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
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
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
- $O(n^2)$ time to construct kernel matrix $K$
- $O(n^3)$ time to compute solution
- $O(n^2)$ 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

- $O(n^2)$ time to construct kernel matrix $K$
- $O(n^3)$ time to compute solution
- $O(n^2)$ 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 scalability or accuracy
Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalability or accuracy. We want both.
Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

**Problem:** Existing DL methods guarantee either **scalability** 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 scalability or accuracy; we want both.

We present **SQUEAK** — a dictionary learning algorithm that guarantees accurate reconstruction of the input in all cases.
Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

Problem: Existing DL methods guarantee either scalability 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 $O(n)$ space/time requirements
on “hard” problems not worse than storing whole input
Distributed sequential sampling for adaptive kernel DL

Why DL for kernel problems?

\textbf{Problem:} Existing DL methods guarantee either scalability or accuracy. We want both.

We present \textbf{SQUEAK} — a dictionary learning algorithm that guarantees:

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

\[ A = A^{1/2}(A^{1/2})^T = \sum_{i=1}^{n} a_i a_i^T \quad \tilde{A} = \sum_{i=1}^{m} w_i x_i x_i^T \]

<table>
<thead>
<tr>
<th>Method</th>
<th>( w_i )</th>
<th>( x_i )</th>
<th>Accuracy</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Input</td>
<td>1</td>
<td>( a_i )</td>
<td>⭐⭐⭐⭐⭐⭐️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

$$A = A^{1/2}(A^{1/2})^T = \sum_{i=1}^{n} a_ia_i^T$$

$$\tilde{A} = \sum_{i=1}^{m} w_ix_ix_i^T$$

<table>
<thead>
<tr>
<th>Method</th>
<th>$w_i$</th>
<th>$x_i$</th>
<th>Accuracy</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Input</td>
<td>1</td>
<td>$a_i$</td>
<td>★★★★★★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empty dictionary</td>
<td>0</td>
<td>0</td>
<td>★★★★★★</td>
<td>★★★★★★</td>
<td></td>
</tr>
</tbody>
</table>
We consider Positive Semi-Definite matrices

\[ A = A^{1/2}(A^{1/2})^T = \sum_{i=1}^{n} a_ia_i^T \]
\[ \tilde{A} = \sum_{i=1}^{m} w_ix_ix_i^T \]

<table>
<thead>
<tr>
<th>Method</th>
<th>( w_i )</th>
<th>( x_i )</th>
<th>Accuracy</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Input</td>
<td>1</td>
<td>( a_i )</td>
<td>★★★★★★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA</td>
<td>( \lambda_i )</td>
<td>( u_i )</td>
<td>★★★★★★</td>
<td>★★★★★</td>
<td>★</td>
</tr>
<tr>
<td>Empty dictionary</td>
<td>0</td>
<td>0</td>
<td>★★★★★★</td>
<td>★★★★★</td>
<td></td>
</tr>
</tbody>
</table>
Distributed sequential sampling for adaptive kernel DL

We consider Positive Semi-Definite matrices

\[
A = A^{1/2}(A^{1/2})^T = \sum_{i=1}^{n} a_i a_i^T \\
\tilde{A} = \sum_{i=1}^{m} w_i x_i x_i^T
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>(w_i)</th>
<th>(x_i)</th>
<th>Accuracy</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Input</td>
<td>1</td>
<td>(a_i)</td>
<td>★★★★★★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA</td>
<td>(\lambda_i)</td>
<td>(u_i)</td>
<td>★★★★★★</td>
<td>★★★★★</td>
<td>★</td>
</tr>
<tr>
<td>RLS (this)</td>
<td>(1/\tau_i)</td>
<td>(a_i)</td>
<td>★★★★★</td>
<td>★★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>Empty dictionary</td>
<td>0</td>
<td>0</td>
<td>★★★★★★</td>
<td>★★★★★★</td>
<td></td>
</tr>
</tbody>
</table>
We consider Positive Semi-Definite matrices

$$A = A^{1/2} (A^{1/2})^T = \sum_{i=1}^{n} a_i a_i^T$$

$$\tilde{A} = \sum_{i=1}^{m} w_i x_i x_i^T$$

<table>
<thead>
<tr>
<th>Method</th>
<th>$w_i$</th>
<th>$x_i$</th>
<th>Accuracy</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Input</td>
<td>1</td>
<td>$a_i$</td>
<td>★★★★★★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA</td>
<td>$\lambda_i$</td>
<td>$u_i$</td>
<td>★★★★★★</td>
<td>★★★★★★</td>
<td>★</td>
</tr>
<tr>
<td>RLS (this)</td>
<td>$1/\tau_i$</td>
<td>$a_i$</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>Uniform</td>
<td>$n/m$</td>
<td>$a_i$</td>
<td>★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>Empty dictionary</td>
<td>0</td>
<td>0</td>
<td>★★★★★★</td>
<td>★★★★★★</td>
<td>★★★★★★</td>
</tr>
</tbody>
</table>
Preliminaries: Setting and Kernels

Indexing $[t] = \{1, \ldots, t\}$, notation $K$ matrices, $k$ vectors, $k$ scalar
Dataset $D_n = \{x_i\}_{i=1}^n$, samples $x_i \in \mathcal{X}$ (e.g., $\mathbb{R}^d$)
Kernel function $K(x_i, x_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
Feature map $\varphi(x_i) : \mathcal{X} \to \mathcal{H} = \phi_i$
Kernel trick
$K(x_i, x_j) = \langle K(x_i, \cdot), K(x_j, \cdot) \rangle_{\mathcal{H}} = \langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}} = \phi_i^T \phi_j$
Feature matrix $\Phi_t = [\phi_1, \phi_2, \ldots, \phi_t] : \mathbb{R}^t \to \mathcal{H}$
Empirical kernel matrix $K_t \in \mathbb{R}^{t \times t} = K[t], [t] = \Phi_t^T \Phi_t$
New column $k_{[t-1], t} \in \mathbb{R}^{t-1} = \Phi_{t-1}^T \phi_t$
Kernel at a point $k_{t, t} \in \mathbb{R} = \phi_t^T \phi_t$

Find a dictionary $I = \{(w_j, \phi_j)\}_{j=1}^m$ such that $\tilde{K} = f(I)$ close to $K$
Preliminaries: Linear Algebra

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

Matrix norms (if omitted, $\ell$-2 norm)

