Pack only the essentials: distributed sequential sampling for adaptive kernel DL

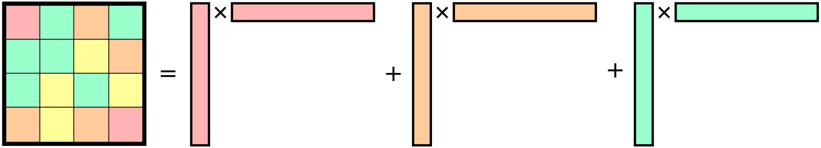
with **Daniele Calandriello** and **Alessandro Lazaric**

SequeL team, Inria Lille - Nord Europe, France

appeared in AISTATS 2017

Distributed sequential sampling for adaptive kernel DL

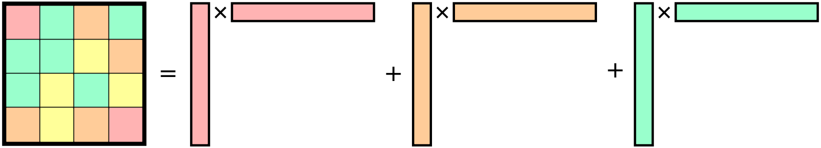
What is Dictionary Learning (DL)?



Finding an accurate representation of the input data as a linear combination of a small set of basic elements (atoms)

Distributed sequential sampling for adaptive kernel DL

What is Dictionary Learning (DL)?

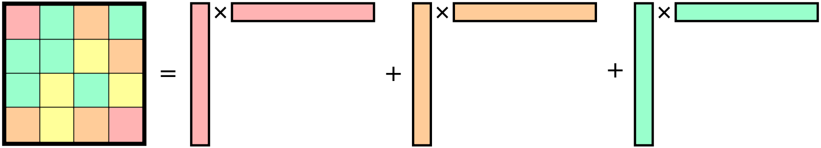


Finding an accurate representation of the input data as a linear combination of a small set of basic elements (atoms)

Representation/Unsupervised learning

Distributed sequential sampling for adaptive kernel DL

What is Dictionary Learning (DL)?



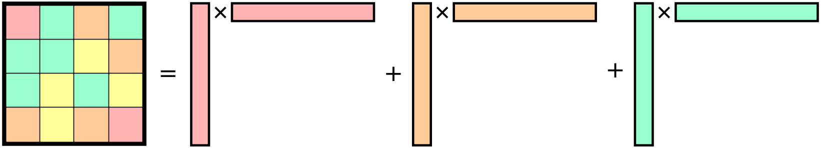
Finding an accurate representation of the input data as a linear combination of a small set of basic elements (atoms)

Representation/Unsupervised learning

> "Most important open problem in ML" **Y. LeCun**, NIPS 2016

Distributed sequential sampling for adaptive kernel DL

What is Dictionary Learning (DL)?



Finding an accurate representation of the input data as a linear combination of a small set of basic elements (atoms)

Representation/Unsupervised learning

“Most important open problem in ML” **Y. LeCun**, NIPS 2016

“Already solved” **J. Schmidhuber**, NIPS 2016

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Kernel methods have huge scalability problem

Problem: for a dataset $\mathcal{D}$ with n samples

- $\mathcal{O}(\mathbf{n^2})$ time to construct kernel matrix $\mathbf{K}$
- $\mathcal{O}(\mathbf{n^3})$ time to compute solution
- $\mathcal{O}(\mathbf{n^2})$ space to store it

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Kernel methods have huge scalability problem

Problem: for a dataset $\mathcal{D}$ with n samples

- $\mathcal{O}(\mathbf{n^2})$ time to construct kernel matrix $\mathbf{K}$
- $\mathcal{O}(\mathbf{n^3})$ time to compute solution
- $\mathcal{O}(\mathbf{n^2})$ space to store it

Solution:
compute accurate, small dictionary $\mathcal{I}$ to represent $\mathcal{D}$
compute approximate solution on $\mathcal{I}$ efficiently

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalablility or accuracy

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalablility or accuracy

we want both

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalablility or accuracy

we want both

We present SQUEAK — a dictionary learning algorithm that guarantees

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalablility or accuracy

we want both

We present SQUEAK — a dictionary learning algorithm that guarantees

In all cases accurate reconstruction of the input

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalablility or accuracy

we want both

We present SQUEAK — a dictionary learning algorithm that guarantees

- In all cases accurate reconstruction of the input
- Adapts to the data:
  - on "easy" problems small $\mathcal{O}(n)$ space/time requirements
  - on "hard" problems not worse than storing whole input

Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalablility or accuracy

we want both

We present SQUEAK — a dictionary learning algorithm that guarantees

- In all cases accurate reconstruction of the input
- Adapts to the data:
  - on "easy" problems small $\mathcal{O}(n)$ space/time requirements
  - on "hard" problems not worse than storing whole input
- Only local data access, distributed version with $\mathcal{O}(\log(n))$ runtime

Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

$$\mathbf{A} = \mathbf{A}^{1/2}(\mathbf{A}^{1/2})^{\mathsf{T}} = \sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^{\mathsf{T}} \qquad \widetilde{\mathbf{A}} = \sum_{i=1}^{m} w_i \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}$$

Method	w_i	$\mathbf{x}_i$	Accuracy	Space	Time
Whole Input	1	$\mathbf{a}_i$	★★★★★		

Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

$$\mathbf{A} = \mathbf{A}^{1/2}(\mathbf{A}^{1/2})^\mathsf{T} = \sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^\mathsf{T} \qquad \widetilde{\mathbf{A}} = \sum_{i=1}^{m} w_i \mathbf{x}_i \mathbf{x}_i^\mathsf{T}$$

Method	w_i	$\mathbf{x}_i$	Accuracy	Space	Time
Whole Input	1	$\mathbf{a}_i$	★★★★★		
Empty dictionary	0	$\mathbf{0}$		★★★★★	★★★★★

Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

$$\mathbf{A} = \mathbf{A}^{1/2}(\mathbf{A}^{1/2})^\mathsf{T} = \sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^\mathsf{T} \qquad \widetilde{\mathbf{A}} = \sum_{i=1}^{m} w_i \mathbf{x}_i \mathbf{x}_i^\mathsf{T}$$

Method	w_i	$\mathbf{x}_i$	Accuracy	Space	Time
Whole Input	1	$\mathbf{a}_i$	★★★★★		
PCA	λ_i	$\mathbf{u}_i$	★★★★★	★★★★★	★
Empty dictionary	0	$\mathbf{0}$		★★★★★	★★★★★

Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

$$\mathbf{A} = \mathbf{A}^{1/2}(\mathbf{A}^{1/2})^\mathsf{T} = \sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^\mathsf{T} \qquad \widetilde{\mathbf{A}} = \sum_{i=1}^{m} w_i \mathbf{x}_i \mathbf{x}_i^\mathsf{T}$$

Method	w_i	$\mathbf{x}_i$	Accuracy	Space	Time
Whole Input	1	$\mathbf{a}_i$	★★★★★		
PCA	λ_i	$\mathbf{u}_i$	★★★★★	★★★★★	★
RLS (this)	$1/\tau_i$	$\mathbf{a}_i$	★★★★	★★★★	★★★
Empty dictionary	0	$\mathbf{0}$		★★★★★	★★★★★

Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

$$\mathbf{A} = \mathbf{A}^{1/2}(\mathbf{A}^{1/2})^\mathsf{T} = \sum_{i=1}^{n} \mathbf{a}_i\mathbf{a}_i^\mathsf{T} \qquad \widetilde{\mathbf{A}} = \sum_{i=1}^{m} w_i\mathbf{x}_i\mathbf{x}_i^\mathsf{T}$$

Method	w_i	$\mathbf{x}_i$	Accuracy	Space	Time
Whole Input	1	$\mathbf{a}_i$	★★★★★		
PCA	λ_i	$\mathbf{u}_i$	★★★★★	★★★★★	★
RLS (this)	$1/\tau_i$	$\mathbf{a}_i$	★★★★	★★★★	★★★
Uniform	n/m	$\mathbf{a}_i$	★★	★★	★★★★
Empty dictionary	0	$\mathbf{0}$		★★★★★	★★★★★

Preliminaries: Setting and Kernels

Indexing $[t] = \{1, \dots, t\}$, notation $\mathbf{K}$ matrices, $\mathbf{k}$ vectors, k scalar

Dataset $\mathcal{D}_n = \{\mathbf{x}_i\}_{i=1}^n$, samples $\mathbf{x}_i \in \mathcal{X}$ (e.g., $\mathbb{R}^d$)

Kernel function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

Feature map $\varphi(\mathbf{x}_i) : \mathcal{X} \to \mathcal{H} = \phi_i$

Kernel trick
$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathcal{K}(\mathbf{x}_i, \cdot), \mathcal{K}(\mathbf{x}_j, \cdot) \rangle_{\mathcal{H}} = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle_{\mathcal{H}} = \phi_i^\mathsf{T} \phi_j$

Feature matrix $\mathbf{\Phi}_t = [\phi_1, \phi_2, \dots, \phi_t] : \mathbb{R}^t \to \mathcal{H}$

Empirical kernel matrix $\mathbf{K}_t \in \mathbb{R}^{t \times t} = \mathbf{K}_{[t],[t]} = \mathbf{\Phi}_t^\mathsf{T} \mathbf{\Phi}_t$

New column $\mathbf{k}_{[t-1],t} \in \mathbb{R}^{t-1} = \mathbf{\Phi}_{t-1}^\mathsf{T} \phi_t$

Kernel at a point $k_{t,t} \in \mathbb{R} = \phi_t^\mathsf{T} \phi_t$

Find a dictionary $\mathcal{I} = \{(w_j, \phi_j)\}_{j=1}^m$ such that $\widetilde{\boldsymbol{K}} = f(\mathcal{I})$ close to $\mathbf{K}$

Preliminaries: Linear Algebra

(Full) Singular Value Decomposition $\mathbf{\Phi} = \mathbf{V\Sigma U}^\mathsf{T}$, $\mathbf{\Sigma}$ rectangular
Eigendecomposition $\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi} = \mathbf{U\Sigma}^\mathsf{T}\mathbf{\Sigma U}^\mathsf{T} = \mathbf{U\Lambda U}^\mathsf{T} = \mathbf{K}$

