:

$$\begin{aligned}&\frac{1}{n}\sum_{j=1}^{n}\sum_{t=\tau+1}^{T} (F_j(\boldsymbol{\theta}_j(t), \boldsymbol{\xi}_j(t)) - F_j(\boldsymbol{\theta}^*, \boldsymbol{\xi}_j(t))) \\ &\qquad\leq \frac{1}{n}\sum_{j=1}^{n}\sum_{t=\tau+1}^{T} (F_j(\boldsymbol{\theta}_j(t), \boldsymbol{\xi}_j(t)) - F_j(\boldsymbol{\omega}(t), \boldsymbol{\xi}_j(t))) \\ &\qquad\quad + \sum_{t=1}^{T} (\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{d}}(t) \\ &\qquad\leq \frac{L_f}{n}\sum_{j=1}^{n}\sum_{t=\tau+1}^{T} \gamma(t)\|\mathbf{z}_j(t) - \bar{\mathbf{z}}(t)\| \\ &\qquad\quad + \sum_{t=1}^{T} (\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)^\top \bar{\mathbf{d}}(t),\end{aligned}$$

which can then be bounded similarly to Proposition 5. Finally, the last term (5.63) is bounded by $2\tau LD$, as in the centralized case. □

5.9.3 Asynchronous Pairwise Dual Averaging

In this section, we focus on a fully asynchronous setting where each node has a local clock. We assume for simplicity that each node has a clock ticking at a Poisson rate equals to 1, so it is equivalent to a global clock ticking at a Poisson rate of n, and then drawing an edge uniformly at random (see BOYD et al., 2006 for more details). Under this assumption, we can state a method detailed in Algorithm 16.

The main difficulty in the asynchronous setting is that each node i has to use a time estimate m_i instead of the global clock reference (that is no longer available in such a context). Even if the time estimate is unbiased,

its variance puts an additional error term in the convergence rate. However, for an iteration T *large enough*, one can bound these estimates as stated bellow.

Lemma 17. There exists $T_1 > 0$ such that for any $t \geq T_1$, any $k \in [n]$ and any $q > 0$,

$$t^- := t - t^{\frac{1}{2}+q} \leq m_k(t) \leq t + t^{\frac{1}{2}+q} =: t^+ \text{ a.s.}$$

Proof. Let $k \in [n]$. For $t \geq 1$, let us define $\delta_k(t)$ such that $\delta_k(t) = 1$ if k is picked at iteration t and $\delta_k(t) = 0$ otherwise. Then one has $m_k(t) = (1/p_k)\sum_{s=1}^{t} \delta_k(t)$. Since $(\delta_k(t))_{t\geq 1}$ is a Bernoulli process of parameter $1/p_k$, by the law of iterative logarithms DUDLEY, 2010, (NEDIĆ, 2011, Lemma 3) one has with probability 1 and for any $q > 0$

$$\lim_{t\to+\infty} \frac{|m_k(t) - t|}{t^{\frac{1}{2}+q}} = 0,$$

and the result holds. □

Theorem 25. *Let $\mathcal{G}$ be a connected and non bipartite graph. Let $(\gamma(t))_{t\geq 1}$ be defined as $\gamma(t) = c/t^{1/2+\alpha}$ for some constant $c > 0$ and $\alpha \in (0, 1/2)$. For $i \in [n]$, let $(\mathbf{d}_i(t))_{t\geq 1}$, $(\mathbf{g}_i(t))_{t\geq 1}$, $(\boldsymbol{\epsilon}_i(t))_{t\geq 1}$, $(\mathbf{z}_i(t))_{t\geq 1}$ and $(\boldsymbol{\theta}_i(t))_{t\geq 1}$ be generated as stated previously. For $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}'\in\mathbb{R}^d} R_n(\boldsymbol{\theta}')$, $i \in [n]$ and $T > 0$, one has for some C:*

$$R_n(\bar{\boldsymbol{\theta}}_i(T)) - R_n(\boldsymbol{\theta}^*) \leq C \max(T^{-\alpha/2}, T^{\alpha-1/2}) + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_t[\bar{\boldsymbol{\epsilon}}(t)^\top \boldsymbol{\omega}(t)]\,.$$

Proof. In the asynchronous case, for $i \in [n]$ and $t \geq 1$, one has

$$\bar{\boldsymbol{\theta}}_i(T) = \frac{1}{m_i(T)}\sum_{t=1}^{T} \frac{\delta_i(t)}{p_i}\boldsymbol{\theta}_i(t).$$

Then, using the convexity of R_n, one has:

$$\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i(T)] - R_n(\boldsymbol{\theta}^*) \leq \mathbb{E}_T\left[\frac{1}{m_i(T)}\sum_{t=1}^{T} \frac{\delta_i(t)}{p_i} R_n(\boldsymbol{\theta}_i(t))\right] - R_n(\boldsymbol{\theta}^*). \tag{5.64}$$

By Lemma 17, one has for $q > 0$

$$\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i(T)] - R_n(\boldsymbol{\theta}^*) \leq \frac{1}{T^-}\sum_{t=1}^{T} \mathbb{E}_T\left[\frac{\delta_i(t)}{p_i} R_n(\boldsymbol{\theta}_i(t))\right] - R_n(\boldsymbol{\theta}^*).$$

Similarly to the synchronous case, one can write

$$\begin{aligned}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}f(\boldsymbol{\theta}_i(t))\right] =& \sum_{j=1}^{n}\frac{1}{n}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}f_j(\boldsymbol{\theta}_i(t))\right]\\ =&\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}(f_j(\boldsymbol{\theta}_i(t))-f_j(\boldsymbol{\theta}_j(t))\right]\\ &+\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}f_j(\boldsymbol{\theta}_j(t))\right].\end{aligned}$$