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

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

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

$$\Phi \Phi^T (\Phi \Phi^T + \gamma I_n)^{-1} = \Phi (\Phi^T \Phi + \gamma I_m)^{-1} \Phi^T$$
Example: Kernel Ridge Regression

\[ \hat{w}_n = (K_n + \gamma I)^{-1} y_n \]

\[ \hat{y}_n = K_n \hat{w}_n = K_n (K_n + \gamma I)^{-1} y_n = P_n y_n \]
Example: Kernel Ridge Regression

\[ \hat{w}_n = (K_n + \gamma I)^{-1} y_n \]
\[ \hat{y}_n = K_n \hat{w}_n = K_n (K_n + \gamma I)^{-1} y_n = P_n y_n \]

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

\[ \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I \]
Example: Kernel Ridge Regression

\[ \hat{w}_n = (K_n + \gamma I)^{-1} y_n \]
\[ \hat{y}_n = K_n \hat{w}_n = K_n (K_n + \gamma I)^{-1} y_n = P_n y_n \]

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

\[ \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I \]

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

\[ \tilde{w}_n = (\tilde{K}_n + \gamma I)^{-1} y_n \]
\[ R(\tilde{w}_n) \leq \left(1 + \frac{1}{1 - \varepsilon}\right) R(\hat{w}_n) \]
Example: Kernel Ridge Regression

\[ \hat{w}_n = (K_n + \gamma I)^{-1} y_n \]
\[ \hat{y}_n = K_n \hat{w}_n = K_n(K_n + \gamma I)^{-1} y_n = P_n y_n \]

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

\[ \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I \]

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

\[ \tilde{w}_n = (\tilde{K}_n + \gamma I)^{-1} y_n \]
\[ R(\tilde{w}_n) \leq \left( 1 + \frac{1}{1 - \varepsilon} \right) R(\hat{w}_n) \]

\[ \mathcal{O}(n^3) \Rightarrow \mathcal{O}(nm + m^3) \text{ time to compute the approx. solution} \]
\[ \mathcal{O}(n^2) \Rightarrow \mathcal{O}(nm) \text{ space to store dictionary} \]
Reconstruction Guarantees

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

$$
\sum_{i=1}^{m} w_i \phi_i \phi_i^T = \sum_{i=1}^{m} (\sqrt{w_i} \phi_i) (\sqrt{w_i} \phi_i)^T = \Phi_n S_n S_n^T \Phi_n^T
$$
Reconstruction guarantees

Consider the regularized projection $\Psi_n$

$$\Psi_n = \Phi_n \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1} = (\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1/2}$$

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

$$\tilde{\Psi}_n = (\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n S_n S_n^T \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1/2} = \sum_{j=1}^{m} w_j \psi_j \psi_j^T$$
Reconstruction guarantees

Consider the regularized projection $\Psi_n$

$$\Psi_n = \Phi_n \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1} = (\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1/2}$$

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

$$\tilde{\Psi}_n = (\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n S_n S_n^T \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1/2} = \sum_{j=1}^m w_j \psi_j \psi_j^T$$

An accurate dictionary satisfies

$$\|\Psi_n - \tilde{\Psi}_n\|_2 \leq \varepsilon$$
Reconstruction guarantees

Consider the regularized projection $\Psi_n$

$$
\Psi_n = \Phi_n \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1} = (\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1/2}
$$

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

$$
\tilde{\Psi}_n = (\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n S_n S_n^T \Phi_n^T (\Phi_n \Phi_n^T + \gamma I)^{-1/2} = \sum_{j=1}^{m} w_j \psi_j \psi_j^T
$$

An accurate dictionary satisfies

$$
\|\Psi_n - \tilde{\Psi}_n\|_2 \leq \varepsilon
$$

equivalent to mixed additive/multiplicative error in quadratic form

$$
(1 - \varepsilon) \Phi_n \Phi_n^T - \varepsilon \gamma I \preceq \Phi_n S_n S_n^T \Phi_n^T \preceq (1 + \varepsilon) \Phi_n \Phi_n^T + \varepsilon \gamma I
$$
Reconstruction guarantees

Why would bounding $\|\psi_n - \tilde{\psi}_n\|_2$ be useful?
Reconstruction guarantees

Why would bounding $\|\Psi_n - \tilde{\Psi}_n\|_2$ be useful?

\[
\|\Psi_n - \tilde{\Psi}_n\|_2 = \|(\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n (I - S_n S_n^T) \Phi_n (\Phi_n \Phi_n^T + \gamma I)^{-1/2}\|_2 \\
= \|(\Sigma \Sigma^T + \gamma I)^{-1/2} \Sigma U^T (I - S_n S_n^T) U \Sigma^T (\Sigma \Sigma^T + \gamma I)^{-1/2}\|_2 \\
= \|(K_n + \gamma I)^{-1/2} K_n^{1/2} (I - S_n S_n^T) K_n^{1/2} (K_n + \gamma I)^{-1/2}\|_2 \\
= \|P_n - \tilde{P}_n\|_2
\]
Reconstruction guarantees

Why would bounding $\|\Psi_n - \tilde{\Psi}_n\|_2$ be useful?

\[
\|\Psi_n - \tilde{\Psi}_n\|_2 = \|(\Phi_n \Phi_n^T + \gamma I)^{-1/2} \Phi_n (I - S_n S_n^T) \Phi_n (\Phi_n \Phi_n^T + \gamma I)^{-1/2}\|_2 \\
= \|(\Sigma \Sigma^T + \gamma I)^{-1/2} \Sigma U^T (I - S_n S_n^T) U \Sigma^T (\Sigma \Sigma^T + \gamma I)^{-1/2}\|_2 \\
= \|(K_n + \gamma I)^{-1/2} K_n^{1/2} (I - S_n S_n^T) K_n^{1/2} (K_n + \gamma I)^{-1/2}\|_2 \\
= \|P_n - \tilde{P}_n\|_2
\]

with

\[
P_n = K_n (K_n + \gamma I)^{-1} \\
\tilde{P}_n = (K_n + \gamma I)^{-1/2} K_n^{1/2} S_n S_n^T K_n^{1/2} (K_n + \gamma I)^{-1/2}
\]
Reconstruction guarantees

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

Why would bounding $\|P_n - \tilde{P}_n\|_2$ be useful?

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

Why would bounding $\|P_n - \tilde{P}_n\|_2$ be useful?