Matrix norms (if omitted, ℓ-2 norm)

$$\ell\text{-2 norm} \quad \|\mathbf{A}\|_2 = \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = \max \lambda_i$$

$$\text{Frobenius norm} \quad \|\mathbf{A}\|_F^2 = \sum a_{i,j}^2 = \sum \lambda_i^2$$

Useful equality for arbitrary $n \times m$ matrix (or operator)

$$\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}(\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} + \gamma \mathbf{I}_n)^{-1} = \mathbf{\Phi}(\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi} + \gamma \mathbf{I}_m)^{-1}\mathbf{\Phi}^\mathsf{T}$$

Example: Kernel Ridge Regression

$$\widehat{w}_n = (\mathbf{K}_n + \gamma \mathbf{I})^{-1} \mathbf{y}_n$$
$$\widehat{y}_n = \mathbf{K}_n \widehat{w}_n = \mathbf{K}_n (\mathbf{K}_n + \gamma \mathbf{I})^{-1} \mathbf{y}_n = \mathbf{P}_n \mathbf{y}_n$$

Example: Kernel Ridge Regression

$$\widehat{w}_n = (\mathbf{K}_n + \gamma\mathbf{I})^{-1}\mathbf{y}_n$$
$$\widehat{y}_n = \mathbf{K}_n\widehat{w}_n = \mathbf{K}_n(\mathbf{K}_n + \gamma\mathbf{I})^{-1}\mathbf{y}_n = \mathbf{P}_n\mathbf{y}_n$$

If we can have accurate low-rank approximations ...

$$\widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

Example: Kernel Ridge Regression

$$\widehat{w}_n = (\mathbf{K}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n$$
$$\widehat{y}_n = \mathbf{K}_n\widehat{w}_n = \mathbf{K}_n(\mathbf{K}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n = \mathbf{P}_n\mathbf{y}_n$$

If we can have accurate low-rank approximations ...

$$\widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

...then we can used them for to get good approximate solutions:

$$\widetilde{w}_n = (\widetilde{\boldsymbol{K}}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n$$
$$R(\widetilde{w}_n) \leq \left(1 + \frac{1}{1-\varepsilon}\right) R(\widehat{w}_n)$$

Example: Kernel Ridge Regression

$$\widehat{w}_n = (\mathbf{K}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n$$
$$\widehat{y}_n = \mathbf{K}_n \widehat{w}_n = \mathbf{K}_n(\mathbf{K}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n = \mathbf{P}_n \mathbf{y}_n$$

If we can have accurate low-rank approximations ...

$$\widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

... then we can used them for to get good approximate solutions:

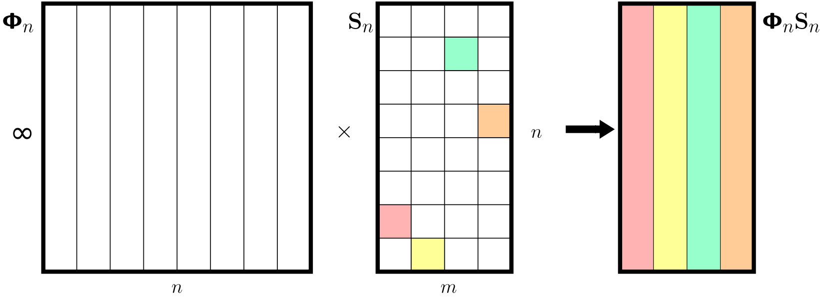
$$\widetilde{w}_n = (\widetilde{\boldsymbol{K}}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n$$
$$R(\widetilde{w}_n) \leq \left(1 + \frac{1}{1-\varepsilon}\right) R(\widehat{w}_n)$$

$\cancel{\mathcal{O}(\mathbf{n^3})} \Rightarrow \mathcal{O}(\mathbf{nm} + \mathbf{m^3})$ time to compute the approx. solution

$\cancel{\mathcal{O}(n^2)} \Rightarrow \mathcal{O}(\mathbf{nm})$ space to store dictionary

Reconstruction Guarantees

Given dataset $\mathcal{D}_n$ and dictionary $\mathcal{I}_n$, the selection matrix $\mathbf{S}_n$ is defined as



$$\sum_{i=1}^{m} w_i \phi_i \phi_i^{\mathsf{T}} = \sum_{i=1}^{m} (\sqrt{w_i}\phi_i)(\sqrt{w_i}\phi_i)^{\mathsf{T}} = \mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}}$$

Reconstruction guarantees

Consider the regularized projection $\mathbf{\Psi}_n$

$$\mathbf{\Psi}_n = \mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1} = (\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}$$

$$= \sum_{i=1}^{n}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\phi_i\phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2} = \sum_{i=1}^{n}\psi_i\psi_i^\mathsf{T}$$

$$\widetilde{\mathbf{\Psi}}_n = (\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n\mathbf{S}_n\mathbf{S}_n^\mathsf{T}\mathbf{\Phi}_n^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2} = \sum_{j=1}^{m} w_j\psi_j\psi_j^\mathsf{T}$$

Reconstruction guarantees

Consider the regularized projection $\mathbf{\Psi}_n$

$$\mathbf{\Psi}_n = \mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}}(\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1} = (\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}}(\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1/2}$$
$$= \sum_{i=1}^{n}(\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1/2}\phi_i\phi_i^{\mathsf{T}}(\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1/2} = \sum_{i=1}^{n}\psi_i\psi_i^{\mathsf{T}}$$
$$\widetilde{\mathbf{\Psi}}_n = (\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n\mathbf{S}_n\mathbf{S}_n^{\mathsf{T}}\mathbf{\Phi}_n^{\mathsf{T}}(\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1/2} = \sum_{j=1}^{m}w_j\psi_j\psi_j^{\mathsf{T}}$$

An accurate dictionary satisfies

$$\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2 \leq \varepsilon$$

Reconstruction guarantees

Consider the regularized projection $\mathbf{\Psi}_n$

$$\mathbf{\Psi}_n = \mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1} = (\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}$$
$$= \sum_{i=1}^{n}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\phi_i\phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2} = \sum_{i=1}^{n}\psi_i\psi_i^\mathsf{T}$$
$$\widetilde{\mathbf{\Psi}}_n = (\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n\mathbf{S}_n\mathbf{S}_n^\mathsf{T}\mathbf{\Phi}_n^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2} = \sum_{j=1}^{m} w_j\psi_j\psi_j^\mathsf{T}$$

An accurate dictionary satisfies

$$\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2 \leq \varepsilon$$

equivalent to mixed additive/multiplicative error in quadratic form

$$(1-\varepsilon)\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} - \varepsilon\gamma\mathbf{I} \preceq \mathbf{\Phi}_n\mathbf{S}_n\mathbf{S}_n^\mathsf{T}\mathbf{\Phi}_n^\mathsf{T} \preceq (1+\varepsilon)\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \varepsilon\gamma\mathbf{I}$$

Reconstruction guarantees

Why would bounding $\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2$ be useful?

Reconstruction guarantees

Why would bounding $\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2$ be useful?

$$
\begin{aligned}
\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2 &= \|(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n(\mathbf{I} - \mathbf{S}_n\mathbf{S}_n^\mathsf{T})\mathbf{\Phi}_n(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\|_2 \\
&= \|(\mathbf{\Sigma}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Sigma}\mathbf{U}^\mathsf{T}(\mathbf{I} - \mathbf{S}_n\mathbf{S}_n^\mathsf{T})\mathbf{U}\mathbf{\Sigma}^\mathsf{T}(\mathbf{\Sigma}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\|_2 \\
&= \|(\mathbf{K}_n + \gamma\mathbf{I})^{-1/2}\mathbf{K}_n^{1/2}(\mathbf{I} - \mathbf{S}_n\mathbf{S}_n^\mathsf{T})\mathbf{K}_n^{1/2}(\mathbf{K}_n + \gamma\mathbf{I})^{-1/2}\|_2 \\
&= \|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2
\end{aligned}
$$

Reconstruction guarantees

Why would bounding $\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2$ be useful?

$$
\begin{aligned}
\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2 &= \|(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Phi}_n(\mathbf{I} - \mathbf{S}_n\mathbf{S}_n^\mathsf{T})\mathbf{\Phi}_n(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\|_2 \\
&= \|(\mathbf{\Sigma}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\mathbf{\Sigma}\mathbf{U}^\mathsf{T}(\mathbf{I} - \mathbf{S}_n\mathbf{S}_n^\mathsf{T})\mathbf{U}\mathbf{\Sigma}^\mathsf{T}(\mathbf{\Sigma}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I})^{-1/2}\|_2 \\
&= \|(\mathbf{K}_n + \gamma\mathbf{I})^{-1/2}\mathbf{K}_n^{1/2}(\mathbf{I} - \mathbf{S}_n\mathbf{S}_n^\mathsf{T})\mathbf{K}_n^{1/2}(\mathbf{K}_n + \gamma\mathbf{I})^{-1/2}\|_2 \\
&= \|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{P}_n &= \mathbf{K}_n(\mathbf{K}_n + \gamma\mathbf{I})^{-1} \\
\widetilde{\mathbf{P}}_n &= (\mathbf{K}_n + \gamma\mathbf{I})^{-1/2}\mathbf{K}_n^{1/2}\mathbf{S}_n\mathbf{S}_n^\mathsf{T}\mathbf{K}_n^{1/2}(\mathbf{K}_n + \gamma\mathbf{I})^{-1/2}
\end{aligned}
$$

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

It appears in many problems e.g., Kernel Ridge Regression

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

It appears in many problems e.g., Kernel Ridge Regression

$$\widehat{w}_n = (\mathbf{K}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n$$
$$\widehat{y}_n = \mathbf{K}_n\widehat{w}_n = \mathbf{K}_n(\mathbf{K}_n + \gamma \mathbf{I})^{-1}\mathbf{y}_n = \mathbf{P}_n\mathbf{y}_n$$

