Tight Kernel Query Complexity of Kernel Ridge Regression and Kernel k-means Clustering

Manuel Fernández [*1] **David P. Woodruff** [*1] **Taisuke Yasuda** [*2]

Abstract

Kernel methods generalize machine learning algorithms that only depend on the pairwise inner products of the data set by replacing inner products with kernel evaluations, a function that passes input points through a nonlinear feature map before taking the inner product in a higher dimensional space. In this work, we present tight lower bounds on the number of kernel evaluations required to approximately solve kernel ridge regression (KRR) and kernel k-means clustering (KKMC) on n input points. For KRR, our bound for relative error approximation to the minimizer of the objective function is $\Omega(nd_{\text{eff}}^\lambda/\varepsilon)$ where d_{eff}^λ is the effective statistical dimension, which is tight up to a $\log(d_{\text{eff}}^\lambda/\varepsilon)$ factor. For KKMC, our bound for finding a k-clustering achieving a relative error approximation of the objective function is $\Omega(nk/\varepsilon)$, which is tight up to a $\log(k/\varepsilon)$ factor. Our KRR result resolves a variant of an open question of El Alaoui and Mahoney, asking whether the effective statistical dimension is a lower bound on the sampling complexity or not. Furthermore, for the important practical case when the input is a mixture of Gaussians, we provide algorithms which bypass the above lower bounds.

1. Introduction

The *kernel trick* in machine learning is a general technique that takes linear learning algorithms that only depend on the dot products of the data, including linear regression, support vector machines, principal component analysis, and k-means clustering, and boosts them to powerful nonlinear algorithms. This is done by replacing the inner product between two data points with their inner product after applying a kernel map, which implicitly maps the points to a higher dimensional space via a non-linear feature map. The simplicity and power of kernel methods has led to wide adoption across the machine learning community: nowadays, kernel methods are a staple both in theory (Friedman et al., 2001) and in practice (Schölkopf et al., 2004; Zhang et al., 2007). We refer the reader to (Schölkopf & Smola, 2001) for more background on kernel methods.

However, one problem with kernel methods is that the computation of the kernel matrix $\mathbf{K}$, the matrix containing all pairs of kernel evaluations between n data points, requires $\Omega(n^2)$ time, which is prohibitively expensive for the large-scale data sets encountered in modern data science. To combat this, a large body of literature in the last decade has been devoted to designing faster algorithms that attempt to trade a small amount of accuracy in exchange for speed and memory, based on techniques such as random Fourier features (Rahimi & Recht, 2008), sampling (Bach, 2013; El Alaoui & Mahoney, 2015; Musco & Musco, 2017; Musco & Woodruff, 2017), sketching (Yang et al., 2017), and incomplete Cholesky factorization (Bach & Jordan, 2002; Fine & Scheinberg, 2001). We refer the reader to the exposition of (Musco & Musco, 2017) for a more extensive overview of recent literature on the approximation of kernel methods.

1.1. Previous work on kernel query complexity

In this work, we consider lower bounds on the query complexity of the kernel matrix. The kernel query complexity is a fundamental information-theoretic parameter of kernel problems and both upper and lower bounds have been studied by a number of works (Lin et al., 2014; Cesa-Bianchi et al., 2015; Musco & Musco, 2017; Musco & Woodruff, 2017).

For kernel ridge regression, a lower bound has been shown for additive error approximation of the objective function value in Corollary 8 of (Cesa-Bianchi et al., 2015), which is a weaker approximation guarantee than what we study in this work. However, their bound is not known to be tight. Furthermore, the best known upper bounds for kernel ridge regression are in terms of a data-dependent quantity

[*] Equal contribution [1] Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [2] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Correspondence to: Taisuke Yasuda <taisukey@andrew.cmu.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

known as the *effective statistical dimension* (El Alaoui & Mahoney, 2015; Musco & Musco, 2017), on which the (Cesa-Bianchi et al., 2015) bound does not depend. The question of whether the effective statistical dimension gives a lower bound on the sample complexity has been posed as an open question by El Alaoui and Mahoney (El Alaoui & Mahoney, 2015). We will answer this question affirmatively under a slightly different approximation guarantee than they use, which is nevertheless satisfied by known algorithms nearly tightly, for instance by (Musco & Musco, 2017).

Another kernel problem for which lower bounds have been shown is the problem of giving a $(1+\varepsilon)$ relative Frobenius norm error rank k approximation of the kernel matrix, which has a bound of $\Omega(nk/\varepsilon)$ by Theorem 13 of (Musco & Woodruff, 2017). For kernel k-means clustering, there are no kernel complexity lower bounds to our knowledge.

Similar cost models have also been studied in the context of semisupervised/interactive learning. Intuitively, kernel evaluations are queries that ask for the similarity between two objects, where the notion of similarity in this context is the implicit notion of similarity recognized by humans, i.e. the "crowd kernel". In such situations, the dominant cost is the number of these queries that must be made to users, making kernel query complexity an important computational parameter. Mazumdar and Saha (Mazumdar & Saha, 2017) study the problem of clustering under the setting where the algorithm obtains information by adaptively asking users whether two data points belong to the same cluster or not. In this setting, the dominant cost that is analyzed is the number of same-cluster queries that the algorithm must make, which exactly corresponds to the kernel query complexity of clustering a set of n points drawn from k distinct points with the indicator function kernel and the 0-1 loss (as opposed to k-means clustering, which uses the ℓ_2 loss). In (Tamuz et al., 2011), the authors consider the problem of learning a "crowd kernel", where the implicit kernel function is crowdsourced and the cost is measured as the number of queries of the form "is a more similar to b than c?" rather than queries that directly access the underlying kernel evaluations.

1.2. Our contributions

In this work, we resolve the kernel query complexity of kernel ridge regression and kernel k-means clustering up to $\log(d_{\text{eff}}^{\lambda}/\varepsilon)$ and $\log(k/\varepsilon)$ factors, respectively. Our lower bounds apply even to *adaptive* algorithms, that is, algorithms that are allowed to decide which kernel entries to query based on the results of previous kernel queries. This is a crucial aspect of our contributions, since some of the most efficient algorithms known for kernel ridge regression and kernel k-means clustering make use of adaptive queries, most notably through the use of a data-dependent sampling technique known as *ridge leverage score sampling* (Musco & Musco, 2017).

Kernel problem	KRR	KKMC
Upper bound (Musco & Musco, 2017)	$\tilde{O}\left(\frac{nd_{\text{eff}}^{\lambda}}{\varepsilon}\right)$ Theorem 15	$\tilde{O}\left(\frac{nk}{\varepsilon}\right)$ Theorem 16
Lower bound (This work)	$\Omega\left(\frac{nd_{\text{eff}}^{\lambda}}{\varepsilon}\right)$ Theorem 3.1	$\Omega\left(\frac{nk}{\varepsilon}\right)$ Theorem 4.2

Figure 1. Table of upper bounds and lower bounds on the kernel query complexity, where $\tilde{O}(\cdot)$ hides logarithmic factors in d_{eff}^{λ}, k, and $1/\varepsilon$.