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

$$\hat{w}_n = (K_n + \gamma I)^{-1}y_n$$
$$\hat{y}_n = K_n\hat{w}_n = K_n(K_n + \gamma I)^{-1}y_n = P_ny_n$$
Reconstruction guarantees

Why would bounding $\|P_n - \tilde{P}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\tilde{K}_n = K_n S_n (S_n K_n S_n + \gamma I)^{-1} S_n K_n$$

then

$$\|P_n - \tilde{P}_n\|_2 \leq \varepsilon \Rightarrow \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I$$
Reconstruction guarantees

Why would bounding $\|P_n - \tilde{P}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\tilde{K}_n = K_n S_n (S_n K_n S_n + \gamma I)^{-1} S_n K_n$$

then

$$\|P_n - \tilde{P}_n\|_2 \leq \varepsilon \Rightarrow \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I$$

e.g., Kernel Ridge Regression

$$\tilde{w}_n = (\tilde{K}_n + \gamma I)^{-1} y_n$$

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

$O(n^3) \Rightarrow O(nm + m^3)$ time to compute the approx. solution

$O(n^2) \Rightarrow O(nm)$ space to store dictionary
Reconstruction guarantees

Why would bounding $\|P_n - \tilde{P}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\tilde{K}_n = K_n S_n (S_n K_n S_n + \gamma I)^{-1} S_n K_n$$

then

$$\|P_n - \tilde{P}_n\|_2 \leq \varepsilon \Rightarrow \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I$$

e.g., Kernel Ridge Regression*

$$\tilde{w}_n = (\tilde{K}_n + \gamma I)^{-1} y_n$$

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

$O(n^3) \Rightarrow O(nm + m^3)$ time to compute the approx. solution

$O(n^2) \Rightarrow O(nm)$ space to store dictionary

---

*Gaussian Processes
Reconstruction guarantees

Why would bounding $\|P_n - \tilde{P}_n\|_2$ be useful?

We can compute accurate low rank approximations. Let

$$\tilde{K}_n = K_n S_n (S_n K_n S_n + \gamma I)^{-1} S_n K_n$$

then

$$\|P_n - \tilde{P}_n\|_2 \leq \varepsilon \Rightarrow \tilde{K}_n \preceq K_n \preceq \tilde{K}_n + \frac{\gamma}{1 - \varepsilon} I$$

e.g., Kernel PCA, $K_n$ and $\tilde{K}_n$ have close leading eigenvalues/vectors

e.g., Kernel $K$-means can be formulated as a quadratic form

$$\min_C \text{Tr}(K_n - CC^T K_n CC^T) \sim \min_{\tilde{C}} \text{Tr}(\tilde{K}_n - \tilde{C}\tilde{C}^T \tilde{K}_n \tilde{C}\tilde{C}^T)$$
Regularized Nyström reconstruction

\[ \tilde{K}_n = K_n S_n (S_n^T K_n S_n + \gamma I)^{-1} S_n^T K_n \]

\[ C = K_n S_n \]

\[ W^{-1} = (S_n^T K_n S_n + \gamma I_m)^{-1} \]

\[ C^T = S_n^T K_n \]
Distributed sequential **sampling** for adaptive kernel DL

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

Sample $m$ points w.p. $p_n, i$, add to $I$ with weight $1/p_n, i$ (unbiased).
Distributed sequential **sampling** for adaptive kernel DL

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

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

\[
\mathcal{I}_n = \{7, 8, 2, 4\}
\]
Distributed sequential sampling for adaptive kernel DL

How do we compute an accurate \((\|\Psi_n - \tilde{\Psi}_n\|_2 \leq \varepsilon)\) 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 $K_n \in \mathbb{R}^{n \times n}$, define

$$
\gamma\text{-RLS} \quad \tau_{n,i} = e_{n,i}K_n^T(K_n + \gamma I_n)^{-1}e_{n,i}
$$

$$
= \phi_i^T(\Phi_n\Phi_n^T + \gamma I)^{-1}\phi_i
$$

effective dim. \quad d_{\text{eff}}(\gamma)_n = \sum_{i=1}^{n} \tau_{n,i} = \text{Tr} \left( K_n(K_n + \gamma I_n)^{-1} \right)

(1)

(2)
Ridge Leverage Scores

Intuitively, RLS capture orthogonality

\[ \tau_{n,i} = e_{n,i} K_n^T (K_n + \gamma I_n)^{-1} e_{n,i} = \phi_i^T (\Phi_n \Phi_n^T + \gamma I)^{-1} \phi_i \]

If all \( \phi_i \) are orthogonal, we have

\[ \tau_{n,i} = \phi_i^T (\Phi_n \Phi_n^T + \gamma I)^{-1} \phi_i = \phi_i^T (\phi_i \phi_i^T + \gamma I)^{-1} \phi_i = \frac{\phi_i^T \phi_i}{\phi_i \phi_i + \gamma} \sim 1 \]

If all \( \phi_i \) are identical (collinear), we have

\[ \tau_{n,i} = \phi_i^T (\Phi_n \Phi_n^T + \gamma I)^{-1} \phi_i = \phi_i^T (n \phi_i \phi_i^T + \gamma I)^{-1} \phi_i = \frac{\phi_i^T \phi_i}{n \phi_i \phi_i + \gamma} \sim \frac{1}{n} \]
Ridge Leverage Scores

Intuitively, RLS capture orthogonality

\[ \tau_{n,i} = e_{n,i}K_n^T(K_n + \gamma I_n)^{-1}e_{n,i} = \phi_i^T(\Phi_n\Phi_n^T + \gamma I)^{-1}\phi_i \]

If all \( \phi_i \) are orthogonal, we have

\[ \tau_{n,i} = \phi_i^T(\Phi_n\Phi_n^T + \gamma I)^{-1}\phi_i = \phi_i^T(\phi_i\phi_i^T + \gamma I)^{-1}\phi_i = \frac{\phi_i^T\phi_i}{\phi_i^T\phi_i + \gamma} \sim 1 \]

If all \( \phi_i \) are identical (collinear), we have

\[ \tau_{n,i} = \phi_i^T(\Phi_n\Phi_n^T + \gamma I)^{-1}\phi_i = \phi_i^T(n\phi_i\phi_i^T + \gamma I)^{-1}\phi_i = \frac{\phi_i^T\phi_i}{n\phi_i^T\phi_i + \gamma} \sim \frac{1}{n} \]