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\widetilde{\boldsymbol{K}}_n = \mathbf{K}_n \mathbf{S}_n (\mathbf{S}_n \mathbf{K}_n \mathbf{S}_n + \gamma \mathbf{I})^{-1} \mathbf{S}_n \mathbf{K}_n$$

then

$$\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2 \leq \varepsilon \Rightarrow \widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\widetilde{\boldsymbol{K}}_n = \mathbf{K}_n\mathbf{S}_n(\mathbf{S}_n\mathbf{K}_n\mathbf{S}_n + \gamma\mathbf{I})^{-1}\mathbf{S}_n\mathbf{K}_n$$

then

$$\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2 \leq \varepsilon \Rightarrow \widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

e.g., Kernel Ridge Regression

$$\widetilde{w}_n = (\widetilde{\boldsymbol{K}}_n + \gamma\mathbf{I})^{-1}\mathbf{y}_n$$

$$R(\widetilde{w}_n) \leq \left(1 + \frac{1}{1-\varepsilon}\right) R(\widehat{w}_n)$$

$\cancel{\mathcal{O}(\mathbf{n^3})} \Rightarrow \mathcal{O}(\mathbf{nm + m^3})$ time to compute the approx. solution

$\cancel{\mathcal{O}(n^2)} \Rightarrow \mathcal{O}(\mathbf{nm})$ space to store dictionary

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\widetilde{\boldsymbol{K}}_n = \mathbf{K}_n\mathbf{S}_n(\mathbf{S}_n\mathbf{K}_n\mathbf{S}_n + \gamma\mathbf{I})^{-1}\mathbf{S}_n\mathbf{K}_n$$

then

$$\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2 \leq \varepsilon \Rightarrow \widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

e.g., Kernel Ridge Regression*

*Gaussian Processes

$$\widetilde{w}_n = (\widetilde{\boldsymbol{K}}_n + \gamma\mathbf{I})^{-1}\mathbf{y}_n$$

$$R(\widetilde{w}_n) \leq \left(1 + \frac{1}{1-\varepsilon}\right) R(\widehat{w}_n)$$

$\cancel{\mathcal{O}(\mathbf{n^3})} \Rightarrow \mathcal{O}(\mathbf{nm} + \mathbf{m^3})$ time to compute the approx. solution

$\cancel{\mathcal{O}(n^2)} \Rightarrow \mathcal{O}(\mathbf{nm})$ space to store dictionary

Reconstruction guarantees

Why would bounding $\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\widetilde{\boldsymbol{K}}_n = \mathbf{K}_n\mathbf{S}_n(\mathbf{S}_n\mathbf{K}_n\mathbf{S}_n + \gamma\mathbf{I})^{-1}\mathbf{S}_n\mathbf{K}_n$$

then

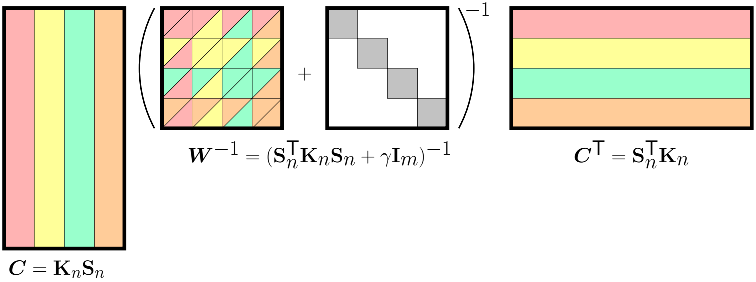
$$\|\mathbf{P}_n - \widetilde{\mathbf{P}}_n\|_2 \leq \varepsilon \Rightarrow \widetilde{\boldsymbol{K}}_n \preceq \mathbf{K}_n \preceq \widetilde{\boldsymbol{K}}_n + \frac{\gamma}{1-\varepsilon}\mathbf{I}$$

e.g., Kernel PCA, $\mathbf{K}_n$ and $\widetilde{\boldsymbol{K}}_n$ have close leading eigenvalues/vectors
e.g., Kernel K-means can be formulated as a quadratic form

$$\min_{\mathbf{C}} \mathrm{Tr}(\mathbf{K}_n - \mathbf{C}\mathbf{C}^\mathsf{T}\mathbf{K}_n\mathbf{C}\mathbf{C}^\mathsf{T}) \sim \min_{\widetilde{\mathbf{C}}} \mathrm{Tr}(\widetilde{\mathbf{K}}_n - \widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^\mathsf{T}\widetilde{\mathbf{K}}_n\widetilde{\mathbf{C}}\widetilde{\mathbf{C}}^\mathsf{T})$$

Regularized Nyström reconstruction

$$\widetilde{\mathbf{K}}_n = \mathbf{K}_n \mathbf{S}_n \left(\mathbf{S}_n^\mathsf{T} \mathbf{K}_n \mathbf{S}_n + \gamma \mathbf{I} \right)^{-1} \mathbf{S}_n^\mathsf{T} \mathbf{K}_n$$



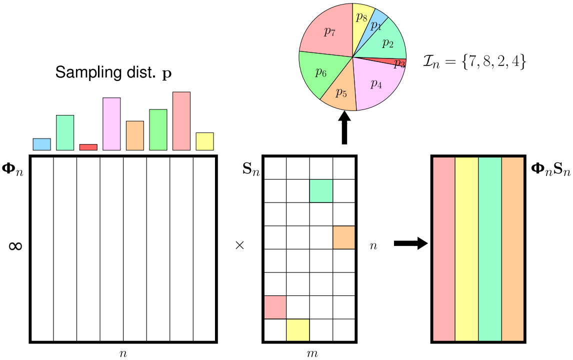
Distributed sequential sampling for adaptive kernel DL

How do we compute an accurate ($\|\boldsymbol{\Psi}_n - \widetilde{\boldsymbol{\Psi}}_n\|_2 \leq \varepsilon$) dictionary?

Distributed sequential sampling for adaptive kernel DL

How do we compute an accurate ($\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2 \leq \varepsilon$) dictionary?

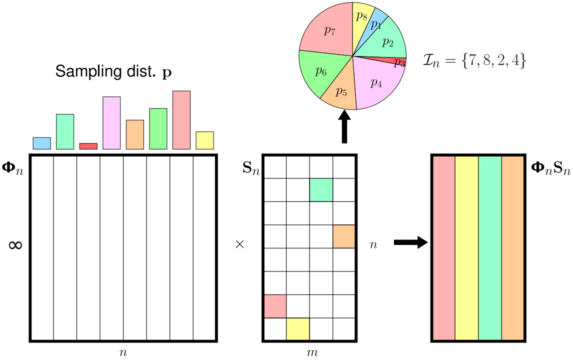
Sample m points w.p. $p_{n,i}$, add to $\mathcal{I}$ with weight $1/p_{n,i}$ (unbiased)



Distributed sequential sampling for adaptive kernel DL

How do we compute an accurate ($\|\mathbf{\Psi}_n - \widetilde{\mathbf{\Psi}}_n\|_2 \leq \varepsilon$) dictionary?

Sample m points w.p. $p_{n,i}$, add to $\mathcal{I}$ with weight $1/p_{n,i}$ (unbiased)



- ? How to choose the sampling distribution?
- ? How to choose m?

Ridge Leverage Scores and Effective Dimension

Definition

Given a kernel matrix $\mathbf{K}_n \in \mathbb{R}^{n \times n}$, define

$$\gamma\text{-RLS} \qquad \tau_{n,i} = \mathbf{e}_{n,i}\mathbf{K}_n^{\mathsf{T}}(\mathbf{K}_n + \gamma\mathbf{I}_n)^{-1}\mathbf{e}_{n,i} = \phi_i^{\mathsf{T}}(\mathbf{\Phi}_n\mathbf{\Phi}_n^{\mathsf{T}} + \gamma\mathbf{I})^{-1}\phi_i \tag{1}$$

$$\text{effective dim.} \qquad d_{\text{eff}}(\gamma)_n = \sum_{i=1}^{n} \tau_{n,i} = \operatorname{Tr}\left(\mathbf{K}_n(\mathbf{K}_n + \gamma\mathbf{I}_n)^{-1}\right) \tag{2}$$

Ridge Leverage Scores

Intuitively, RLS capture orthogonality

$$\tau_{n,i} = \mathbf{e}_{n,i}\mathbf{K}_n^\mathsf{T}(\mathbf{K}_n + \gamma\mathbf{I}_n)^{-1}\mathbf{e}_{n,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i$$

If all ϕ_i are orthogonal, we have

$$\tau_{n,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \phi_i^\mathsf{T}(\phi_i\phi_i^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \frac{\phi_i^\mathsf{T}\phi_i}{\phi_i^\mathsf{T}\phi_i + \gamma} \sim \mathbf{1}$$

If all ϕ_i are identical (collinear), we have

$$\tau_{n,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \phi_i^\mathsf{T}(n\phi_i\phi_i^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \frac{\phi_i^\mathsf{T}\phi_i}{n\phi_i^\mathsf{T}\phi_i + \gamma} \sim \frac{1}{n}$$

Ridge Leverage Scores

Intuitively, RLS capture orthogonality

$$\tau_{n,i} = \mathbf{e}_{n,i}\mathbf{K}_n^\mathsf{T}(\mathbf{K}_n + \gamma\mathbf{I}_n)^{-1}\mathbf{e}_{n,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i$$

If all ϕ_i are orthogonal, we have

$$\tau_{n,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \phi_i^\mathsf{T}(\phi_i\phi_i^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \frac{\phi_i^\mathsf{T}\phi_i}{\phi_i^\mathsf{T}\phi_i + \gamma} \sim \mathbf{1}$$

If all ϕ_i are identical (collinear), we have

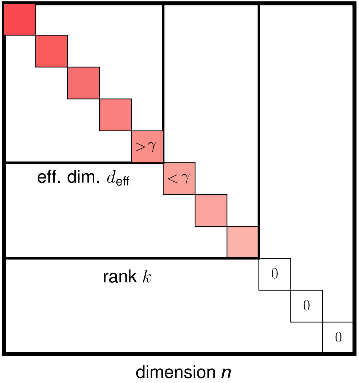
$$\tau_{n,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_n\mathbf{\Phi}_n^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \phi_i^\mathsf{T}(n\phi_i\phi_i^\mathsf{T} + \gamma\mathbf{I})^{-1}\phi_i = \frac{\phi_i^\mathsf{T}\phi_i}{n\phi_i^\mathsf{T}\phi_i + \gamma} \sim \frac{1}{n}$$

