Trade-offs in Large-Scale Distributed Tuplewise Estimation and Learning

Robin Vogel[1,2] Aurélien Bellet[3] Stephan Clémençon[1]
Ons Jelassi[1] Guillaume Papa[1]

[1] Telecom Paris, [2] IDEMIA, [3] Inria

ECML-PKDD 2019

Outline

Large scale distributed data processing

Very large datasets are common nowadays in ML.
$\rightarrow$ Distribute the data in **partitions** over several machines.

Cluster computing frameworks:

· Abstract network and communication aspects of distribution. ☺

· Restrict the types of operations efficiently achieved. ☹

E.g. **Apache Spark** [Meng et al., 2016], **Petuum** [Xing et al., 2015], ...

Most ML techniques optimize **standard means** $\hat{L} = \sum_i \ell(x_i)/n$,
Those are **separable across partitions**.
$\rightarrow$ One can **efficiently estimate** those.

Very common statistics - e.g. *U*-**statistics** - are not.
$\rightarrow$ Estimation can be **slow or inaccurate**.

Our contribution

Problem:
When a statistic is **not separable across partitions**,
frameworks may be **unsuited to computing accurate estimators.**

Contribution: Quantified analysis for *U*-statistics,

1. **Efficient estimators** of *U*-statistics in a distributed setting.
2. The analysis of their **accuracy-vs-time tradeoff**.
3. Learning experiment with those as **gradient estimators**.

Plan:

- Properties of *U*-statistics, see [Hoeffding, 1948],
- Contributions 1-3.

Outline

U-statistics: definition and examples

Introduce K independent i.i.d. samples $\mathcal{D}_k = \{X_1^{(k)}, \dots, X_{n_k}^{(k)}\} \subset \mathcal{X}_k$, and the **kernel** $h : \mathcal{X}_1^{d_1} \times \cdots \times \mathcal{X}_K^{d_K} \to \mathbb{R}$ with $h \in L^2$.

The **generalized** U**-statistic** of degrees $(d_1, \dots, d_K)$ is defined as:

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

where $\sum_{I_k}$ is the sum over all $\binom{n_k}{d_k}$ subsets of d_k elements of $\mathcal{D}_k$.

Examples:

- a sample variance $h(x_1, x_2) = (x_1 - x_2)^2$,
- Kendall's tau $h((x_1, y_1), (x_2, y_2)) = \mathbb{I}\{(x_1 - x_2) \cdot (y_1 - y_2) > 0\}$,
- **clustering**, **metric learning** and **ranking** criterions.

Here: study quantitatively a common case in a distributed setting.

Two-sample U-statistics

Introduce, with $m \ll n$:

- the **abundant positive** i.i.d. sample $\mathcal{D}_n = \{X_1, \ldots, X_n\} \subset \mathcal{X}$,
- the **scarce negative** i.i.d. sample $\mathcal{Q}_m = \{Z_1, \ldots, Z_m\} \subset \mathcal{Z}$,
- a **kernel** $h : \mathcal{X} \times \mathcal{Z} \mapsto \mathbb{R}$.

Objective: Estimate $U(h) = \mathbb{E}[h(X_1, Z_1)]$.

With $\mathbf{n} = (n, m)$, the **U-statistic** $U_{\mathbf{n}}$, where:

$$U_{\mathbf{n}} := \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} h(X_i, Z_j),$$

is the **unbiased** estimator of $U(h)$ with **lowest variance**.

$U_{\mathbf{n}}$ is an average of nm elements $\rightarrow$ **What if** $n = 10^9$**,** $m = 10^3$**?**

One answer is **incomplete** U**-statistics** [Clémençon et al., 2016],

$$\widetilde{U}_B := \frac{1}{B} \sum_{(i,j) \in \mathcal{D}_B} h(X_i, Z_j),$$

where $\mathcal{D}_B$ is a set of B elements sampled WR in $\{(i, j)\}_{i \in [\![n]\!], j \in [\![m]\!]}$.

Properties of two-sample U-statistics

With the **Hajèk projections** $h_1(x) = \mathbb{E}[h(x, Z_1)], h_2(z) = \mathbb{E}[h(X_1, z)]$,

$$\text{and} \quad h_0(x,z) = h(x,z) - h_1(x) - h_2(z) + U(h),$$

then $U_{\mathbf{n}} = T_1 + T_2 + W_0 - U$ with

$$T_1 = \frac{1}{n}\sum_{i=1}^{n} h_1(X_i) \text{ and } T_2 = \frac{1}{m}\sum_{j=1}^{m} h_2(Z_j) \text{ and } W_0 = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} h_0(X_i, Z_j),$$

which is called the **second Hoeffding decomposition**.

A U-statistic $U_{\mathbf{n}}$ is called **degenerate** when $h_1 = U$ and $h_2 = U$ *a.s.*.