Given \( \Phi_{t-1} \), adding a new column to it can only reduce the RLS of columns already in \( \Phi_{t-1} \)

\[ \tau_{t,i} \leq \tau_{t-1,i} \]
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 $\Phi_{t-1}$ can only increase $d_{\text{eff}}(\gamma)_t$

$$d_{\text{eff}}(\gamma)_t \geq d_{\text{eff}}(\gamma)_{t-1}$$
Nyström Sampling

**Theorem (Alaoui, Mahoney, 2015)**

Given $\gamma$ be the Nyström regularization, $\varepsilon$ the accuracy, $\delta$ 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

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

then with probability $1 - \delta$

$$\| P_n - \tilde{P}_n \|_2 \leq \varepsilon$$
Nyström Sampling

Theorem (Alaoui, Mahoney, 2015)

Given $\gamma$ be the Nyström regularization, $\varepsilon$ the accuracy, $\delta$ the confidence. If the dictionary $I_n$ is computed using the sampling distribution $p_{n,i} \propto \tau_{n,i}$ and using at least $m$ columns

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

then with probability $1 - \delta$

$$\|P_n - \tilde{P}_n\|_2 \leq \varepsilon$$

Done!
Nyström Sampling

**Theorem (Alaoui, Mahoney, 2015)**

Given $\gamma$ be the Nyström regularization, $\varepsilon$ the accuracy, $\delta$ the confidence.

If the dictionary $I_n$ is computed using the sampling distribution $p_{n,i} \propto \tau_{n,i}$ and using at least $m$ columns

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

then with probability $1 - \delta$

$$\|P_n - \tilde{P}_n\|_2 \leq \varepsilon$$

Done!

If someone gave us the RLS

Computing $\tau_{n,i} = e_{n,i}K_n^T(K_n + \gamma I_n)^{-1}e_{n,i}$ also requires storing and inverting the full $K_n$
Idea 1: Instead of computing exact RLS, compute good approximations

Lemma

Assume that the dictionary \( I^{t-1} \) is accurate, and let \( S^t \) be constructed by adding \((1, \varphi_t)\) to \( I^{t-1} \). Then, denoting \( \alpha = \frac{1 + \varepsilon}{1 - \varepsilon} \), for all \( i \in \{I^{t-1} \cup \{t\}\} \),

\[
\tilde{\tau}_{t,i} = 1 + \varepsilon \alpha \gamma \left( k_i - k_t \right) S^t \left( S^t K_t S^t + \gamma I \right)^{-1} S^t k_t, i
\]

is an \( \alpha \)-approximation of the RLS \( \tau_{t,i} \), that is \( \tau_{t,i}(\gamma) / \alpha \leq \tilde{\tau}_{t,i} \leq \tau_{t,i}(\gamma) \).
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 $I_{t-1}$ is accurate, and let $S_t$ be constructed by adding $(1, \phi_t)$ to $I_{t-1}$. Then, denoting $\alpha = \frac{1 + \epsilon}{1 - \epsilon}$, for all $i$ such that $i \in \{I_{t-1} \cup \{t\}\}$, $\tilde{\tau}_{t,i} = 1 + \epsilon \alpha \gamma(k_i, i - k_t, S^T K_t S + \gamma I_S) - 1 S^T k_t, i)$, (3) is an $\alpha$-approximation of the RLS $\tau_{t,i}$, that is $\tau_{t,i}(\gamma) / \alpha \leq \tilde{\tau}_{t,i} \leq \tau_{t,i}(\gamma)$.
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 $\bar{S}_t$ be constructed by adding $(1, \phi_t)$ to $\mathcal{I}_{t-1}$. Then, denoting $\alpha = (1 + \varepsilon)/(1 - \varepsilon)$, for all $i$ such that $i \in \{\mathcal{I}_{t-1} \cup \{t\}\}$,

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

(3)

is an $\alpha$-approximation of the RLS $\tau_{t,i}$, that is $\tau_{t,i}(\gamma)/\alpha \leq \tilde{\tau}_{t,i} \leq \tau_{t,i}(\gamma)$.
The problem of estimating RLS

Subsampled columns $\mathcal{I}_t$

$$\bar{K}_{t+1} \quad \tilde{K}_t \quad \bar{k}_{t+1}$$

$$\bar{k}_{t+1}^T$$

$k_{t+1} = \mathcal{K}(x_{t+1}, x_{t+1})$
The problem of estimating RLS

Subsampled columns $\mathcal{I}_t$

$\overline{K}_{t+1}$

$\tilde{K}_{t}$

$\overline{k}_{t+1}$

$\overline{k}_{t+1}^T$

$k_{t+1} = \mathcal{K}(x_{t+1}, x_{t+1})$

Approximate sampling distribution $\mathbf{p}_{t+1}$
The problem of estimating RLS

Subsampled columns $\mathcal{I}_t$

\[
\begin{array}{ccc}
\bar{K}_{t+1} & t & \tilde{K}_t \\
\hline
\hline
k_{t+1}^T & k_{t+1} \\
\end{array}
\]

$k_{t+1} = \mathcal{K}(\mathbf{x}_{t+1}, \mathbf{x}_{t+1})$

Approximate sampling distribution $p_{t+1}$

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

\[ \tilde{\tau}_{t,i} = \frac{1 + \varepsilon}{\alpha \gamma} \left( k_{t,i} - k_{t,i} S \left( S^T K_t S + \gamma I \right)^{-1} S^T k_{t,i} \right), \]

- \( \tilde{\tau}_{t,i} = e_i^T K_t (K_t + \gamma I)^{-1} e_i \) would fail
- Instead, approximate \( \tau_{t,i} \) directly in \( \mathcal{H} \), and then reformulate using kernel trick
  \[ \tilde{\tau}_{t,i} = \phi_i^T (\Phi S S^T \Phi^T + \gamma I)^{-1} \phi_i \]
- \( \tilde{\tau}_{t,i} \) can be computed in \( O(|I_t|^2) \) space and \( O(|I_t|^3) \) time
  \[ \text{independent from } t \]
- \( \tilde{\tau}_{t,i} \) for \( i \in I_t \) can be computed using only samples contained in \( I_t \).
Estimating RLS incrementally

$$p_{t+1} = p_t - p_{t-1}$$

At each time step $t$, construct $\tilde{K}_t$ as if it was drawn from $p_t$. Update the sampling set $I_t$ incrementally as $p_t$ changes.
At each time step $t$ construct $\tilde{K}_t$ as if it was drawn from $p_t$.
Estimating RLS incrementally