Given $\mathbf{\Phi}_{t-1}$, adding a new column to it can only reduce the RLS of columns already in $\mathbf{\Phi}_{t-1}$

$$\tau_{\mathbf{t},\mathbf{i}} \leq \tau_{\mathbf{t-1},\mathbf{i}}$$

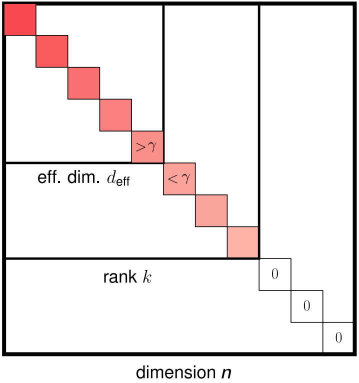
Effective Dimension

Intuitively, the effective dimension is a soft version of matrix rank



Effective Dimension

Intuitively, the effective dimension is a soft version of matrix rank



Given $d_{\text{eff}}(\gamma)_{t-1}$, adding a new column to $\mathbf{\Phi}_{t-1}$ can only increase $d_{\text{eff}}(\gamma)_t$

$$\mathbf{d}_{\text{eff}}(\gamma)_{\mathbf{t}} \geq \mathbf{d}_{\text{eff}}(\gamma)_{\mathbf{t-1}}$$

Nyström Sampling

Theorem (Alaoui, Mahoney, 2015)

Given γ be the Nyström regularization, ε the accuracy, δ the confidence.
If the dictionary $\mathcal{I}_n$ is computed using the sampling distribution $p_{n,i} \propto \tau_{n,i}$ and using at least m columns

$$m \geq \left(\frac{2\mathbf{d}_{\mathit{eff}}(\gamma)_{\mathbf{n}}}{\varepsilon^{\mathbf{2}}}\right) \log\left(\frac{n}{\delta}\right),$$

then with probability $1 - \delta$

$$\|\mathbf{P_n} - \widetilde{\mathbf{P}}_{\mathbf{n}}\|_{\mathbf{2}} \leq \varepsilon$$

Nyström Sampling

Theorem (Alaoui, Mahoney, 2015)

Given γ *be the Nyström regularization,* ε *the accuracy,* δ *the confidence.*
If the dictionary $\mathcal{I}_n$ *is computed using the sampling distribution* $p_{n,i} \propto \tau_{n,i}$ *and using at least m columns*

$$m \geq \left(\frac{2\mathbf{d}_{eff}(\gamma)_\mathbf{n}}{\varepsilon^2} \right) \log \left(\frac{n}{\delta} \right),$$

then with probability $1 - \delta$

$$\|\mathbf{P_n} - \widetilde{\mathbf{P}}_\mathbf{n}\|_2 \leq \varepsilon$$

Done!

Nyström Sampling

Theorem (Alaoui, Mahoney, 2015)

Given γ be the Nyström regularization, ε the accuracy, δ the confidence.
If the dictionary $\mathcal{I}_n$ is computed using the sampling distribution $p_{n,i} \propto \tau_{n,i}$ and using at least m columns

$$m \geq \left(\frac{2\mathbf{d}_{\mathit{eff}}(\gamma)_\mathbf{n}}{\varepsilon^2} \right) \log \left(\frac{n}{\delta} \right),$$

then with probability $1 - \delta$

$$\|\mathbf{P_n} - \widetilde{\mathbf{P}}_\mathbf{n}\|_2 \leq \varepsilon$$

Done!

If someone gave us the RLS

Computing $\tau_{n,i} = \mathbf{e}_{n,i}\mathbf{K}_n^{\mathsf{T}}(\mathbf{K}_n + \gamma\mathbf{I}_n)^{-1}\mathbf{e}_{n,i}$ also requires storing and inverting the full $\mathbf{K}_n$

Estimating RLS

Idea 1: Instead of computing exact RLS, compute good approximations

Estimating RLS

Idea 1: Instead of computing exact RLS, compute good approximations
Idea 2: When all you have is a dictionary, you use the dictionary

Estimating RLS

Idea 1: Instead of computing exact RLS, compute good approximations
Idea 2: When all you have is a dictionary, you use the dictionary

Lemma

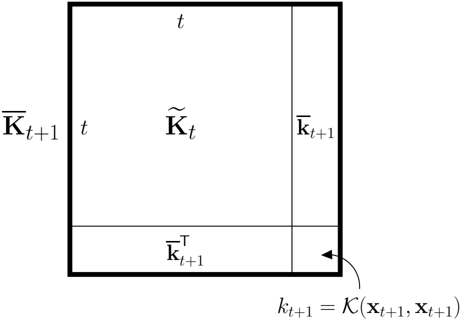
Assume that the dictionary $\mathcal{I}_{t-1}$ is accurate, and let $\overline{\mathbf{S}}_t$ be constructed by adding $(1, \phi_t)$ to $\mathcal{I}_{t-1}$. Then, denoting $\alpha = (1+\varepsilon)/(1-\varepsilon)$, for all i such that $i \in \{\mathcal{I}_{t-1} \cup \{t\}\}$,

$$\widetilde{\tau}_{t,i} = \frac{1+\varepsilon}{\alpha\gamma}\left(k_{i,i} - \mathbf{k_{t,i}}\overline{\mathbf{S}}\left(\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{K_t}\overline{\mathbf{S}} + \gamma\mathbf{I}\right)^{-1}\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{k_{t,i}}\right), \tag{3}$$

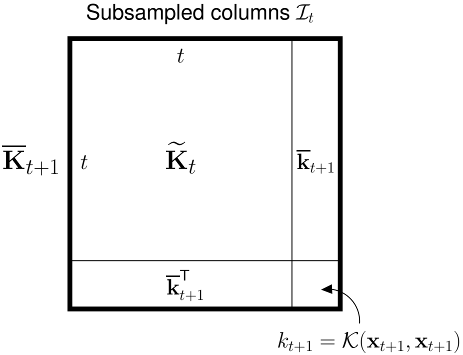
is an α-approximation of the RLS $\tau_{t,i}$, that is $\tau_{t,i}(\gamma)/\alpha \le \widetilde{\tau}_{t,i} \le \tau_{t,i}(\gamma)$.

The problem of estimating RLS

Subsampled columns $\mathcal{I}_t$

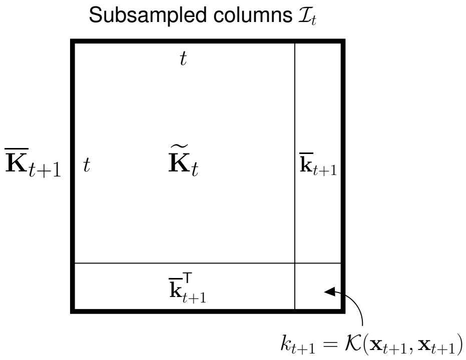


The problem of estimating RLS



Approximate sampling distribution $\mathbf{p}_{t+1}$

The problem of estimating RLS



Approximate sampling distribution $\mathbf{p}_{t+1}$

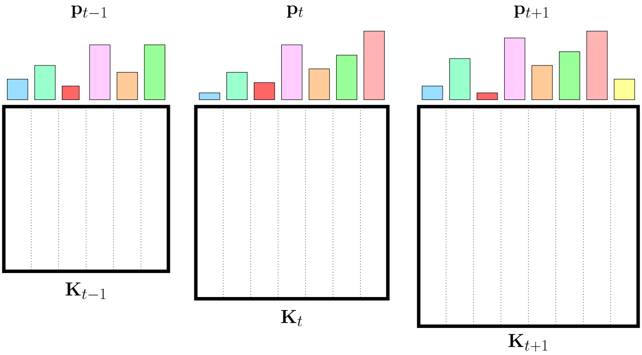
$\Rightarrow$ since $p_{i,t+1} \propto \tau_{i,t+1}$, *approximate* $\tau_{i,t+1}$

Estimating RLS

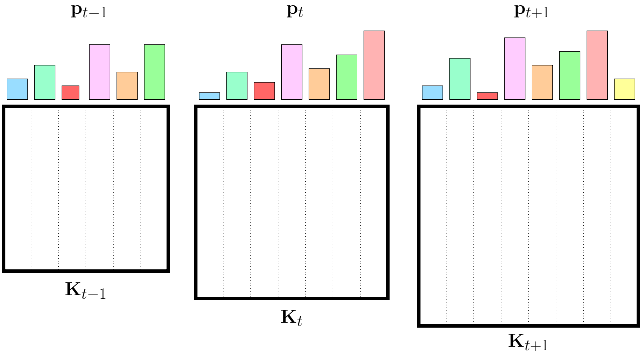
$$\widetilde{\tau}_{t,i} = \frac{1+\varepsilon}{\alpha\gamma}\left(k_{i,i} - \mathbf{k_{t,i}}\overline{\mathbf{S}}\left(\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{K_t}\overline{\mathbf{S}} + \gamma\mathbf{I}\right)^{-1}\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{k_{t,i}}\right),$$

- $\widetilde{\tau}_{t,i} = \mathbf{e}_i^{\mathsf{T}}\widetilde{\mathbf{K}}_{\mathbf{t}}(\widetilde{\mathbf{K}}_{\mathbf{t}} + \gamma\mathbf{I})^{-1}\mathbf{e}_i$ would fail
- Instead, approximate $\tau_{t,i}$ directly in $\mathcal{H}$, and then reformulate using kernel trick $\widetilde{\tau}_{\mathbf{t,i}} = \phi_{\mathbf{i}}^{\mathsf{T}}(\mathbf{\Phi}\overline{\mathbf{S}}\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{\Phi}^{\mathsf{T}} + \gamma\mathbf{I})^{-1}\phi_{\mathbf{i}}$
- $\widetilde{\tau}_{t,i}$ can be computed in $\mathcal{O}(|\mathcal{I}_{\mathbf{t}}|^{\mathbf{2}})$ space and $\mathcal{O}(|\mathcal{I}_{\mathbf{t}}|^{\mathbf{3}})$ time
  - ↳ independent from t
- $\widetilde{\tau}_{t,i}$ for $i \in \mathcal{I}_t$ can be computed using only samples contained in $\mathcal{I}_t$.