In order to use the gradient inequality, we need to introduce $\delta_j(t)f_j(\boldsymbol{\theta}_j(t))$ instead of $\delta_i(t)f_j(\boldsymbol{\theta}_j(t))$. For $j\in[n]$, one has:

$$\begin{aligned}\frac{1}{T^-}\sum_{t=1}^{T}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}f_j(\boldsymbol{\theta}_j(t))\right] =&\frac{1}{T^-}\sum_{t=1}^{T}\mathbb{E}_T\left[\left(\frac{\delta_i(t)}{p_i}-\frac{\delta_j(t)}{p_j}\right)f_j(\boldsymbol{\theta}_j(t))\right]\\ &+\frac{1}{T^-}\sum_{t=1}^{T}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}f_j(\boldsymbol{\theta}_j(t))\right].\end{aligned}$$

Let $N_j=\sum_{t=1}^{T}\delta_j(t)$ and let $1\leq t_1<\ldots<t_{N_j}\leq T$ be such that $\delta_j(t_k)=1$ for $k\in[N_j]$. One can write

$$\begin{aligned}&\frac{1}{T^-}\sum_{t=1}^{T}\mathbb{E}_T\left[\left(\frac{\delta_i(t)}{p_i}-\frac{\delta_j(t)}{p_j}\right)f_j(\boldsymbol{\theta}_j(t))\right] \qquad (5.65)\\ &\quad=\frac{1}{T^-}\mathbb{E}_T\left[\sum_{k=1}^{N_j-1}\left(\left(\sum_{t=t_k}^{t_{k+1}-1}\frac{\delta_i(t)}{p_i}\right)-\frac{1}{p_j}\right)f_j(\boldsymbol{\theta}_j(t_k))\right]\\ &\qquad+\frac{1}{T^-}\mathbb{E}_T\left[\left(\sum_{t=0}^{t_1}\frac{\delta_i(t)}{p_i}\right)f_j(\boldsymbol{\theta}_j(0))\right]\\ &\qquad+\frac{1}{T^-}\mathbb{E}_T\left[\left(\left(\sum_{t=t_{N_j}}^{T}\frac{\delta_i(t)}{p_i}\right)-\frac{1}{p_j}\right)f_j(\boldsymbol{\theta}_j(t_{N_j}))\right]\\ &\quad\leq\frac{1}{T^-}\mathbb{E}_T\left[\sum_{k=1}^{N_j-1}\left(\left(\sum_{t=t_k}^{t_{k+1}-1}\frac{\delta_i(t)}{p_i}\right)-\frac{1}{p_j}\right)f_j(\boldsymbol{\theta}_j(t_k))\right]\\ &\qquad+\frac{f_j(0)}{p_ip_jT^-}+\frac{L_f^2\mathbb{E}_T[\gamma(t_{N_j}-1)]}{p_ip_j}.\end{aligned}$$

We need to study the behavior of δ_i and δ_j in the first term of the right hand side. One can check that

$$\begin{aligned}&\mathbb{E}_T\left[\sum_{k=1}^{N_j-1}\left(\left(\sum_{t=t_k}^{t_{k+1}-1}\frac{\delta_i(t)}{p_i}\right)-\frac{1}{p_j}\right)f_j(\boldsymbol{\theta}_j(t_k))\right]\\ &\qquad=\mathbb{E}_T\left[\sum_{k=1}^{N_j-1}\left(\mathbb{E}\left[\sum_{t=t_k}^{t_{k+1}-1}\frac{\delta_i(t)}{p_i}\middle|t_k,t_{k+1}\right]-\frac{1}{p_j}\right)f_j(\boldsymbol{\theta}_j(t_k))\right].\end{aligned}$$

$\delta_i(t)$ will not have the same dependency in t_k whether i and j are connected or not. Let us first assume that $(i,j) \in \mathcal{E}$. Then,

$$\mathbb{E}[\delta_i(t_k)|t_k] = \mathbb{E}[\delta_i(t)|\delta_j(t)=1] = \frac{1}{d_j}.$$

Also, for $t_k < t < t_{k+1}$, we get:

$$\mathbb{E}[\delta_i(t)|t_k] = \mathbb{E}[\delta_i(t)|\delta_j(t)=0] = \frac{p_i - 2/|\mathcal{E}|}{1-p_j}.$$

Finally, if $(i,j) \in \mathcal{E}$, we obtain

$$\mathbb{E}\left[\sum_{t=t_k}^{t_{k+1}-1} \frac{\delta_i(t)}{p_i} \middle| t_k, t_{k+1}\right] = \left(\frac{1}{d_j} + (t_{k+1}-t_k-1)\frac{p_i - 2/|\mathcal{E}|}{1-p_j}\right)\frac{1}{p_i}.$$

Before using this relation in the full expectation, let us denote that since $t_{k+1} - t_k$ is independent from t_k, one can write

$$\begin{aligned}\mathbb{E}\left[\left(\frac{1}{d_j} + (t_{k+1}-t_k-1)\frac{p_i - 2/|\mathcal{E}|}{1-p_j}\right)\frac{1}{p_i} \mid t_k\right] \\ &= \left(\frac{1}{d_j} + \left(\frac{1-p_j}{p_j}\right)\frac{p_i - 2/|\mathcal{E}|}{1-p_j}\right)\frac{1}{p_i} \\ &= \frac{1}{p_j}.\end{aligned}$$

We can now use this relation in the full expectation

$$\begin{aligned}&\mathbb{E}_T\left[\left(\frac{\delta_i(t)}{p_i} - \frac{\delta_j(t)}{p_j}\right) f_j(\boldsymbol{\theta}_j(t))\right] \\ &= \mathbb{E}_T\left[\sum_{k=1}^{N_j-1}\left(\mathbb{E}\left[\mathbb{E}\left[\sum_{t=t_k}^{t_{k+1}-1}\frac{\delta_i(t)}{p_i}\middle| t_{k+1}-t_k\right]\middle| t_k\right] - \frac{1}{p_j}\right) f_j(\boldsymbol{\theta}_j(t_k))\right] \\ &= 0. \qquad (5.66)\end{aligned}$$

Similarly if $(i,j) \notin \mathcal{E}$, one has

$$\mathbb{E}[\delta_i(t_k)|t_k] = \mathbb{E}[\delta_i(t)|\delta_j(t)=1] = 0,$$

and for $t_k < t < t_{k+1}$,

$$\mathbb{E}[\delta_i(t)|t_k] = \mathbb{E}[\delta_i(t)|\delta_j(t)=0] = \frac{p_i}{1-p_j},$$

so the result of Equation (5.66) holds in this case. We have just shown that for every $j \in [n]$, we can use $\delta_j(t) f_j(\boldsymbol{\theta}_j(t))/p_j$ instead of $\delta_i(t) f_j(\boldsymbol{\theta}_j(t))/p_i$.