For kernel ridge regression, we present Theorem 3.1, in which we construct a distribution over kernel ridge regression instances such that any randomized algorithm requires $\Omega(nd_{\text{eff}}^{\lambda}/\varepsilon)$ adaptive kernel evaluations. This matches the upper bound given in Theorem 15 of (Musco & Musco, 2017) up to a $\log(d_{\text{eff}}^{\lambda}/\varepsilon)$ factor. Although we present the main ideas of the proof using the kernel as the dot product kernel, our proof in fact applies to a more general class of kernels, including the polynomial kernel and the Gaussian kernel (Theorem 3.7). This result resolves a variant of an open question posed by (El Alaoui & Mahoney, 2015), which asks whether the effective statistical dimension is a lower bound on the sampling complexity or not. In their paper, they consider the approximation guarantee of a $(1+\varepsilon)$ relative error in the statistical risk, while we consider a $(1+\varepsilon)$ relative error approximation of the minimizer of the KRR objective function. By providing tight bounds on the query complexity in terms of the effective statistical dimension d_{eff}^{λ}, we establish the fundamental importance of the quantity as a computational parameter, in addition to its established significance as a statistical parameter in the statistics literature (Friedman et al., 2001).

For kernel k-means clustering, we present Theorem 4.2, which shows a lower bound of $\Omega(nk/\varepsilon)$ for the problem of outputting a clustering which achieves a $(1+\varepsilon)$ relative error value in the objective function. This matches the upper bound given in Theorem 16 of (Musco & Musco, 2017) up to a $\log(k/\varepsilon)$ factor.

Although our lower bounds show that existing upper bounds for kernel ridge regression and kernel k-means clustering are optimal, up to logarithmic factors, in their query complexity, one could hope that for important input distributions that may occur in practice, that better query complexities are possible. We show specifically in the case of kernel k-means that when the n points are drawn from a mixture of k Gaussians with $1/\operatorname{poly}(k/\varepsilon)$ mixing probabilities and a separation between their means that matches the information-theoretically best possible for learning the means given by (Regev & Vijayaraghavan, 2017), one can bypass the $\Omega(nk/\varepsilon)$ lower bound, achieving an $(n/\varepsilon)\operatorname{poly}(\log(k/\varepsilon))$

query upper bound, effectively saving a factor of k from the lower bounds for worst-case input distributions. This is our Theorem 5.1.

2. Preliminaries

2.1. Notation

We denote the set $\{1, 2, \ldots, n\}$ by $[n]$. For $j \in [d]$, we write $\mathbf{e}_j \in \mathbb{R}^d$ for the standard Euclidean basis vectors. We write $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ for the $n \times n$ identity matrix and $\mathbf{1}_n \in \mathbb{R}^n$ for the vector of all ones in n dimensions.

Let $\mathcal{X}$ be the input space of a data set and $\mathcal{F}$ a reproducing kernel Hilbert space with kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We write $\varphi : \mathcal{X} \to \mathcal{F}$ for the feature map, i.e. the φ such that $k(\mathbf{x}, \mathbf{y}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle_{\mathcal{F}}$. For a set of vectors $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ and a kernel map $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we write $\mathbf{K} \in \mathbb{R}^{n \times n}$ for the kernel matrix, i.e. the matrix with $\mathbf{e}_i^\top \mathbf{K} \mathbf{e}_j \coloneqq k(\mathbf{x}_i, \mathbf{x}_j)$. Note that $\mathbf{K}$ is symmetric and positive semidefinite (PSD). We refer the reader to (Schölkopf & Smola, 2001) for more details on the general theory of kernel methods. For all of our lower bound constructions, we will take $\mathcal{X} = \mathbb{R}^d$ and our kernel to be the linear kernel, i.e. the standard dot product on $\mathbb{R}^d$, $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$. Hence, we will frequently refer to kernel queries alternatively as inner product queries.

2.2. Kernel ridge regression

The kernel ridge regression (KRR) task is defined as follows. We parameterize an instance of KRR by a triple $(\mathbf{K}, \mathbf{z}, \lambda)$, where $\mathbf{K} \in \mathbb{R}^n$ is the kernel matrix of a data set $\{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{z} \in \mathbb{R}^n$ is the target vector, and λ is the regularization parameter. The problem is to compute

$$\boldsymbol{\alpha}_{\text{opt}} \coloneqq \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^n} \lVert \mathbf{K}\boldsymbol{\alpha} - \mathbf{z} \rVert_2^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}. \tag{2.1}$$

It is well-known that the solution to the above is given in closed form by

$$\boldsymbol{\alpha}_{\text{opt}} = (\mathbf{K} + \lambda \mathbf{I}_n)^{-1} \mathbf{z} \tag{2.2}$$

which can be shown for example by completing the square.

An important parameter to the KRR instance $(\mathbf{K}, \mathbf{z}, \lambda)$ is the *effective statistical dimension*:

Definition 2.1 (Effective statistical dimension ((Friedman et al., 2001; Zhang, 2005)). *Given a rank r kernel matrix $\mathbf{K}$ with eigenvalues σ_i^2 for $i \in [r]$ and a regularization parameter λ, we define the effective statistical dimension as*

$$d_{\text{eff}}^{\lambda}(\mathbf{K}) \coloneqq \operatorname{tr}\Big(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I}_n)^{-1}\Big) = \sum_{i=1}^{r} \frac{\sigma_i^2}{\sigma_i^2 + \lambda}. \tag{2.3}$$

We simply write d_{eff}^{λ} when the kernel matrix $\mathbf{K}$ is clear from context.

The effective statistical dimension was first introduced to measure the statistical capacity of the KRR instance, but has since been used to parameterize its computational properties as well, in the form of bounds on sketching dimension (Avron et al., 2017) and sampling complexity (El Alaoui & Mahoney, 2015; Musco & Musco, 2017).

2.2.1. Approximate solutions

In the literature, various notions of approximation guarantees for KRR have been studied, including $(1+\varepsilon)$ relative error approximations in the objective function cost (Avron et al., 2017) and $(1+\varepsilon)$ relative error approximations in the statistical risk (Bach, 2013; El Alaoui & Mahoney, 2015; Musco & Musco, 2017). In our paper, we consider a slightly different approximation guarantee, namely a $(1+\varepsilon)$ relative error approximation of the argmin of the KRR objective function.

Definition 2.2 ($(1+\varepsilon)$-approximate solution to kernel ridge regression)**.** *Given a kernel ridge regression instance $(\mathbf{K}, \mathbf{z}, \lambda)$, we say that $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^n$ is a $(1+\varepsilon)$-approximate solution to kernel ridge regression if*

$$\lVert \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}_{\text{opt}} \rVert_2 \leq \varepsilon \lVert \boldsymbol{\alpha}_{\text{opt}} \rVert_2 = \varepsilon \big\lVert (\mathbf{K} + \lambda \mathbf{I}_n)^{-1} \mathbf{z} \big\rVert_2. \tag{2.4}$$

This approximation guarantee is natural, and we note that it is achieved by the estimator of (Musco & Musco, 2017), the proof of which is provided in the supplementary material.

2.3. Kernel k-means clustering

Recall the feature map $\varphi : \mathcal{X} \to \mathcal{F}$ for an input space $\mathcal{X}$ and a reproducing kernel Hilbert space $\mathcal{F}$. The problem of kernel k-means clustering (KKMC) involves forming a partition of the data set $\{\mathbf{x}_i\}_{i=1}^n$ into k clusters $\mathcal{C} \coloneqq \{C_j\}_{j=1}^k$ with centroids $\boldsymbol{\mu}_j \coloneqq \frac{1}{\lvert C_j \rvert} \sum_{\mathbf{x} \in C_j} \varphi(\mathbf{x})$ such that the objective function

$$\operatorname{cost}(\mathcal{C}) \coloneqq \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} \big\lVert \varphi(\mathbf{x}) - \boldsymbol{\mu}_j \big\rVert_{\mathcal{F}}^2 \tag{2.5}$$

is minimized. The problem of finding exact solutions are known to be NP-hard (Aloise et al., 2009), but it has nonetheless proven to be an extremely popular model in practice (Hartigan, 1975).

With an abuse of notation, we will also talk about the cost of a single cluster, which is just the above sum taken only over one cluster:

$$\operatorname{cost}(C_j) \coloneqq \sum_{\mathbf{x} \in C_j} \big\lVert \varphi(\mathbf{x}) - \boldsymbol{\mu}_j \big\rVert_{\mathcal{F}}^2. \tag{2.6}$$

As done in (Boutsidis et al., 2009; Cohen et al., 2015; Musco & Musco, 2017) and many other works, we consider the approximation guarantee of finding a clustering $\{C'_j\}_{j=1}^k$ that achieves a $(1+\varepsilon)$ relative error in the objective function cost, i.e. $\mathrm{cost}(\{C'_j\}_{j=1}^k) \leq (1+\varepsilon)\min_{\mathcal{C}} \mathrm{cost}(\mathcal{C})$.

3. Lower bound for kernel ridge regression

We present our lower bound on the number of kernel entries required in order to compute a $(1+\varepsilon)$-approximate solution to kernel ridge regression (see definition 2.2).

Theorem 3.1 (Query lower bound for kernel ridge regression)**.** *Consider a possibly randomized algorithm $\mathcal{A}$ that correctly outputs a $(1+\varepsilon)$-approximate solution $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^n$ (see definition 2.2) to any kernel ridge regression instance $(\mathbf{K}, \mathbf{z}, \lambda)$ with probability at least $2/3$. Then there exists an input instance $(\mathbf{K}, \mathbf{z}, \lambda)$ on which $\mathcal{A}$ reads at least $\Omega(nd_{\mathrm{eff}}^{\lambda}/\varepsilon)$ entries of $\mathbf{K}$, possibly adaptively, in expectation.*

3.1. Main lower bound

We introduce the following hard distribution.

Definition 3.2 (Hard input distribution – kernel ridge regression)**.** *Let $J, n \in \mathbb{N}$ and assume for simplicity that $4 \mid J$. We define a distribution $\mu_{\mathrm{KRR}}(n, J)$ on binary PSD matrices $\mathbf{K} \in \mathbb{R}^{n\times n}$ defined as follows. We first define a distribution $\nu_{\mathrm{KRR}}(J)$ over standard basis vectors $\{\mathbf{e}_j \in \mathbb{R}^{3J/4} : j \in [3J/4]\}$, where with probability $1/2$ we draw a uniformly random $\mathbf{e}_j$ from $S_1 := \{\mathbf{e}_j : j \in [J/2]\}$ and with probability $1/2$ we draw a uniformly random $\mathbf{e}_j$ from $S_2 := \{\mathbf{e}_{j+J/2} : j \in [J/4]\}$. We then generate $\mathbf{K}$ by drawing n i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n$ from $\nu_{\mathrm{KRR}}(J)$ and letting $\mathbf{K}$ be the inner product matrix of $\{\mathbf{x}_i\}_{i=1}^n$, that is, $\mathbf{e}_i^\top \mathbf{K}\mathbf{e}_j := \mathbf{x}_i \cdot \mathbf{x}_j$.*

Using the above distribution, we prove the following:

Theorem 3.3. *Let $\varepsilon \in (0, 1/2)$ and $J = k/\varepsilon$ with $J^2 = O(n)$ and k a parameter. Suppose that there exists a possibly randomized algorithm $\mathcal{A}$ that, with probability at least $2/3$ over its random coin tosses and random kernel matrix drawn from $\mathbf{K} \sim \mu_{\mathrm{KRR}}(n, J)$, correctly outputs a $(1+\varepsilon/100)$-approximate solution $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^n$ (see definition 2.2) to the kernel ridge regression instance $(\mathbf{K}, \mathbf{z}, \lambda)$ with $\mathbf{z} = \mathbf{1}_n$ and $\lambda = n/k$. Furthermore, suppose that $\mathcal{A}$ reads at most r positions of $\mathbf{K}$ on any input, possibly adaptively. Then, $d_{\mathrm{eff}}^{\lambda}(\mathbf{K}) = \Theta(k)$ and $r = \Omega(nd_{\mathrm{eff}}^{\lambda}/\varepsilon)$.*

We prove Theorem 3.3, via a reduction to a hardness lemma.

Lemma 3.4. *Recall $\mu_{\mathrm{KRR}}(n, J), \nu_{\mathrm{KRR}}(J), S_1, S_2$ from definition 3.2. Suppose that there exists a possibly randomized algorithm $\mathcal{A}$ that, with probability at least $2/3$ over its random coin tosses and random inputs drawn from $\mu_{\mathrm{KRR}}(n, k/\varepsilon)$, correctly outputs whether $\mathbf{x}_i$ corresponds to $\mathbf{e}_j$ with $j \in S_1$ or $j \in S_2$ for at least a $9/10$ fraction of rows $\mathbf{e}_i^\top \mathbf{K}$ for $i \in [n]$. Further, suppose that $\mathcal{A}$ reads at most r positions of $\mathbf{K}$ on any input, possibly adaptively. Then, $r = \Omega(nJ)$.*

This lemma follows from standard techniques, including reductions to hypothesis testing and total variation distance computations, and its proof is deferred to the supplementary material. The lemma is used as follows:

Proof of Theorem 3.3. Assume that $nJ = o(n^2)$, since otherwise the lower bound is $\Omega(n^2)$, which is best possible. Note that for $\mathbf{x} \sim \nu_{\mathrm{KRR}}(J)$, $\mathbf{x} = \mathbf{e}_j$ with probability $\frac{1}{2}\frac{1}{J/2} = \frac{1}{J}$ if $\mathbf{e}_j \in S_1$ and $\frac{1}{2}\frac{1}{J/4} = \frac{2}{J}$ if $\mathbf{e}_j \in S_2$. For a fixed $j \in [3J/4]$, let n_j be the number of $\mathbf{e}_j$ sampled in $\mathbf{K}$ and $\mu_j := \mathbf{E}_{\mathbf{K}\sim\mu_{\mathrm{KRR}}(n,J)}(n_j)$. Note that $\mu_j = n/J$ for $j \in [J/2]$ and $\mu_j = 2n/J$ for $j \in [J/4] + J/2$. Then by Chernoff bounds,

$$\Pr_{\mathbf{K}}\left(\left\{|n_j - \mu_j| \geq \frac{\mu_j}{100}\right\}\right) \leq 2\exp\left(-\frac{1}{100}\frac{n/J}{3}\right) \tag{3.1}$$

so by a union bound over $j \in [3J/4]$, the above holds for all $\mathbf{e}_j$ with probability at most $1/100$ for n/J large enough. Dismiss this event as a failure and assume that $|n_j - \mu_j| \leq \frac{1}{100}\mu_j$ for all $j \in [3J/4]$.

Now let $\mathbf{K} = \mathbf{U\Sigma U}^\top$ be the full SVD of $\mathbf{K}$. Note that the first $3J/4$ singular values are n_j with corresponding singular vectors $\mathbf{U}\mathbf{e}_j = \frac{1}{\sqrt{n_j}}\mathbf{K}\mathbf{e}_j$, and the rest are all 0s. Then, the target vector $\mathbf{z} = \mathbf{1}_n$ can be written as

$$\mathbf{z} = \sum_{j\in[3J/4]} \mathbf{K}\mathbf{e}_j = \sum_{j\in[3J/4]} \sqrt{n_j}\mathbf{U}\mathbf{e}_j, \tag{3.2}$$

since each coordinate $i \in [n]$ belongs to exactly one of the $3J/4$ input points drawn from $\nu_{\mathrm{KRR}}(n, J)$. The optimal solution can then be written as

$$\begin{aligned}\boldsymbol{\alpha}_{\mathrm{opt}} &= (\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{z} = \mathbf{U}(\boldsymbol{\Sigma} + \lambda\mathbf{I}_n)^{-1}\mathbf{U}^\top\mathbf{z} \\ &= \sum_{j\in[3J/4]} \sqrt{n_j}\mathbf{U}(\boldsymbol{\Sigma} + \lambda\mathbf{I}_n)^{-1}\mathbf{U}^\top\mathbf{U}\mathbf{e}_j \\ &= \sum_{j\in[3J/4]} \frac{1}{n_j + \lambda}\left(\sqrt{n_j}\mathbf{U}\mathbf{e}_j\right).\end{aligned} \tag{3.3}$$

Thus, for $i \in [n]$, the optimal solution takes the value $(\boldsymbol{\alpha}_{\mathrm{opt}})_i = (n_{j_i} + \lambda)^{-1}$ where $j_i \in [3J/4]$ is the index of the standard basis vector that the ith input point corresponds to.

Now by multiplying the $(1+\varepsilon/100)$-approximation guarantee by n/k and squaring, we have that

$$\begin{aligned}&\left\|\frac{n}{k}\hat{\boldsymbol{\alpha}} - \frac{n}{k}\boldsymbol{\alpha}_{\mathrm{opt}}\right\|_2^2 \leq \frac{\varepsilon^2}{100^2}\left\|\frac{n}{k}\boldsymbol{\alpha}_{\mathrm{opt}}\right\|_2^2 \\ &= \frac{\varepsilon^2}{100^2}\sum_{j\in[3J/4]}\left\|\frac{n/k}{n_j + \lambda}\left(\sqrt{n_j}\mathbf{U}\mathbf{e}_j\right)\right\|_2^2 \leq \frac{\varepsilon^2}{100^2}n\end{aligned} \tag{3.4}$$

so by averaging, we have that $\left(\frac{n}{k}(\hat{\boldsymbol{\alpha}})_i - \frac{n}{k}(\boldsymbol{\alpha}_{\text{opt}})_i\right)^2 \leq \varepsilon^2/100$ for at least a $99/100$ fraction of the n coordinates of i. Then on these coordinates, $\left|\frac{n}{k}(\hat{\boldsymbol{\alpha}})_i - \frac{n}{k}(\boldsymbol{\alpha}_{\text{opt}})_i\right| \leq \varepsilon/10$. Now note that on these coordinates, we have that

$$\begin{aligned}
&\left|\frac{n}{k}(\hat{\boldsymbol{\alpha}})_i - \frac{n}{k}\frac{1}{\mu_j+\lambda}\right| \\
&\leq \left|\frac{n}{k}(\hat{\boldsymbol{\alpha}})_i - \frac{n}{k}(\boldsymbol{\alpha}_{\text{opt}})_i\right| + \left|\frac{n}{k}(\boldsymbol{\alpha}_{\text{opt}})_i - \frac{n}{k}\frac{1}{\mu_j+\lambda}\right| \\
&\leq \frac{\varepsilon}{10} + \frac{n}{k}\left|\frac{1}{n_j+n/k} - \frac{1}{\mu_j+n/k}\right| \\
&\leq \frac{\varepsilon}{10} + \frac{\mu_j/100}{n/k} \leq \frac{\varepsilon}{10} + \frac{2n\varepsilon/(100k)}{n/k} = \frac{6}{50}\varepsilon.
\end{aligned} \tag{3.5}$$

Since

$$\frac{n}{k}\frac{1}{n\varepsilon/k+n/k} - \frac{n}{k}\frac{1}{2n\varepsilon/k+n/k} > \frac{\varepsilon}{3} > 2\frac{6}{50}\varepsilon \tag{3.6}$$

for $\varepsilon \in (0, 1/2)$, we can distinguish whether the ith input point has $\mu_j = n\varepsilon/k$ or $\mu_j = 2n\varepsilon/k$ on these coordinates and thus we can solve the problem of Lemma 3.4 without reading any more entries of $\mathbf{K}$ after solving the KRR instance. Thus, we have that $\mathcal{A}$ reads $\Omega(nk/\varepsilon)$ kernel entries by a reduction to Lemma 3.4.

Finally, to obtain the statement of the theorem, it remains to show that $d_{\text{eff}}^{\lambda} = \Theta(k)$. Indeed,

$$d_{\text{eff}}^{\lambda} = \sum_{j\in[3J/4]} \frac{n_j}{n_j+\lambda} = \Theta\left(\sum_{j\in[3J/4]} \frac{n\varepsilon/k}{n\varepsilon/k+n/k}\right) = \Theta(k) \tag{3.7}$$

as desired. □

We now obtain Theorem 3.1 by scaling parameters by constant factors.

Remark 3.5. *The setting of the regularization parameter in the above construction is a bit unnatural as the top $d_{\text{eff}}^{\lambda} = \Theta(k)$ singular values of the kernel matrix are of order $n\varepsilon/k$ while the regularization is of order n/k, which is $1/\varepsilon$ times larger. One can easily fix this as follows. We add $(n/k)\mathbf{e}_i$ to the end of our data set for $i = k/\varepsilon+1, k/\varepsilon+2, \ldots, k/\varepsilon+k$. This only increases our effective statistical dimension to*

$$d_{\text{eff}}^{\lambda} = \sum_{j\in[3J/4]} \frac{n_j}{n_j+\lambda} + \sum_{i=1}^{k} \frac{n/k}{n/k+\lambda} = \Theta(k) \tag{3.8}$$

and our hardness argument is clearly unaffected. Now the setting of the regularization is such that it scales as the top d_{eff}^{λ} singular values, so that it reduces the effects of the next k/ε noisy directions, which is natural.

3.2. Extensions to other kernels

The above lower bound was proven just for the dot product kernel, but essentially the same proof applies to more general kernels as well. To this end, we introduce the notion of *indicator kernels*:

Definition 3.6 (Indicator kernels)**.** *We say that $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is an* indicator kernel *if there exist $c_1 > 0$ and $c_0 < c_1$ such that $k(\mathbf{e}_i, \mathbf{e}_j) = (c_1 - c_0)\mathbb{1}(\mathbf{e}_i = \mathbf{e}_j) + c_0$ for all standard basis vectors $\mathbf{e}_i, \mathbf{e}_j$ for $i, j \in [d]$.*

Examples of such kernels include generalized dot product kernels and distance kernels, i.e. kernels of the form $k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}\cdot\mathbf{x}')$ and $k(\mathbf{x}, \mathbf{x}') = f(\|\mathbf{x}-\mathbf{x}'\|_2)$ for an appropriate function $f : \mathbb{R} \to \mathbb{R}$, which in turn include important kernels such as the polynomial kernel, the Gaussian kernel, etc. For these kernels, we show the following:

Theorem 3.7 (Query lower bound for kernel ridge regression for indicator kernels)**.** *The lower bound of Theorem 3.1 continues to hold for any algorithm computing a $(1+\varepsilon)$ relative error solution to a KRR instance with an indicator kernel (Definition 3.6) instead of the dot product kernel.*

We obtain this result by showing that for indicator kernels, α_{opt} is exactly a constant factor away from the above analysis. We defer the details to the supplementary material.

4. Lower bound for kernel k-means clustering

Next, we present our lower bound on the number of kernel entries required in order to compute a $(1+\varepsilon)$-approximate solution to kernel k-means clustering.

4.1. Finding the cost vs. assigning points

Recall that (Musco & Musco, 2017) present an algorithm for solving KKMC with a kernel querying complexity of $O\left(\frac{nk}{\varepsilon}\log\frac{k}{\varepsilon}\right)$. We now briefly present some intuition on how we would like to match this up to $\log\frac{k}{\varepsilon}$. We first note that the hardness cannot come from finding the centers of an approximately optimal clustering or approximating the cost of the optimal clustering up to $(1\pm\varepsilon)$, since there is an algorithm for finding these in $O(nk + \text{poly}(k, 1/\varepsilon, \log n))$ kernel queries: indeed, Theorem 15.5 of (Feldman & Langberg, 2011) shows how to find a strong ε-coreset of size $\text{poly}(k\log n/\varepsilon)$ in $O(nk+\text{poly}(k, 1/\varepsilon, \log n))$ kernel queries, which can then be used to compute both approximate centers and the cost (in fact, we show in the supplementary material that there is a lower bound of $\Omega(nk)$ kernel queries for the problem of computing a $(1+\varepsilon)$ relative error approximation to the cost of KKMC, so this is tight). Thus, intuitively, in order to achieve a lower bound of $\Omega(nk/\varepsilon)$ which nearly matches the dominant term in the upper bound of (Musco & Musco, 2017), we must design a hard point set in which the hardness is not in computing the cost nor the centers, but rather in assigning the n input points to their appropriate clusters.

4.2. Main lower bound

We describe our hard input distribution $\mu_{\mathrm{KKMC}}(n,k,\varepsilon)$, formed as an inner product matrix of points drawn from the ambient space $\mathbb{R}^{k/\varepsilon}$.

Definition 4.1 (Hard input distribution – kernel k-means clustering)**.** *Let $\varepsilon > 0, k, n$ be such that $k\binom{\varepsilon^{-1}}{2} = o(n)$ and $k/\varepsilon = \omega(1)$. We first define a distribution $\nu_{\mathrm{KKMC}}(k,\varepsilon)$ over vectors in $\mathbb{R}^{k/\varepsilon}$ as follows. First divide the k/ε coordinates into k blocks of $1/\varepsilon$ dimensions. Then, we sample our point set as follows: first uniformly select some block $j \in [k]$, and then uniformly select one of the $\binom{1/\varepsilon}{2}$ pairs (j_1, j_2) where $j_1, j_2 \in [1/\varepsilon]$ with $j_1 \neq j_2$, and then output $\mathbf{v}_{j,j_1,j_2} := (\mathbf{e}_{\ell_1} + \mathbf{e}_{\ell_2})/\sqrt{2}$, where $\ell_1 = j/\varepsilon + j_1, \ell_2 = j/\varepsilon + j_2$. We then generate an i.i.d. sample $\{\mathbf{x}_i\}_{i=1}^n$ of n points drawn from $\nu_{\mathrm{KKMC}}(k,\varepsilon)$ and then generate $\mathbf{K} \sim \mu_{\mathrm{KKMC}}(n,k,\varepsilon)$ by setting it to be the inner product matrix of $\{\mathbf{x}_i\}_{i=1}^n$, i.e. $\mathbf{e}_i^\top \mathbf{K} \mathbf{e}_j := \mathbf{x}_i \cdot \mathbf{x}_j$. For $\mathbf{x}$ in the support of $\nu_{\mathrm{KKMC}}(k,\varepsilon)$, we let* $\mathrm{block}(\mathbf{x})$ *denote the $j \in [k]$ such that $\mathbf{x} = \mathbf{v}_{j,j_1,j_2}$.*

Intuitively, we add "edges" between pairs of coordinates in the same block of $1/\varepsilon$ coordinates, so that clusterings that associate points in the same block together have lower cost.

In this section, we will prove the following main theorem:

Theorem 4.2 (Query lower bound for kernel k-means clustering)**.** *Let ε, k, n be such that $k\binom{\varepsilon^{-1}}{2} = o(n)$. Suppose an algorithm $\mathcal{A}$ finds a $(1 \pm \varepsilon)$-approximate solution to a kernel k-means clustering instance drawn from $\mu_{\mathrm{KKMC}}(n,k,\varepsilon)$ with probability at least $2/3$ over its random coin tosses and the input distribution. Then, $\mathcal{A}$ makes at least $\Omega(nk/\varepsilon)$ kernel queries.*

The main thrust of this proof are cost computations that show that a set of points of size at most $2n/5$ drawn from $\nu_{\mathrm{KKMC}}(k,\varepsilon)$ must have large cost. This is then related to hardness results via a reduction, using the observation that sampling from clusters with low cost must have a high probability of drawing vectors that have positive inner product.

4.3. Cost computations

4.3.1. The cost of a good clustering

Consider the clustering that assigns all points supported in the same block with each other. We first do our cost computations for the average case, where every vector $\mathbf{v}_{j,j_1,j_2}$ is drawn the same number of times. Then, the first block has center

$$\frac{1}{\binom{\varepsilon^{-1}}{2}} \sum_{(i,j)\in\binom{[\varepsilon^{-1}]}{2}} \frac{\mathbf{e}_i+\mathbf{e}_j}{\sqrt{2}} = \sqrt{2}\varepsilon \sum_{i\in[\varepsilon^{-1}]} \mathbf{e}_i \qquad (4.1)$$

and the center for the rest of the blocks is similar. Then, the cost of a single point $(\mathbf{e}_{i^*} + \mathbf{e}_{j^*})/\sqrt{2}$ is

$$\left\| \frac{\mathbf{e}_{i^*}+\mathbf{e}_{j^*}}{\sqrt{2}} - \sqrt{2}\varepsilon \sum_{i\in[\varepsilon^{-1}]} \mathbf{e}_i \right\|_2^2 = 1 - 2\varepsilon - 4\varepsilon^2. \qquad (4.2)$$

Thus, the cost of this clustering is like $n(1-2\varepsilon)$. Note that this computation also gives a lower bound on the cost of a cluster containing n/k points, since for any cluster of size n/k, we can clearly improve its cost while we can swap points to be supported on the same block.

Now by Chernoff bounds, with probability tending to 1 as $n/k\binom{\varepsilon^{-1}}{2}$ tends to infinity, the cost of this clustering is bounded above by

$$n\left(1 - \left(1 - \frac{1}{40}\right)2\varepsilon\right) = n\left(1 - \frac{79}{40}\varepsilon\right). \qquad (4.3)$$

and the cost of any cluster of size n/k is bounded below by

$$\frac{n}{k}\left(1 - \left(1 + \frac{1}{40}\right)2\varepsilon\right) = \frac{n}{k}\left(1 - \frac{81}{40}\varepsilon\right). \qquad (4.4)$$

This proves the following lemmas.

Lemma 4.3 (Cost bound for an optimal clustering)**.** *With probability at least $99/100$, the cost of an optimal clustering is at most $n(1 - (79/40)\varepsilon)$.*

Lemma 4.4 (Cost bound for a large cluster)**.** *Let C be a cluster of size at least n/k. Then with probability at least $99/100$, the cost per point of C is bounded below by $1 - (81/40)\varepsilon$.*

4.3.2. Cost lower bounds

If we cluster all n points, we will clearly not achieve lower bounds beyond the cost of the optimal clustering. However, it turns out that if restrict our attention to at most $2n/5$ points, we can find meaningful cost lower bounds for any clustering. Formally, we present the following lemma:

Lemma 4.5 (Cost bound for ℓ clusters)**.** *Suppose S is a set of at most $|S| \leq 2n/5$ points drawn from $\nu_{\mathrm{KKMC}}(k,\varepsilon)$. Then, for any clustering $\mathcal{C}_S$ of S into $\ell \leq k$ clusters,*

$$\mathrm{cost}(\mathcal{C}_S) \geq |S| - \frac{77}{40}n\varepsilon. \qquad (4.5)$$

The estimates and computations for this lemma are cumbersome and are deferred to the supplementary material.

4.4. Hardness

We now prove a lemma that translates our cost computations into a statement about sampling nonzero inner products.

Lemma 4.6 (Sampling probability of an approximate solution)**.** *Suppose that $\mathcal{C}$ is a $(1+\varepsilon/40)$-approximate solution to a kernel k-means clustering instance drawn as $\mathbf{K}\sim\mu_{\mathrm{KKMC}}(n,k,\varepsilon)$. Then for at least $2n/5$ of the points, if we uniformly sample dot products between the point and other points in its cluster, then there is at least an $\varepsilon/80$ probability of sampling a point that has nonzero inner product with the point.*

Proof. Suppose for contradiction that there are at most $2n/5$ points belonging to a cluster such that sampling uniformly from the cluster yields at least an $\varepsilon/80$ probability of sampling a point that has nonzero inner product with that point, which we refer to as a *neighbor*. Let S be the set of points that belong to such a cluster with at least probability $\varepsilon/80$ of sampling a neighbor, and let $\overline{S}$ be the complement. We first compute the cost of a point $(\mathbf{e}_i+\mathbf{e}_j)/\sqrt{2}$ in $\overline{S}$. Let C be the point's cluster and let n_i, n_j be the number of points in the cluster that has support on the ith coordinate. Then, $n_i/|C|$ and $n_j/|C|$ are both at most $\varepsilon/80$. Now note that the i and jth coordinates of the center are $n_i/(\sqrt{2}|C|)$ and $n_j/(\sqrt{2}|C|)$, so the cost of that point is at least

$$\left(\frac{1}{\sqrt{2}}-\frac{1}{\sqrt{2}}\frac{n_i}{|C|}\right)^2+\left(\frac{1}{\sqrt{2}}-\frac{1}{\sqrt{2}}\frac{n_j}{|C|}\right)^2\geq 1-\frac{1}{40}\varepsilon. \tag{4.6}$$

Then $|S|\leq 2n/5$, so we may use the bounds from Lemma 4.5 to bound the cost from below by

$$|\overline{S}|\left(1-\frac{1}{40}\varepsilon\right)+|S|-\frac{77}{40}n\varepsilon\geq n\left(1-\frac{78}{40}\varepsilon\right). \tag{4.7}$$

Now recall that by Lemma 4.3, the optimal solution has cost at most $n(1-(79/40)\varepsilon)$, so a $(1+\varepsilon/40)$-approximate solution needs to have cost at most

$$n\left(1-\frac{79}{40}\varepsilon\right)\left(1+\frac{1}{40}\varepsilon\right)<n\left(1-\frac{78}{40}\varepsilon\right) \tag{4.8}$$

which the above solution does not. □

Finally, we give the hardness result. Recall the definition of ν_{KKMC} and block from Definition 4.1 and consider the following computational problem LABELKKMC.

Definition 4.7 (LABELKKMC)**.** *We first sample n points $\{\mathbf{x}_i\}_{i=1}^n$ from our hard point set $\nu_{\mathrm{KKMC}}(k,\varepsilon)$, label the identity of the first $n/2$ points, and then ask an algorithm to correctly give* $\mathrm{block}(\mathbf{x}_i)$ *for $1/6$ of the remaining $n/2$ points.*

One can show that this problem requires reading $\Omega(nk/\varepsilon)$ kernel entries, again using standard techniques (details in the supplementary material).

Lemma 4.8 (Hardness of LABELKKMC)**.** *Suppose an algorithm $\mathcal{A}$, possibly randomized, solves* LABELKKMC *with probability at least $2/3$ over the input distribution $\nu_{\mathrm{KKMC}}(k,\varepsilon)$ and the algorithm's random coin tosses. Then, $\mathcal{A}$ makes $\Omega(nk/\varepsilon)$ kernel queries.*

Finally, we use the above lemma to show the hardness of kernel k-means clustering.

Corollary 4.9. *Suppose an algorithm $\mathcal{A}$ gives a $(1+\varepsilon/40)$-approximate kernel k-means clustering solution with probability at least $2/3$ over the input distribution $\mathbf{K}\sim\mu_{\mathrm{KKMC}}(n,k,\varepsilon)$ and the algorithm's random coin tosses. Then, $\mathcal{A}$ makes $\Omega(nk/\varepsilon)$ kernel queries.*

Proof. Using a $(1+\varepsilon/40)$-approximate algorithm for k-means clustering, we solve LABELKKMC as follows. First cluster all the points using $\mathcal{A}$. Then by Lemma 4.6, at least $2/5$ of the points must belong to a cluster such that sampling $O(1/\varepsilon)$ points within its cluster allows us to find a point such that at least one coordinate matches a labeled point's coordinate. Then, on average, we will get $1/5$ of these correct and thus $1/6$ of these with high probability by Chernoff bounds. This used $Q+O(n/\varepsilon)$ kernel queries, where Q is the number of kernel queries that the KKMC step used. Then, $Q+O(n/\varepsilon)=\Omega(nk/\varepsilon)$ so $Q=\Omega(nk/\varepsilon)$. □

Finally, Theorem 4.2 follows by rescaling ε by a constant.

5. Clustering mixtures of Gaussians

In this section, we show that our worst case kernel query complexity lower bounds for the kernel k-means clustering problem are pessimistic by a factor of k when our input instance is mixture of k Gaussians. More specifically, we prove the following theorem:

Theorem 5.1 (Clustering mixtures of Gaussians)**.** *Let $m=\Omega(\varepsilon^{-1}\log n)$ as specified by Corollary 5.3. Suppose we have a mixture of k Gaussians with isotropic covariance $\sigma^2\mathbf{I}_d$ and means $(\boldsymbol{\mu}_\ell)_{\ell=1}^k$ in $\mathbb{R}^d$. Furthermore, suppose that the Gaussian means $\boldsymbol{\mu}_\ell$ are all separated by at least $\left\|\boldsymbol{\mu}_{\ell_1}-\boldsymbol{\mu}_{\ell_2}\right\|_2\geq\Omega(\sigma\sqrt{\log k})$ as specified by Theorem 5.1 of (Regev & Vijayaraghavan, 2017) and $\left\|\boldsymbol{\mu}_{\ell_1}-\boldsymbol{\mu}_{\ell_2}\right\|_2\geq\Omega(\sigma\sqrt{\log\log n+\log\varepsilon^{-1}})$ as specified by Lemma 5.2 with $\delta=(2m+k)^{-3}$. Finally, suppose that we are in the parameter regime of* $\mathrm{poly}(k,1/\varepsilon,d,\log n)=O(\sqrt{n})$, *$d\varepsilon\geq 1$, and $k/\varepsilon\leq d\leq n/10$. Then, there exists an algorithm outputting a $(1+O(\varepsilon))$-approximate k-means clustering solution with probability at least $2/3$.*

5.1. Proof overview

By Theorem 5.1 of (Regev & Vijayaraghavan, 2017), we can in $s=\mathrm{poly}(k,1/\varepsilon,d)$ samples compute approximations $(\hat{\boldsymbol{\mu}}_\ell)_{\ell=1}^k$ to the true Gaussian means $(\boldsymbol{\mu}_\ell)_{\ell=1}^k$

$$\|\hat{\boldsymbol{\mu}}_\ell-\boldsymbol{\mu}_\ell\|_2^2\leq\sigma^2. \tag{5.1}$$

Set $t := \max\{s, 2m+k, d\}$. Then, we extract the t underlying points in $t^2 = O(n)$ kernel queries by reading a $t \times t$ submatrix of the kernel matrix and retrieving the underlying Gaussian points themselves from the inner product matrix up to a rotation, for instance by Cholesky decomposition. Since we have a sample of size at least s, we may approximate the Gaussian means. Now, of the t Gaussian points sampled, we show that we can assign which points belong to which Gaussians for $2m+k$ input points in Lemma 5.2.

Now let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two input points with the same mean. Then note that $\mathbf{x}_1 - \mathbf{x}_2 \sim \mathcal{N}(0, 2\sigma^2\mathbf{I}_d)$ and that we may compute its inner product with another input point $\mathbf{x}'$ in two kernel queries, i.e. by computing $\mathbf{x}_1 \cdot \mathbf{x}'$ and $\mathbf{x}_2 \cdot \mathbf{x}'$ individually and subtracting them. Now let $\mathbf{S} \in \mathbb{R}^{m \times d}$ be the matrix formed by placing m pairs of the above difference of pairs of Gaussians drawn from the same mean, scaled by $(2\sigma^2)^{-1}$. Then $\mathbf{S}$ is an $m \times d$ matrix of i.i.d. Gaussians, and for $n - 2m$ input points $\mathbf{x}_i$, we may compute $\mathbf{S}\mathbf{x}_i$ with $O(nm)$ kernel queries total. We then prove that for well-separated Gaussian means, $\mathbf{S}\mathbf{x}_i$ can be used to identify which true Gaussian mean $\mathbf{x}_i$ came from in corollary 5.3.

Finally, we show that clustering points to their Gaussian means is nearly optimal. We then implement this assigning for Gaussian means that are separated by more than $\varepsilon\sigma^2 d$. Otherwise, assigning to a wrong mean only $\varepsilon\sigma^2 d$ away is still nearly optimal.

5.2. Assigning input points to Gaussian means

We first present a lemma which allows us to distinguish the mean of a Gaussian point.

Lemma 5.2 (Distinguishing Gaussian means)**.** *Let $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$ be two Gaussian means separated by $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 \geq C\sigma^2 \log \delta^{-1}$ for a constant C large enough and $\delta \in (0, 1/2)$. Furthermore, let $\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2$ be approximations to the Gaussian means with $\left\|\hat{\boldsymbol{\theta}}_b - \boldsymbol{\theta}_b\right\|_2 \leq \sigma$ for $b \in \{1, 2\}$. Let $\hat{\mathbf{c}} := (\hat{\boldsymbol{\theta}}_1 + \hat{\boldsymbol{\theta}}_2)/2$. Then*

$$\begin{cases} (\mathbf{x} - \hat{\mathbf{c}}) \cdot (\hat{\boldsymbol{\theta}}_1 - \hat{\mathbf{c}}) > 0 & \textit{if } \mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}_1, \sigma^2\mathbf{I}_d) \textit{ w.p.} \geq 1 - \delta \\ (\mathbf{x} - \hat{\mathbf{c}}) \cdot (\hat{\boldsymbol{\theta}}_1 - \hat{\mathbf{c}}) < 0 & \textit{if } \mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}_2, \sigma^2\mathbf{I}_d) \textit{ w.p.} \geq 1 - \delta \end{cases}. \tag{5.2}$$

The estimates for this proof are straightforward, and the details can be found in the supplementary material.

Using Lemma 5.2, we identify the true Gaussian mean of a point with probability at least $1 - (2m+k)^{-3}$ with squared separation only $O(\sigma^2(\log\log n + \log \varepsilon^{-1} + \log k))$. Then by a union bound, we identify the true Gaussian means of $2m + k$ points simultaneously with high probability. We may then form the matrix $\mathbf{S}$ of i.i.d. Gaussians as described previously and apply it to the $n - 2m$ remaining points.

As a corollary of Lemma 5.2, we show that for Gaussian means that are separated more, with squared distance at least $\varepsilon\sigma^2 d$, we may distinguish the means with a Gaussian sketch of dimension $m = O(\varepsilon^{-1}\log\delta^{-1})$ with probability at least $1 - \delta$. In particular, we may choose the failure probability to be $\delta = (nk)^{-3}$ so that with a sketch dimension of $m = O(\varepsilon^{-1}\log(nk)^3) = O(\varepsilon^{-1}\log n)$, we can identify the correct Gaussian mean for all $n - 2m$ remaining input points simultaneously by the union bound, as claimed. That is, using $\mathbf{S}\mathbf{x}$, we can find the correct mean of $\mathbf{x}$ for Gaussians with large enough separation.

Corollary 5.3. *Let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$ be two Gaussian means separated by $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 \geq \varepsilon\sigma^2 d$, and let $\delta \in (0, 1/2)$. Let $\mathbf{S} \in \mathbb{R}^{m \times d}$ be a matrix of i.i.d. standard Gaussians. If $m \geq C\varepsilon^{-1}\log(\delta^{-1})$, for some constant C large enough, then there exists an algorithm that can decide whether $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_1, \sigma^2\mathbf{I}_d)$ or $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_2, \sigma^2\mathbf{I}_d)$ given only $\mathbf{S}$, $\mathbf{S}\mathbf{x}$, and the approximate means $\hat{\boldsymbol{\mu}}_j$, with probability at least $1 - \delta$.*

The proof comes from combining the sketching guarantees of $\mathbf{S}$ with Lemma 5.2, and the details can be found in the supplementary material.

We now put corollary 5.3 to algorithmic use by using it to assign to each point a center withing $\varepsilon\sigma^2 d$ (details in the supplementary material).

Lemma 5.4. *With probability at least $99/100$, we may simultaneously assign for each $\mathbf{x}_i$ for $i \in [n]$ a center $\boldsymbol{\mu}_{\ell_i}$ with $\left\|\boldsymbol{\mu}_{\ell_i} - \boldsymbol{\mu}_{\ell_i^*}\right\|_2^2 \leq \varepsilon\sigma^2 d$, where $\boldsymbol{\mu}_{\ell_i^*}$ is the true Gaussian mean that generated $\mathbf{x}_i$. Furthermore, the assignment algorithm that we describe only depends on $\mathbf{S}$, $\mathbf{S}\mathbf{x}_i$, and approximate means $\hat{\boldsymbol{\mu}}_j$.*

5.3. Clustering the points

Now that we have approximately assigned input points to Gaussian means in $O(nm) = \tilde{O}(n/\varepsilon)$ kernel queries, it remains to show that this information suffices to give a $(1+\varepsilon)$-approximate solution to the KKMC problem.

Theorem 5.5. *Let $d\varepsilon \geq 1$ and $k/\varepsilon \leq d \leq n/10$ and let our data set $\{\mathbf{x}_i\}_{i=1}^n$ be distributed as a mixture of k Gaussians as described before. Then assigning the $\mathbf{x}_i$ to approximate means as in Lemma 5.4 gives a $(1+8\varepsilon)$-approximate k-means clustering solution with probability at least $98/100$.*

The proof of the theorem essentially follows from noting that the cost of the optimal clustering is at least the cost of an optimal rank $2k$ approximation of the noise component of the mixture of Gaussians, which can be shown to be $(1+\varepsilon)$ within the total Frobenius norm of the noise component. We then show that finding the Gaussian means optimally allows us to approximate this cost from above as well. The details are elaborated in the supplementary material.

Acknowledgements

We would like to thank the anonymous reviewers for helpful feedback. D. Woodruff would like to thank support from the Office of Naval Research (ONR) grant N00014-18-1-2562. This work was also partly done while D. Woodruff was visiting the Simons Institute for the Theory of Computing.

References

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

Avron, H., Clarkson, K. L., and Woodruff, D. P. Sharper bounds for regularized data fitting. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, pp. 27:1–27:22, 2017.

Bach, F. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pp. 185–209, 2013.

Bach, F. R. and Jordan, M. I. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul): 1–48, 2002.

Boutsidis, C., Drineas, P., and Mahoney, M. W. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems*, pp. 153–161, 2009.

Cesa-Bianchi, N., Mansour, Y., and Shamir, O. On the complexity of learning with kernels. In *Conference on Learning Theory*, pp. 297–325, 2015.

Cohen, M. B., Elder, S., Musco, C., Musco, C., and Persu, M. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pp. 163–172. ACM, 2015.

El Alaoui, A. and Mahoney, M. W. Fast randomized kernel methods with statistical guarantees. In *Advances in Neural Information Processing Systems*, pp. 775–783, 2015.

Feldman, D. and Langberg, M. A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pp. 569–578. ACM, 2011.

Fine, S. and Scheinberg, K. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2(Dec):243–264, 2001.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

Hartigan, J. A. *Clustering algorithms*. Wiley, 1975.

Lin, M., Weng, S., and Zhang, C. On the sample complexity of random Fourier features for online learning: How many random Fourier features do we need? *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8 (3):13, 2014.

Mazumdar, A. and Saha, B. Clustering with noisy queries. In *Advances in Neural Information Processing Systems*, pp. 5788–5799, 2017.

Musco, C. and Musco, C. Recursive sampling for the Nyström method. In *Advances in Neural Information Processing Systems*, pp. 3833–3845, 2017.

Musco, C. and Woodruff, D. P. Sublinear time low-rank approximation of positive semidefinite matrices. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, pp. 672–683. IEEE, 2017.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.

Regev, O. and Vijayaraghavan, A. On learning mixtures of well-separated Gaussians. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, pp. 85–96. IEEE, 2017.

Schölkopf, B. and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Schölkopf, B., Tsuda, K., Vert, J.-P., Istrail, D. S., Pevzner, P. A., Waterman, M. S., et al. *Kernel methods in computational biology*. MIT press, 2004.

Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. Adaptively learning the crowd kernel. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 673–680. ACM, 2011. ISBN 978-1-4503-0619-5.

Yang, Y., Pilanci, M., and Wainwright, M. J. Randomized sketches for kernels: Fast and optimal nonparametric regression. *The Annals of Statistics*, 45(3):991–1023, 2017.

Zhang, J., Marszałek, M., Lazebnik, S., and Schmid, C. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

Zhang, T. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17 (9):2077–2098, 2005.