**Scalable Kernel K-Means Clustering with Nystro…m Approximation: Relative-Error Bounds**

**Shusen Wang** shusen.wang@stevens.edu

Department of Computer Science Stevens Institute of Technology

Hoboken, NJ 07030, USA

**Alex Gittens** gittea@rpi.edu

Computer Science Department Rensselaer Polytechnic Institute Troy, NY 12180, USA

**Michael W. Mahoney** mmahoney@stat.berkeley.edu International Computer Science Institute and Department of Statistics

University of California at Berkeley Berkeley, CA 94720, USA

**Editor:** Sanjiv Kumar

**Abstract**

Kernel k-means clustering can correctly identify and extract a far more varied collection of cluster structures than the linear k-means clustering algorithm. However, kernel k- means clustering is computationally expensive when the non-linear feature map is high- dimensional and there are many input points.

Kernel approximation,e.g., the Nystro…m method, has been applied in previous works to approximately solve kernel learning problems when both of the above conditions are present. This work analyzes the application of this paradigm to kernel k-means clustering, and shows that applying the linear k-means clustering algorithm to  (1 + o(1)) features constructed using a so-called rank-restricted Nystro…m approximation results in cluster assignments that satisfy a 1 + ∈ approximation ratio in terms of the kernel k-means cost function, relative to the guarantee provided by the same algorithm without the use of the Nystro…m method. As part of the analysis, this work establishes a novel 1 + ∈ relative-error trace norm guarantee for low-rank approximation using the rank-restricted Nystro…m approximation.

Empirical evaluations on the 8.1 million instance MNIST8M dataset demonstrate the scalability and usefulness of kernel k-means clustering with Nystro…m approximation. This work argues that spectral clustering using Nystro…m approximation—a popular and computationally e伍cient, but theoretically unsound approach to non-linear clustering— should be replaced with the e伍cient and theoretically sound combination of kernel k-means clustering with Nystro…m approximation. The superior performance of the latter approach is empirically veriﬁed.

**Keywords:** kernel k-means clustering, the Nystro…m method, randomized linear algebra

oc2019 Shusen Wang, Alex Gittens, and Michael W. Mahoney.

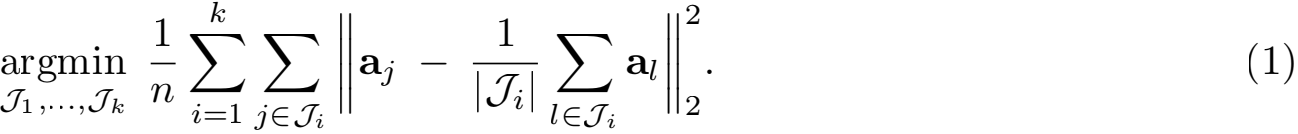
License: CC-BY 4.0, see [https://creativecommons.org/licenses/by/4.0/.](https://creativecommons.org/licenses/by/4.0/) Attribution requirements are provided at [http://jmlr.org/papers/v20/17-517.html.](http://jmlr.org/papers/v20/17-517.html)

Wang, Gittens, and Mahoney

**1. Introduction**

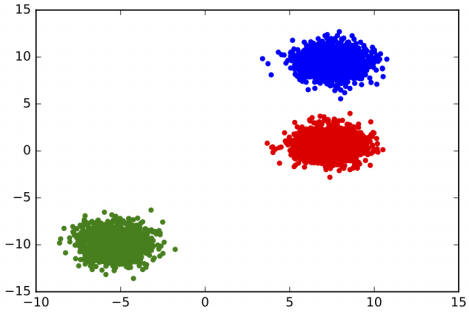
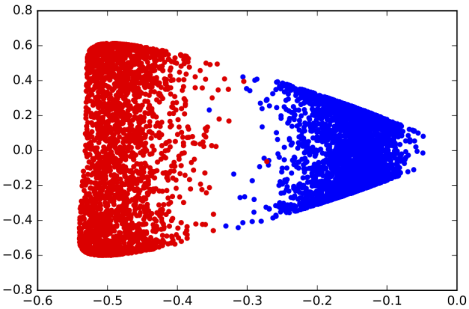
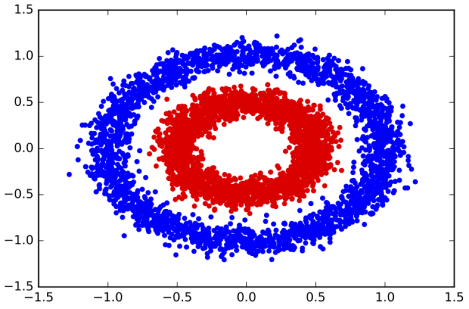
Cluster analysis divides a data set into several groups using information found only in the data points. Clustering can be used in an exploratory manner to discover meaningful groupings within a data set, or it can serve as the starting point for more advanced analyses. As such, applications of clustering abound in machine learning and data analysis, including, [inter alia: genetic expression analysis (Sharan et al.,](#bookmark2) [2002), market](#bookmark2) [segmentation (Chaturvedi et al.,](#bookmark3) [1997), social network analysis (Handcock et al.,](#bookmark3) [2007),](#bookmark4) [image segmentation (Haralick and Shapiro,](#bookmark5) [1985), anomaly detection (Chandola et al.,](#bookmark5) [2009), collaborative ﬁltering (Ungar and Foster,](#bookmark6) [1998), and fast approximate learning of](#bookmark7) [non-linear models (Si et al.,](#bookmark8) [2014)](#bookmark8).