Combining (5.64) and (5.65) yields:

$$\mathbb{E}_T[R_n(\bar{\boldsymbol{\theta}}_i(T))] - R_n(\boldsymbol{\theta}^*) \leq \frac{1}{nT^-}\sum_{t=2}^{T}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}(f_j(\boldsymbol{\theta}_i(t)) - f_j(\boldsymbol{\theta}_j(t))\right] \tag{5.67}$$

$$+ \frac{1}{nT^-}\sum_{t=2}^{T}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\left(f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*)\right)\right]$$

$$+ \frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}(\psi(\boldsymbol{\theta}_i(t)) - \psi(\boldsymbol{\theta}^*))\right] \tag{5.68}$$

$$+ \frac{f_j(0)}{p_i p_j T^-} + \frac{L_f^2\mathbb{E}_T[\gamma(t_{N_j}-1)]}{p_i p_j}.$$

Let us focus on the second term of the right hand side. For $t \geq 2$, one can write

$$\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\left(f_j(\boldsymbol{\theta}_j(t)) - f_j(\boldsymbol{\theta}^*)\right)\right] \leq \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)^\top(\boldsymbol{\theta}_j(t) - \boldsymbol{\theta}^*)\right]$$

$$= \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)^\top(\boldsymbol{\theta}_j(t) - \boldsymbol{\omega}(t))\right] \tag{5.69}$$

$$+ \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)^\top(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)\right] \tag{5.70}$$

Here we control the term from (5.70) using $\boldsymbol{\omega}(t) := \pi_{m_i(t)}(\bar{\mathbf{z}}(t))$

$$\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)^\top(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)\right] = \mathbb{E}_T\left[\left(\frac{1}{n}\sum_{j=1}^{n}\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)\right)^\top(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)\right]$$

$$= \mathbb{E}_T\left[\bar{\mathbf{g}}(t)^\top(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)\right],$$

and the reasoning of the synchronous case can be applied to obtain

$$\frac{1}{nT^-}\sum_{t=2}^{T}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)^\top(\boldsymbol{\omega}(t) - \boldsymbol{\theta}^*)\right] \leq \frac{L_f^2}{2T^-}\sum_{t=2}^{T}\gamma(t-1) + \frac{\|\boldsymbol{\theta}^*\|^2}{2\gamma(T)}$$

$$+ \frac{1}{T}\sum_{t=2}^{T}\mathbb{E}_t[\bar{\boldsymbol{\epsilon}}^n(t)^\top\boldsymbol{\omega}(t)]$$

$$+ \frac{1}{T^-}\sum_{t=2}^{T}(\psi(\boldsymbol{\theta}^*) - \mathbb{E}_T[\psi(\boldsymbol{\omega}(t))]). \tag{5.71}$$

Let us regroup the term from (5.71) and (5.68) together:

$$\begin{aligned}
&\frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}(\psi(\boldsymbol{\theta}_i(t))-\psi(\boldsymbol{\theta}^*))\right]+\frac{1}{T^-}\sum_{t=2}^{T}(\psi(\boldsymbol{\theta}^*)-\mathbb{E}_T[\psi(\boldsymbol{\omega}(t))])\\
&\quad=\frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}\psi(\boldsymbol{\theta}_i(t))-\psi(\boldsymbol{\omega}(t))\right]\\
&\quad=\frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}(\psi(\boldsymbol{\theta}_i(t))-\psi(\boldsymbol{\omega}(t)))\right]\\
&\qquad+\frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[(\frac{\delta_i(t)}{p_i}-1)\psi(\boldsymbol{\omega}(t))\right]\\
&\quad=\frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[\frac{\delta_i(t)}{p_i}(\psi(\boldsymbol{\theta}_i(t))-\psi(\boldsymbol{\omega}(t)))\right],
\end{aligned}$$

where we have used for the last term the same arguments as in (5.66) to state $\frac{1}{T^-}\sum_{t=2}^{T}\mathbb{E}_T\left[(\frac{\delta_i(t)}{p_i}-1)\psi(\boldsymbol{\omega}(t))\right]=0$. Then, one can use the fact that π_t is $\gamma(t)$-Lipschitz to write:

$$\frac{1}{p_iT^-}\sum_{t=2}^{T}\mathbb{E}_T\left[2L_f\gamma(m_i(t-1))\|\bar{\mathbf{z}}(t)-\mathbf{z}_i(t)\|+\frac{\gamma(m_i(t-1))\|\bar{\mathbf{z}}(t)-\mathbf{z}_i(t)\|^2}{2(m_i(t-1))}\right].$$