At each time step $t$ construct $\tilde{K}_t$ as if it was drawn from $p_t$

⇒ update the sampling set $\mathcal{I}_t$ incrementally as $p_t$ changes
Estimating RLS incrementally by rejection sampling

\[ p_{t+1} \]

Accept \( w_p \)

\[ P_{1,t+1} \]

\[ K_{t+1} \]
Estimating RLS incrementally by rejection sampling

$m$ calls to a multinomial $p_{t+1}$

$\approx$ 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

\[
q_{i,i} \sim B(\tilde{p}_{i,i}, \overline{q})
\]

\[
q_{t,i} \sim B(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}, q_{t-1,i})
\]
Distributed **sequential** sampling for adaptive kernel DL

Instead of sampling from multinomial consider the sampling process

\[ q_{i,i} \sim \mathcal{B}(\tilde{p}_{i,i}, \overline{q}) \]
\[ q_{t,i} \sim \mathcal{B}(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}, q_{t-1,i}) \]

Similar to **importance sampling**. If the \( \tilde{p}_{t,i} \) were fixed in advance

\[
\mathbb{P}(z_{t,i,j} = 1) = \mathbb{P}(\mathcal{B}(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}) = 1)z_{t-1,i,j} \\
= \mathbb{P}(\mathcal{B}(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}) = 1)\mathbb{P}(\mathcal{B}(\tilde{p}_{t-1,i}/\tilde{p}_{t-2,i}) = 1)z_{t-2,i,j} \\
= \frac{\tilde{p}_{t,i}}{\tilde{p}_{t-1,i}} \frac{\tilde{p}_{t-1,i}}{\tilde{p}_{t-2,i}} \cdots \frac{\tilde{p}_{i+1,i}}{\tilde{p}_{i,i}} \frac{\tilde{p}_{i,i}}{1} = \tilde{p}_{t,i}
\]
Distributed sequential sampling for adaptive kernel DL

Instead of sampling from multinomial consider the sampling process

\[
q_{i,t} \sim \mathcal{B}(\tilde{p}_{t,i}, q)
\]
\[
q_{t,i} \sim \mathcal{B}(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}, q_{t-1,i})
\]

Similar to importance sampling. If the \(\tilde{p}_{t,i}\) were fixed in advance

\[
\mathbb{P}(z_{t,i,j} = 1) = \mathbb{P}(\mathcal{B}(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}) = 1)z_{t-1,i,j}
\]
\[
= \mathbb{P}(\mathcal{B}(\tilde{p}_{t,i}/\tilde{p}_{t-1,i}) = 1)\mathbb{P}(\mathcal{B}(\tilde{p}_{t-1,i}/\tilde{p}_{t-2,i}) = 1)z_{t-2,i,j}
\]
\[
= \frac{\tilde{p}_{t,i}}{\tilde{p}_{t-1,i}} \frac{\tilde{p}_{t-1,i}}{\tilde{p}_{t-2,i}} \ldots \frac{\tilde{p}_{i+1,i}}{\tilde{p}_{i,i}} \frac{\tilde{p}_{i,i}}{1} = \tilde{p}_{t,i}
\]
Dictionary $\mathcal{I}_t = \{(j, \phi_j, q_{t,j}, \tilde{p}_{t,j})\}$, weights $w_i = \frac{q_{t,j}}{\tilde{p}_{t,j}q}$

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

1. Initialize $\mathcal{I}_0$ as empty, $\tilde{p}_{1,0} = 1$
2. **for** $t = 1, \ldots, n$ **do**
3. Receive new sample $x_t$
4. Compute $\alpha$-app. RLS $\{\tilde{\tau}_{t,i} : i \in \mathcal{I}_{t-1} \cup \{t\}\}$, using $\mathcal{I}_{t-1}$, $x$, and Eq. 3
5. Set $\tilde{p}_{t,i} = \min\{\tilde{\tau}_{t,i}, \tilde{p}_{t-1,i}\}$
6. Initialize $\mathcal{I}_t = \emptyset$
7. **for all** $j \in \{1, \ldots, t-1\}$ **do**
8. **if** $q_{t-1,j} \neq 0$ **then**
9. \quad $q_{t,j} \sim B(\tilde{p}_{t,j}/\tilde{p}_{t-1,j}, q_{t-1,j})$
10. **end if**
11. Add $(j, \phi_j, q_{t,j}, \tilde{p}_{t,j})$ to $\mathcal{I}_t$.
12. **end for**
13. $q_{t,t} \sim B(\tilde{p}_{t,t}, \bar{q})$
14. Add $q_{t,t}$ copies of $(t, \phi_t, q_{t,t}, \tilde{p}_{t,t})$ to $\mathcal{I}_t$
15. **end for**
**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 \( \overline{q} = O\left( \frac{\alpha}{\varepsilon^2} \log\left( \frac{n}{\delta} \right) \right) \), then w.p. \( 1 - \delta \), for all \( t \in [n] \):

1. \( \| P_t - \tilde{P}_t \|_2 \leq \varepsilon \).
2. \( |I_t| = \sum_i q_{t,i} \leq O\left( \overline{q}d_{\text{eff}}(\gamma)_t \right) \leq O\left( \frac{\alpha}{\varepsilon^2} d_{\text{eff}}(\gamma)_n \log\left( \frac{n}{\delta} \right) \right) \).
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 $\bar{q} = O\left(\frac{\alpha}{\varepsilon^2} \log\left(\frac{n}{\delta}\right)\right)$, then w.p. $1 - \delta$, for all $t \in [n]$

(1) $\|P_t - \tilde{P}_t\|_2 \leq \varepsilon$.

(2) $|I_t| = \sum_i q_{t,i} \leq O(q_{\text{d}eff(\gamma)_t}) \leq O\left(\frac{\alpha}{\varepsilon^2} d_{\text{d}eff(\gamma)_t} n \log\left(\frac{n}{\delta}\right)\right)$.

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

Theorem

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

(1) $\|P_t - \tilde{P}_t\|_2 \leq \epsilon$.

(2) $|I_t| = \sum_i q_{t,i} \leq \mathcal{O}(\bar{q}d_{\text{eff}}(\gamma)t) \leq \mathcal{O}(\frac{\alpha}{\epsilon^2} d_{\text{eff}}(\gamma)n \log(\frac{n}{\delta}))$.

▶ Only need to compute $\tilde{r}_{t,i}$ if $i \in I_t$, never recompute after dropping

Never construct the whole $K_n$

Subquadratic runtime $\mathcal{O}(n^3) \Rightarrow \mathcal{O}(n|I_n|^3)$

▶ Store points directly in the dictionary

$\tilde{O}(d_{\text{eff}}(\gamma)^2_n + d_{\text{eff}}(\gamma)_n d)$ space constant in $n$

Single pass over the dataset (streaming)
Proof sketch

Need to bound

\[ \mathbb{P}\left( \exists t \in \{1, \ldots, n\} : \|P_t - \tilde{P}_t\|_2 \geq \varepsilon \cup |I_t| \geq 3q_{\text{eff}}(\gamma)_t \right) \]
Proof sketch

Need to bound

\[ \mathbb{P}\left( \exists t \in \{1, \ldots, n\} : \|P_t - \tilde{P}_t\|_2 \geq \varepsilon \cup |I_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t \right) \]

After a union bound

\[
\sum_{t=1}^{n} \mathbb{P}\left( \|P_t - \tilde{P}_t\|_2 \geq \varepsilon \right) + \sum_{t=1}^{n} \mathbb{P}\left( |I_t| \geq 3\overline{q}d_{\text{eff}}(\gamma)_t \cap \left\{ \forall t' \in \{1, \ldots, t\} : \|P_t - \tilde{P}_t\|_2 \leq \varepsilon \right\} \right)
\]
Proof sketch

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

$$z_{s,i,j} = \mathbb{I} \left\{ u_{s,i,j} \leq \frac{\tilde{p}_{s,i}}{\tilde{p}_{s-1,i}} \right\} z_{s-1,i,j}, \quad v_i = (K_t + \gamma I)^{-1} K_t^{1/2} e_{t,i}$$

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

$$Y_t = P_t - \tilde{P}_t = \frac{1}{\bar{q}} \sum_{i=1}^{t} \sum_{j=1}^{\bar{q}} \left( 1 - \frac{z_{t,i,j}}{\tilde{p}_{t,i}} \right) v_i v_i^T$$
Proof sketch

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

$$z_{s,i,j} = \mathbb{I} \left\{ u_{s,i,j} \leq \frac{\tilde{p}_{s,i}}{\tilde{p}_{s-1,i}} \right\} z_{s-1,i,j}, \quad v_i = (K_t + \gamma I)^{-1} K_t^{1/2} e_{t,i}$$

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

$$Y_t = P_t - \tilde{P}_t = \frac{1}{q} \sum_{i=1}^{t} \sum_{j=1}^{q} \left( 1 - \frac{z_{t,i,j}}{\tilde{p}_{t,i}} \right) v_i v_i^T$$

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

$X_{\{s,i,j\}} = \left(\frac{Z_{s-1,i,j}}{\tilde{p}_{s-1,i}} - \frac{Z_{t,i,j}}{\tilde{p}_{s,i}}\right) v_i v_i^T$

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

$$\|X_{\{s,i,j\}}\| \leq \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} \|v_i v_i^T\| \leq \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} \|v_i\|^2 \leq \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} \left(\tilde{p}_t \sum \gamma I\right)^{-1} \tilde{p}_{t/2} e_i$$

$$\leq \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} e_i^T P_t e_i = \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} e_i^T P_t e_i = \frac{1}{q} \frac{\tau_{t,i}}{\tilde{p}_{s,i}} \leq \frac{\alpha}{q} \frac{\tau_{t,i}}{\tilde{p}_{s,i}} = \frac{\alpha}{q} \frac{\tau_{t,i}}{\tilde{p}_{s,i}} \leq \frac{\alpha}{q} := R,$$

Michal Valko: Distributed sequential sampling for adaptive DL
SequeL, Inria - 28/39
**Proof sketch**

Build the martingale

\[
X_{\{s,i,j\}} = \left( \frac{Z_{s-1,i,j}}{\tilde{p}_{s-1,i}} - \frac{Z_{t,i,j}}{\tilde{p}_{s,i}} \right) v_i v_i^T
\]

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

\[
\|X_{\{s,i,j\}}\| = \frac{1}{q} \left\| \left( \frac{Z_{s-1,i,j}}{\tilde{p}_{s-1,i}} - \frac{Z_{s,i,j}}{\tilde{p}_{s,i}} \right) \right\| \|v_i v_i^T\| \leq \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} \|v_i\|^2
\]

\[
\leq \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} v_i^T v_i = \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} e_i^T K_t^{1/2} (K_t + \gamma I)^{-1} K_t^{1/2} e_i
\]

\[
= \frac{1}{q} \frac{1}{\tilde{p}_{s,i}} e_i^T P_t e_i = \frac{1}{q} \frac{\tau_{t,i}}{\tilde{p}_{s,i}} \leq \frac{\alpha}{q} \frac{\tau_{t,i}}{\tilde{p}_{s,i}} = \frac{\alpha}{q} \frac{\tau_{t,i}}{\tau_{s,i}} \leq \frac{\alpha}{q} := R,
\]

**RLS normalize our r.v.**
Proof sketch

Now bound the total variation

\[ W = \sum_{s,i,j} \mathbb{E} \left[ X_{\{s,i,j\}}^2 \right] \left\{ X_r \right\}_{r=0}^{\{s,i,j\}-1} \]

\[ = \frac{1}{q^2} \sum_{j=1}^{q} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{Z_{s-1,i,j}}{\tilde{p}_{s-1,i}} \left( \frac{1}{\tilde{p}_s,i} - \frac{1}{\tilde{p}_{s-1,i}} \right) \mathbf{v}_i \mathbf{v}_i^T \mathbf{v}_i \mathbf{v}_i^T \]
Proof sketch

Now bound the total variation

\[ W = \sum \mathbb{E} \left[ X^2_{\{s,i,j\}} \mid \{ X_r \}_{r=0}^{\{s,i,j\}-1} \right] \]

\[ = \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{z_{s-1,i,j}}{\tilde{p}_{s-1,i}} \left( \frac{1}{\tilde{p}_s,i} - \frac{1}{\tilde{p}_{s-1,i}} \right) v_i v_i^T v_i v_i^T \]

Deterministically

\[ \| W \| = \left\| \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{z_{s-1,i,j}}{\tilde{p}_{s-1,i}} \left( \frac{1}{\tilde{p}_s,i} - \frac{1}{\tilde{p}_{s-1,i}} \right) v_i v_i^T v_i v_i^T \right\| \leq \left\| \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \frac{v_i^T v_i}{\tilde{p}_t,i} v_i v_i^T \right\| \leq \left\| \frac{\alpha}{q} \sum_{i=1}^{t} \frac{1}{\tilde{p}_t,i} v_i v_i^T \right\| \leq \left\| \frac{\alpha^2}{\bar{q}} \sum_{i=1}^{t} I \right\| = \frac{\alpha^2}{\bar{q}} t \]
Proof sketch

Now bound the total variation

\[ W = \sum \mathbb{E} \left[ X_{\{s,i,j\}}^2 \mid \{X_r\}_{r=0}^{\{s,i,j\} - 1} \right] \]

\[ = \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{Z_{s-1,i,j}}{\tilde{p}_{s-1,i}} \left( \frac{1}{\tilde{p}_{s,i}} - \frac{1}{\tilde{p}_{s-1,i}} \right) v_i v_i^T \]

Deterministically

\[ \|W\| = \left\| \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \sum_{s=1}^{t} \frac{Z_{s-1,i,j}}{\tilde{p}_{s-1,i}} \left( \frac{1}{\tilde{p}_{s,i}} - \frac{1}{\tilde{p}_{s-1,i}} \right) v_i v_i^T \right\| \]

\[ \leq \left\| \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \frac{v_i^T v_i}{\tilde{p}_{t,i}^2} v_i v_i^T \right\| \leq \left\| \frac{\alpha}{q} \sum_{i=1}^{t} \frac{1}{\tilde{p}_{t,i}} v_i v_i^T \right\| \]

\[ \leq \left\| \frac{\alpha^2}{\bar{q}} \sum_{i=1}^{t} 1 \right\| = \frac{\alpha^2}{\bar{q}} t \quad \text{Deterministic bound on variance too large} \]
Proof sketch

This looks too pessimistic. When \( \frac{1}{p_{s,i}} \) is large, \( z_{s,i,j} \) should be zero. We should take advantage of that.
Proof sketch

This looks too pessimistic. When $\frac{1}{p_{s,i}}$ 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.

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

This looks **too pessimistic**. When $\frac{1}{\tilde{p}_{s,i}}$ 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 $W$ itself as a random variable.

$$
\Pr \left( \|Y_t\| \geq \varepsilon \cap \|W\| \leq \sigma^2 \right) \leq t \exp\{ - \ldots \}
$$

Starting from an upper bound on $W$ that is still a r.v.

$$
W \leq \frac{1}{q^2} \sum_{j=1}^{\bar{q}} \sum_{i=1}^{t} \left( \operatorname{max}_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\tilde{p}_{s,i}^2} \right\} \right)^{v_i v_i^T} v_i v_i^T
$$
Proof sketch

This looks too pessimistic. When $\frac{1}{p_{s,i}}$ 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 $W$ itself as a random variable.

$$P(\|Y_t\| \geq \varepsilon \cap \|W\| \leq \sigma^2) \leq t \exp\{-\ldots\}$$

Starting from an upper bound on $W$ that is still a r.v.

$$W \lesssim \frac{1}{q^2} \sum_{j=1}^{\tilde{q}} \sum_{i=1}^{t} \left\{ \max_{s=0}^{t-1} \frac{z_{s,i,j}}{\tilde{p}_{s,i}^2} \right\} v_i v_i^T v_j v_j^T$$

This still has high variance: cannot simply apply martingale Bernstein
Proof sketch

\[ \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{\rho_{s,i}^2} \right\} \text{ is still hard to analyze, since it is the maximum of dependent variables} \]
Proof sketch

\[
\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\}
\]
is still hard to analyze, since it is the maximum of dependent variables.

Moreover \[
\max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\}
\]
depends on \[
\max_{s=0}^{t-1} \left\{ \frac{z_{s,i',j'}}{p_{s,i'}^2} \right\}
\]
Proof sketch

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

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

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

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

Moreover \[ \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\} \] depends on \[ \max_{s=0}^{t-1} \left\{ \frac{z_{s,i',j'}}{p_{s,i'}^2} \right\} \]

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

\[ \mathbb{P}(A \geq a) \geq \mathbb{P}(B \geq a) \iff \mathbb{P}(A \leq a) \leq \mathbb{P}(B \leq a). \]
Proof sketch

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

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

![Graph showing the distribution of $z_{s,i,j}$]
Proof sketch

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

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

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

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

$$\mathcal{P}\left(\max\left\{ \frac{1}{\bar{p}_{t,e}^{z_{t,e,j}}}, \frac{z_{t,e,j}+1}{\bar{p}_{t+1,e}} \right\} \leq a \mid \mathcal{F}_{\{t,m,N\}} \right)$$

$$\mathcal{P}\left(\frac{1}{w_{t,e,j}} \leq a \mid \mathcal{F}_{\{t,m,N\}} \right)$$

$$z_{t,e,j} = 1$$
Proof sketch

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

\[
P(\max\{ \frac{1}{\tilde{p}_{t,e}}, \frac{z_{t+1,e,j}}{\tilde{p}_{t+1,e}} \} \leq a \mid \mathcal{F}_{\{t,m,N\}}) = \begin{cases} 
0 & \text{for } a < 1 \\
1 - \frac{1}{a} & \text{for } 1 \leq a < \frac{\alpha}{p_{t,i}} \\
1 & \text{for } \frac{\alpha}{p_{t,i}} \leq a
\end{cases}
\]
Proof sketch

We can now unwind the proof

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

We can now unwind the proof

5 dominate \( \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\} \) with \( 1/w_{i,j} \)

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

We can now unwind the proof

5 dominate \( \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\} \) with \( 1/w_{i,j} \)

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

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

We can now unwind the proof

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

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

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

2. apply another stochastic dominance argument to bound

$$\mathbb{P}\left( |I_t| \geq 3\bar{q}d_{\text{eff}}(\gamma)_t \cap \left\{ \forall t' \in \{1, \ldots, t\} : \|P_t - \tilde{P}_t\|_2 \leq \varepsilon \right\} \right)$$
Proof sketch

We can now unwind the proof

5 dominate \( \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\} \) with \( 1/w_{i,j} \)

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

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

2 apply another stochastic dominance argument to bound
\[
\mathbb{P}\left( |I_t| \geq 3\overline{q}_d(\gamma)_t \cap \left\{ \forall t' \in \{1, \ldots, t\} : \|P_t - \tilde{P}_t\|_2 \leq \varepsilon \right\} \right)
\]