Introducing $\sigma_1^2 = \text{Var}(h_1(X)), \sigma_2^2 = \text{Var}(h_2(Z)), \sigma_0^2 = \text{Var}(h_0(X_1, Z_1))$:

$$\text{Var}(U_{\mathbf{n}}) = \frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m} + \frac{\sigma_0^2}{nm}.$$

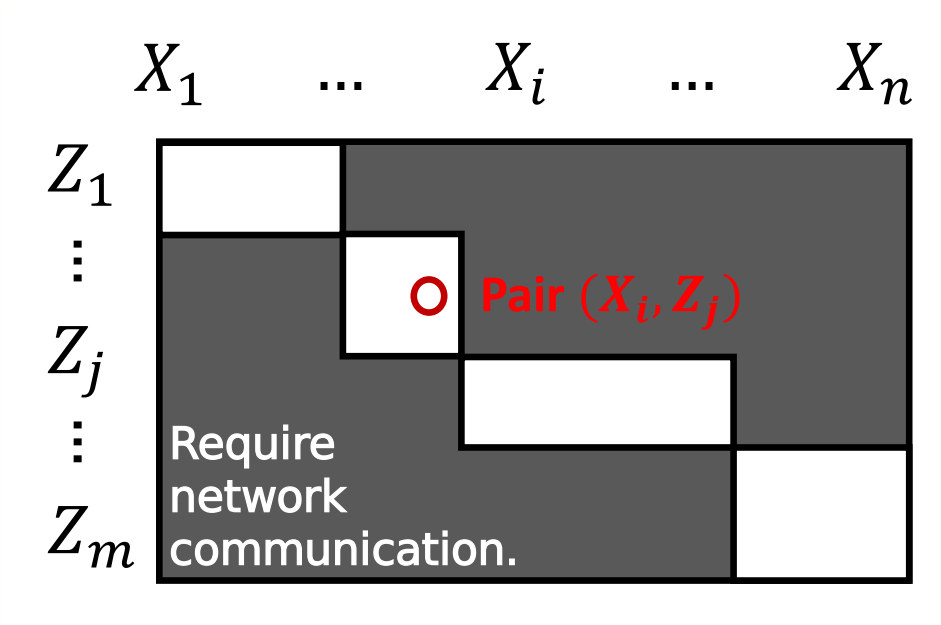
Examples with $X_1, Z_1 \sim \mathcal{U}[-1, 1]$**:**

$h(x,z) = x + z$ gives $\sigma_0^2 = 0$. | $h(x,z) = x \cdot z$ gives $\sigma_1^2, \sigma_2^2 = 0$.

Outline

Distributed data

We distribute the data on N workers.



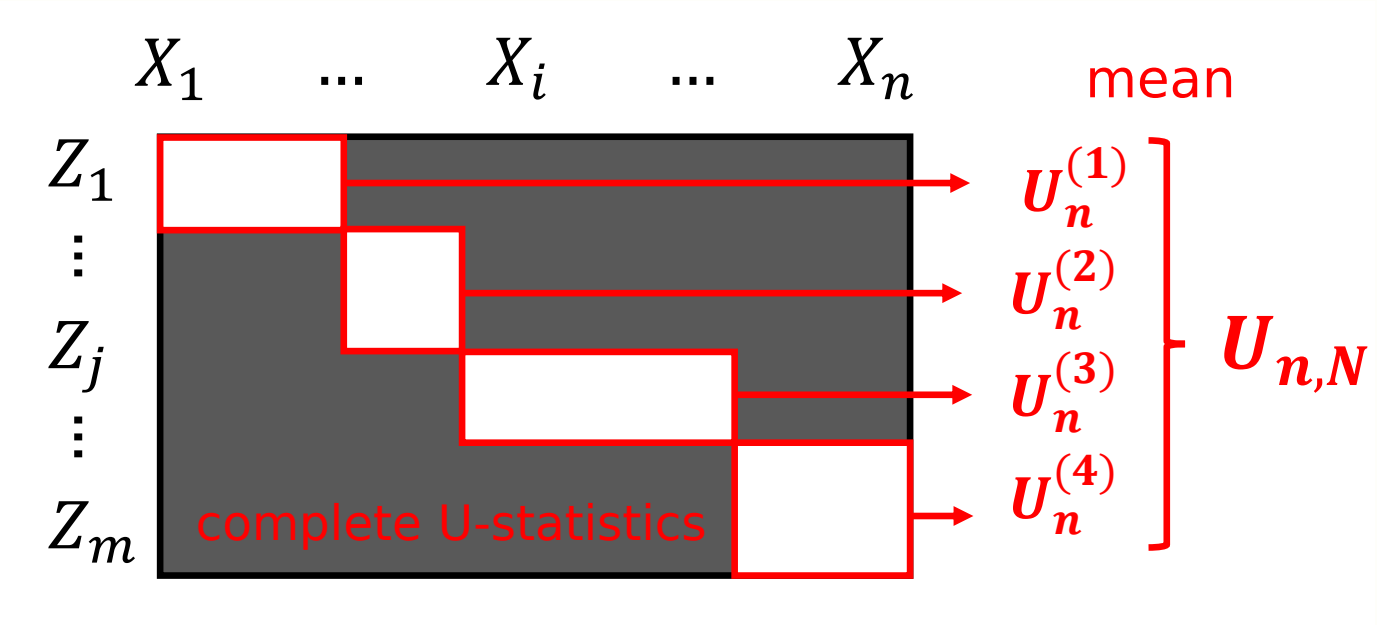
One can distribute the instances:

- with **SWOR** (**SWR**): sampling without (with) replacement,
- **proportionally**: each cluster contains n/N instances from $\mathcal{D}_n$, and m/M from $\mathcal{Q}_m$.