Provided that $\gamma(t)\leq C/\sqrt{t}$ for some constant C, then using Lemma 17 we can bound this term by $C'/\sqrt{T}$. Let us now control the term in (5.69):

$$\begin{aligned}
&\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_T\left[\frac{\delta_j(t)}{p_j}\mathbf{g}_j(t)^\top(\boldsymbol{\theta}_j(t)-\boldsymbol{\omega}(t))\right]\\
&\qquad\leq\frac{L_f}{np_j}\sum_{j=1}^{n}\mathbb{E}_T\left[\|\boldsymbol{\theta}_j(t)-\boldsymbol{\omega}(t)\|\right] \qquad (5.72)\\
&\qquad\leq\frac{L_f}{np_j}\sum_{j=1}^{n}\mathbb{E}_T\left[\|\boldsymbol{\theta}_j(t)-\tilde{\boldsymbol{\theta}}_j(t)\|+\|\tilde{\boldsymbol{\theta}}_j(t)-\boldsymbol{\omega}(t)\|\right]\\
&\qquad\leq\frac{L_f}{np_j}\sum_{j=1}^{n}\mathbb{E}_T\left[\gamma(m_j(t-1))\|z_j(t)-\bar{\mathbf{z}}(t)\|+\|\tilde{\boldsymbol{\theta}}_j(t)-\boldsymbol{\omega}(t)\|\right].
\end{aligned}$$

where $\tilde{\boldsymbol{\theta}}_j(t)=\pi_{m_j(t-1)}(-\bar{\mathbf{z}}(t))$. We can apply Lemma 18 with the choice $\boldsymbol{\theta}_1=\tilde{\boldsymbol{\theta}}_j(t)$, $\boldsymbol{\theta}_2=\boldsymbol{\omega}(t)$, $t_1=m_j(t)$, $t_2=m_i(t)$ and $z=\bar{\mathbf{z}}(t)$:

$$\begin{aligned}
&\|\boldsymbol{\omega}(t)-\tilde{\boldsymbol{\theta}}_j(t)\|\\
&\leq\|\bar{\mathbf{z}}(t)\||\gamma(m_i(t))-\gamma(m_j(t))|\\
&\quad+\|\bar{\mathbf{z}}(t)\|\left(\frac{3}{2}+\max\left(\frac{\gamma(m_j(t))}{\gamma(m_i(t))},\frac{\gamma(m_i(t))}{\gamma(m_j(t))}\right)\right)\left|\gamma(m_j(t))-\frac{m_i(t)}{m_j(t)}\gamma(m_i(t))\right|\Bigg)\\
&\quad+\|\bar{\mathbf{z}}(t)\|\left(\frac{3}{2}+\max\left(\frac{\gamma(m_j(t))}{\gamma(m_i(t))},\frac{\gamma(m_i(t))}{\gamma(m_j(t))}\right)\right)\left|\gamma(m_i(t))-\frac{m_j(t)}{m_i(t)}\gamma(m_j(t))\right|\Bigg).
\end{aligned}$$

We use Lemma 17 with the choice $q=\alpha/2$, so we can bound for t large

enough the former expression by a term of order $\|\bar{\mathbf{z}}(t)\||\gamma(m_i(t))-\gamma(m_j(t))|$. Note also that $\|\bar{\mathbf{z}}(t)\| \leq L_f \max_{k=1,\dots,n} m_k(t)$, so for t large enough we obtain:

$$\|\boldsymbol{\omega}(t) - \tilde{\boldsymbol{\theta}}_j(t)\| \leq L_F t^+ |\gamma(t^-) - \gamma(t^+)|.$$

With the additional constraint that the step size is of the form $\gamma(t) = Ct^{-1/2-\alpha}$, the term $\|\boldsymbol{\omega}(t) - \tilde{\boldsymbol{\theta}}_j(t)\|$ is bounded by $C' t^{-\alpha/2}$ for t large enough, and so is $(1/n)\sum_{j=1}^n \mathbb{E}_T\left[\frac{\delta_j(t)}{p_j} g_j(t)^\top (\boldsymbol{\theta}_j(t) - \boldsymbol{\omega}(t))\right]$.

To control the term in (5.67) we use that f_j is L_f-Lipschitz

$$\begin{aligned}|f_j(\boldsymbol{\theta}_i(t)) - f_j(\boldsymbol{\theta}_j(t)| \leq & L_f \|\boldsymbol{\theta}_i(t) - \boldsymbol{\theta}_j(t)\| \\ \leq & L_f(\|\boldsymbol{\theta}_i(t) - \boldsymbol{\omega}(t)\| + \|\boldsymbol{\omega}(t) - \boldsymbol{\theta}_j(t)\|).\end{aligned}$$

and we use now the same control as for (5.72), hence the result.

□

Lemma 18. Let $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ be a non-increasing positive function and let $\mathbf{z} \in \mathbb{R}^d$. For any $t_1, t_2 > 0$, one has

$$\begin{aligned}\|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\| \leq & \|\mathbf{z}\||\gamma(t_2) - \gamma(t_1)| \\ & + \|\mathbf{z}\| \left(\frac{3}{2} + \max(\frac{\gamma(t_1)}{\gamma(t_2)}, \frac{\gamma(t_2)}{\gamma(t_1)})\right)\left(\frac{1}{t_1} + \frac{1}{t_2}\right)|t_1\gamma(t_1) - t_2\gamma(t_2)|,\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{\theta}_1 = \pi_{t_1}(\mathbf{z}) &:= \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{\mathbf{z}^\top\boldsymbol{\theta} - \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t_1)} - t_1\psi(\boldsymbol{\theta})\right\} \\ \boldsymbol{\theta}_2 = \pi_{t_2}(\mathbf{z}) &:= \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{\mathbf{z}^\top\boldsymbol{\theta} - \frac{\|\boldsymbol{\theta}\|^2}{2\gamma(t_2)} - t_2\psi(\boldsymbol{\theta})\right\}.\end{aligned}$$

Proof. Using the optimality property of the minimizers, for any $s_1 \in \partial\psi(\boldsymbol{\theta}_1)$ (resp. $s_2 \in \partial\psi(\boldsymbol{\theta}_2)$):

$$\begin{aligned}(\gamma(t_1)\mathbf{z} - t_1\gamma(t_1)s_1 - \boldsymbol{\theta}_1)^\top(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \leq 0 \\ (\gamma(t_2)\mathbf{z} - t_2\gamma(t_2)s_2 - \boldsymbol{\theta}_2)^\top(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \leq 0\end{aligned}$$