Estimating RLS incrementally

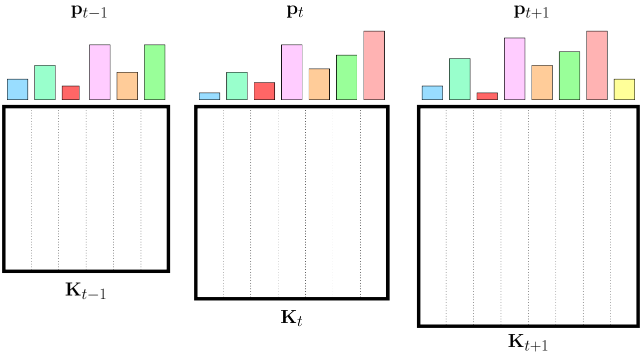


Estimating RLS incrementally



At each time step t construct $\widetilde{\mathbf{K}}_t$ ***as if*** it was drawn from $\mathbf{p}_t$

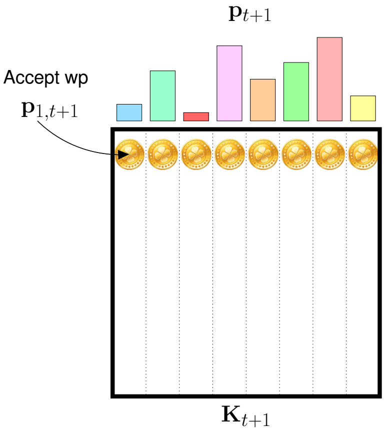
Estimating RLS incrementally



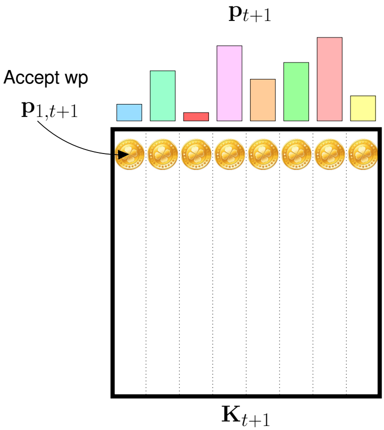
At each time step t construct $\widetilde{\mathbf{K}}_t$ ***as if*** it was drawn from $\mathbf{p}_t$

$\Rightarrow$ update the sampling set $\mathcal{I}_t$ incrementally as $\mathbf{p}_t$ changes

Estimating RLS incrementally by rejection sampling



Estimating RLS incrementally by rejection sampling



m calls to a multinomial $\mathbf{p}_{t+1}$

$\approx$ calls to $t + 1$ binomials each with probability $p_{i,t+1}$

Distributed sequential sampling for adaptive kernel DL

Instead of sampling from multinomial consider the sampling process

$$q_{i,i} \sim \mathcal{B}(\widetilde{p}_{i,i}, \overline{q})$$
$$q_{t,i} \sim \mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}, q_{t-1,i})$$

Distributed sequential sampling for adaptive kernel DL

Instead of sampling from multinomial consider the sampling process

$$q_{i,i} \sim \mathcal{B}(\widetilde{p}_{i,i}, \overline{q})$$
$$q_{t,i} \sim \mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}, q_{t-1,i})$$

Similar to importance sampling. If the $\widetilde{p}_{t,i}$ were fixed in advance

$$\begin{aligned}
\mathbb{P}(z_{t,i,j} = 1) &= \mathbb{P}(\mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}) = 1) z_{t-1,i,j} \\
&= \mathbb{P}(\mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}) = 1)\mathbb{P}(\mathcal{B}(\widetilde{p}_{t-1,i}/\widetilde{p}_{t-2,i}) = 1) z_{t-2,i,j} \\
&= \frac{\widetilde{p}_{t,i}}{\widetilde{p}_{t-1,i}} \frac{\widetilde{p}_{t-1,i}}{\widetilde{p}_{t-2,i}} \cdots \frac{\widetilde{p}_{i+1,i}}{\widetilde{p}_{i,i}} \frac{\widetilde{p}_{i,i}}{1} = \widetilde{p}_{t,i}
\end{aligned}$$

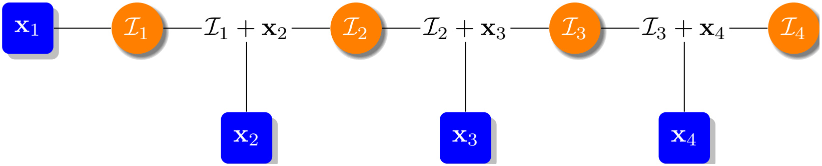
Distributed sequential sampling for adaptive kernel DL

Instead of sampling from multinomial consider the sampling process

$$q_{i,i} \sim \mathcal{B}(\widetilde{p}_{i,i}, \overline{q})$$
$$q_{t,i} \sim \mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}, q_{t-1,i})$$

Similar to importance sampling. If the $\widetilde{p}_{t,i}$ were fixed in advance

$$\begin{aligned}
\mathbb{P}(z_{t,i,j} = 1) &= \mathbb{P}(\mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}) = 1) z_{t-1,i,j} \\
&= \mathbb{P}(\mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}) = 1)\mathbb{P}(\mathcal{B}(\widetilde{p}_{t-1,i}/\widetilde{p}_{t-2,i}) = 1) z_{t-2,i,j} \\
&= \frac{\widetilde{p}_{t,i}}{\widetilde{p}_{t-1,i}} \frac{\widetilde{p}_{t-1,i}}{\widetilde{p}_{t-2,i}} \cdots \frac{\widetilde{p}_{i+1,i}}{\widetilde{p}_{i,i}} \frac{\widetilde{p}_{i,i}}{1} = \widetilde{p}_{t,i}
\end{aligned}$$



SQUEAK

Dictionary $\mathcal{I}_t = \{(j, \phi_j, q_{t,j}, \widetilde{p}_{t,j})\}$, weights $w_i = \frac{q_{t,j}}{\widetilde{p}_{t,j}\overline{q}}$

Input: $\mathcal{D}$, regularization $\gamma, \overline{q}$, ε, **Output**: $\mathcal{I}_n$

Initialize $\mathcal{I}_0$ as empty, $\widetilde{p}_{1,0} = 1$
for $t = 1, \dots, n$ **do**
    Receive new sample $\mathbf{x}_t$
    Compute α-app. RLS $\{\widetilde{\tau}_{t,i} : i \in \mathcal{I}_{t-1} \cup \{t\}\}$, using $\mathcal{I}_{t-1}$, $\mathbf{x}$, and Eq. 3
    Set $\widetilde{\mathbf{p}}_{\mathbf{t,i}} = \min\left\{\widetilde{\tau}_{\mathbf{t,i}},\ \widetilde{\mathbf{p}}_{\mathbf{t-1,i}}\right\}$
    Initialize $\mathcal{I}_t = \emptyset$
    for all $j \in \{1, \dots, t-1\}$ **do**
        if $q_{t-1,j} \neq 0$ **then**
            $\mathbf{q}_{\mathbf{t,j}} \sim \mathcal{B}(\widetilde{\mathbf{p}}_{\mathbf{t,j}}/\widetilde{\mathbf{p}}_{\mathbf{t-1,j}}, \mathbf{q}_{\mathbf{t-1,j}})$
            Add $(j, \phi_j, q_{t,j}, \widetilde{p}_{t,j})$ to $\mathcal{I}_t$.
        end if
    end for
    $\mathbf{q}_{\mathbf{t,t}} \sim \mathcal{B}(\widetilde{\mathbf{p}}_{\mathbf{t,t}}, \overline{\mathbf{q}})$
    Add $q_{t,t}$ copies of $(t, \phi_t, q_{t,t}, \widetilde{p}_{t,t})$ to $\mathcal{I}_t$
end for

SHRINK

EXPAND

DICT-UPDATE

SQUEAK

Theorem

Let $\alpha = (\frac{1+\varepsilon}{1-\varepsilon})$ *and* $\gamma > 1$. *For any* $0 \leq \varepsilon \leq 1$, *and* $0 \leq \delta \leq 1$, *if we run* SQUEAK *with* $\overline{\mathbf{q}} = \mathcal{O}(\frac{\alpha}{\varepsilon^2} \log(\frac{\mathbf{n}}{\delta}))$, *then w.p.* $1 - \delta$, *for all* $t \in [n]$

(1) $\|\mathbf{P_t} - \widetilde{\mathbf{P}}_\mathbf{t}\|_\mathbf{2} \leq \varepsilon$.

(2) $|\mathcal{I}_\mathbf{t}| = \sum_i q_{t,i} \leq \mathcal{O}(\overline{q} d_{eff}(\gamma)_t) \leq \mathcal{O}(\frac{\alpha}{\varepsilon^2} \mathbf{d}_{eff}(\gamma)_\mathbf{n} \log(\frac{\mathbf{n}}{\delta}))$.

SQUEAK

Theorem

Let $\alpha = (\frac{1+\varepsilon}{1-\varepsilon})$ and $\gamma > 1$. For any $0 \leq \varepsilon \leq 1$, and $0 \leq \delta \leq 1$, if we run SQUEAK *with $\overline{\mathbf{q}} = \mathcal{O}(\frac{\alpha}{\varepsilon^2} \log(\frac{\mathbf{n}}{\delta}))$, then w.p. $1-\delta$, for all $t \in [n]$*

(1) $\|\mathbf{P_t} - \widetilde{\mathbf{P}}_\mathbf{t}\|_2 \leq \varepsilon$.

(2) $|\mathcal{I}_\mathbf{t}| = \sum_i q_{t,i} \leq \mathcal{O}(\overline{q} d_{\textit{eff}}(\gamma)_t) \leq \mathcal{O}(\frac{\alpha}{\varepsilon^2} \mathbf{d}_{\textit{eff}}(\gamma)_\mathbf{n} \log(\frac{\mathbf{n}}{\delta})))$.

- Accuracy and space/time guarantees
- Anytime risk guarantees
- In worst case, no space gain (stores full $\mathbf{K}_n$)
- In worst case, no space overhead (stores full $\mathbf{K}_n$)
- RLS estimator not incremental, not easy because of changing weights
- Unnormalized $\widetilde{p}_{t,i}$, no need for appr. $d_{\text{eff}}(\gamma)_t$

SQUEAK

Theorem

Let $\alpha = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)$ *and* $\gamma > 1$. *For any* $0 \leq \varepsilon \leq 1$, *and* $0 \leq \delta \leq 1$, *if we run* SQUEAK *with* $\overline{\mathbf{q}} = \mathcal{O}(\frac{\alpha}{\varepsilon^2} \log(\frac{\mathbf{n}}{\delta}))$, *then w.p.* $1-\delta$, *for all* $t \in [n]$

(1) $\|\mathbf{P_t} - \widetilde{\mathbf{P}}_\mathbf{t}\|_\mathbf{2} \leq \varepsilon$.