Here: Focus on proportional SWOR (see paper for others).

Naive estimators (1/2)

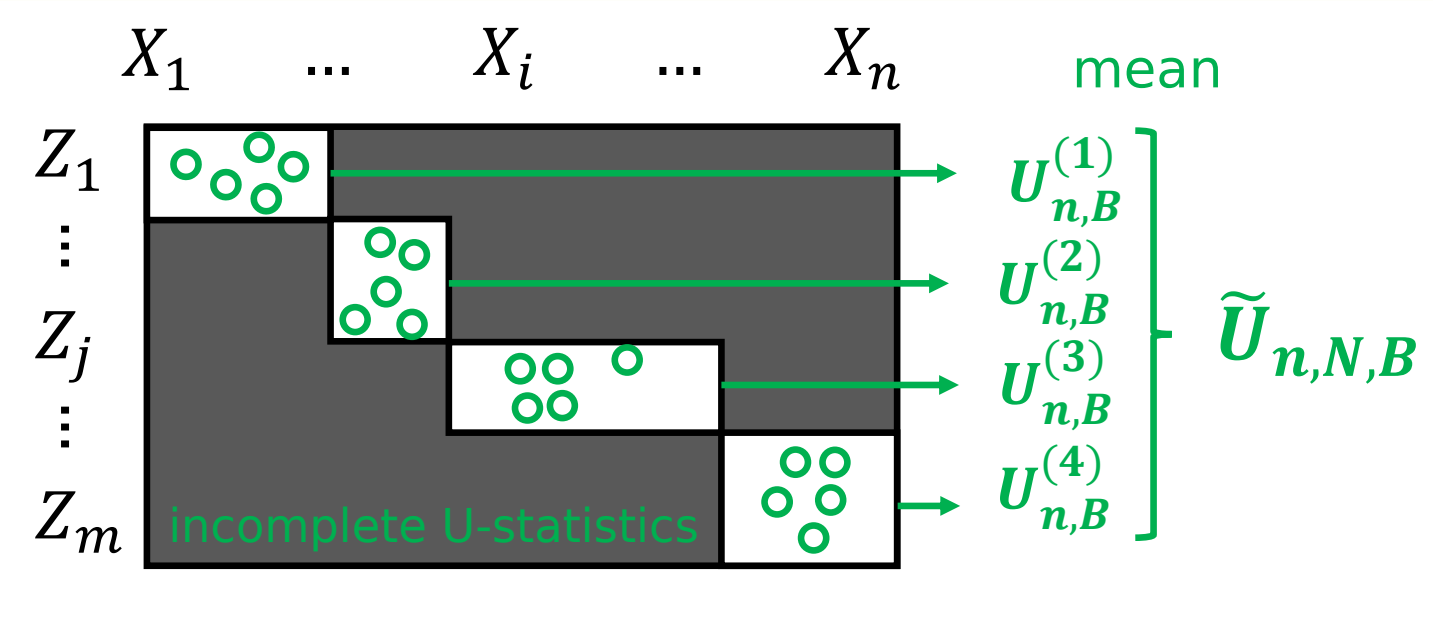
Averaging full *U*-statistics from each cluster gives $U_{\mathbf{n},N}$.



Proposed statistics: $U_{\mathbf{n},N}$

Naive estimators (2/2)

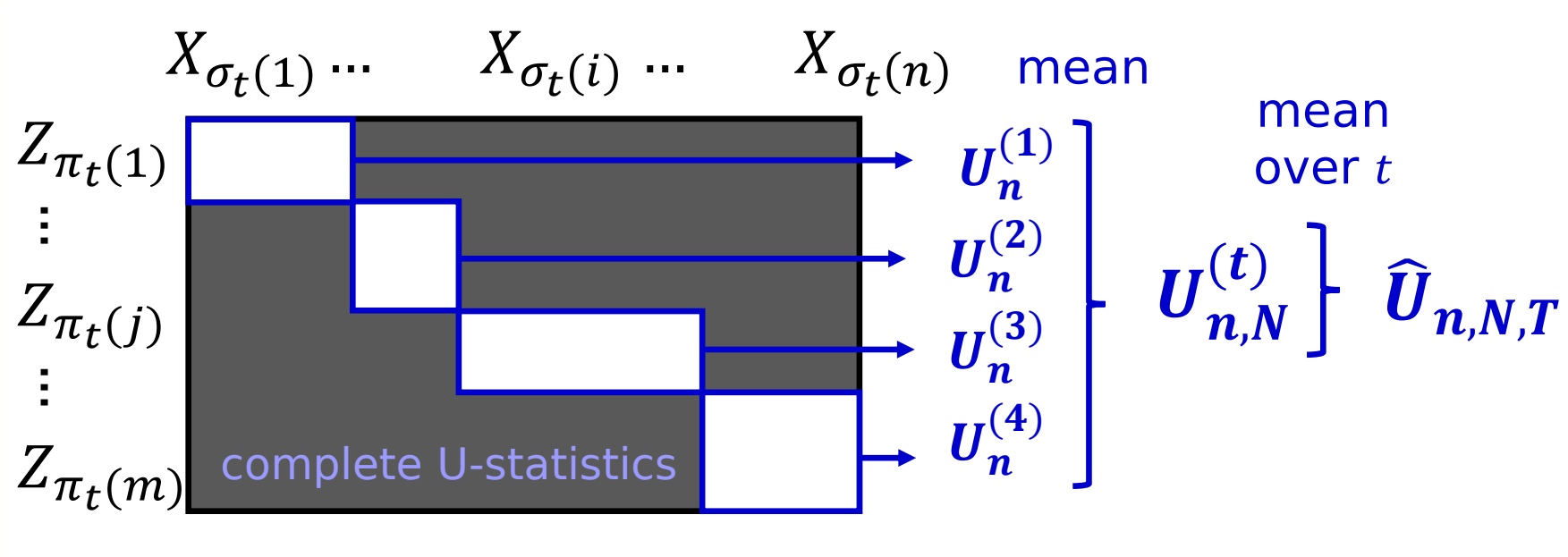
Averaging B pairs SWR from each cluster gives $\widetilde{U}_{\mathbf{n},N,B}$.