Re-arranging the terms, and using properties of sub-gradients yields:

$$\begin{aligned}\|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|^2 \leq & (\gamma(t_2) - \gamma(t_1))\mathbf{z}^\top(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) + (t_1\gamma(t_1)s_1 - t_2\gamma(t_2)s_2)^\top(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) \\ \leq & (\gamma(t_2) - \gamma(t_1))\mathbf{z}^\top(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1) + (t_1\gamma(t_1) - t_2\gamma(t_2))(\psi(\boldsymbol{\theta}_2) - \psi(\boldsymbol{\theta}_1))\end{aligned} \tag{5.73}$$

Also, using the definition of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, one has:

$$|\psi(\boldsymbol{\theta}_1) - \psi(\boldsymbol{\theta}_1)| \leq \|\mathbf{z}\|\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\| \left(\frac{3}{2} + \max(\frac{\gamma(t_1)}{\gamma(t_2)}, \frac{\gamma(t_2)}{\gamma(t_1)})\right)\left(\frac{1}{t_1} + \frac{1}{t_2}\right). \tag{5.74}$$

With relations (5.73) and (5.74) we bound the distance between $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ as follows:

$$\begin{aligned}\|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\| \leq & \|\mathbf{z}\||\gamma(t_2) - \gamma(t_1)| \\ & + \|\mathbf{z}\| \left(\frac{3}{2} + \max(\frac{\gamma(t_1)}{\gamma(t_2)}, \frac{\gamma(t_2)}{\gamma(t_1)})\right) \left(\frac{1}{t_1} + \frac{1}{t_2}\right) |t_1\gamma(t_1) - t_2\gamma(t_2)|\end{aligned}$$

□

Conclusion

In a wide variety of statistical learning problems, U-statistics are natural estimates of the risk measure one seeks to optimize. As the sizes of the samples increase, the computation of such functionals involves summing a rapidly exploding number of terms and becomes numerically unfeasible. In this work, we tackled several problems that were only explored in the case of a sample mean empirical risk. First, we provided theoretical guarantees for different sampling schemes for U-statistic-based ERM, showing that sampling according to the incomplete U-statistic scheme can preserve the learning rate while involving far fewer terms than the naive complete U-statistic from a subsample. Furthermore, we have extended these results to U-statistics based on different sampling schemes (Bernoulli sampling, sampling without replacement) and shown how such functionals can be used for the purpose of model selection and for implementing ERM iterative procedures based on stochastic gradient descent. Beyond theoretical rate bounds, the efficiency of the approach we promote is illustrated by several numerical experiments. Then, we tackled problem of the decentralized estimation. We introduced a new gossip algorithm for both synchronous and fully asynchronous setting for estimating U-statistics. We have shown that the expected convergence rate outperforms the state of the art and numerical experiments confirmed the practical interest of the proposed algorithm. Finally, we introduced synchronous and asynchronous algorithms for optimizing convex objectives depending on pairs of observations. The proposed methods are based on dual averaging and can readily accommodate various popular regularization terms. We provided an analysis showing that they behave similarly to the centralized dual averaging algorithm, with additional terms reflecting the network connectivity and the gradient bias. The numerical experiments on AUC maximization and metric learning illustrated the performance of the proposed algorithms, as well as the influence of network topology.

In future work, one could investigate the possibility of adaptive communication schemes, in order to speed-up to convergence for both decentralized estimation and optimization. One could also extend asynchronous decentralized algorithms to the case where nodes do not know their relative degree, estimating it on the run and using an ergodic analysis to show the convergence.

Publication List

Journal

JMLR [3]	Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics.

Conference

NIPS [4] 2016	Decentralized Topic Modelling with Latent Dirichlet Allocation.
ICML [5] 2016	Gossip Dual Averaging for Decentralized Optimization of Pairwise Functions.
NIPS[4] 2015	Extending Gossip Algorithms to Distributed Estimation of U-statistics.

3. Journal of Machine Learning Research
4. Advances in Neural Information Processing Systems
5. International Conference on Machine Learning

Bibliography

AGARWAL, Alekh, Martin J WAINWRIGHT, and John C DUCHI (2010). "Distributed dual averaging in networks". In: *Advances in Neural Information Processing Systems*, pp. 550–558.

BEKKERMAN, R., M. BILENKO, and J. LANGFORD (2011). *Scaling Up Machine Learning*. Cambridge.

BELLET, A. and A. HABRARD (2015). "Robustness and Generalization for Metric Learning". In: *Neurocomputing* 151.1, pp. 259–267.

BELLET, A., A. HABRARD, and M. SEBBAN (2013). "A Survey on Metric Learning for Feature Vectors and Structured Data". In: *ArXiv e-prints*.

BELLET, Aurélien, Amaury HABRARD, and Marc SEBBAN (2015). *Metric Learning*. Morgan & Claypool.

BERTAIL, P. and J. TRESSOU (2006). "Incomplete generalized U-Statistics for food risk assessment". In: *Biometrics* 62.1, pp. 66–74.

BIANCHI, P. et al. (2013). "On-Line Learning Gossip Algorithm in Multi-Agent Systems with Local Decision Rules". In: *Proceedings of the IEEE International Conference on Big Data*.

BIANCHI, Pascal and Jérémie JAKUBOWICZ (2013). "Convergence of a Multi-Agent Projected Stochastic Gradient Algorithm for Non-Convex Optimization". In: *IEEE Trans. Autom. Control* 58.2, pp. 391–405.

BIAU, Gérard and Kevin BLEAKLEY (2006). "Statistical Inference on Graphs". In: *Statistics & Decisions* 24, pp. 209–232.