(2) $|\mathcal{I}_\mathbf{t}| = \sum_i q_{t,i} \leq \mathcal{O}(\overline{q} d_{eff}(\gamma)_t) \leq \mathcal{O}(\frac{\alpha}{\varepsilon^2} \mathbf{d}_{eff}(\gamma)_\mathbf{n} \log(\frac{\mathbf{n}}{\delta}))$.

- Only need to compute $\widetilde{\tau}_{t,i}$ if $i \in \mathcal{I}_t$, never recompute after dropping
  - ↳ Never construct the whole $\mathbf{K}_n$
    - ↳ subquadratic runtime $\cancel{\mathcal{O}(\mathbf{n}^3)} \Rightarrow \mathcal{O}(\mathbf{n}|\mathcal{I}_\mathbf{n}|^\mathbf{3}) \leq \widetilde{\mathcal{O}}(\mathbf{n}\mathbf{d}_{\text{eff}}(\gamma)^\mathbf{3}_\mathbf{n})$
- Store points directly in the dictionary
  - ↳ $\widetilde{\mathcal{O}}(\mathbf{d}_{\text{eff}}(\gamma)^\mathbf{2}_\mathbf{n} + \mathbf{d}_{\text{eff}}(\gamma)_\mathbf{n}\mathbf{d})$ space constant in n
    - ↳ single pass over the dataset (streaming)

Proof sketch

Need to bound

$$\mathbb{P}\left(\exists t \in \{1,\ldots,n\} : \|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \geq \varepsilon \cup |\mathcal{I}_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t\right)$$

Proof sketch

Need to bound

$$\mathbb{P}\left(\exists t \in \{1,\dots,n\} : \|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \geq \varepsilon \cup |\mathcal{I}_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t\right)$$

After a union bound

$$\sum_{t=1}^{n} \mathbb{P}\left(\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \geq \varepsilon\right)$$
$$+ \sum_{t=1}^{n} \mathbb{P}\left(|\mathcal{I}_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t \cap \left\{\forall t' \in \{1,\dots,t\} : \|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \leq \varepsilon\right\}\right)$$

Proof sketch

We start by bounding $\mathbb{P}\left(\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \geq \varepsilon\right)$. Let

$$z_{s,i,j} = \mathbb{I}\left\{u_{s,i,j} \leq \frac{\widetilde{p}_{s,i}}{\widetilde{p}_{s-1,i}}\right\} z_{s-1,i,j}, \qquad \mathbf{v}_i = (\mathbf{K}_t + \gamma \mathbf{I})^{-1} \mathbf{K}_t^{1/2} \mathbf{e}_{t,i}$$

with $u_{s,i,j} \sim \mathcal{U}(0,1)$. Then

$$\mathbf{Y}_t = \mathbf{P}_t - \widetilde{\mathbf{P}}_t = \frac{1}{\overline{q}} \sum_{i=1}^{t} \sum_{j=1}^{\overline{q}} \left(1 - \frac{z_{t,i,j}}{\widetilde{p}_{t,i}}\right) \mathbf{v}_i \mathbf{v}_i^{\mathsf{T}}$$

Proof sketch

We start by bounding $\mathbb{P}\left(\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \geq \varepsilon\right)$. Let

$$z_{s,i,j} = \mathbb{I}\left\{u_{s,i,j} \leq \frac{\widetilde{p}_{s,i}}{\widetilde{p}_{s-1,i}}\right\} z_{s-1,i,j}, \qquad \mathbf{v}_i = (\mathbf{K}_t + \gamma \mathbf{I})^{-1}\mathbf{K}_t^{1/2}\mathbf{e}_{t,i}$$

with $u_{s,i,j} \sim \mathcal{U}(0,1)$. Then

$$\mathbf{Y}_t = \mathbf{P}_t - \widetilde{\mathbf{P}}_t = \frac{1}{\overline{q}} \sum_{i=1}^{t} \sum_{j=1}^{\overline{q}} \left(1 - \frac{z_{t,i,j}}{\widetilde{p}_{t,i}}\right) \mathbf{v}_i \mathbf{v}_i^\mathsf{T}$$

Cannot use concentrations for independent r.v., because $z_{t,i,j}$ and $z_{t,i',j'}$ are both dependent on $z_{t-1,i'',j''}$ through the estimates.

Proof sketch

Build the martingale

$$\mathbf{X}_{\{s,i,j\}} = \left(\frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} - \frac{z_{t,i,j}}{\widetilde{p}_{s,i}} \right) \mathbf{v}_i \mathbf{v}_i^\mathsf{T}$$

We can use variants of Bernstein's inequality for matrix martingales, we need a bound on the range

$$\begin{aligned} \|\mathbf{X}_{\{s,i,j\}}\| &= \frac{1}{q} \left| \left(\frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} - \frac{z_{s,i,j}}{\widetilde{p}_{s,i}} \right) \right| \|\mathbf{v}_i \mathbf{v}_i^\mathsf{T}\| \leq \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \|\mathbf{v}_i\|^2 \\ &\leq \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \mathbf{v_i}^\mathsf{T} \mathbf{v_i} = \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \mathbf{e_i}^\mathsf{T} \mathbf{K_t}^{1/2} (\mathbf{K_t} + \gamma \mathbf{I})^{-1} \mathbf{K_t}^{1/2} \mathbf{e_i} \\ &= \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \mathbf{e_i}^\mathsf{T} \mathbf{P_t} \mathbf{e_i} = \frac{1}{q} \frac{\tau_{\mathbf{t,i}}}{\widetilde{p}_{s,i}} \leq \frac{\alpha}{q} \frac{\tau_{t,i}}{p_{s,i}} = \frac{\alpha}{q} \frac{\tau_{t,i}}{\tau_{s,i}} \leq \frac{\alpha}{q} := R, \end{aligned}$$

Proof sketch

Build the martingale

$$\mathbf{X}_{\{s,i,j\}} = \left(\frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} - \frac{z_{t,i,j}}{\widetilde{p}_{s,i}} \right) \mathbf{v}_i \mathbf{v}_i^{\mathsf{T}}$$

We can use variants of Bernstein's inequality for matrix martingales, we need a bound on the range

$$\begin{aligned}
\|\mathbf{X}_{\{s,i,j\}}\| &= \frac{1}{q} \left| \left(\frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} - \frac{z_{s,i,j}}{\widetilde{p}_{s,i}} \right) \right| \|\mathbf{v}_i \mathbf{v}_i^{\mathsf{T}}\| \leq \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \|\mathbf{v}_i\|^2 \\
&\leq \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \mathbf{v_i^{\mathsf{T}} v_i} = \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \mathbf{e_i^{\mathsf{T}} K_t^{1/2} (K_t + \gamma I)^{-1} K_t^{1/2} e_i} \\
&= \frac{1}{q} \frac{1}{\widetilde{p}_{s,i}} \mathbf{e_i^{\mathsf{T}} P_t e_i} = \frac{1}{q} \frac{\boldsymbol{\tau_{t,i}}}{\widetilde{p}_{s,i}} \leq \frac{\alpha}{q} \frac{\tau_{t,i}}{p_{s,i}} = \frac{\alpha}{q} \frac{\tau_{t,i}}{\tau_{s,i}} \leq \frac{\alpha}{q} := R,
\end{aligned}$$

RLS normalize our r.v.

Proof sketch

Now bound the total variation

$$\mathbf{W} = \sum \mathbb{E}\left[\mathbf{X}^2_{\{s,i,j\}} \;\middle|\; \{\mathbf{X}_r\}_{r=0}^{\{s,i,j\}-1}\right]$$

$$= \frac{1}{\overline{q}^2} \sum_{j=1}^{\overline{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} \left(\frac{1}{\widetilde{p}_{s,i}} - \frac{1}{\widetilde{p}_{s-1,i}}\right) \mathbf{v}_i\mathbf{v}_i^{\mathsf{T}}\mathbf{v}_i\mathbf{v}_i^{\mathsf{T}}$$

Proof sketch

Now bound the total variation

$$\mathbf{W} = \sum \mathbb{E}\left[\mathbf{X}_{\{s,i,j\}}^2 \;\middle|\; \{\mathbf{X}_r\}_{r=0}^{\{s,i,j\}-1}\right]$$
$$= \frac{1}{\overline{q}^2} \sum_{j=1}^{\overline{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} \left(\frac{1}{\widetilde{p}_{s,i}} - \frac{1}{\widetilde{p}_{s-1,i}}\right) \mathbf{v}_i\mathbf{v}_i^\mathsf{T}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}$$

Deterministically

$$\|\mathbf{W}\| = \left\| \frac{1}{\overline{q}^2} \sum_{j=1}^{\overline{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}} \left(\frac{1}{\widetilde{p}_{s,i}} - \frac{1}{\widetilde{p}_{s-1,i}}\right) \mathbf{v}_i\mathbf{v}_i^\mathsf{T}\mathbf{v}_i\mathbf{v}_i^\mathsf{T} \right\|$$
$$\leq \left\| \frac{1}{\overline{q}^2} \sum_{j=1}^{\overline{q}} \sum_{i=1}^{t} \frac{\mathbf{v}_i^\mathsf{T}\mathbf{v}_i}{\widetilde{p}_{t,i}^2} \mathbf{v}_i\mathbf{v}_i^\mathsf{T} \right\| \leq \left\| \frac{\alpha}{\overline{q}} \sum_{i=1}^{t} \frac{1}{\widetilde{p}_{t,i}} \mathbf{v}_i\mathbf{v}_i^\mathsf{T} \right\|$$
$$\leq \left\| \frac{\alpha^2}{\overline{q}} \sum_{i=1}^{t} \mathbf{I} \right\| = \frac{\alpha^2}{\overline{q}} t$$

Proof sketch

Now bound the total variation

$$\mathbf{W} = \sum \mathbb{E}\left[\mathbf{X}_{\{s,i,j\}}^2 \;\middle|\; \{\mathbf{X}_r\}_{r=0}^{\{s,i,j\}-1}\right]$$
$$= \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{t}\boxed{\sum_{s=1}^{t}\frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}}\left(\frac{1}{\widetilde{p}_{s,i}}-\frac{1}{\widetilde{p}_{s-1,i}}\right)}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}$$

Deterministically

$$\|\mathbf{W}\| = \left\|\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{t}\sum_{s=1}^{t}\frac{z_{s-1,i,j}}{\widetilde{p}_{s-1,i}}\left(\frac{1}{\widetilde{p}_{s,i}}-\frac{1}{\widetilde{p}_{s-1,i}}\right)\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\right\|$$
$$\leq \left\|\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{t}\frac{\mathbf{v}_i^\mathsf{T}\mathbf{v}_i}{\widetilde{p}_{t,i}^2}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\right\| \leq \left\|\frac{\alpha}{\overline{q}}\sum_{i=1}^{t}\frac{1}{\widetilde{p}_{t,i}}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\right\|$$
$$\leq \left\|\frac{\alpha^2}{\overline{q}}\sum_{i=1}^{t}\mathbf{I}\right\| = \frac{\alpha^2}{\overline{q}}t$$

Deterministic bound on variance too large

Proof sketch

This looks too pessimistic. When $\frac{1}{\widetilde{p}_{s,i}}$ is large, $z_{s,i,j}$ should be zero.
We should take advantage of that.