Proposed statistics: $U_{\mathbf{n},N}$, $\widetilde{U}_{\mathbf{n},N,B}$,

Estimators with redistribution (1/2)

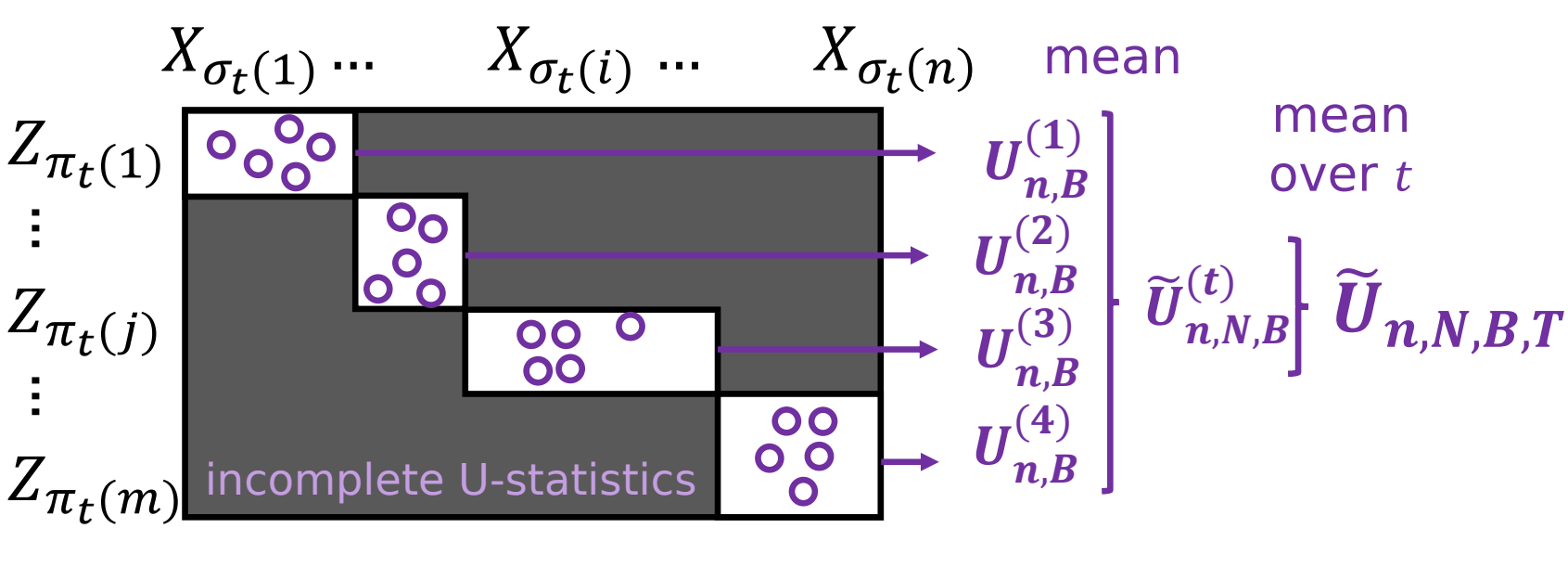
Averaging $U_{\mathbf{n},N}$ on T redistributions of the data gives $\widehat{U}_{\mathbf{n},N,T}$.



Proposed estimators: $U_{\mathbf{n},N}$, $\widetilde{U}_{\mathbf{n},N,B}$, $\widehat{U}_{\mathbf{n},N,T}$.
With σ_t, π_t random permutations at time t.

Estimators with redistribution (2/2)

Averaging $\widetilde{U}_{\mathbf{n},N,B}$ on T redistributions of the data gives $\widetilde{U}_{\mathbf{n},N,B,T}$.



Proposed estimators: $U_{\mathbf{n},N}$, $\widetilde{U}_{\mathbf{n},N,B}$, $\widehat{U}_{\mathbf{n},N,T}$, and $\widetilde{U}_{\mathbf{n},N,B,T}$.
With σ_t, π_t random permutations at time t.

All of the estimators are **unbiased**.

Variance expressions

We have: Variances in **closed form**.