1 union bound
Proof sketch

We can now unwind the proof

5 dominate \( \max_{s=0}^{t-1} \left\{ \frac{z_{s,i,j}}{p_{s,i}^2} \right\} \) with \( 1/w_{i,j} \)

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

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

2 apply another stochastic dominance argument to bound
\[
P \left( |I_t| \geq 3\bar{q}d_{eff}(\gamma)_t \cap \left\{ \forall t' \in \{1, \ldots, t\} : \|P_t - \tilde{P}_t\|_2 \leq \varepsilon \right\} \right)
\]

1 union bound

0 Q.E.D.
Distributed sequential sampling for adaptive kernel DL

SQUEAK is a strictly sequential algorithm
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

\[ I_{1,2,3,4} = I_{1,2,3} + I_4 \]
\[ I_{1,2,3} + I_4 = I_{1,2} + I_3 \]
\[ I_{1,2} + I_3 = I_1 + I_2 \]
\[ I_1 + I_2 = I_1 \]
\[ D_1 = I_1 \]
\[ D_2 = I_2 \]
\[ D_3 = I_3 \]
\[ D_4 = I_4 \]

\( s = 4 \)
\( s = 3 \)
\( s = 2 \)
\( s = 1 \)
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
SQUEAK is a strictly sequential algorithm

DISQUEAK is the distributed equivalent

$h = 4$

$h = 3$

$h = 2$

$h = 1$
DISQUEAK

**Input**: Dataset $\mathcal{D}$, regularization $\gamma$, $\bar{q}$, $\epsilon$, **Output**: $\mathcal{I}_D$

1: Partition $\mathcal{D}$ into disjoint sub-datasets $\mathcal{D}_i$
2: Run SQUEAK on each $\mathcal{D}_i$, build set $S_1 = \{\mathcal{I}_{D_i}\}^k_{i=1}$
3: \textbf{for} $h = 1, \ldots, k - 1$ \textbf{do}
4: \hspace{1em} \textbf{if} $|S_h| > 1$ \textbf{then} \hspace{1em} $\triangleright$ Dict-Merge
5: \hspace{2em} Pick two dictionaries $\mathcal{I}_D, \mathcal{I}_D'$ from $S_h$
6: \hspace{2em} $\mathcal{I} = \mathcal{I}_D \cup \mathcal{I}_D'$
7: \hspace{2em} $\mathcal{I}_{D,D'} = \text{Dict-Update}(\mathcal{I})$ using Eq. (4)
8: \hspace{2em} Place $\mathcal{I}_{D,D'}$ back into $S_{h+1}$
9: \hspace{1em} \textbf{else}
10: \hspace{2em} $S_{h+1} = S_h$
11: \hspace{1em} \textbf{end if}
12: \hspace{1em} \textbf{end for}
13: Return $\mathcal{I}_D$, the last dictionary in $S_k$

\[
\tilde{\tau}_{D \cup D', i} = \frac{1 - 2\epsilon}{\gamma} (k_{i,i} - k_i^T S (S^T K S + \gamma I)^{-1} S^T k_i), \quad (4)
\]
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 DISQUEAK with \( \bar{q} = \mathcal{O}(\frac{\alpha}{\varepsilon^2} \log(\frac{n}{\delta})) \), then w.p. \( 1 - \delta \), for all nodes \( \{h, l\} \) in the merge tree

\[
(1) \quad \|\mathbf{P}_{\{h, l\}} - \tilde{\mathbf{P}}_{\{h, l\}}\|_2 \leq \varepsilon.
\]

\[
(2) \quad |\mathcal{I}_{\{h, l\}}| \leq \mathcal{O}(\bar{q}d_{\text{eff}}(\gamma)_{\{h, l\}}) \leq \mathcal{O}(\frac{\alpha}{\varepsilon^2} d_{\text{eff}}(\gamma)n \log(\frac{n}{\delta})).
\]

- 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                                                                 | $|\mathcal{I}_n|$                                                                 | Increm. |
|---------------|----------------------------------------------------------------------|---------------------------------------------------------------------------------|---------|
| **EXACT**     | $n^3 / n \cdot d_{\text{max}_n}^2 / \varepsilon$                   | $n / d_{\text{max}_n}^2 / \varepsilon$                                         | -       |
| Bach’13       | $n(|\mathcal{I}_n|)^2$                                               | $\left( \frac{\lambda_{\text{min}} + n \gamma \varepsilon}{\lambda_{\text{min}} - n \gamma \varepsilon} \right) d_{\text{eff}_n} + \frac{\text{Tr}(K_n)}{\gamma \varepsilon}$ | No      |
| A&M’15        | $\frac{\lambda_{\text{max}}^2}{\gamma^2} \cdot \frac{n^2 d_{\text{eff}_n}^3}{\varepsilon^2}$ | $\frac{\lambda_{\text{max}}}{\gamma} \cdot \frac{d_{\text{eff}_n}}{\varepsilon^2}$ | No      |
| Cal&al’16     | $\frac{n d_{\text{eff}_n}^3}{\varepsilon^2}$                        | $\frac{d_{\text{eff}_n}}{\varepsilon^2}$                                      | Yes     |
| SQUEAK        | $\frac{n d_{\text{eff}_n}^2}{\varepsilon^2}$                        | $\frac{d_{\text{eff}_n}}{\varepsilon^2}$                                      | Yes     |
| RLS-sampling  | $\frac{n d_{\text{eff}_n}^3}{\varepsilon^2}$                        | $\frac{d_{\text{eff}_n}}{\varepsilon^2}$                                      | -       |
| 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 $O(n)$ time barrier using DISQUEAK, with M&M’16 first to break $O(n^2)$ barrier

Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks
Conclusions

SQUEAK and DISQUEAK

First method (with guarantees) to break $O(n)$ time barrier using DISQUEAK, with M&M’16 first to break $O(n^2)$ barrier

Strong reconstruction guarantees, suitable for many downstream kernel (and not) tasks

Final dictionary can be updated if new samples arrive
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 $O(n)$ time barrier using DISQUEAK, with M&M’16 first to break $O(n^2)$ 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 $O(n)$ time barrier using DISQUEAK, with M&M’16 first to break $O(n^2)$ 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 $O(n)$ time barrier using DISQUEAK, with M&M’16 first to break $O(n^2)$ 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