Proof sketch

This looks too pessimistic. When $\frac{1}{\widetilde{p}_{s,i}}$ is large, $z_{s,i,j}$ should be zero. We should take advantage of that.

We can use a finer concentration, Freedman's inequality, that treats $\mathbf{W}$ itself as a random variable.

$$\mathbb{P}\left(\|\mathbf{Y}_t\| \geq \varepsilon \ \cap \ \|\mathbf{W}\| \leq \sigma^2\right) \leq t \exp\{-\dots\}$$

Proof sketch

This looks too pessimistic. When $\frac{1}{\widetilde{p}_{s,i}}$ is large, $z_{s,i,j}$ should be zero. We should take advantage of that.

We can use a finer concentration, Freedman's inequality, that treats $\mathbf{W}$ itself as a random variable.

$$\mathbb{P}\left(\|\mathbf{Y}_t\| \geq \varepsilon \ \cap \ \|\mathbf{W}\| \leq \sigma^2\right) \leq t \exp\{-\dots\}$$

Starting from an upper bound on $\mathbf{W}$ that is still a r.v.

$$\mathbf{W} \preceq \frac{1}{\overline{q}^2} \sum_{j=1}^{\overline{q}} \sum_{i=1}^{t} \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2} \right\} \mathbf{v}_i \mathbf{v}_i^\mathsf{T} \mathbf{v}_i \mathbf{v}_i^\mathsf{T}$$

Proof sketch

This looks too pessimistic. When $\frac{1}{\widetilde{p}_{s,i}}$ is large, $z_{s,i,j}$ should be zero. We should take advantage of that.

We can use a finer concentration, Freedman's inequality, that treats $\mathbf{W}$ itself as a random variable.

$$\mathbb{P}\left(\|\mathbf{Y}_t\| \geq \varepsilon \ \cap \ \|\mathbf{W}\| \leq \sigma^2\right) \leq t\exp\{-\dots\}$$

Starting from an upper bound on $\mathbf{W}$ that is still a r.v.

$$\mathbf{W} \preceq \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{t}\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2}\right\}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\mathbf{v}_i\mathbf{v}_i^\mathsf{T}$$

This still has high variance: cannot simply apply martingale Bernstein

Proof sketch

$\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2} \right\}$ is still hard to analyze, since it is the maximum of dependent variables

Proof sketch

$\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2} \right\}$ is still hard to analyze, since it is the maximum of dependent variables

Moreover $\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2} \right\}$ depends on $\max_{s=0}^{t-1} \left\{ \frac{z_{s,i',j'}}{\widetilde{p}_{s,i'}^2} \right\}$

Proof sketch

$\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2}\right\}$ is still hard to analyze, since it is the maximum of dependent variables

Moreover $\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2}\right\}$ depends on $\max_{s=0}^{t-1}\left\{\frac{z_{s,i',j'}}{\widetilde{p}_{s,i'}^2}\right\}$

We will find another set of dominating r.v. $1/w_{i,j}$, indep. from each other

Then apply Bernstein for indep. r.v.

Proof sketch

$\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^{2}}\right\}$ is still hard to analyze, since it is the maximum of dependent variables

Moreover $\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^{2}}\right\}$ depends on $\max_{s=0}^{t-1}\left\{\frac{z_{s,i',j'}}{\widetilde{p}_{s,i'}^{2}}\right\}$

We will find another set of dominating r.v. $1/w_{i,j}$, indep. from each other
Then apply Bernstein for indep. r.v.

Random variable A stochastically dominates random variable B, if for all values a the two equivalent conditions are verified

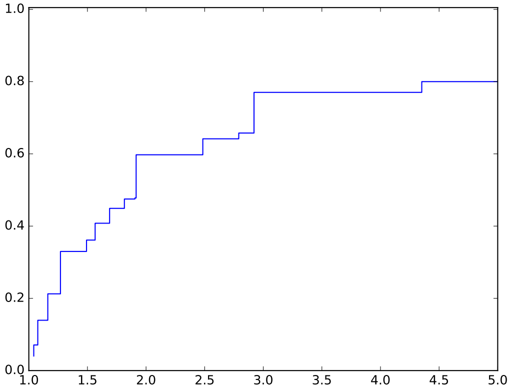
$$\mathbb{P}(A \geq a) \geq \mathbb{P}(B \geq a) \Leftrightarrow \mathbb{P}(A \leq a) \leq \mathbb{P}(B \leq a).$$

Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.

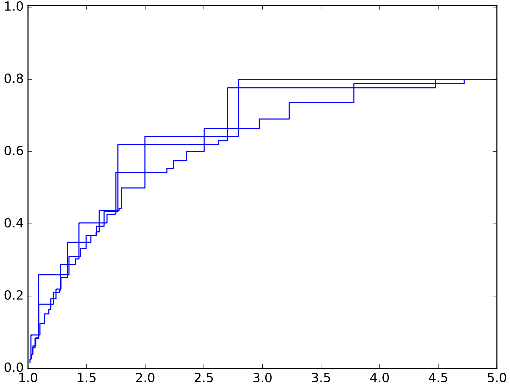
Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.



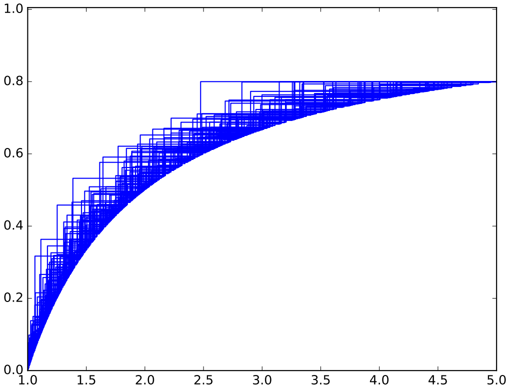
Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.



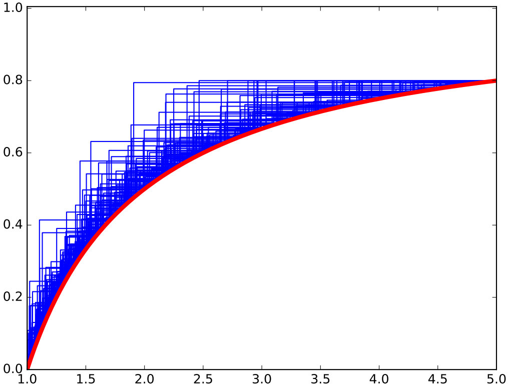
Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.



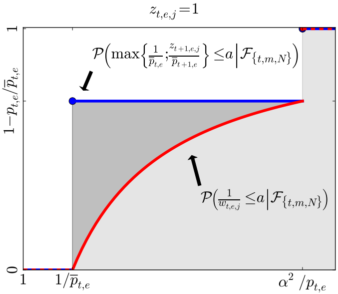
Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.



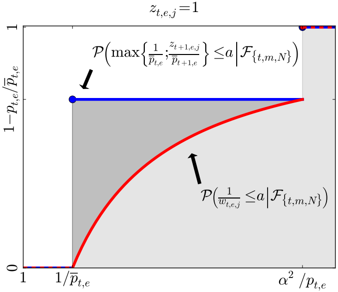
Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.



Proof sketch

Imagine the sequence $\widetilde{p}_{s,i}$ was fixed in advance. I can compute exactly the distribution of all $z_{s,i,j}$.



$$\mathbb{P}\left(\frac{1}{w_{0,i,j}} \leq a\right) = \begin{cases} 0 & \text{for} \quad a < 1 \\ 1 - \frac{1}{a} & \text{for} \quad 1 \leq a < \alpha/p_{t,i} \\ 1 & \text{for} \quad \alpha/p_{t,i} \leq a \end{cases},$$

Proof sketch

We can now unwind the proof

5 dominate $\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^{2}}\right\}$ with $1/w_{i,j}$

Proof sketch

We can now unwind the proof

5 dominate $\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2}\right\}$ with $1/w_{i,j}$

4 apply Bernstein inequality for indep. r.v. to bound $\mathbb{P}(\|\mathbf{W}\| \geq \sigma^2)$

Proof sketch

We can now unwind the proof

5 dominate $\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2}\right\}$ with $1/w_{i,j}$

4 apply Bernstein inequality for indep. r.v. to bound $\mathbb{P}(\|\mathbf{W}\| \geq \sigma^2)$

3 apply Freedman inequality to bound $\mathbb{P}(\|\mathbf{Y}\| \geq \varepsilon \cap \|\mathbf{W}\| \leq \sigma^2)$

Proof sketch

We can now unwind the proof

5 dominate $\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2} \right\}$ with $1/w_{i,j}$

4 apply Bernstein inequality for indep. r.v. to bound $\mathbb{P}(\|\mathbf{W}\| \geq \sigma^2)$

3 apply Freedman inequality to bound $\mathbb{P}(\|\mathbf{Y}\| \geq \varepsilon \cap \|\mathbf{W}\| \leq \sigma^2)$