For the **naive estimators**:

$$\mathrm{Var}(U_{\mathbf{n},N}) = \mathrm{Var}(U_{\mathbf{n}}) + (N-1)\frac{\sigma_0^2}{nm},$$
$$\mathrm{Var}(\widetilde{U}_{\mathbf{n},N,B}) = \left(1 - \frac{1}{B}\right)\mathrm{Var}(U_{\mathbf{n},N}) + \frac{\sigma^2}{NB}.$$

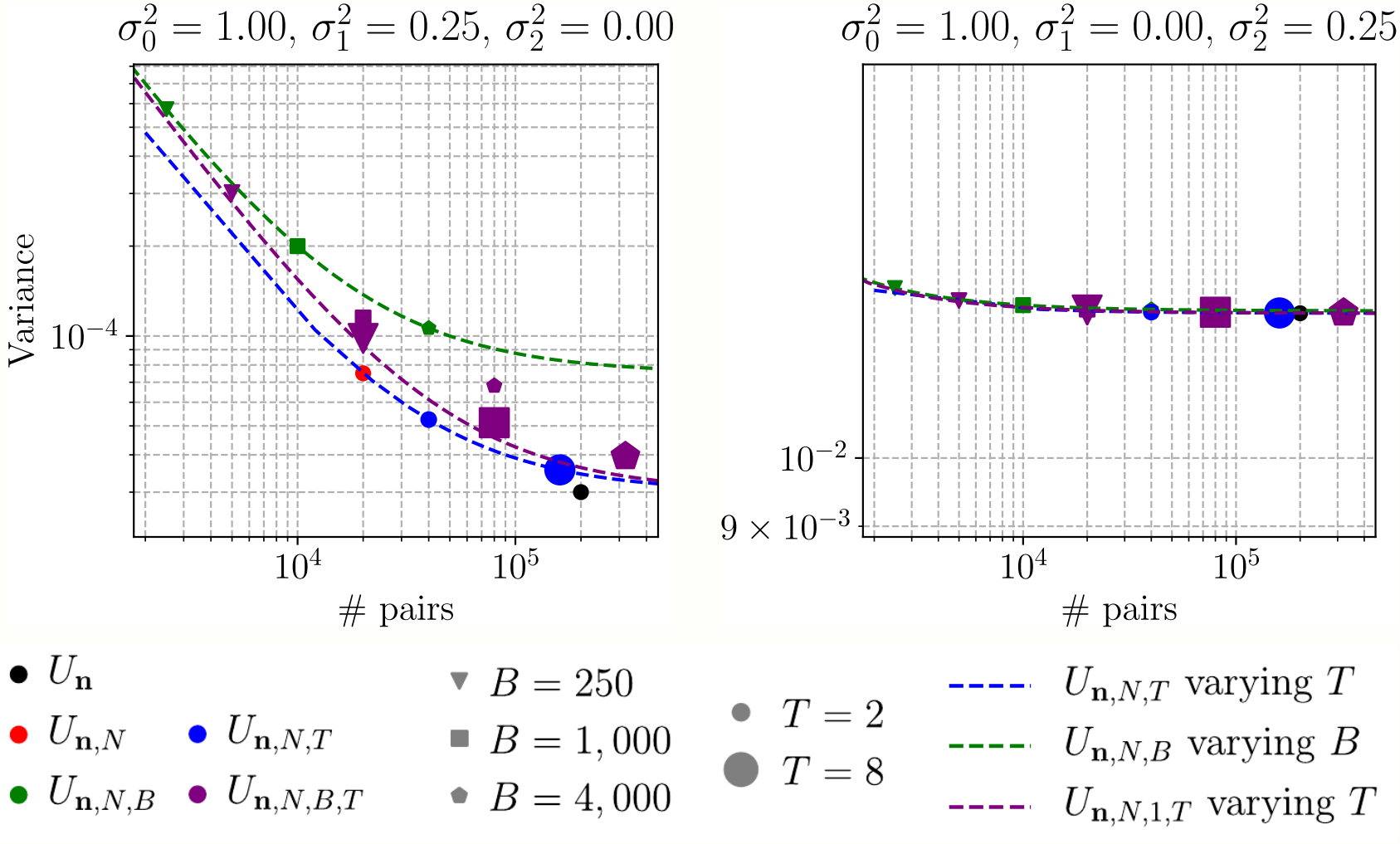
$\rightarrow$ Term in $N\sigma_0^2/nm$.

For the **estimators with redistribution**:

$$\mathrm{Var}(\widehat{U}_{\mathbf{n},N,T}) = \mathrm{Var}(U_{\mathbf{n}}) + (N-1)\frac{\sigma_0^2}{nmT},$$
$$\mathrm{Var}(\widetilde{U}_{\mathbf{n},N,B,T}) = \mathrm{Var}(\widehat{U}_{\mathbf{n},N,T}) - \frac{1}{TB}\mathrm{Var}(U_{\mathbf{n},N}) + \frac{\sigma^2}{NTB}.$$

$\rightarrow$ Term in $N\sigma_0^2/Tnm$.

Illustrations of the variances



Conclusions:

· Redistribution is useful when $\sigma_2^2 \ll \sigma_0^2$,

· For any same # of pairs, $\widetilde{U}_{\mathbf{n},N,B,T}$ is worse than $\widehat{U}_{\mathbf{n},N,T}$.