BLOM, G. (1976). "Some properties of incomplete U-statistics". In: *Biometrika* 63.3, pp. 573–580.

BOLLOBÁS, Béla (1998). *Modern Graph Theory*. Vol. 184. Springer.

BOTTOU, L. (1998). *Online Algorithms and Stochastic Approximations: Online Learning and Neural Networks*. Cambridge University Press.

BOUCHERON, S., O. BOUSQUET, and G. LUGOSI (2005). "Theory of Classification: A Survey of Some Recent Advances". In: *ESAIM: Probability and Statistics* 9, pp. 323–375.

BOYD, Stephen et al. (2006). "Randomized gossip algorithms". In: *IEEE Trans. Inf. Theory* 52.6, pp. 2508–2530.

BRÉMAUD, Pierre (1999). *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Vol. 31. Springer Science & Business Media.

BROWN, B. M. and D. G. KILDEA (1978). "Reduced U-statistics and the Hodges-Lehmann estimator". In: *The Annals of Statistics* 6, pp. 828–835.

CAO, Q., Z.-C. GUO, and Y. YING (2012). *Generalization Bounds for Metric and Similarity Learning*. Tech. rep. arXiv:1207.5437. University of Exeter.

CHECHIK, G. et al. (2010). "Large Scale Online Learning of Image Similarity Through Ranking". In: *Journal of Machine Learning Research* 11, pp. 1109–1135.

CHUNG, Fan (1997). *Spectral Graph Theory*. Vol. 92. Amer. Math. Soc.

CLÉMENÇON, S. (2014). "A statistical view of clustering performance through the theory of U-processes". In: *Journal of Multivariate Analysis* 124, pp. 42–56.

CLÉMENÇON, S., G. LUGOSI, and N. VAYATIS (2005). "Ranking and scoring using empirical risk minimization". In: *Proceedings of COLT*.

— (2008). "Ranking and empirical risk minimization of U-statistics". In: *The Annals of Statistics* 36.2, pp. 844–874.

CLÉMENÇON, S. and S. ROBBIANO (2014). "Building confidence regions for the ROC surface". In: *To appear in Pattern Recognition Letters*.

CLÉMENÇON, S., S. ROBBIANO, and N. VAYATIS (2013). "Ranking Data with Ordinal Labels: Optimality and Pairwise Aggregation". In: *Machine Learning* 91.1, pp. 67–104.

CLÉMENÇON, S. and N. VAYATIS (2009). "Tree-based ranking methods". In: *IEEE Transactions on Information Theory* 55.9, pp. 4316–4336.

CLÉMENÇON, Stéphan (2011). "On U-processes and clustering performance". In: *NIPS*, pp. 37–45.

CLÉMENÇON, Stéphan, Gàbor LUGOSI, and Nicolas VAYATIS (2008). "Ranking and Empirical Minimization of U-statistics". In: *Ann. Stat.* 36.2, pp. 844–874.

COCHRAN, W. G. (1977). *Sampling techniques*. Wiley, NY.

DEVILLE, J. C. (1987). "Réplications d'échantillons, demi-échantillons, Jackknife, bootstrap dans les sondages". In: *Les Sondages*. Economica et al.

DEVROYE, L., L. GYÖRFI, and G. LUGOSI (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.

DIMAKIS, Alexandros G., Anand D. SARWATE, and Martin J. WAINWRIGHT (2008). "Geographic Gossip: Efficient Averaging for Sensor Networks". In: *IEEE Transactions on Signal Processing* 56.3, pp. 1205–1216.

DIMAKIS, Alexandros G. et al. (2010). "Gossip Algorithms for Distributed Signal Processing". In: *Proceedings of the IEEE* 98.11, pp. 1847–1864.

DUCHI, John, Alekh AGARWAL, and Martin WAINWRIGHT (2012). "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling". In: *IEEE Trans. Autom. Control* 57.3, pp. 592–606.

DUCHI, John C et al. (2012). "Ergodic mirror descent". In: *SIAM Journal on Optimization* 22.4, pp. 1549–1578.

DUDLEY, Richard M. (2010). "Distances of probability measures and random variables". In: *Selected Works of RM Dudley*, pp. 28–37.

ENQVIST, E. (1978). "On sampling from sets of random variables with application to incomplete U-statistics". PhD thesis. Lund University.

FIEDLER, Miroslav (1973). "Algebra connectivity of graphs". In: *Czechoslovake Mathematical Journal* 23.98, pp. 298–305.

FRIEDMAN, J., T. HASTIE, and R. TIBSHIRANI (2009). *The Elements of Statistical Learning*. Springer.

FUK, D. K. and S. V. NAGAEV (1971). "Probability Ineaqualities for Sums of Independent Random Variables". In: *Prob. Th. Appl.* 16.4, 643Ð660.

GINÉ, E. and J. ZINN (1984). "Some limit theorems for empirical processes". In: *The Annals of Probability* 12.4, pp. 929–989.

GRAMS, W. and R. SERFLING (1973). "Convergence rates for U-statistics and related statistics". In: *Ann. Stat.* 1.1, pp. 153–160.

HÁJEK, J. (1964). "Asymptotic theory of rejective sampling with varying probabilities from a finite population". In: *The Annals of Mathematical Statistics* 35.4, pp. 1491–1523.

— (1968). "Asymptotic normality of simple linear rank statistics under alternatives". In: *Ann. Math. Stat.* 39, pp. 325–346.

HANLEY, James A. and Barbara J. MCNEIL (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." In: *Radiology* 143.1, pp. 29–36.

HEDETNIEMI, Sandra M, Stephen T HEDETNIEMI, and Arthur L LIESTMAN (1988). "A survey of gossiping and broadcasting in communication networks". In: *Networks* 18.4, pp. 319–349.

HOEFFDING, W. (1948). "A class of statistics with asymptotically normal distribution". In: *Ann. Math. Stat.* 19, pp. 293–325.

