Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

COLUMN SUBSET SELECTION AND NYSTRÖM APPROXIMATION VIA CONTINUOUS OPTIMIZATION

Anant Mathur
Sarat Moka
Zdravko Botev

School of Mathematics and Statistics University of
New South Wales
High Street
Kensington Sydney, NSW 2052, AUSTRALIA

ABSTRACT

We propose a continuous optimization algorithm for the Column Subset Selection Problem (CSSP) and Nyström approximation. The CSSP and Nyström method construct low-rank approximations of matrices based on a predetermined subset of columns. It is well known that choosing the best column subset of size k is a difficult combinatorial problem. In this work, we show how one can approximate the optimal solution by defining a penalized continuous loss function that is minimized via stochastic gradient descent. We show that the gradients of this loss function can be estimated efficiently using matrix-vector products with a data matrix $\mathbf{X}$ in the case of the CSSP or a kernel matrix $\mathbf{K}$ in the case of the Nyström approximation. We provide numerical results for a number of real datasets showing that this continuous optimization is competitive against existing methods.

1 INTRODUCTION

Recent advances in the technological ability to capture and collect data have meant that high-dimensional datasets are now ubiquitous in the fields of engineering, economics, finance, biology, and health sciences to name a few. In the case where the data collected is not labeled it is often desirable to obtain an accurate low-rank approximation for the data which is relatively low-cost to obtain and memory efficient. Such an approximation is useful to speed up downstream matrix computations that are often required in large-scale learning algorithms. The Column Subset Selection Problem (CSSP) and its popular variant Nyström method are two such tools that generate low-rank approximations based on a subset of columns that typically represent either data instances or features of a dataset. The chosen subset of columns are commonly referred to as "landmark" points. The choice of landmark points determines how accurate the low-rank approximation is.

The challenge in the CSSP is to select the best k columns of a data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ that span its column space. That is, for any binary vector $\boldsymbol{s} \in \{0,1\}^n$, compute

$$\underset{\mathbf{s}\in\{0,1\}^n}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{P}_s\mathbf{X}\|_F^2, \quad \text{subject to } \|\boldsymbol{s}\|_0 \le k, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius matrix norm, $\|\boldsymbol{s}\|_0 = \sum_{j=1}^n I(s_j = 1)$ and $\mathbf{P}_s$ is the projection matrix onto $\operatorname{span}\{\boldsymbol{x}_j : s_j = 1, j = 1,\dots,n\}$ ($\boldsymbol{x}_j$ being the j-th column of $\mathbf{X}$).

Solving this combinatorial problem exactly is known to be NP-complete (Shitov 2021), and is practically infeasible even when k is of moderate size. We propose a novel continuous optimization algorithm to approximate the exact solution to this problem. While an optimization approach via Group Lasso exists

for the convex relaxation of this problem (Bien et al. 2010), to the best of our knowledge, no continuous optimization method has been developed to solve the highly non-convex combinatorial problem (1). To introduce our approach for the CSSP, instead of searching over binary vectors $\boldsymbol{s} \in \{0,1\}^n$, we consider the hyper-cube $[0,1]^n$ and define for each $\boldsymbol{t} \in [0,1]^n$ a matrix $\widetilde{\mathbf{P}}(\boldsymbol{t})$ which allows the following well-defined penalized continuous extension of the exact problem,

$$\underset{\boldsymbol{t}\in[0,1]^n}{\operatorname{argmin}} \|\mathbf{X} - \widetilde{\mathbf{P}}(\boldsymbol{t})\mathbf{X}\|_F^2 + \lambda \sum_{j=1}^{n} t_j.$$

The parameter $\lambda > 0$ plays an analogous role to that of the regularization parameter in regularized linear regression methods (Tibshirani 1996) and controls the sparsity of the solution, that is, the size of k. Two aspects of this continuous extension make it useful for approximating the exact solution. Firstly, the continuous loss agrees with the discrete loss at every corner point $\boldsymbol{s} \in \{0,1\}^n$ of the hypercube $[0,1]^n$, and secondly, for large datasets the gradient can be estimated via an unbiased stochastic estimate. To obtain an approximate solution to the exact problem, *stochastic gradient descent* (SGD) is implemented on the penalized loss. After starting at an interior point of the hyper-cube, under SGD, the vector $\boldsymbol{t}$ moves towards a corner point, and some of the $\boldsymbol{t}_j$'s exhibit shrinkage to zero. It is these values that indicate which columns in $\mathbf{X}$ should not be selected as landmark points.

The Nyström approximation (Williams and Seeger 2000; Drineas et al. 2005) is a popular variant of the CSSP for positive semi-definite kernel matrices. The Nyström method also constructs a low-rank approximation $\widehat{\mathbf{K}} \in \mathbb{R}^{n\times n}$ to the true kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ using a subset of columns. Once the k columns are selected, $\widehat{\mathbf{K}}$ (in factored form) takes $O(k^3)$ additional time to compute, requires $O(nk)$ space to store, and can be manipulated quickly in downstream applications, e.g., inverting $\widehat{\mathbf{K}}$ takes $O(nk^2)$ time. In addition to the continuous extension for the CSSP, in this paper, we provide a continuous optimization algorithm that can approximate the best k columns to be used to construct $\widehat{\mathbf{K}}$

The continuous algorithm for the CSSP formulated in this paper utilizes SGD where at each iteration one can estimate the gradient with a cost of $O(mn)$. We show that the gradients of the penalized continuous loss can be estimated via linear solves with random vectors that are approximated with the conjugate gradient algorithm (CG) (Golub and Van Loan 1996), which itself is an iterative algorithm that only requires matrix-vector multiplications (MVMs) with the $m \times n$ matrix $\mathbf{X}$. Similarly, for the Nyström method we show that at each step of the gradient descent, the gradient can be estimated in $O(n^2)$ time requiring only matrix-vector multiplications with the kernel matrix $\mathbf{K}$. This is especially useful in cases where we only have access to a black-box MVM function. The fact that both these algorithms require only matrix-vector multiplications to estimate the gradients lends itself to utilizing GPU hardware acceleration. Moreover, the computations in the proposed algorithm can exploit the sparsity that is achieved by working only with the columns of $\mathbf{X}$ that are selected by the algorithm at any given iteration. We refer the reader to (Mathur et al. 2023) for proofs of the results presented in this paper.

1.1 Related Work

There exists extensive literature on random sampling methods for the approximation of the exact CSSP and Nyström problem. Sampling techniques such as adaptive sampling (Deshpande and Vempala 2006), ridge leverage scores (Gittens and Mahoney 2013; Musco and Musco 2017; Alaoui and Mahoney 2015) attempt to sample "important" and "diverse" columns. In particular, recent attention has been paid to Determinantal Point Processes (DPPs) (Hough et al. 2006; Derezinski and Mahoney 2021). DPPs provide strong theoretical guarantees (Derezinski et al. 2020) for the CSSP and Nyström approximation and are amenable to efficient numerical implementation (Calandriello et al. 2020). Outside of sampling methods, iterative methods such as Greedy selection (Farahat et al. 2011; Farahat et al. 2013) have been shown to perform well in practice and exhibit provable guarantees (Altschuler et al. 2016).