2 apply another stochastic dominance argument to bound
$\mathbb{P}\left(|\mathcal{I}_t| \geq 3\overline{q} d_{\text{eff}}(\gamma)_t \cap \left\{\forall t' \in \{1, \ldots, t\} : \|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \leq \varepsilon\right\}\right)$

Proof sketch

We can now unwind the proof

5 dominate $\max_{s=0}^{t-1}\left\{\frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2}\right\}$ with $1/w_{i,j}$

4 apply Bernstein inequality for indep. r.v. to bound $\mathbb{P}(\|\mathbf{W}\| \geq \sigma^2)$

3 apply Freedman inequality to bound $\mathbb{P}(\|\mathbf{Y}\| \geq \varepsilon \cap \|\mathbf{W}\| \leq \sigma^2)$

2 apply another stochastic dominance argument to bound $\mathbb{P}\left(|\mathcal{I}_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t \cap \left\{\forall t' \in \{1,\ldots,t\} : \|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \leq \varepsilon\right\}\right)$

1 union bound

Proof sketch

We can now unwind the proof

5 dominate $\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\widetilde{p}_{s,i}^2} \right\}$ with $1/w_{i,j}$

4 apply Bernstein inequality for indep. r.v. to bound $\mathbb{P}(\|\mathbf{W}\| \geq \sigma^2)$

3 apply Freedman inequality to bound $\mathbb{P}(\|\mathbf{Y}\| \geq \varepsilon \cap \|\mathbf{W}\| \leq \sigma^2)$

2 apply another stochastic dominance argument to bound $\mathbb{P}\left(|\mathcal{I}_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t \cap \left\{\forall t' \in \{1, \dots, t\} : \|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|_2 \leq \varepsilon\right\}\right)$

1 union bound

0 Q.E.D.

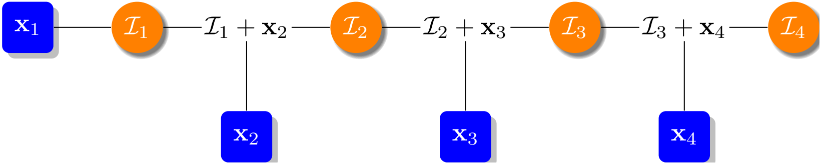
Distributed sequential sampling for adaptive kernel DL

SQUEAK is a strictly sequential algorithm

Distributed sequential sampling for adaptive kernel DL

SQUEAK is a strictly sequential algorithm

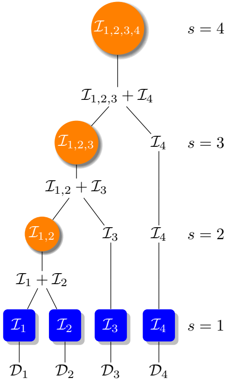
We just did a sequential analysis



Distributed sequential sampling for adaptive kernel DL

SQUEAK is a strictly sequential algorithm

We just did a sequential analysis



Distributed sequential sampling for adaptive kernel DL

SQUEAK is a strictly sequential algorithm

Distributed sequential sampling for adaptive kernel DL

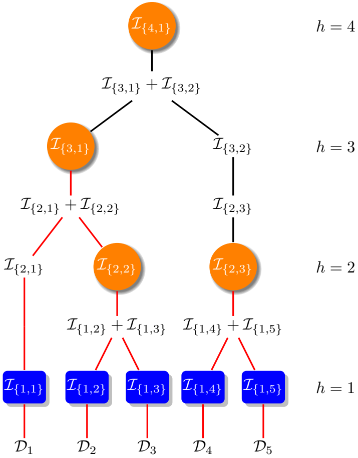
SQUEAK is a strictly sequential algorithm

DISQUEAK is the distributed equivalent

Distributed sequential sampling for adaptive kernel DL

SQUEAK is a strictly sequential algorithm

DISQUEAK is the distributed equivalent



DISQUEAK

Input: Dataset $\mathcal{D}$, regularization $\gamma, \overline{q}$, ε, **Output**: $\mathcal{I}_\mathcal{D}$

Partition $\mathcal{D}$ into disjoint sub-datasets $\mathcal{D}_i$
Run SQUEAK on each $\mathcal{D}_i$, build set $\mathcal{S}_1 = \{\mathcal{I}_{\mathcal{D}_i}\}_{i=1}^k$
for $h = 1, \ldots, k-1$ **do**
    if $|\mathcal{S}_h| > 1$ **then** ▷DICT-MERGE
        Pick two dictionaries $\mathcal{I}_\mathcal{D}, \mathcal{I}_{\mathcal{D}'}$ from $\mathcal{S}_h$
        $\overline{\mathcal{I}} = \mathcal{I}_\mathcal{D} \cup \mathcal{I}_{\mathcal{D}'}$
        $\mathcal{I}_{\mathcal{D},\mathcal{D}'} = \textsc{Dict-Update}(\overline{\mathcal{I}})$ using Eq. (4)
        Place $\mathcal{I}_{\mathcal{D},\mathcal{D}'}$ back into $\mathcal{S}_{h+1}$
    else
        $\mathcal{S}_{h+1} = \mathcal{S}_h$
    end if
end for
Return $\mathcal{I}_\mathcal{D}$, the last dictionary in $\mathcal{S}_k$

$$\widetilde{\tau}_{\mathcal{D}\cup\mathcal{D}',i} = \frac{1-\mathbf{2}\varepsilon}{\gamma}(k_{i,i} - \mathbf{k}_i^\mathsf{T}\overline{\mathbf{S}}(\overline{\mathbf{S}}^\mathsf{T}\mathbf{K}\overline{\mathbf{S}} + \gamma\mathbf{I})^{-1}\overline{\mathbf{S}}^\mathsf{T}\mathbf{k}_i), \tag{4}$$

DISQUEAK

Theorem

Let $\alpha = \left(\frac{1+2\varepsilon}{1-2\varepsilon}\right)$ and $\gamma > 1$. For any $0 \leq \varepsilon \leq 1$, and $0 \leq \delta \leq 1$, if we run $\mathrm{DISQUEAK}$ *with $\overline{\mathbf{q}} = \mathcal{O}(\frac{\alpha}{\varepsilon^2}\log(\frac{\mathbf{n}}{\delta}))$, then w.p. $1-\delta$, for all nodes $\{h, l\}$ in the merge tree*

(1) $\|\mathbf{P}_{\{\mathbf{h},\mathbf{l}\}} - \widetilde{\mathbf{P}}_{\{\mathbf{h},\mathbf{l}\}}\|_{\mathbf{2}} \leq \varepsilon$.

(2) $|\mathcal{I}_{\{\mathbf{h},\mathbf{l}\}}| \leq \mathcal{O}(\overline{q} d_{\mathit{eff}}(\gamma)_{\{h,l\}}) \leq \mathcal{O}(\frac{\alpha}{\varepsilon^2} \mathbf{d}_{\mathit{eff}}(\gamma)_{\mathbf{n}} \log(\frac{\mathbf{n}}{\delta}))$.

- Same accuracy as SQUEAK but much faster
- Space/accuracy guarantees for all nodes
- Much more space used, but spread across many machines
- Runtime depends on exact merge tree
  - ↳ Fully unbalanced tree: $\mathcal{O}(n|\mathcal{I}_n|^3)$, same as SQUEAK
    - ↳ Fully balanced tree: $\mathcal{O}(\log(n)|\mathcal{I}_n|^3)$ time, $\mathcal{O}(n|\mathcal{I}_n|^3)$ work!

Comparison

	Time	$\lvert\mathcal{I}_n\rvert$	Increm.
Exact	n^3	n	-
Bach'13	$\frac{n {d_{\max}}_n^2}{\varepsilon}$	$\frac{d_{\max,n}}{\varepsilon}$	No
A&M'15	$n(\lvert\mathcal{I}_n\rvert)^2$	$\left(\frac{\lambda_{\min} + n\gamma\varepsilon}{\lambda_{\min} - n\gamma\varepsilon}\right) {d_{\text{eff}}}_n + \frac{\mathrm{Tr}(\mathbf{K}_n)}{\gamma\varepsilon}$	No
Cal&al'16	$\frac{\lambda_{\max}^2}{\gamma^2} \frac{n^2 {d_{\text{eff}}}_n^3}{\varepsilon^2}$	$\frac{\lambda_{\max}}{\gamma} \frac{{d_{\text{eff}}}_n}{\varepsilon^2}$	Yes
SQUEAK	$\frac{n {d_{\text{eff}}}_n^3}{\varepsilon^2}$	$\frac{{d_{\text{eff}}}_n}{\varepsilon^2}$	Yes
RLS-sampling	$\frac{n {d_{\text{eff}}}_n^2}{\varepsilon^2}$	$\frac{{d_{\text{eff}}}_n}{\varepsilon^2}$	-
M&M'16	$\frac{n {d_{\text{eff}}}_n^3}{\varepsilon^2}$	$\frac{{d_{\text{eff}}}_n}{\varepsilon^2}$	No

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
- Final dictionary can be updated if new samples arrive

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
- Final dictionary can be updated if new samples arrive
- Novel analysis, potentially useful for general importance sampling

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
- Final dictionary can be updated if new samples arrive
- Novel analysis, potentially useful for general importance sampling

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
- Final dictionary can be updated if new samples arrive
- Novel analysis, potentially useful for general importance sampling

Future work

- Experiments
  - ↳ Trivial to implement: 328 lines of python, single file, including distributed task queue

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
- Final dictionary can be updated if new samples arrive
- Novel analysis, potentially useful for general importance sampling

Future work

Experiments

↳ Trivial to implement: 328 lines of python, single file, including distributed task queue

Preliminary results promising, easily scales to 100k of samples

Conclusions

SQUEAK and DISQUEAK

- First method (with guarantees) to break $\mathcal{O}(n)$ time barrier using DISQUEAK, with M&M'16 first to break $\mathcal{O}(n^2)$ barrier
- Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
- Final dictionary can be updated if new samples arrive
- Novel analysis, potentially useful for general importance sampling

Future work

- Experiments
  - ↳ Trivial to implement: 328 lines of python, single file, including distributed task queue
  - Preliminary results promising, easily scales to 100k of samples
- Beyond closed formulas: SQUEAK for gradient based methods