HORVITZ, D. G. and D. J. THOMPSON (1951). "A generalization of sampling without replacement from a finite universe". In: *JASA* 47, pp. 663–685.

IUTZELER, Franck et al. (2013). "Asynchronous Distributed Optimization using a Randomized Alternating Direction Method of Multipliers". In: *IEEE CDC*, pp. 3671–3676.

JANSON, S. (1984). "The asymptotic distributions of Incomplete U-statistics". In: *Z. Wahrsch. verw. Gebiete* 66, pp. 495–505.

JIN, R., S. WANG, and Y. ZHOU (2009). "Regularized Distance Metric Learning: Theory and Algorithm". In: *Advances in Neural Information Processing Systems 22*, pp. 862–870.

JOHANSSON, Björn, Maben RABI, and Mikael JOHANSSON (2010). "A Randomized Incremental Subgradient Method for Distributed Optimization in Networked Systems". In: *SIAM J. Optimiz.* 20.3, pp. 1157–1170.

JOHNSON, R. and T. ZHANG (2013). "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: *Advances in Neural Information Processing Systems 26*, pp. 315–323.

KANUNGO, Tapas et al. (2004). "A local search approximation algorithm for k-means clustering". In: *Computational Geometry* 28.2–3, pp. 89–112.

KAR, Soummya and José MF MOURA (2009). "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise". In: *IEEE Transactions on Signal Processing* 57.1, pp. 355–369.

KARP, Richard et al. (2000). "Randomized rumor spreading". In: *Symposium on Foundations of Computer Science*. IEEE, pp. 565–574.

KEMPE, David, Alin DOBRA, and Johannes GEHRKE (2003). "Gossip-Based Computation of Aggregate Information". In: *FOCS*, pp. 482–491.

KOWALCZYK, Wojtek and Nikos A. VLASSIS (2004). "Newscast EM". In: *NIPS*, pp. 713–720.

KUMAR, Abhishek et al. (2012). "A Binary Classification Framework for Two-Stage Multiple Kernel Learning". In: *ICML*.

LE ROUX, N., M. W. SCHMIDT, and F. BACH (2012). "A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets". In: *Advances in Neural Information Processing Systems 25*, pp. 2672–2680.

LEDOUX, M. and M. TALAGRAND (1991). *Probability in Banach Spaces*. New York: Springer.

LEE, A. J. (1990a). *U-statistics: Theory and practice*. New York: Marcel Dekker, Inc.

LEE, Alan J. (1990b). *U-Statistics: Theory and Practice*. Marcel Dekker, New York.

LEE, Soomin and Angelia NEDIĆ (2015). "Asynchronous gossip-based random projection algorithms over networks". In: *IEEE Trans. Autom. Control*.

LEE, Soomin, Angelia NEDIĆ, and Maxim RAGINSKY (2015). "Decentralized Online Optimization with Global Objectives and Local Communication". In: *arXiv preprint arXiv:1508.07933*.

LI, Wenjun, Huaiyu DAI, and Yanbing ZHANG (2010). "Location-Aided Fast Distributed Consensus in Wireless Networks". In: *IEEE Transactions on Information Theory* 56.12, pp. 6208–6227.

LOIZOU, Nicolas and Peter RICHTÁRIK (2016). "A NEW PERSPECTIVE ON RANDOMIZED GOSSIP ALGORITHMS". In:

LUGOSI, G. (2002). "Pattern Classification and learning theory". In: *Principles of Nonparametric Learning*. Ed. by L. GYÖRFI. Springer, NY, pp. 1–56.

MANN, Henry B. and Donald R. WHITNEY (1947). "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other". In: *Annals of Mathematical Statistics* 18.1, pp. 50–60.

MASSART, P. (2006). "Concentration inequalities and model selection". In: Lecture Notes in Mathematics. Springer.

MCDIARMID, C. (1989). "On the method of bounded differences". In: *Surveys in Combinatorics*. Cambridge Univ. Press, pp. 148–188.

MOSK-AOYAMA, Damon and Devavrat SHAH (2008). "Fast Distributed Algorithms for Computing Separable Functions". In: *IEEE Trans. Inf. Theory* 54.7, pp. 2997–3007.

NEDIĆ, Angelia (2011). "Asynchronous broadcast-based convex optimization over a network". In: *Automatic Control, IEEE Transactions on* 56.6, pp. 1337–1351.

NEDIC, Angelia and Asuman OZDAGLAR (2009). "Distributed subgradient methods for multi-agent optimization". In: *IEEE Transactions on Automatic Control* 54.1, pp. 48–61.

NEDIĆ, Angelia and Asuman E. OZDAGLAR (2009). "Distributed Subgradient Methods for Multi-Agent Optimization". In: *IEEE Trans. Autom. Control* 54.1, pp. 48–61.

NESTEROV, Yurii (2009). "Primal-dual subgradient methods for convex problems". In: *Math. Program.* 120.1, pp. 261–283.

PAPA, Guillaume, Stéphan CLÉMENÇON, and Aurélien BELLET (2015). "SGD Algorithms based on Incomplete U-statistics: Large-Scale Minimization of Empirical Risk". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. CORTES et al. Curran Associates, Inc., pp. 1027–1035.

PEDREGOSA, Fabian et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

PELCKMANS, Kristiaan and Johan SUYKENS (2009). "Gossip Algorithms for Computing U-Statistics". In: *NecSys*, pp. 48–53.

PEÑA, V. de la and E. GINÉ (1999). *Decoupling: from Dependence to Independence*. Springer.

RAM, S., Angelia NEDIĆ, and V. VEERAVALLI (2010). "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization". In: *J. Optimiz. Theory. App.* 147.3, pp. 516–545.

SERFLING, R. J. (1974). "Probability inequalities for the sum in sampling without replacement". In: *The Annals of Statistics* 2.1, pp. 39–48.

— (1980). *Approximation theorems of mathematical statistics*. Wiley.