Outline

Optimize the convexified AUC

Objective: optimize the AUC using a its convex upper-bound:

$$U_n(h_{\text{AUC}}) := \frac{1}{nm} \sum_{i,j} \mathbb{I}\{w^\top (X_i - Z_j) > 0\},$$
$$\leq \frac{1}{nm} \sum_{i,j} \left[1 + w^\top (X_i - Z_j)\right]_+ =: U_n(h_{\text{conv}}).$$

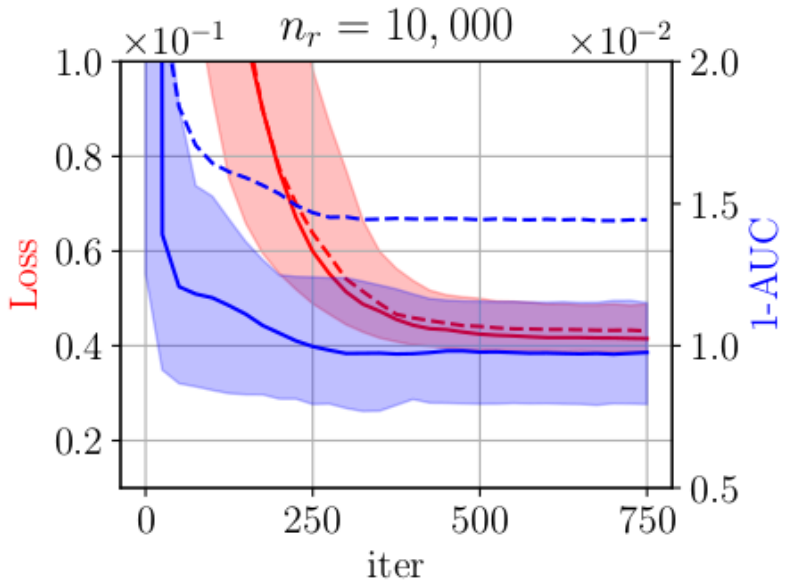
Optimizer: Momentum BGD with LR 10^{-2}, mom. 0.9 for 5×10^3 iter.

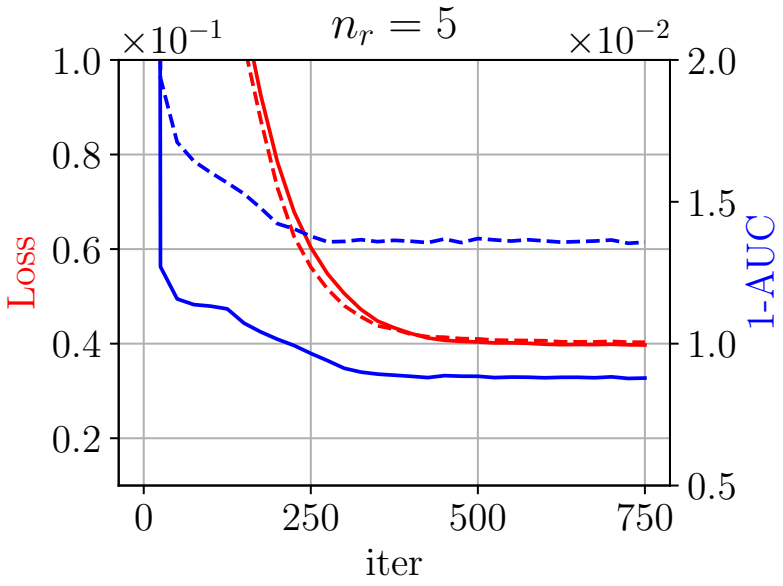
Loss: $U_{\mathbf{n},N,B}(h_{\text{conv}}) + \lambda \|w\|_2^2$, where we redistribute each n_r iter.

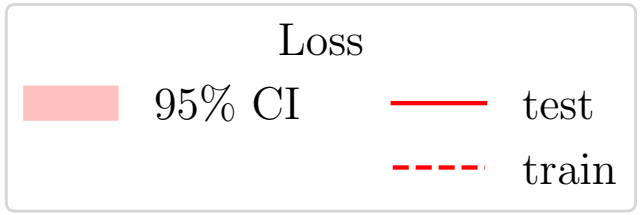
Parameters: $N = 100, B = 500, \lambda = 0.01, n_r \in \{1, 5, 25, +\infty\}$.

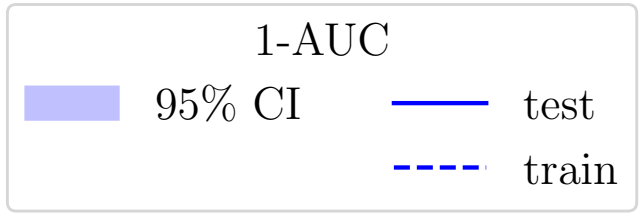
Dataset: Shuttle (outlier dataset), $n = 45,000$, $m = 3,500$,
20% as test set, train monitored on a fixed set of $450K$ pairs,
see [Rayana, 2016].

Results



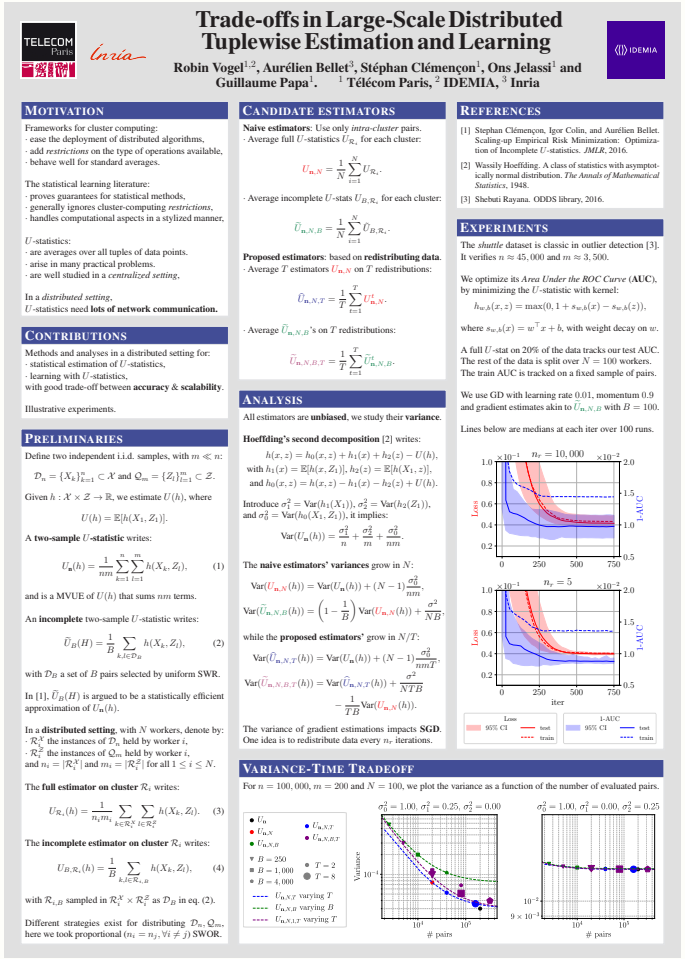






- Not redistributing $\rightarrow$ end performance very noisy ☹
- Redistributing $\rightarrow$ better performance (with high. prob.) ☺

Danke, please come see us at the poster session !



Trade-offs in Large-Scale Distributed Tuplewise Estimation and Learning

Robin Vogel[1,2], Aurélien Bellet[3], Stéphan Clémençon[1], Ons Jelassi[1] and Guillaume Papa[1]. [1] Télécom Paris, [2] IDEMIA, [3] Inria

Motivation

Frameworks for cluster computing:
· ease the deployment of distributed algorithms,
· add *restrictions* on the type of operations available,
· behave well for standard averages.

The statistical learning literature:
· proves guarantees for statistical methods,
· generally ignores cluster-computing *restrictions*,
· handles computational aspects in a stylized manner,

U-statistics:
· are averages over all tuples of data points.
· arise in many practical problems,
· are well studied in a *centralized setting*,

In a *distributed setting*, U-statistics need **lots of network communication.**

Contributions

Methods and analyses in a distributed setting for:
· statistical estimation of U-statistics,
· learning with U-statistics,
with good trade-off between **accuracy** & **scalability**.

Illustrative experiments.

Preliminaries

Define two independent i.i.d. samples, with $m \ll n$:

$$\mathcal{D}_n = \{X_k\}_{k=1}^n \subset \mathcal{X} \text{ and } \mathcal{Q}_m = \{Z_l\}_{l=1}^m \subset \mathcal{Z}.$$

Given $h : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$, we estimate $U(h)$, where

$$U(h) = \mathbb{E}[h(X_1, Z_1)].$$

A **two-sample U-statistic** writes:

$$U_{\mathbf{n}}(h) = \frac{1}{nm} \sum_{k=1}^{n} \sum_{l=1}^{m} h(X_k, Z_l), \quad (1)$$

and is a MVUE of $U(h)$ that sums nm terms.

An **incomplete** two-sample U-statistic writes:

$$\widetilde{U}_B(H) = \frac{1}{B} \sum_{k,l \in \mathcal{D}_B} h(X_k, Z_l), \quad (2)$$

with $\mathcal{D}_B$ a set of B pairs selected by uniform SWR.

In [1], $\widetilde{U}_B(H)$ is argued to be a statistically efficient approximation of $U_{\mathbf{n}}(h)$.

In a **distributed setting**, with N workers, denote by:
· $\mathcal{R}_i^X$ the instances of $\mathcal{D}_n$ held by worker i,
· $\mathcal{R}_i^Z$ the instances of $\mathcal{Q}_m$ held by worker i,
and $n_i = |\mathcal{R}_i^X|$ and $m_i = |\mathcal{R}_i^Z|$ for all $1 \leq i \leq N$.

The **full estimator on cluster** $\mathcal{R}_i$ writes:

$$U_{\mathcal{R}_i}(h) = \frac{1}{n_i m_i} \sum_{k \in \mathcal{R}_i^X} \sum_{l \in \mathcal{R}_i^Z} h(X_k, Z_l). \quad (3)$$

The **incomplete estimator on cluster** $\mathcal{R}_i$ writes:

$$U_{B,\mathcal{R}_i}(h) = \frac{1}{B} \sum_{k,l \in \mathcal{R}_{i,B}} h(X_k, Z_l), \quad (4)$$

with $\mathcal{R}_{i,B}$ sampled in $\mathcal{R}_i^X \times \mathcal{R}_i^Z$ as $\mathcal{D}_B$ in eq. (2).

Different strategies exist for distributing $\mathcal{D}_n, \mathcal{Q}_m$, here we took proportional ($n_i = n_j, \forall i \neq j$) SWOR.

Candidate estimators

Naive estimators: Use only *intra-cluster* pairs.
· Average full U-statistics $U_{\mathcal{R}_i}$ for each cluster:

$$U_{\mathbf{n},N} = \frac{1}{N} \sum_{i=1}^{N} U_{\mathcal{R}_i}.$$

· Average incomplete U-stats $U_{B,\mathcal{R}_i}$ for each cluster:

$$\widetilde{U}_{\mathbf{n},N,B} = \frac{1}{N} \sum_{i=1}^{N} \widetilde{U}_{B,\mathcal{R}_i}.$$

Proposed estimators: based on **redistributing data**.
· Average T estimators $U_{\mathbf{n},N}$ on T redistributions:

$$\widetilde{U}_{\mathbf{n},N,T} = \frac{1}{T} \sum_{t=1}^{T} U_{\mathbf{n},N}^t.$$

· Average $\widetilde{U}_{\mathbf{n},N,B}$'s on T redistributions:

$$\widetilde{U}_{\mathbf{n},N,B,T} = \frac{1}{T} \sum_{t=1}^{T} \widetilde{U}_{\mathbf{n},N,B}^t.$$

Analysis

All estimators are **unbiased**, we study their **variance**.

Hoeffding's second decomposition [2] writes:

$$h(x,z) = h_0(x,z) + h_1(x) + h_2(z) - U(h),$$
with $h_1(x) = \mathbb{E}[h(x, Z_1)]$, $h_2(z) = \mathbb{E}[h(X_1, z)]$,
and $h_0(x,z) = h(x,z) - h_1(x) - h_2(z) + U(h)$.

Introduce $\sigma_1^2 = \text{Var}(h_1(X_1))$, $\sigma_2^2 = \text{Var}(h_2(Z_1))$, and $\sigma_0^2 = \text{Var}(h_0(X_1, Z_1))$, it implies:

$$\text{Var}(U_{\mathbf{n}}(h)) = \frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m} + \frac{\sigma_0^2}{nm}.$$

The **naive estimators' variances** grow in N:

$$\text{Var}(U_{\mathbf{n},N}(h)) = \text{Var}(U_{\mathbf{n}}(h)) + (N-1)\frac{\sigma_0^2}{nm},$$

$$\text{Var}(\widetilde{U}_{\mathbf{n},N,B}(h)) = \left(1 - \frac{1}{B}\right)\text{Var}(U_{\mathbf{n},N}(h)) + \frac{\sigma^2}{NB},$$

while the **proposed estimators'** grow in N/T:

$$\text{Var}(\widetilde{U}_{\mathbf{n},N,T}(h)) = \text{Var}(U_{\mathbf{n}}(h)) + (N-1)\frac{\sigma_0^2}{nmT},$$

$$\text{Var}(\widetilde{U}_{\mathbf{n},N,B,T}(h)) = \text{Var}(\widetilde{U}_{\mathbf{n},N,T}(h)) + \frac{\sigma^2}{NTB} - \frac{1}{TB}\text{Var}(U_{\mathbf{n},N}(h)).$$

The variance of gradient estimations impacts **SGD**. One idea is to redistribute data every n_r iterations.

Variance-Time Tradeoff

For $n = 100,000$, $m = 200$ and $N = 100$, we plot the variance as a function of the number of evaluated pairs.

References

[1] Stephan Clémençon, Igor Colin, and Aurélien Bellet. Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics. *JMLR*, 2016.

[2] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 1948.

[3] Shebuti Rayana. ODDS library, 2016.

Experiments

The *shuttle* dataset is classic in outlier detection [3]. It verifies $n \approx 45,000$ and $m \approx 3,500$.

We optimize its *Area Under the ROC Curve* (**AUC**), by minimizing the U-statistic with kernel:

$$h_{w,b}(x,z) = \max(0, 1 + s_{w,b}(x) - s_{w,b}(z)),$$

where $s_{w,b}(x) = w^\top x + b$, with weight decay on w.

A full U-stat on 20% of the data tracks our test AUC. The rest of the data is split over $N = 100$ workers. The train AUC is tracked on a fixed sample of pairs.

We use GD with learning rate 0.01, momentum 0.9 and gradient estimates akin to $\widetilde{U}_{\mathbf{n},N,B}$ with $B = 100$.

Lines below are medians at each iter over 100 runs.

References

Clémençon, S., Colin, I., and Bellet, A. (2016).
Scaling-up Empirical Risk Minimization: Optimization of Incomplete *U*-statistics.
Journal of Machine Learning Research, 17(76):1–36.

Hoeffding, W. (1948).
A class of statistics with asymptotically normal distribution.
The Annals of Mathematical Statistics, 19:293–325.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M. J., Zadeh, R., Zaharia, M., and Talwalkar, A. (2016).
MLlib: Machine Learning in Apache Spark.
Journal of Machine Learning Research, 17(34):1–7.

Rayana, S. (2016).
Odds library.

Xing, E. P., Ho, Q., Dai, W., Kim, J. K., Wei, J., Lee, S., Zheng, X., Xie, P., Kumar, A., and Yu, Y. (2015).
Petuum: A New Platform for Distributed Machine Learning on Big Data.
IEEE Transactions on Big Data, 1(2):49–67.