Column selection has been extensively studied in the supervised context of linear regression (more commonly referred to as feature or variable selection). Penalized regression methods such as the Lasso (Tibshirani 1996) have been widely applied to select columns of a predictor matrix that best explain a response vector. The canonical k-best subset or l_0-penalized regression problem is another penalized regression method, where the goal is to find the best subset of k predictors that best fit a response $\boldsymbol{y}$ (Beale et al. 1967; Hocking and Leslie 1967). The recently proposed *Continuous Optimization Method Towards Best Subset Selection* (COMBSS) algorithm (Moka et al. 2022) attempts to solve the l_0-penalized regression problem, which falls under supervised learning, by minimizing a continuous loss that approximates the exact solution. The algorithm we propose for the CSSP in this paper can be viewed as an adaptation of COMBSS to the unsupervised setting. In this setting, the goal is to find the best subset of size k for a multiple multivariate regression model where both the response and predictor matrix are $\mathbf{X}$. Interestingly, this framework can be extended to include a continuous selection loss for the Nyström approximation.

The rest of the paper is structured as follows. In Section 2 we describe the continuous extension for the CSSP and the Nyström method. In Section 3 we provide steps for the efficient implementation of our proposed continuous algorithm on large matrices and in Section 3.3 we provide numerical results on a variety of real datasets.

2 CONTINUOUS LOSS FOR LANDMARK SELECTION

In this section, we formally define the CSSP and the best size k-Nyström approximation. Then, we provide the mathematical setup for the continuous extension of the exact problem.

2.1 Column Subset Selection

Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ and for any binary vector $\boldsymbol{s} = (s_1,\ldots,s_n)^\top \in \{0,1\}^n$, let $\mathbf{X}_{[\boldsymbol{s}]}$ denote the matrix of size $m \times \|\boldsymbol{s}\|_0$ keeping only columns j of $\mathbf{X}$ where $s_j = 1$, for $j = 1,\ldots,n$. Then for every integer $k \le n$ the CSSP finds the solution to (1) where $\mathbf{P}_{\boldsymbol{s}} := \mathbf{X}_{[\boldsymbol{s}]}\mathbf{X}^{\dagger}_{[\boldsymbol{s}]}$ († denotes Moore–Penrose inverse) is the projection matrix onto $\text{span}\{\mathbf{x}_j : s_j = 1\}$ and $\boldsymbol{x}_j$ is the j-th column of $\mathbf{X}$. By expanding the Frobenius norm it is easy to see that the discrete problem (1) can be re-formulated as,

$$\underset{\mathbf{s}\in\{0,1\}^n}{\text{argmin}} -\text{tr}\left[\mathbf{X}^\top \mathbf{P}_{\boldsymbol{s}}\mathbf{X}\right], \quad \text{subject to } \|\boldsymbol{s}\|_0 \le k.$$

We now define a new matrix function on $\boldsymbol{t} \in [0,1]^n$ which acts as a continuous generalization of $\mathbf{P}_{\boldsymbol{s}}$.

Definition 1 For $\boldsymbol{t} = (t_1,\ldots,t_n)^\top \in [0,1]^n$, define $\mathbf{T} := \text{Diag}(\boldsymbol{t})$ as the diagonal matrix with diagonal elements $t_1,\ldots,t_n$ and

$$\widetilde{\mathbf{P}}(\boldsymbol{t}) := \mathbf{XT}\left[\mathbf{TX}^\top\mathbf{XT} + \delta(\mathbf{I}-\mathbf{T}^2)\right]^{\dagger}\mathbf{TX}^\top,$$

where $\delta > 0$ is a fixed constant.

Although not explicitly stated in Moka et al. (2022), $\widetilde{\mathbf{P}}(\boldsymbol{t})$ is used as the continuous generalization for the hat matrix $\mathbf{P}_{\boldsymbol{s}}$ to solve the l_0-penalized regression problem.

The main difference between this definition and traditional sampling methods is that instead of multiplying $\mathbf{X}$ by a sampling matrix to obtain a smaller matrix $\mathbf{X}_{[\boldsymbol{s}]}$ we compute the matrix $\mathbf{XT}$ which weights column j of $\mathbf{X}$ by the parameter $t_j \in [0,1]$. This difference allows one to construct a generalized projection matrix that has smooth transitions from one corner to another, thus allowing the use of continuous optimization methods. Intuitively, the matrix $\mathbf{TX}^\top\mathbf{XT} + \delta(\mathbf{I}-\mathbf{T}^2)$ can be viewed as a convex combination of the matrices $\mathbf{X}^\top\mathbf{X}$ and $\delta\mathbf{I}$.

From an evaluation standpoint, the pseudo-inverse need not be evaluated for any interior point in this newly defined function. We remark that for any $\boldsymbol{t} \in [0,1)^n$ the matrix inverse in Definition 1 exists and

therefore,

$$\widetilde{\mathbf{P}}(\boldsymbol{t}) = \mathbf{X}\mathbf{T}\left[\mathbf{T}\mathbf{X}^\top\mathbf{X}\mathbf{T} + \delta(\mathbf{I}-\mathbf{T}^2)\right]^{-1}\mathbf{T}\mathbf{X}^\top.$$

We now state two results for the function $\widetilde{\mathbf{P}}(\boldsymbol{t})$ and its relationship with the projection matrix $\mathbf{P}_s$. The following Lemmas (1 and 2) are extensions of the results stated in Moka et al. (2022).

Lemma 1 For any binary vector $\boldsymbol{s} \in \{0,1\}^n$, $\widetilde{\mathbf{P}}(\boldsymbol{s})$ exists and

$$\widetilde{\mathbf{P}}(\boldsymbol{s}) = \mathbf{P}_{\boldsymbol{s}} = \mathbf{X}_{[\boldsymbol{s}]}\mathbf{X}^\dagger_{[\boldsymbol{s}]}.$$

Lemma 2 $\widetilde{\mathbf{P}}(\boldsymbol{t})$ is continuous element-wise over $[0,1]^n$. Moreover, for any sequence $\boldsymbol{t}^{(1)}, \boldsymbol{t}^{(2)} \cdots \in [0,1)^n$ converging to $\boldsymbol{t} \in [0,1]^n$, the limit $\lim_{l\to\infty}\widetilde{\mathbf{P}}(\boldsymbol{t}^{(l)})$ exists and

$$\lim_{l\to\infty}\widetilde{\mathbf{P}}(\boldsymbol{t}^{(l)}) = \widetilde{\mathbf{P}}(\boldsymbol{t}).$$

We note that the proof of Lemma 2 follows identically to the proof of *Theorem 3* in Moka et al. (2022) where it is stated that the function $\|\boldsymbol{y} - \widetilde{\mathbf{P}}(\boldsymbol{t})\boldsymbol{y}\|_2^2$ is continuous over $[0,1]^n$ for any fixed vector $\boldsymbol{y} \in \mathbb{R}^n$.

Given $\widetilde{\mathbf{P}}(\boldsymbol{t})$ is continuous on $[0,1]^n$ and agrees with $\mathbf{P}_{\boldsymbol{s}}$ at every corner point we can define the continuous generalization of the exact problem (1),

$$\operatorname*{argmin}_{\boldsymbol{t}\in[0,1]^n} -\operatorname{tr}\left[\mathbf{X}^\top\widetilde{\mathbf{P}}(\boldsymbol{t})\mathbf{X}\right], \quad \text{subject to } \sum_{j=1}^n t_j \le k.$$

Instead of solving this constrained problem, for a tunable parameter λ, we consider minimizing the Lagrangian function,

$$\operatorname*{argmin}_{\boldsymbol{t}\in[0,1]^n} f_\lambda(\boldsymbol{t}), \quad f_\lambda(\boldsymbol{t}) := -\operatorname{tr}\left[\mathbf{X}^\top\widetilde{\mathbf{P}}(\boldsymbol{t})\mathbf{X}\right] + \lambda\sum_{j=1}^n t_j.$$

In Section 3 we reformulate this box-constrained problem into an equivalent unconstrained problem via a nonlinear mapping $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{w})$ for $\boldsymbol{w} \in \mathbb{R}^n$ that forces $\boldsymbol{t}$ to be in the hypercube $[0,1]^n$. We solve this optimization via continuous gradient descent. To this end, we need to evaluate the gradient $\nabla f_\lambda(\boldsymbol{t})$ for any interior point.

Lemma 3 Let $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$, $\mathbf{Z} = \mathbf{K} - \delta\mathbf{I}$ and $\mathbf{L}_{\boldsymbol{t}} = \mathbf{T}\mathbf{Z}\mathbf{T} + \boldsymbol{\delta}\mathbf{I}$. Then, for $\boldsymbol{t} \in (0,1)^n$,

$$\nabla f_\lambda(\boldsymbol{t}) = 2\operatorname{Diag}\left[\mathbf{L}_{\boldsymbol{t}}^{-1}\mathbf{T}\mathbf{K}^2\left(\mathbf{T}\mathbf{L}_{\boldsymbol{t}}^{-1}\mathbf{T}\mathbf{Z} - \mathbf{I}\right)\right] + \lambda\mathbf{1}.$$

Evaluating $\nabla f_\lambda(\boldsymbol{t})$ has a computational complexity of $O(n^3)$ due to the required inversion of $\mathbf{L}_t$. In Section 3 we detail an unbiased estimate for $\nabla f_\lambda(\boldsymbol{t})$ which utilizes the CG algorithm, where the most expensive operations involved are matrix-vector multiplications with $\mathbf{X}$ and $\mathbf{X}^\top$, which reduces the computational complexity to $O(mn)$.

2.2 Nyström Method

We now turn our attention to defining a continuous objective for the landmark points in the Nyström approximation. We consider optimizing the landmark points first with respect to the trace matrix norm and then to the Frobenius matrix norm.

In many applications, we are interested in obtaining a low-rank approximation to a *kernel* matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$. Consider an input space $\mathscr{X}$ and a positive semi-definite kernel function $h : \mathscr{X}\times\mathscr{X} \to \mathbb{R}$. Given a set of n input points $\boldsymbol{x}'_1, ..., \boldsymbol{x}'_n \in \mathscr{X}$, the kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ is defined by $\mathbf{K}_{i,j} = h(\boldsymbol{x}'_i, \boldsymbol{x}'_j)$ and is positive semi-definite.

For any binary vector $\boldsymbol{s} \in \{0,1\}^n$ let $\mathbf{K}_{[\boldsymbol{s}]}$ be the $n \times \|\boldsymbol{s}\|_0$ matrix with columns indexed by $\{j : s_j = 1\}$ and $\mathbf{K}_{[\boldsymbol{s},\boldsymbol{s}]}$ be the $\|\boldsymbol{s}\|_0 \times \|\boldsymbol{s}\|_0$ principal sub-matrix indexed by $\{j : s_j = 1\}$. The Nyström low-rank approximation for $\mathbf{K}$ is given by,

$$\widehat{\mathbf{K}}_s := \mathbf{K}_{[\boldsymbol{s}]} \mathbf{K}^{\dagger}_{[\boldsymbol{s},\boldsymbol{s}]} \mathbf{K}^{\top}_{[\boldsymbol{s}]}.$$

The following observation appearing in Derezinski et al. (2020) connects the CSSP and the Nyström approximation with respect to the trace matrix norm.

Suppose we have the decomposition of the kernel matrix $\mathbf{K} = \mathbf{X}^{\top}\mathbf{X}$ where $\mathbf{X} \in \mathbb{R}^{m \times n}$. Then, the Nyström approximation is given by $\widehat{\mathbf{K}}_{\boldsymbol{s}} = (\mathbf{P}_{\boldsymbol{s}}\mathbf{X})^{\top} \mathbf{P}_{\boldsymbol{s}}\mathbf{X}$ and

$$\|\mathbf{K} - \widehat{\mathbf{K}}_{\boldsymbol{s}}\|_* = \|\mathbf{X} - \mathbf{P}_{\boldsymbol{s}}\mathbf{X}\|_F^2.$$

where $\|\mathbf{A}\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(\mathbf{A})$ for $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the trace matrix norm. This connection is used in (Derezinski et al. 2020) to provide shared approximation bounds for both the CSSP and Nyström approximation. Given that the kernel matrix is always positive semi-definite, the decomposition $\mathbf{K} = \mathbf{X}^{\top}\mathbf{X}$ always exists and one can solve the CSSP for $\mathbf{X}$ to obtain the best k-landmark Nyström approximation with respect to the trace norm. We note that such a decomposition is not unique, e.g., it can be the Cholesky decomposition or the symmetric square-root decomposition.

The matrix $\mathbf{X}$ does not need explicit evaluation in order to perform CSSP as one can attain $\nabla f_{\lambda}(\boldsymbol{t})$ with the matrix $\mathbf{K}$ instead (see, Lemma 3). Therefore, finding the decomposition $\mathbf{K} = \mathbf{X}^{\top}\mathbf{X}$ is not required, and one can approximately solve the CSSP by minimizing $\nabla f_{\lambda}(\boldsymbol{t})$ with the kernel matrix $\mathbf{K}$.

Suppose instead we want to use the Frobenius matrix norm to find the best choice of columns of the matrix $\mathbf{K}$ to construct the Nyström approximation. This problem is formulated as

$$\operatorname*{argmin}_{\boldsymbol{s} \in \{0,1\}^n} \|\mathbf{K} - \widehat{\mathbf{K}}_{\boldsymbol{s}}\|_F^2, \quad \text{subject to } \|\boldsymbol{s}\|_0 \leq k. \tag{2}$$

Similar to $\widetilde{\mathbf{P}}(\boldsymbol{t})$ we can weight each column j of $\mathbf{K}$ by $t_j \in [0,1]$ instead of sampling the columns $\mathbf{K}_{[\boldsymbol{s}]}$ for the Nyström approximation. We define continuous generalization for the Nyström approximation,

Definition 2 For $\boldsymbol{t} = (t_1, \ldots, t_n)^{\top} \in [0,1]^n$ let $\mathbf{T} := \mathrm{Diag}(\boldsymbol{t})$ and

$$\widetilde{\mathbf{K}}(\boldsymbol{t}) := \mathbf{K}\mathbf{T}\left[\mathbf{T}\mathbf{K}\mathbf{T} + \delta(\mathbf{I} - \mathbf{T}^2)\right]^{\dagger}\mathbf{T}\mathbf{K},$$

where $\delta > 0$ is a fixed constant.