SHAH, Devavrat (2009). "Gossip Algorithms". In: *Foundations and Trends in Networking* 3.1, pp. 1–125.

TILLÉ, Y. (2006). *Sampling algorithms*. Springer Series in Statistics.

TSIANOS, Konstantinos, Sean LAWLOR, and Michael RABBAT (2015). "Push-Sum Distributed Dual Averaging for convex optimization". In: *IEEE CDC*.

TSITSIKLIS, John (1984). "Problems in decentralized decision making and computation". PhD thesis. Massachusetts Institute of Technology.

VAPNIK, V. N. (1999). "An Overview of Statistical Learning Theory". In: *IEEE Transactions on Neural Networks* 10.5, pp. 988–999.

WARD, Joe H. (1963). "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58.301, pp. 236–244.

WATTS, Duncan J and Steven H STROGATZ (1998). "Collective dynamics of 'small-world'networks". In: *Nature* 393.6684, pp. 440–442.

WEI, Ermin and Asuman OZDAGLAR (2012). "Distributed Alternating Direction Method of Multipliers". In: *IEEE CDC*, pp. 5445–5450.

— (2013). "On the O(1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers". In: *IEEE GlobalSIP*.

WEINBERGER, K. Q. and L. K. SAUL (2009). "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *Journal of Machine Learning Research* 10, pp. 207–244.

XIAO, Lin (2009). "Dual averaging method for regularized stochastic learning and online optimization". In: *NIPS*, pp. 2116–2124.

— (2010). "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization". In: *JMLR* 11, pp. 2543–2596. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=141578.

YUAN, Deming et al. (2012). "Distributed dual averaging method for multi-agent optimization with quantized communication". In: *Systems & Control Letters* 61.11, pp. 1053–1061.

ZHAO, Peilin et al. (2011). "AUC Maximization". In: *ICML*.

Adaptation des méthodes d'apprentissage aux U-statistiques

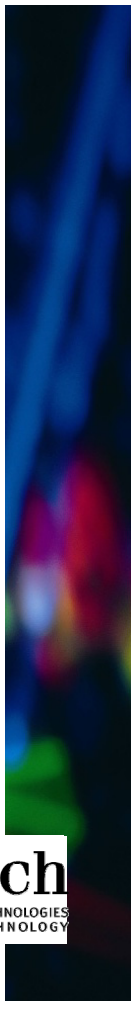
Igor COLIN

RESUME : L'explosion récente des volumes de données disponibles a fait de la complexité algorithmique un élément central des méthodes d'apprentissage automatique. Les algorithmes d'optimisation stochastique ainsi que les méthodes distribuées et décentralisées ont été largement développés durant les dix dernières années. Ces méthodes ont permis de faciliter le passage à l'échelle pour optimiser des risques empiriques dont la formulation est séparable en les observations associées. Pourtant, dans de nombreux problèmes d'apprentissage statistique, l'estimation précise du risque s'effectue à l'aide de U-statistiques, des fonctions des données prenant la forme de moyennes sur des d-uplets. Nous nous intéressons tout d'abord au problème de l'échantillonage pour la minimisation du risque empirique. Nous montrons que le risque peut être remplacé par un estimateur de Monte-Carlo, intitulé U-statistique incomplète, basé sur seulement $O(n)$ termes et permettant de conserver un taux d'apprentissage du même ordre. Nous établissons des bornes sur l'erreur d'approximation du U-processus et les simulations numériques mettent en évidence l'avantage d'une telle technique d'échantillonnage. Nous portons par la suite notre attention sur l'estimation décentralisée, où les observations sont désormais distribuées sur un réseau connexe. Nous élaborons des algorithmes dits *gossip*, dans des cadres synchrones et asynchrones, qui diffusent les observations tout en maintenant des estimateurs locaux de la U-statistique à estimer. Nous démontrons la convergence de ces algorithmes avec des dépendances explicites en les données et la topologie du réseau. Enfin, nous traîtons de l'optimisation décentralisée de fonctions dépendant de paires d'observations. De même que pour l'estimation, nos méthodes sont basées sur la concomitance de la propagation des observations et l'optimisation local du risque. Notre analyse théorique souligne que ces méthodes conservent une vitesse de convergence du même ordre que dans le cas centralisé. Les expériences numériques confirment l'intérêt pratique de notre approche.

MOTS-CLEFS : U-statistique, gossip, optimisation décentralisée, graphe

ABSTRACT : With the increasing availability of large amounts of data, computational complexity has become a keystone of many machine learning algorithms. Stochastic optimization algorithms and distributed/decentralized methods have been widely studied over the last decade and provide increased scalability for optimizing an empirical risk that is separable in the data sample. Yet, in a wide range of statistical learning problems, the risk is accurately estimated by U-statistics, *i.e.*, functionals of the training data with low variance that take the form of averages over d-tuples. We first tackle the problem of sampling for the empirical risk minimization problem. We show that empirical risks can be replaced by drastically computationally simpler Monte-Carlo estimates based on $O(n)$ terms only, usually referred to as incomplete U-statistics, without damaging the learning rate. We establish uniform deviation results and numerical examples show that such approach surpasses more naive subsampling techniques. We then focus on the decentralized estimation topic, where the data sample is distributed over a connected network. We introduce new synchronous and asynchronous randomized gossip algorithms which simultaneously propagate data across the network and maintain local estimates of the U-statistic of interest. We establish convergence rate bounds with explicit data and network dependent terms. Finally, we deal with the decentralized optimization of functions that depend on pairs of observations. Similarly to the estimation case, we introduce a method based on concurrent local updates and data propagation. Our theoretical analysis reveals that the proposed algorithms preserve the convergence rate of centralized dual averaging up to an additive bias term. Our simulations illustrate the practical interest of our approach.

KEY-WORDS : U-statistic, gossip, decentralized optimization, graph



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY