Pack only the essentials: Adaptive dictionary learning for kernel ridge regression

Daniele Calandriello, Alessandro Lazaric, Michal Valko

Pack only the essentials: Adaptive dictionary learning for kernel ridge regression

Daniele Calandriello **Alessandro Lazaric** **Michal Valko**
SequeL team, INRIA Lille - Nord Europe, France
{daniele.calandriello, alessandro.lazaric, michal.valko}@inria.fr

1 Introduction

One of the major limits of kernel ridge regression (KRR) is that for n samples storing and manipulating the kernel matrix $\mathbf{K}_n$ requires $\mathcal{O}(n^2)$ space, which becomes rapidly unfeasible for large n. Many solutions focus on how to scale KRR by reducing its space (and time) complexity without compromising the prediction accuracy. A popular approach is to construct low-rank approximations of the kernel matrix by randomly selecting a subset of m columns from $\mathbf{K}_n$, thus reducing the space complexity to $\mathcal{O}(nm)$. These methods, often referred to as *Nyström approximations*, mostly differ in the distribution used to sample the columns of $\mathbf{K}_n$ and the construction of low-rank approximations. Both of these choices significantly affect the accuracy of the resulting approximation [5]. Bach [2] showed that uniform sampling preserves the prediction accuracy of KRR (up to ε) only when the number of columns m is proportional to the maximum degree of freedom of the kernel matrix. This may require sampling $\mathcal{O}(n)$ columns in datasets with high coherence [4] (i.e., a kernel matrix with weakly correlated columns). Alternatively, Alaoui and Mahoney [1] showed that sampling columns according to their ridge leverage scores (RLS) (i.e., a measure of the influence of a point on the regression) produces an accurate Nyström approximation with only a number of columns m proportional to the average degrees of freedom of the matrix, called *effective dimension*. Unfortunately, the complexity of computing RLS is comparable to solving KRR itself, making this approach unfeasible. However, Alaoui and Mahoney [1] proposed a fast method to compute a constant-factor approximation of the RLS and showed that accuracy and space complexity are close to the case of sampling with exact RLS at the cost of an extra dependency on the inverse of the minimal eigenvalue of the kernel matrix. Unfortunately, the minimal eigenvalue can be arbitrarily small in many problems. Calandriello et al. [3] addressed this issue by processing the dataset *incrementally* and updating estimates of the ridge leverage scores, effective dimension, and Nyström approximations on-the-fly. Although the space complexity of the resulting algorithm (INK-ESTIMATE) does not depend on the minimal eigenvalue anymore, it introduces a dependency on the largest eigenvalue of $\mathbf{K}_n$, which in the worst case can be as big as n. This can potentially reduce the advantage of the method. In this paper we introduce SQUEAK, a new algorithm that builds on INK-ESTIMATE, but uses *unnormalized* RLS and an improved RLS estimator. As a consequence, the algorithm is simpler, does not need to compute an estimate of the effective dimension for normalization, and it achieves a space complexity that is only a constant factor worse than sampling according to the exact RLS.

2 Background

Notation. We use curly capital letters $\mathcal{A}$ for collections and $|\mathcal{A}|$ for the number of entries in $\mathcal{A}$, upper-case bold letters $\mathbf{A}$ for matrices and lower-case bold letters $\mathbf{a}$ for vectors. We denote by $[\mathbf{A}]_{ij}$ and $[\mathbf{a}]_i$ the (i,j) element of a matrix and i-th element of a vector respectively. We use $\mathbf{e}_{n,i} \in \mathbb{R}^n$ for the i-th indicator vector of dimension n. Finally, the set of the first n integers is $[n] := \{1, \dots, n\}$.

Kernel regression. We consider a regression dataset $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$, with input $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ and output $y_t = f^\star(x_t) + \eta_t$, where $f^\star$ is an unknown target function and η_t is a zero-mean i.i.d. noise. We denote by $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive definite kernel function. Given the first t samples in $\mathcal{D}$, the kernel matrix $\mathbf{K}_t \in \mathbb{R}^{t\times t}$ is obtained as $[\mathbf{K}_t]_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ for any $i, j \in [t]$ and we denote by $\mathbf{y}_t, \mathbf{f}_t^\star \in \mathbb{R}^t$ the vectors with components y_i and $f^\star(\mathbf{x}_i)$, $i \in [t]$. Whenever a new point $\mathbf{x}_{t+1}$ arrives, the kernel matrix $\mathbf{K}_{t+1} \in \mathbb{R}^{t+1\times t+1}$ is obtained by *bordering* $\mathbf{K}_t$ as

$$\mathbf{K}_{t+1} = \left[\begin{array}{c|c} \mathbf{K}_t & \overline{\mathbf{k}}_{t+1} \\ \hline \overline{\mathbf{k}}_{t+1}^\intercal & k_{t+1} \end{array}\right] \tag{1}$$

where $\overline{\mathbf{k}}_{t+1} \in \mathbb{R}^t$ is such that $[\overline{\mathbf{k}}_{t+1}]_i = \mathcal{K}(\mathbf{x}_{t+1}, \mathbf{x}_i)$ for any $i \in [t]$ and $k_{t+1} = \mathcal{K}(\mathbf{x}_{t+1}, \mathbf{x}_{t+1})$. At any time t, the objective of kernel regression is to find the vector $\widehat{\mathbf{w}}_t \in \mathbb{R}^t$ that minimizes the regularized quadratic loss

$$\widehat{\mathbf{w}}_t = \arg\min_{\mathbf{w}} \|\mathbf{y}_t - \mathbf{K}_t\mathbf{w}\|^2 + \mu\|\mathbf{w}\|^2 = (\mathbf{K}_t + \mu\mathbf{I})^{-1}\mathbf{y}_t, \tag{2}$$

where $\mu \in \mathbb{R}$ is a regularization parameter. If μ is properly tuned, then $\widehat{\mathbf{w}}_t$ achieves a near-optimal risk $\mathcal{R}(\widehat{\mathbf{w}}_t) = \mathbb{E}_\eta\big[||\mathbf{f}_t^\star - \mathbf{K}_t\widehat{\mathbf{w}}_t||_2^2\big]$. Nonetheless, the computation of the final $\widehat{\mathbf{w}}_n$ requires $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space, which is infeasible for large datasets.

Nyström approximation. A common approach to reduce the complexity is to (randomly) select m columns of $\mathbf{K}_t$ according to some distribution $\mathbf{p}_t = \{p_{t,i}\}_{i=1}^t$ and construct the dictionary $\mathcal{I}_t = \{(i_j, \mathbf{k}_{t,i_j}, \widetilde{p}_{t,i_j})\}_{j=1}^m$, which contains the set of indices $i_j \in [t]$, the corresponding columns and their weights. Given a dictionary $\mathcal{I}_t$, the regularized Nyström approximation of $\mathbf{K}_t$ is obtained as

$$\widetilde{\mathbf{K}}_t = \mathbf{K}_t\mathbf{S}_t(\mathbf{S}_t^\mathsf{T}\mathbf{K}_t\mathbf{S}_t + \gamma\mathbf{I}_m)^{-1}\mathbf{S}_t^\mathsf{T}\mathbf{K}_t, \tag{3}$$

where the selection matrix $\mathbf{S}_t \in \mathbb{R}^{t\times m}$ is defined as $\mathbf{S}_t = [(\overline{q}\widetilde{p}_{t,i_1})^{-1/2}\mathbf{e}_{t,i_1}, \dots, (\overline{q}\widetilde{p}_{t,i_m})^{-1/2}\mathbf{e}_{t,i_m}]$, $\overline{q}$ is a constant, and γ is a regularization term (possibly different from μ). At this point, $\widetilde{\mathbf{K}}_t$ can be used to compute $\widetilde{\mathbf{w}}_t = (\widetilde{\mathbf{K}}_t + \mu\mathbf{I}_t)^{-1}\mathbf{y}_t$ efficiently using block inversion, reducing the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2 + m^3)$ time and from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$ space.

Ridge leverage scores. The accuracy of $\widetilde{\mathbf{K}}_t$ is strictly related to the distribution $\mathbf{p}_t$ used to construct the dictionary $\mathcal{I}_t$. In particular, Alaoui and Mahoney [1] showed that sampling according to the γ-ridge leverage scores (RLS) of $\mathbf{K}_t$ leads to an accurate Nyström approximation.

Definition 1. *Given $\mathbf{K}_t = \mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^\mathsf{T}$, the γ-ridge leverage score (RLS) of column $i \in [t]$ is*

$$\tau_{t,i} = \mathbf{k}_{t,i}^\mathsf{T}(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1}\mathbf{e}_{t,i} = \mathbf{e}_{t,i}^\mathsf{T}\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1}\mathbf{e}_{t,i}, \tag{4}$$

Furthermore, the effective dimension of the kernel is defined as $d_{\mathit{eff}}(\gamma)_t = \sum_{i=1}^t \tau_{t,i}$.

Similar to standard leverage scores (i.e., $\sum_j [U]_{i,j}^2$), RLSs measure the importance of each point $\mathbf{x}_i$ for the kernel regression. Furthermore, the sum of the RLSs is the effective dimension $d_{\text{eff}}(\gamma)_t$, which measures the intrinsic capacity of the kernel $\mathbf{K}_t$ when its spectrum is soft-thresholded by a regularization γ. Using RLS in constructing a Nyström approximation leads to the following result.

Proposition 1 (Alaoui and Mahoney [1])**.** *Let $\varepsilon \in [0,1]$ and $\mathcal{I}_n$ be the dictionary built with m columns selected proportionally to RLSs $\{\tau_{n,i}\}$. If $m = \mathcal{O}(\frac{1}{\varepsilon^2} d_{\mathit{eff}}(\gamma)_n \log(\frac{n}{\delta}))$, the Nyström approximation $\widetilde{\mathbf{K}}_n$ is a γ-approximation of $\mathbf{K}_t$, that is $\mathbf{0} \preceq \mathbf{K}_t - \widetilde{\mathbf{K}}_t \preceq \frac{\gamma}{1-\varepsilon}\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I})^{-1} \preceq \frac{\gamma}{1-\varepsilon}\mathbf{I}$ and the risk of $\widetilde{\mathbf{w}}_t$ is $\mathcal{R}(\widetilde{\mathbf{w}}_t) \le (1 + \frac{\gamma}{\mu}\frac{1}{1-\varepsilon})\mathcal{R}(\widehat{\mathbf{w}}_t)$.*

Unfortunately, computing exact RLS requires storing $\mathbf{K}_n$, and has the same $\mathcal{O}(n^2)$ space requirement as solving Eq. 2. In the next section, we introduce SQUEAK, an RLS-based incremental algorithm able to preserve the same accuracy of Prop. 1 without requiring to know the RLS in advance, and that generates a dictionary only a constant factor larger than exact RLS sampling.

3 Incremental Nyström approximation with ridge leverage scores

SQUEAK (Alg. 1) builds on the INK-ESTIMATE algorithm [3] with the major algorithmic difference that the sampling probabilities are computed directly on estimates $\tau_{t,i}$ without renormalizing them by an estimate of $d_{\text{eff}}(\gamma)_t$. SQUEAK introduces two key elements: **1)** an improved, accurate estimator of the RLS and **2)** an incremental sampling scheme for the construction of the dictionary $\mathcal{I}_t$.

1) Estimation of RLS. We introduce an RLS estimator that improves on [3], showing that it can be efficiently computed. At any time t, let $Q_t = \sum_i Q_{t,i}$ be the number of columns $|\mathcal{I}_t|$ contained in the dictionary at time t, and $\mathbf{S}_t \in \mathbb{R}^{t\times Q_t}$ the selection matrix constructed so far. Let $\overline{\mathbf{S}}_{t+1} \in \mathbb{R}^{(t+1)\times(Q_t+\overline{q})}$ be constructed as $[\mathbf{S}_t, (\overline{q})^{-1/2}\mathbf{e}_{t+1,t+1}, \dots, (\overline{q})^{-1/2}\mathbf{e}_{t+1,t+1}]$ by adding $\overline{q}$ copies of $\mathbf{e}_{t+1,t+1}$ to the selection matrix. Denoting $\alpha = (1+\varepsilon)/(1-\varepsilon)$, we define the RLS estimator as

$$\widetilde{\tau}_{t+1,i} = \frac{1+\varepsilon}{\alpha\gamma}\left(k_{i,i} - \mathbf{k}_{t+1,i}\overline{\mathbf{S}}\left(\overline{\mathbf{S}}^\mathsf{T}\mathbf{K}_{t+1}\overline{\mathbf{S}} + \gamma\mathbf{I}\right)^{-1}\overline{\mathbf{S}}^\mathsf{T}\mathbf{k}_{t+1,i}\right). \tag{5}$$

Algorithm 1 The SQUEAK algorithm

Input: Dataset $\mathcal{D}$, regularization $\gamma, \mu, \overline{q}$

Output: $\widetilde{\mathbf{K}}_n, \widetilde{\mathbf{w}}_n$

1: Initialize $\mathcal{I}_0$ as empty, $\widetilde{p}_{1,0} = 1$
2: **for** $t = 0, \dots, n-1$ **do**
3: Receive new column $[\overline{\mathbf{k}}_{t+1}, k_{t+1}]$
4: Compute α-approximate RLS $\{\widetilde{\tau}_{t+1,i} : i \in \mathcal{I}_t \cup \{t+1\}\}$, using $\mathcal{I}_t$, $[\overline{\mathbf{k}}_{t+1}, k_{t+1}]$, and Eq. 5
5: Set $\widetilde{p}_{t+1,i} = \max\{\min\{\widetilde{\tau}_{t+1,i},\ \widetilde{p}_{t,i}\},\ \widetilde{p}_{t,i}/2\}$
6: Initialize $\mathcal{I}_{t+1} = \emptyset$ — DICT-UPDATE
7: **for all** $j \in \{1, \dots, t\}$ **do** — SHRINK
8: $Q_{t,j} = |\{i = j : i \in \mathcal{I}_t\}|$
9: **if** $Q_{t,j} \neq 0$ **then**
10: $Q_{t+1,j} \sim \mathcal{B}(\widetilde{p}_{t+1,j}/\widetilde{p}_{t,j}, Q_{t,j})$
11: Add $Q_{t+1,j}$ copies of $(j, \mathbf{k}_{t+1,j}, \widetilde{p}_{t+1,j})$ to $\mathcal{I}_{t+1}$.
12: **end if**
13: **end for**
14: $Q_{t+1,t+1} \sim \mathcal{B}(\widetilde{p}_{t+1,t+1}, \overline{q})$ — EXPAND
15: Add $Q_{t+1,t+1}$ copies of $(t+1, \mathbf{k}_{t+1,t+1}, \widetilde{p}_{t+1,t+1})$ to $\mathcal{I}_{t+1}$
16: **end for**
17: Compute $\widetilde{\mathbf{K}}_n$ using $\mathcal{I}_n$ and Eq. 3
18: Compute $\widetilde{\mathbf{w}}_n$ using $\widetilde{\mathbf{K}}_n, \mathbf{y}_n$

If $Q_t \geq \overline{q}$, then $\widetilde{\tau}_{t+1,i}$ can be computed in $\mathcal{O}(Q_t^3)$ time ($\mathcal{O}(Q_t)$ to compute $\mathbf{k}_{t+1,i}\overline{\mathbf{S}}$ and $\mathcal{O}(Q_t)^3$ to invert the inner matrix) and $\mathcal{O}(Q_t^2)$ space. If $Q_t < \overline{q}$ the same applies with $\overline{q}$ replacing Q_t. Furthermore, we have the following guarantee.

Lemma 1. *Assume that the dictionary $\mathcal{I}_t$ induces a γ-approximate kernel $\widetilde{\mathbf{K}}_t$. Then for all i such that $i \in \{\mathcal{I}_t \cup \{t+1\}\}$, $\widetilde{\tau}_{t+1,i}$ computed using Eq.5 is an α-approximation of the RLS $\tau_{t,i}$, that is $\tau_{t+1,i}(\gamma)/\alpha \leq \widetilde{\tau}_{t+1,i} \leq \tau_{t+1,i}(\gamma)$.*

2) Sequential sampling. At each time step t, SQUEAK receives a new column $[\overline{\mathbf{k}}_{t+1}, k_{t+1}]$. This can be implemented either by having a separate algorithm that constructs each column sequentially and streams it to SQUEAK, or by storing just the samples (with an additional $\mathcal{O}(td)$ space complexity) and computing the column once. Adding a new column to the matrix can either decrease the importance of columns already observed (i.e., if they are correlated to the new column) or leave it unchanged (i.e., if they are orthogonal) and thus the RLS evolves as $\tau_{t+1,i} \leq \tau_{t,i}$ [3, App. A, Lem. 4]. In the DICT-UPDATE loop, the dictionary is updated to reflect the change in importance of old columns (e.g., $p_{t,i} = \tau_{t,i}$ may decrease) and to add the new column proportionally to its RLS $\tau_{t+1,t+1}$. The dictionary $\mathcal{I}_t$, and the new column are used to compute new approximate RLS $\widetilde{\tau}_{t+1,i}$ as in Eq. 5, which in turn define the new sampling probabilities $\widetilde{p}_{t+1,i}$. The DICT-UPDATE phase is composed of two steps. For each index $i \in [t]$, the SHRINK step counts the number of copies $Q_{t,i}$ present in $\mathcal{I}_t$, and then draws a sample from the binomial $\mathcal{B}(\widetilde{p}_{t+1,i}/\widetilde{p}_{t,i}, Q_{t,i})$, where taking $\widetilde{p}_{t+1,i} = \min\{\widetilde{\tau}_{t+1,i},\ \widetilde{p}_{t,i}\}$ ensures that the binomial probability at L10 is well defined. The more $\widetilde{p}_{t+1,i}$ is lower than $\widetilde{p}_{t,i}$, the more $Q_{t+1,i}$ will be lower than $Q_{t,i}$. If the probability $\widetilde{p}_{t+1,i}$ continues to decrease over time, it is also possible that $Q_{t+1,i}$ is decreased to zero, and column i is completely dropped from the dictionary. Intuitively, the SHRINK step stochastically reduces the size of the dictionary to reflect the reductions of the RLSs. Conversely, the EXPAND step add the new column to the dictionary with a number of copies (from 0 to $\overline{q}$) which depends on its estimated relevance $\widetilde{p}_{t+1,t+1}$. Unlike in [3], the approximate probabilities $\widetilde{p}_{t,i}$ are not obtained by normalizing the approximate $\widetilde{\tau}_{t,i}$ by an estimate of the effective dimension and thus they do not necessarily sum to one. Yet, we guarantee that $\widetilde{p}_{t,i} \leq p_{t,i} \leq 1$ by construction. Note that SQUEAK *never* estimates again the RLS of a columns dropped from $\mathcal{I}_t$. Moreover, computing Eq. (5) requires only to construct the kernel sub-matrix for samples whose indices are in $\mathcal{I}_t$. Therefore, if we are only interested in estimating the approximate RLS $\widetilde{\tau}_{t,i}$ and not the regression weights $\widetilde{\mathbf{w}}_t$, SQUEAK is the first RLS sampling algorithm that can operate in a single pass over the dataset (store and access only the samples in $\mathcal{I}_t$ instead of the whole $\mathcal{D}_t$), without ever constructing the whole matrix. Thm. 1 guarantees that SQUEAK succeeds in returning a γ-approximate matrix $\widetilde{\mathbf{K}}_n$ with high probability.

	Time	$\lvert\mathcal{I}_n\rvert$ (Total space = $\mathcal{O}(n\lvert\mathcal{I}_n\rvert)$)	Acc. loss	Increm.
EXACT	n^3	n	1	N/A
Bach [2]	$\frac{nd_{\max}n^2}{\varepsilon} + \frac{d_{\max}n^3}{\varepsilon}$	$\frac{d_{\max},n}{\varepsilon}$	$(1+4\varepsilon)$	No
Alaoui and Mahoney [1]	$n(\lvert\mathcal{I}_n\rvert)^2$	$\left(\frac{\lambda_{\min}+n\mu\varepsilon}{\lambda_{\min}-n\mu\varepsilon}\right) d_{\text{eff}}(\gamma)_n + \frac{\mathrm{Tr}(\mathbf{K}_n)}{\mu\varepsilon}$	$(1+2\varepsilon)^2$	No
Calandriello et al. [3]	$\frac{\lambda_{\max}^2}{\gamma^2}\frac{n^2 d_{\text{eff}}(\gamma)_n^2}{\varepsilon^2}$	$\frac{\lambda_{\max}}{\gamma}\frac{d_{\text{eff}}(\gamma)_n}{\varepsilon^2}$	$(1+2\varepsilon)^2$	Yes
SQUEAK	$\frac{n^2 d_{\text{eff}}(\gamma)_n^2}{\varepsilon^2}$	$\frac{d_{\text{eff}}(\gamma)_n}{\varepsilon^2}$	$(1+2\varepsilon)^2$	Yes
RLS-SAMPLING	$\frac{nd_{\text{eff}}(\gamma)_n^2}{\varepsilon^2}$	$\frac{d_{\text{eff}}(\gamma)_n}{\varepsilon^2}$	$(1+2\varepsilon)^2$	N/A

Table 1: Comparison of Nyström methods. $\lambda_{\max}$ and $\lambda_{\min}$ refer to largest and smallest eigenvalues of $\mathbf{K}_n$.

Theorem 1. *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 Alg. 1 with parameter $\overline{q} = \mathcal{O}(\frac{\alpha}{\varepsilon^2}\log(\frac{n}{\delta}))$ to compute a sequence of random dictionaries $\mathcal{I}_t$ each with a random number of entries $\lvert\mathcal{I}_t\rvert$, then with probability $1-\delta$, for all iterations $t \in [n]$*

(1) *The Nyström approximation $\widetilde{\mathbf{K}}_t$ (Eq. 3) associated with $\mathcal{I}_t$ is a γ-approximation of $\mathbf{K}_t$.*

(2) *The number of stored columns is $\lvert\mathcal{I}_t\rvert = \sum_i Q_{t,i} \leq \mathcal{O}(\overline{q} d_{\text{eff}}(\gamma)_t) \leq \mathcal{O}(\frac{\alpha}{\varepsilon^2} d_{\text{eff}}(\gamma)_n \log(\frac{n}{\delta}))$.*

(3) *The solution $\widetilde{\mathbf{w}}_t$ satisfies $\mathcal{R}(\widetilde{\mathbf{w}}_t) \leq (1 + \frac{\gamma}{\mu}\frac{1}{1-\varepsilon})\mathcal{R}(\widehat{\mathbf{w}}_t)$.*

As the previous theorem holds for any $t \in [n]$, SQUEAK has any-time guarantees on its space complexity, approximation, and risk performance. In fact, **(1)** combined with Lem. 1 shows that, at all steps, $\widetilde{\tau}_{t,i}$ are α-approximate RLSs estimates. Since adding a column to $\mathbf{K}_t$ can only increase the effective dimension (i.e., $d_{\text{eff}}(\gamma)_t \leq d_{\text{eff}}(\gamma)_{t+1}$) [3, App. A, Lem. 5], from **(2)** we see that the number of columns stored by SQUEAK over iterations never exceeds the budget $\mathcal{O}(d_{\text{eff}}(\gamma)_n \log(n))$ required by sampling columns according to the exact RLS computed over the whole dataset. Notice that this is obtained by automatically increasing the dictionary size (and space occupation) over time to adapt to the growth in effective dimension of the data, which does not need to be known in advance. Furthermore, if the size of the dictionary grows too large w.r.t. the memory available, we can still terminate the algorithm knowing that the intermediate dictionary returned is a good approximation of the part of dataset processed. We can also restart the process with a larger γ, since $d_{\text{eff}}(\gamma)_n$ is inversely proportional to γ. The tradeoffs of this approach are quantified by **(3)**, which shows that all solutions $\widetilde{\mathbf{w}}_t$ incur a risk only a factor roughly $(1+\gamma/\mu)$ away from the corresponding exact solution $\widehat{\mathbf{w}}_t$. This means that choosing a small $\gamma < \mu$ allows to achieve a risk close to the exact solution for a large range of μ, at the cost of increasing the space, while larger γ require less space but it may prevent from tuning μ optimally. Finally, it is important to notice that even in the worst case $d_{\text{eff}}(\gamma)_n = n$, SQUEAK requires only $\log(n)$ more space than storing the whole matrix.

4 Discussion

Table 1 compares several Nyström approximation methods w.r.t. their space complexity and risk. For all methods, we omit $\mathcal{O}(\log(n))$ factors. The space complexity of uniform sampling [2] scales with the maximal degree of freedom $d_{\max}$. Since $d_{\max} = n\max_i \tau_{n,i} \geq \sum_i \tau_{n,i} = d_{\text{eff}}(\gamma)_n$, uniform sampling is often outperformed by RLS sampling. While Alaoui and Mahoney [1] also sample according to RLS, their two-pass estimator is not very accurate. In particular, the first pass requires to sample $\mathcal{O}\left(n\mu\varepsilon/(\lambda_{\min} - n\mu\varepsilon)\right)$ columns, which quickly grows above n^2 when $\lambda_{\min}$ becomes small. Finally, [3] require that the maximum dictionary size is fixed in advance, which implies some knowledge of the effective dimensions $d_{\text{eff}}(\gamma)_n$, and requires estimating both $\widetilde{\tau}_{t,i}$ and $\widetilde{d}_{\text{eff}}(\gamma)_t$. In particular, this extra estimation effort causes an additional $\lambda_{\max}/\gamma$ factor to appear in the space complexity. This factor cannot be easily estimated, and causes a space complexity of n^3 in the worst case. We also include RLS-SAMPLING, a fictitious algorithm that receives the exact RLS in input, as an ideal baseline for all RLS sampling algorithms. From the table, we can therefore see that SQUEAK achieves the same space complexity (up to constant factors) as knowing the RLS in advance. Moreover, although in this paper we only considered fixed design KRR, γ-approximation guarantees for $\widetilde{\mathbf{K}}_n$ are commonly used in similar problems such as random design KRR, or Kernel PCA. Finally, with a more careful analysis, we can generalize SQUEAK and its guarantees to the distributed setting, where multiple machines construct dictionaries in parallel on separate datasets, and then recursively merge them to construct a dictionary for the union of the datasets.

References

[1] Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel methods with statistical guarantees. In *Neural Information Processing Systems*, 2015.

[2] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, 2013.

[3] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Analysis of Nyström method with sequential ridge leverage scores. In *Uncertainty in Artificial Intelligence*, 2016.

[4] Alex Gittens and Michael W Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *International Conference on Machine Learning*, 2013.

[5] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Neural Information Processing Systems*, 2015.