Similar to $\widetilde{\mathbf{P}}(\boldsymbol{t})$, for any $\boldsymbol{t} \in [0,1)^n$ the matrix $\mathbf{T}\mathbf{K}\mathbf{T} + \delta(\mathbf{I} - \mathbf{T}^2)$ is invertible. In the following two results, we state that $\widetilde{\mathbf{K}}(\boldsymbol{t})$ is a continuous function on $[0,1]^n$ and agrees with the exact Nyström approximation at every corner point.

Lemma 4 For any corner point $\boldsymbol{s} \in \{0,1\}^n$, $\widetilde{\mathbf{K}}(\boldsymbol{s})$ exists and

$$\widetilde{\mathbf{K}}(\boldsymbol{s}) = \widehat{\mathbf{K}}_s = \mathbf{K}_{[\boldsymbol{s}]} \mathbf{K}^{\dagger}_{[\boldsymbol{s},\boldsymbol{s}]} \mathbf{K}^{\top}_{[\boldsymbol{s}]}.$$

Lemma 5 $\widetilde{\mathbf{K}}(\boldsymbol{t})$ is continuous element-wise over $[0,1]^n$. Moreover, for any sequence $\boldsymbol{t}^{(1)}, \boldsymbol{t}^{(2)} \cdots \in [0,1)^n$ converging to $\boldsymbol{t} \in [0,1]^n$, the limit $\lim_{l \to \infty} \widetilde{\mathbf{K}}(\boldsymbol{t}^{(l)})$ exists and

$$\lim_{l \to \infty} \widetilde{\mathbf{K}}(\boldsymbol{t}^{(l)}) = \widetilde{\mathbf{K}}(\boldsymbol{t}).$$

We therefore have the continuous generalization of the exact problem (2),

$$\operatorname*{argmin}_{\boldsymbol{t} \in [0,1]^n} \|\mathbf{K} - \widetilde{\mathbf{K}}(\boldsymbol{t})\|_F^2, \quad \text{subject to } \sum_{j=1}^{n} t_j \leq k.$$

Instead of solving this constrained problem, for a tunable parameter λ, we consider minimizing the Lagrangian function,

$$\operatorname*{argmin}_{\boldsymbol{t}\in[0,1]^n} g_\lambda(\boldsymbol{t}), \quad g_\lambda(\boldsymbol{t}) := \|\mathbf{K}-\widetilde{\mathbf{K}}(\boldsymbol{t})\|_F^2 + \lambda \sum_{j=1}^n t_j.$$

As with the continuous extension for CSSP we use a gradient descent method to solve the above problem. The following result provides an expression for $\nabla g_\lambda(\boldsymbol{t})$ for $\boldsymbol{t} \in (0,1)^n$.

Lemma 6 Let $\mathbf{Z} = \mathbf{K} - \delta\mathbf{I}$, $\mathbf{L}_{\boldsymbol{t}} = \mathbf{TZT} + \delta\mathbf{I}$ and $\mathbf{D} = \widetilde{\mathbf{K}}(\boldsymbol{t}) - \mathbf{K}$. Then, for $\boldsymbol{t} \in (0,1)^n$,

$$\nabla g_\lambda(\boldsymbol{t}) = 4\,\mathrm{Diag}\left[\mathbf{L}_{\boldsymbol{t}}^{-1}\mathbf{TKDK}\left(\mathbf{I} - \mathbf{TL}_{\boldsymbol{t}}^{-1}\mathbf{TZ}\right)\right] + \lambda\mathbf{1}.$$

Evaluating $\nabla g_\lambda(\boldsymbol{t})$ has a computational complexity of $O(n^3)$ due to the required inversion of $\mathbf{L}$ and evaluation of $\mathbf{K}(\boldsymbol{t})$. As with $\nabla f_\lambda(\boldsymbol{t})$ we detail an unbiased estimate for $\nabla g_\lambda(\boldsymbol{t})$ in Section 3 which utilizes matrix-vector multiplications with $\mathbf{K}$ and that helps in reducing the computational cost.

3 IMPLEMENTATION

In this section, we detail how to efficiently solve the continuous problems posed in Section 2. In particular, we detail a non-linear transformation that was also used in Moka et al. (2022) to make both the CSSP and Nyström approximation optimization problems unconstrained. We then show how one can estimate the gradients using MVMs with $\mathbf{X}$ and $\mathbf{K}$.

3.1 Handling Box Constraints

The continuous extension of the CSSP and Nyström approximation requires minimizing the functions $f_\lambda(\boldsymbol{t})$ and $g_\lambda(\boldsymbol{t})$ over $\boldsymbol{t} \in [0,1]^n$. We now consider a non-linear transformation, as proposed in Moka et al. (2022), to make both optimization problems unconstrained. Consider the mapping $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{w})$ given by,

$$t_j(w_j) = 1 - \exp(-w_j^2), \quad j = 1,\ldots,n,$$

then if we consider the optimization of continuous CSSP,

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^p} f_\lambda(\boldsymbol{t}(\boldsymbol{w})),$$

we attain the solution to (2.1) by evaluating $\boldsymbol{t}(\boldsymbol{w}^*)$. This is true because for any $a, b \in \mathbb{R}$,

$$1 - \exp(-a^2) < 1 - \exp(-b^2) \quad \text{if and only if } a^2 < b^2.$$

In vector form the transformation is $\boldsymbol{t}(\boldsymbol{w}) = \mathbf{1} - \exp(-\boldsymbol{w}\odot\boldsymbol{w})$ (here $\odot$ denotes element-wise multiplication) and using the chain rule we obtain for $\boldsymbol{w} \in \mathbb{R}^p$,

$$\frac{\partial f_\lambda(\boldsymbol{t}(\boldsymbol{w}))}{\partial \boldsymbol{w}} = \frac{\partial f_\lambda(\boldsymbol{t}(\boldsymbol{w}))}{\partial \boldsymbol{t}} \odot (2\boldsymbol{w}\odot\exp(-\boldsymbol{w}\odot\boldsymbol{w})).$$

We can now implement a gradient descent algorithm to approximately obtain $\boldsymbol{t}(\boldsymbol{w}^*)$. Using this approximation we can select an appropriate binary vector as a solution to the exact problem (1). The same transformation can be applied to solve $g_\lambda(\boldsymbol{t}(\boldsymbol{w}))$ over $\boldsymbol{w} \in \mathbb{R}^n$.

3.2 Stochastic Estimate for the Gradient

As discussed in Section 2, $\nabla f_\lambda(\boldsymbol{t})$ and $\nabla g_\lambda(\boldsymbol{t})$ are problematic to compute for large n due to the $O(n^3)$ complexity of inverting a matrix. Here we show that we can implement a stochastic gradient descent (SGD)

which has strong theoretical guarantees (Robbins and Monro 1951) by using an unbiased estimate for $\nabla f_\lambda(\boldsymbol{t})$ and $\nabla g_\lambda(\boldsymbol{t})$.

The method we employ is a factorized estimator $\hat{\ell}$ for the diagonal of a square matrix. Suppose we wish to estimate the diagonal of the matrix $\mathbf{A} = \mathbf{B}\mathbf{C}^\top$ where $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{n\times n}$. Let $\boldsymbol{z} \in \mathbb{R}^n$ be a random vector sampled from the Rademacher distribution, whose entries are either -1 or 1, each with probability $1/2$. Then an unbiased estimate for $\mathrm{Diag}(\mathbf{A})$ is $\hat{\ell} = \mathbf{B}\boldsymbol{z} \odot \mathbf{C}\boldsymbol{z}$, see Martens et al. (2012). Further analysis of its properties including its variance can be found in Mathur et al. (2021). We note that when $\mathbf{B} = \mathbf{A}$ and $\mathbf{C} = \mathbf{I}$, this estimator reduces to the well-known Bekas et al. (2007) estimator for the diagonal.

The two following results provide an unbiased estimate for $\nabla f_\lambda(\boldsymbol{t})$ and $\nabla g_\lambda(\boldsymbol{t})$ using the factorized estimator for the diagonal of a matrix. As mentioned earlier, proofs are provided in Mathur et al. (2023).

Lemma 7 Recall that in the continuous CSSP optimization for $\mathbf{X}$, we have the definitions $\mathbf{T} = \mathrm{Diag}(\boldsymbol{t})$ for $t \in [0,1]^n$, $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$, $\mathbf{Z} = \mathbf{K} - \delta\mathbf{I}_n$ and $\mathbf{L}_{\boldsymbol{t}} = \mathbf{T}\mathbf{Z}\mathbf{T} + \delta\mathbf{I}_n$. Suppose $\boldsymbol{z} \in \mathbb{R}^n$ follows a Rademacher distribution and let: $(1)\,\boldsymbol{a} = \mathbf{K}\boldsymbol{z}$, $(2)\,\boldsymbol{b} = \mathbf{L}_{\boldsymbol{t}}^{-1}(\boldsymbol{t} \odot \boldsymbol{a})$ and

$$\boldsymbol{\phi} = \boldsymbol{b} \odot \mathbf{Z}(\boldsymbol{t} \odot \boldsymbol{b}) - \boldsymbol{a} \odot \boldsymbol{b}.$$

Then for $\boldsymbol{t} \in (0,1)^n$,

$$\nabla f_\lambda(\boldsymbol{t}) = 2\mathbb{E}[\boldsymbol{\phi}] + \lambda\mathbf{1}.$$

Lemma 8 Recall that in the continuous Nyström optimization for a kernel matrix $\mathbf{K}$, we have the definitions $\mathbf{T} = \mathrm{Diag}(\boldsymbol{t})$ for $t \in [0,1]^n$, $\mathbf{Z} = \mathbf{K} - \delta\mathbf{I}$ and $\mathbf{L}_{\boldsymbol{t}} = \mathbf{T}\mathbf{Z}\mathbf{T} + \delta\mathbf{I}$. Suppose $\boldsymbol{z} \in \mathbb{R}^n$ follows a Rademacher distribution and let: $(1)\,\boldsymbol{a} = \mathbf{K}\boldsymbol{z}$, $(2)\,\boldsymbol{b} = \mathbf{L}_{\boldsymbol{t}}^{-1}(\boldsymbol{t} \odot \boldsymbol{a})$, $(3)\,\boldsymbol{c} = \mathbf{K}(\boldsymbol{t} \odot \boldsymbol{b}) - \boldsymbol{a}$, $(4)\,\boldsymbol{d} = \mathbf{K}\boldsymbol{c}$, $(5)\,\boldsymbol{e} = \mathbf{L}_{\boldsymbol{t}}^{-1}(\boldsymbol{t} \odot \boldsymbol{d})$ and

$$\boldsymbol{\psi} = \boldsymbol{b} \odot \boldsymbol{d} + \boldsymbol{a} \odot \boldsymbol{e} - \boldsymbol{e} \odot \mathbf{Z}(\boldsymbol{t} \odot \boldsymbol{b}) - \boldsymbol{b} \odot \mathbf{Z}(\boldsymbol{t} \odot \boldsymbol{e}).$$

Then for $\boldsymbol{t} \in (0,1)^n$,

$$\nabla g_\lambda(\boldsymbol{t}) = 2\mathbb{E}[\boldsymbol{\psi}] + \lambda\mathbf{1}.$$

Using these results, we can obtain for a Monte-Carlo size M, the approximations $\nabla f_\lambda(\boldsymbol{t}) \approx 2\left(\frac{1}{M}\sum_{i=1}^M \boldsymbol{\phi}^{(i)}\right) + \lambda\mathbf{1}$ and $\nabla g_\lambda(\boldsymbol{t}) \approx 2\left(\frac{1}{M}\sum_{i=1}^M \boldsymbol{\psi}^{(i)}\right) + \lambda\mathbf{1}$, where $\boldsymbol{\phi}^{(i)}$ and $\boldsymbol{\psi}^{(i)}$ are evaluated using a sample $\boldsymbol{z}^{(i)}$ drawn from the Rademacher distribution.

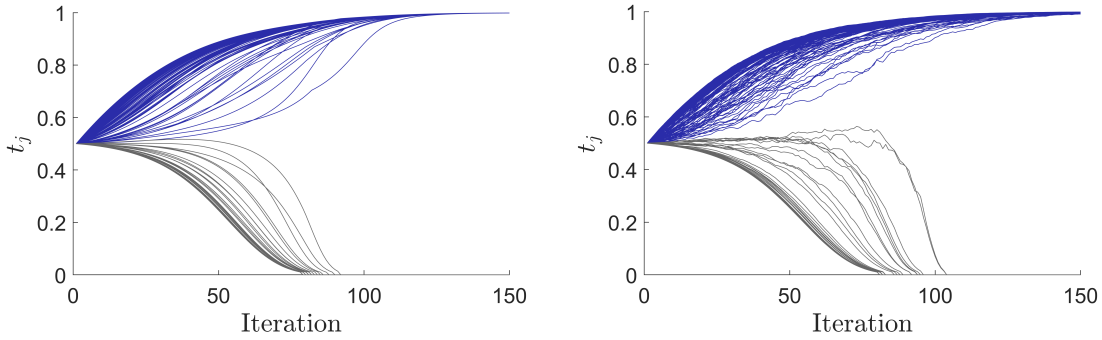
These results show that to evaluate stochastic gradients one needs to solve linear systems efficiently with the matrix $\mathbf{L}_{\boldsymbol{t}}$. These systems can be iteratively solved using the conjugate gradient (CG) algorithm (Golub and Van Loan 1996) which uses a sequence of MVMs with $\mathbf{L}_{\boldsymbol{t}}$. Multiplying a vector with $\mathbf{L}_{\boldsymbol{t}}$ can be reduced to a single MVM with the matrix $\mathbf{K}$ and a sequence of element-wise vector multiplications and additions.

3.3 Obtaining a Solution

While we have re-framed both the CSSP and the Nyström problem as an optimization over $\boldsymbol{t} \in [0,1]^n$, the priority remains to obtain an approximate solution $\boldsymbol{s} \in \{0,1\}^n$ to (1) and (2). To obtain such a binary vector, we first initialize SGD from a starting point $\boldsymbol{t}^{(0)}$ and return the final value $\boldsymbol{t}^*$ after a termination condition for SGD has been satisfied. Under SGD the iterative sequence $\{\boldsymbol{t}^{(i)}\}_{i\geq 0}$ moves towards a corner point of the hypercube. To obtain the closest corner point $\boldsymbol{s} \in \{0,1\}^n$, we map the insignificant t_j^*'s to 0 and all the other t_j^*'s to 1 for some tolerance parameter $\tau \in (0,1)$. This implementation is shown Algorithm 1. In Figure 1 we provide example solution paths $\{\boldsymbol{t}^{(i)}\}_{i\geq 0}$ under both batch gradient descent and SGD.

When choosing the value for $\boldsymbol{t}^{(0)}$ it is important to consider the following true statements: $t_j = 0$ if and only if $w_j = 0$ and

$$\lim_{w_j\to 0} \frac{\partial f_\lambda(\boldsymbol{t}(\boldsymbol{w}))}{\partial w_j} = \lim_{w_j\to 0} \frac{\partial g_\lambda(\boldsymbol{t}(\boldsymbol{w}))}{\partial w_j} = 0.$$



(a) Gradient Descent

(b) Stochastic Gradient Descent

Figure 1: Convergence of $\boldsymbol{t}$ for continuous Column Subset Selection using the MNIST dataset. Blue trajectories correspond to selected columns. Only a subset of 300 randomly chosen column trajectories (out of 784) are displayed. For both (a) and (b), $\lambda = 10$ and $\delta = 10$. In (b) the Monte-Carlo size is $M = 5$.

Algorithm 1 Continous Landmark Selection

1: **input:** Data matrix: $\mathbf{X} \in \mathbb{R}^{m\times n}$ (CSSP) or Kernel matrix: $\mathbf{K} \in \mathbb{R}^{n\times n}$ (Nyström method), Tuning parameters: δ and λ, Monte Carlo size: M, Termination Condition: TermCond, Threshold value: $\tau \in [0,1]$.
2: Set $\boldsymbol{t}^{(0)} = (1/2, \dots 1/2)^\top$
3: $\boldsymbol{w}^{(0)} \leftarrow \sqrt{-\ln(1-\boldsymbol{t}^{(0)})}$
4: $\boldsymbol{w}^* \leftarrow$ SGD ($\boldsymbol{w}^{(0)}$, M, $\mathbf{X}$ or $\mathbf{K}$, TermCond)
5: $\boldsymbol{t}^* \leftarrow 1 - \exp(-\boldsymbol{w}^* \odot \boldsymbol{w}^*)$
6: **for** $i = 1$ **to** n **do**
7: $\quad s_j \leftarrow I(t_j^* > \tau)$
8: **end for**
9: **return:** $\boldsymbol{s}^* = (s_1, \dots, s_n)^\top$

These facts imply that if t_j is set to zero during the course of the optimization it will remain unchanged thereafter. Therefore, it is important to choose $\boldsymbol{t}^{(0)}$ that is away from any corner point. It is for this reason, we set $\boldsymbol{t}^{(0)} = (1/2, \dots, 1/2)^\top$ in all our experiments.

3.4 Complexity Analysis

The main computational cost of our algorithm is the complexity attributed to estimating the gradients at each iteration of SGD. The cost to solve (2) and (5) in either Lemma 7 or 8 via CG is $O(T_{mult} M \ell)$ flops where ℓ is the number of CG iterations and T_{mult} is the cost of computing a matrix-vector product with either $\mathbf{X}^\top\mathbf{X}$ (CSSP) or kernel matrix $\mathbf{K}$ (Nyström). Generally, only $\ell \ll n$ iterations of CG are required to obtain an accurate solution to the linear system.

The cost T_{mult} is $O(mn)$ and $O(n^2)$ via direct computation for $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{K}$ respectively. For kernel matrices with specific structure, this cost can be reduced. For example, for Toeplitz matrices or for matrices constructed from a kernel function that is analytic and isotropic, the cost can be reduced to quasi-linear complexity (Dietrich and Newsam 1997). Utilizing GPU hardware for accelerating matrix computations has gained significant recent attention and numerous software regimes (Charlier et al. 2021) have been proposed to accelerate kernel MVMs. These methods can be implemented out-of-the-box and allow MVMs to be feasible on very large datasets ($n \sim 10^8$). Another advantage of these algorithms is that, as long as the

kernel function $h(\boldsymbol{x}_i', \boldsymbol{x}_j')$ is given, MVMs can be computed directly without ever storing the kernel matrix $\mathbf{K}$. This is an advantage of our method when compared to other methods such as the greedy selection method for the Nyström approximation in (Farahat et al. 2011), which has a cost of $O(n^2k)$ and requires the full explicit matrix to be stored in memory.

3.5 Role of Parameters δ and λ

The tuning parameter λ controls the size of the penalty $\|\boldsymbol{t}\|_1$ which is added to the Frobenius matrix loss. It is intuitive then that for a larger value of λ a stronger shrinkage is applied to $\boldsymbol{t}$ during the course of the continuous optimization. In terms of curvature, as λ increases so does the directional slope of $f_\lambda(\boldsymbol{t}(\boldsymbol{w}))$ and $g_\lambda(\boldsymbol{t}(\boldsymbol{w}))$ in the region around $w_j = 0$. For this reason, it is likelier that more w_j's will be pushed towards zero when the value for λ is large. This behavior is similar to that of the parameter λ in the COMBSS method (Moka et al. 2022) where a more formal analysis can be found. We note that the relationship between λ and k is data dependent and it is suggested that the user apply an efficient grid search regime to obtain an appropriate λ for their use.

With respect to the parameter δ we first note that Lemma 1 and Lemma 4 remain true regardless of the choice of δ. Therefore, the value of δ affects the behavior of the penalized loss only at the interior points $\boldsymbol{t} \in (0,1)^n$. We would like a choice of δ such that for all the interior $\boldsymbol{t}$ the functions $f_\lambda(\boldsymbol{t})$ and $g_\lambda(\boldsymbol{t})$ are well-behaved. When δ is very small the linear systems that require solving at $\boldsymbol{t} \in (0,1)^n$ may be close to singular and numerical issues can arise more frequently. Moreover, when δ is large we observe large shifts in the value of the objective approaching a corner point. Our simulations indicate that $\delta = 1$ produces a well-behaved function.

4 NUMERICAL EXPERIMENTS AND RESULTS

In this section, we provide numerical examples with real data designed to demonstrate that our proposed continuous optimization method outperforms well-known sampling-based methods for small and large datasets. Moreover, we demonstrate that when it is feasible to run greedy selection, our continuous method exhibits very similar performance.

Numerical experiments were conducted on the small to medium-sized datasets: Residential and Building dataset ($m = 372$, $n = 109$), MNIST1K ($m = 1000$, $n = 784$), Arrhythmia ($m = 452$, $n = 279$) and SECOM ($m = 1567$, $n = 591$). Numerical experiments for Nyström landmark selection were also conducted on the larger datasets: Power Plant dataset ($m = 4$, $n = 9568$), HTRU2 dataset ($m = 8$, $n = 17898$) and Protein dataset ($m = 9$, $n = 45730$). All datasets except MNIST (LeCun et al. 1994) are downloaded from UCI ML Repository (Asuncion and Newman 2007). All datasets were standardized such that all columns had mean zero and variance equal to one.

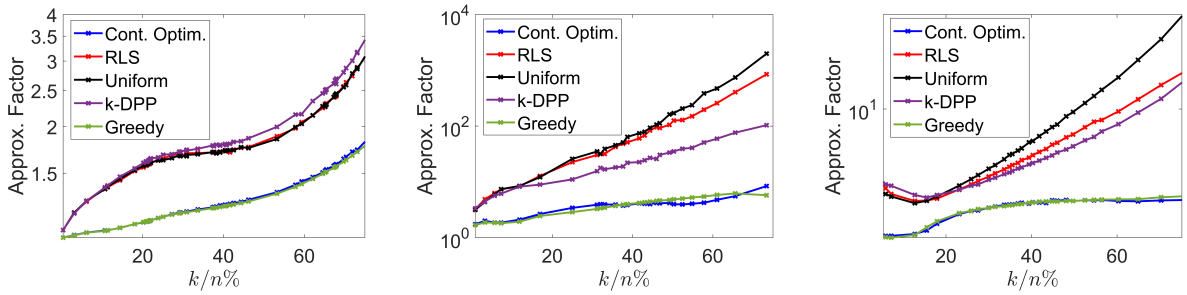
For the small to medium-sized datasets, we use the best rank-k approximation factor to compare our method to existing methods (see Figure 2 and Figure 3). The best rank-k approximation factor is given by

$$\text{Approximation Factor} := \frac{\|\mathbf{A} - \widehat{\mathbf{A}}_s\|_F^2}{\|\mathbf{A} - \widehat{\mathbf{G}}\|_F^2}.$$

where $\widehat{\mathbf{A}}_s$ is either the Nyström or CSSP low-rank matrix and $\widehat{\mathbf{G}}$ is the best rank-k approximation computed using the Singular Value Decomposition (SVD) of $\mathbf{A}$.

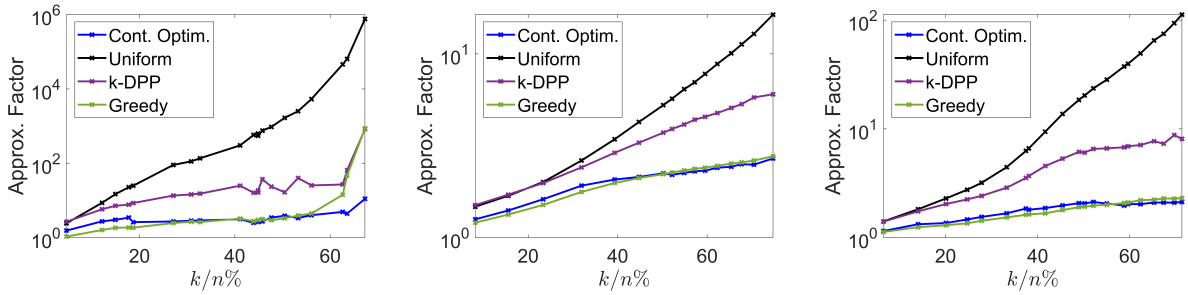
In these experiments, we compare the proposed continuous landmark selection method executed with SGD ($M = 10$) with the following four well-known methods: Uniform Sampling (Williams and Seeger 2000), Recursive RLS (Ridge Leverage Scores) - Nyström sampling (Musco and Musco 2017), k-DPP sampling (Derezinski and Mahoney 2021) and Greedy selection (Farahat et al. 2011; Farahat et al. 2013).

For the experiments conducted on the larger datasets (see Figure 4) we exclude the k-DPP sampling and greedy methods as it is either too costly to compute the choice of landmark points or too costly to store the full kernel matrix on a GPU. In our implementation of continuous Nyström landmark selection, we use



(a) Residential ($\sigma = 1$) (b) Residential ($\sigma = 5$) (c) MNIST1K ($\sigma = 20$)

Figure 2: The mean Nyström empirical approximation factor over 50 trials for the UCI Residential Building and MNIST dataset where $\mathbf{K}$ is constructed using the Gaussian Radial Basis Function (RBF) kernel: $\mathbf{K}_{i,j} = h(\boldsymbol{x}'_i, \boldsymbol{x}'_j) = \exp\left(-\|\boldsymbol{x}'_i - \boldsymbol{x}'_j\|^2\right)/\sigma^2$. Approximation factor is plotted on a logarithmic scale.



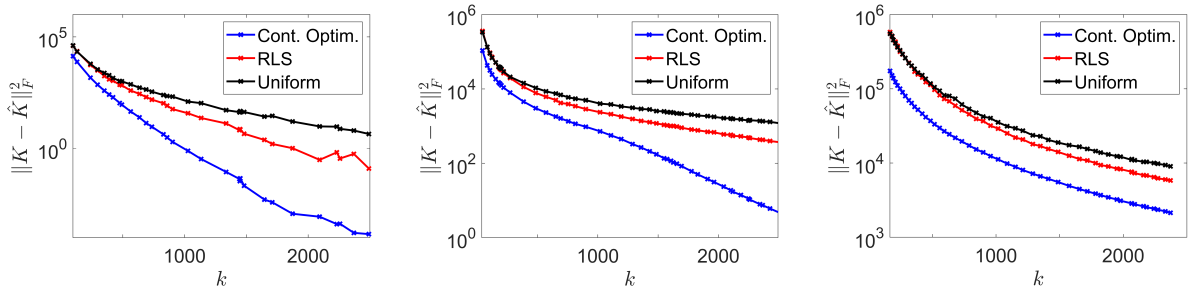
(a) Residential (b) Arrhythmia (c) SECOM

Figure 3: The mean CSSP empirical approximation factor over 50 trials for the MNIST dataset and three UCI datasets for different methods. Approximation factor is plotted on a logarithmic scale.

the KeOps library (Charlier et al. 2021) to efficiently compute MVMs and linear solves on a GPU without ever storing the matrix $\mathbf{K}$, thus negating the need to store any $O(n^2)$ objects. These experiments were run using an NVIDIA Tesla T4 GPU with 16GB memory.

In Figure 2 and Figure 3 we observe the approximation factor for Nyström and CSSP landmark selection with different subset sizes k. A lower approximation factor indicates a better approximation and an approximation factor close to one implies near-best-case performance for the given subset size k. The results indicate that the continuous optimization method is better than every tested sampling method and is very similar to greedy selection in performance (whenever the greedy selection is feasible). In most cases, for the CSSP, as the proportion of selected columns increases the continuous method starts to marginally outperform the greedy method.

In Figure 4, we observe for all three datasets (Power Plant, HTRU2 and Protein) that the continuous landmark selection achieves better accuracy than the Recursive RLS (Ridge Leverage Scores) - Nyström sampling and Uniform sampling methods. While Recursive RLS sampling (complexity: $O(nk^2)$) and uniform sampling are faster at selecting landmark points, for a fixed k the continuous method obtains a more accurate Nyström approximation. For instance, in the Power Plant dataset experiment, when 50% of the columns are selected as landmark points, the continuous method obtains a squared error that is six orders of magnitude smaller than the uniform sampling method and a squared error that is four orders of magnitude smaller than the Recursive RLS sampling method. Thus, if a memory budget for the size of the Nyström approximation is given, as is often the case, the continuous method will compute a superior



(a) Power Plant (b) HTRU2 (c) Protein

Figure 4: The mean empirical squared Frobenius error $\|\mathbf{K} - \widehat{\mathbf{K}}_{\boldsymbol{s}^*}\|_F^2$ over 10 trials for the UCI datasets Power Plant, HTRU2 and Protein for different methods. The kernel matrix $\mathbf{K}$ for all datasets is constructed using the RBF kernel function with $\sigma = 0.5$. Error is plotted on a logarithmic scale.

approximation. We note that in the most demanding experiment (Protein dataset, $n = 45730$), a single gradient update was computed in 1 to 5 seconds.

5 CONCLUSION

In this paper, we have introduced a novel algorithm that exploits unconstrained continuous optimization to select columns for both the CSSP and Nyström approximation. The algorithm selects columns by minimizing an extended objective which is defined over the hypercube $[0,1]^n$ rather than iterating over the corner points of the hypercube which correspond to all of the $\binom{n}{k}$ subsets. The extended objective for both the CSSP and Nyström approximation can be minimized via SGD where the gradients are estimated with an unbiased estimator which requires only MVMs with either $\mathbf{X}$ (CSSP) or $\mathbf{K}$ (Nyström). On the real-world examples that we considered in this article, the proposed method has proven to be more accurate without incurring higher computational cost.

REFERENCES

Alaoui, A., and M. W. Mahoney. 2015. "Fast Randomized Kernel Ridge Regression with Statistical Guarantees". In *Advances in Neural Information Processing Systems*, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Volume 28: Curran Associates, Inc.

Altschuler, J., A. Bhaskara, G. Fu, V. Mirrokni, A. Rostamizadeh, and M. Zadimoghaddam. 2016. "Greedy Column Subset Selection: New Bounds and Distributed Algorithms". In *Proceedings of The 33rd International Conference on Machine Learning*, edited by M. F. Balcan and K. Q. Weinberger, 2539–2548. New York: PMLR.

Asuncion, A., and D. J. Newman. 2007. "UCI Machine Learning Repository". https://archive.ics.uci.edu/ml/, accessed 1st April 2023.

Beale, E., M. Kendall, and D. Mann. 1967. "The Discarding of Variables in Multivariate Analysis". *Biometrika* 54(3-4):357–366.

Bekas, C., E. Kokiopoulou, and Y. Saad. 2007. "An Estimator for the Diagonal of a Matrix". *Applied numerical mathematics* 57(11-12):1214–1229.

Bien, J., Y. Xu, and M. W. Mahoney. 2010. "CUR from a Sparse Optimization Viewpoint". In *Advances in Neural Information Processing Systems*, edited by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Volume 23: Curran Associates, Inc.

Calandriello, D., M. Derezinski, and M. Valko. 2020. "Sampling from a k-DPP Without Looking at All Items". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Volume 33, 6889–6899: Curran Associates, Inc.

Charlier, B., J. Feydy, J. A. Glaunes, F.-D. Collin, and G. Durif. 2021. "Kernel Operations on the GPU, with Autodiff, without Memory Overflows.". *J. Mach. Learn. Res.* 22(74):1–6.

Derezinski, M., R. Khanna, and M. W. Mahoney. 2020. "Improved Guarantees and a Multiple-Descent Curve for Column Subset Selection and the Nystrom method". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Volume 33, 4953–4964: Curran Associates, Inc.

Derezinski, M., and M. W. Mahoney. 2021. "Determinantal Point Processes in Randomized Numerical Linear Algebra". *Notices of the American Mathematical Society* 68(1):34–45.

Deshpande, A., and S. Vempala. 2006. "Adaptive Sampling and Fast Low-Rank Matrix Approximation". In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 292–303. Springer.

Dietrich, C. R., and G. N. Newsam. 1997. "Fast and Exact Simulation of Stationary Gaussian Processes Through Circulant Embedding of the Covariance Matrix". *SIAM Journal on Scientific Computing* 18(4):1088–1107.

Drineas, P., M. W. Mahoney, and N. Cristianini. 2005. "On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning.". *Journal of Machine Learning Research* 6(12).

Farahat, A., A. Ghodsi, and M. Kamel. 2011. "A Novel Greedy Algorithm for Nyström Approximation". In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by G. Gordon, D. Dunson, and M. Dudík, 269–277. Fort Lauderdale, USA: PMLR.

Farahat, A. K., A. Ghodsi, and M. S. Kamel. 2013. "Efficient Greedy Feature Selection for Unsupervised Learning". *Knowledge and information systems* 35(2):285–310.

Gittens, A., and M. Mahoney. 2013, 17–19 Jun. "Revisiting the Nystrom Method for Improved Large-Scale Machine Learning". In *Proceedings of the 30th International Conference on Machine Learning*, edited by S. Dasgupta and D. McAllester, 567–575. Atlanta, Georgia, USA: PMLR.

Golub, G. H., and C. F. Van Loan. 1996. *Matrix Computations*. 3rd ed. The Johns Hopkins University Press.

Hocking, R. R., and R. Leslie. 1967. "Selection of the Best Subset in Regression Analysis". *Technometrics* 9(4):531–540.

Hough, J. B., M. Krishnapur, Y. Peres, and B. Virág. 2006. "Determinantal Processes and Independence". *Probability surveys* 3:206–229.

LeCun, Y., C. Cortez, and C. C. Burges. 1994. "The MNIST Handwritten Digit Database". Yann LeCun's Website yann.lecun.com.

Martens, J., I. Sutskever, and K. Swersky. 2012. "Estimating the Hessian by Back-Propagating Curvature". In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, 963–970. Madison, WI, USA: Omnipress.

Mathur, A., S. Moka, and Z. Botev. 2021. "Variance Reduction for Matrix Computations with Applications to Gaussian Processes". In *EAI International Conference on Performance Evaluation Methodologies and Tools*, 243–261. Springer.

Mathur, A., S. Moka, and Z. Botev. 2023. "Column Subset Selection and Nyström Approximation via Continuous Optimization". *arXiv preprint arXiv:2304.09678*.

Moka, S., B. Liquet, H. Zhu, and S. Muller. 2022. "COMBSS: Best Subset Selection via Continuous Optimization". *arXiv preprint arXiv:2205.02617*.

Musco, C., and C. Musco. 2017. "Recursive Sampling for the Nystrom Method". In *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Volume 30: Curran Associates, Inc.

Robbins, H., and S. Monro. 1951. "A Stochastic Approximation Method". *The Annals of Mathematical Statistics* 22(3):400 – 407.

Shitov, Y. 2021. "Column Subset Selection is NP-Complete". *Linear Algebra and its Applications* 610:52–58.

Tibshirani, R. 1996. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288.

Williams, C., and M. Seeger. 2000. "Using the Nyström Method to Speed Up Kernel Machines". In *Advances in Neural Information Processing Systems*, edited by T. Leen, T. Dietterich, and V. Tresp, Volume 13: MIT Press.

AUTHOR BIOGRAPHIES

ANANT MATHUR is a PhD student in the School of Mathematics and Statistics at UNSW Sydney. He received a Bachelor of Science (Hons) from UNSW Sydney. His current work focuses on feature selection algorithms and their efficient implementation. More broadly he is interested in problems in Data Science and Statistics. His email address is anant.mathur@unsw.edu.au.

SARAT MOKA is a Lecturer in the School of Mathematics at UNSW Sydney. Prior to this, he was a Research Fellow at Macquarie University (2021–2023). His research broadly focuses on problems in Probability Theory, Data Science, Statistics, Deep Learning, and Monte Carlo Simulation. His website is https://saratmoka.com.

ZDRAVKO BOTEV is a Lecturer of Statistics at UNSW Sydney. His research interest include: 1) Monte Carlo variance reduction methods, especially for rare-event probability estimation; 2) nonparametric kernel density estimation, and more recently 3) fast model-selection algorithms for large-scale statistical learning. His email address is botev@unsw.edu.au.