Linear k-means clustering is a standard and well-regarded approach to cluster analysis that partitions input vectors {**a**1 ; . . . ; **a**n} C Rd into k clusters, in an unsupervised manner, by assigning each vector to the cluster with the nearest centroid. Formally, linear k-means clustering seeks to partition the set [n] = {1; . . . ; n} into k disjoint sets J1 ; . . . ; Jk by solving



[Lloyd’s algorithm (Lloyd,](#bookmark10) [1982) is a standard approach for ﬁnding local minimizers of (1),](#bookmark10) and is a staple in data mining and machine learning.

Despite its popularity, linear k-means clustering is not a universal solution to all clustering problems. In particular, linear k-means clustering strongly biases the recovered clusters towards isotropy and sphericity. Applied to the data in Figure [1(a), Lloyd’s](#bookmark11) algorithm is perfectly capable of partitioning the data into three clusters which ﬁt these assumptions. However, the data in Figure [1(b)](#bookmark12) do not ﬁt these assumptions: the clusters are ring-shaped and have coincident centers, so minimizing the linear k-means objective does not recover these clusters.



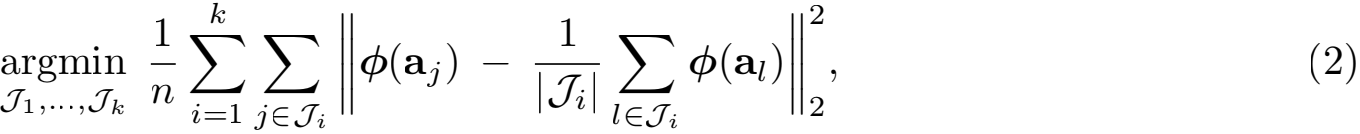
(a) (b) (c)

Figure 1: Figures [1(a)](#bookmark11) and [1(b)](#bookmark12) show two sets of two-dimensional points, with diferent colors indicating diferent clusters. Figure [1(a)](#bookmark11) is separable using linear k-means; Figures [1(b)](#bookmark12) is inseparable using linear k-means. Figure [1(c)](#bookmark12) shows the ﬁrst two dimensions of the feature vectors **k**1 ; . . . ; **k**n ∈ Rn derived from a kernel.

To extend the scope of k-means clustering to include anistotropic, non-spherical clusters such as those depicted in Figure [1(b),](#bookmark12) [Scho…lkopf et al.](#bookmark13) [(1998) proposed to perform linear](#bookmark13) k-means clustering in a nonlinear feature space instead of the input space. After choosing a feature function **φ** : Rd }→ F to map the input vectors non-linearly into feature vectors,

Scalable Kernel K-Means Clustering with Nystrm Approximation

they propose minimizing the objective function



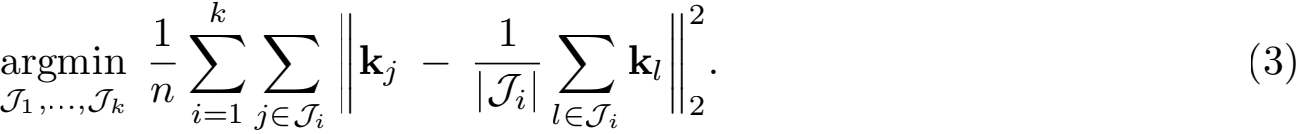
where {J1 , . . . , Jk} denotes a k-partition of [n]. The “kernel trick” enables us to minimize this objective without explicitly computing the potentially high-dimensional features, as

inner products in feature space can be computed implicitly by evaluating the kernel function

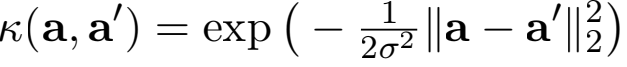
h (**a**i, **a**j) = 〈**φ**(**a** i), **φ**(**a**j)〉.

Thus the information required to solve the kernel [k-means problem (2), is present in the](#bookmark14) kernel matrix **K** = [h(**a**i, **a**j)]ij ∈ Rn ×n.

Let **K** = **VΛV**T be the full eigenvalue decomposition (EVD) of the kernel matrix and **k**1 , . . . , **k**n ∈ Rn be the rows of **VΛ**1/2 ∈ Rn ×n. It can be shown (see Appendix [B.3) that](#bookmark15) [the solution of (2) is identical to the solution of](#bookmark14)



To demonstrate the power of kernel k-means clustering, consider the dataset in Figure [1(b).](#bookmark12) We use the Gaussian RBF kernel



with σ = 0.3, and form the corresponding kernel matrix of the data in Figure [1(b).](#bookmark12) Figure [1(c)](#bookmark12) scatterplots the ﬁrst two coordinates of the feature vectors **k**1 , . . . , **k**n. Clearly, the ﬁrst coordinate of the feature vectors already separates the two classes well, so k-means clustering using the non-linear features **k**1 , · · · , **k**n has a better chance of separating the two classes.

Although it is more generally applicable than linear k-means clustering, kernel k- means clustering is computationally expensive. As a baseline, we consider the cost of [optimizing (3)](#bookmark16). The formation of the kernel matrix **K** given the input vectors **a**1 , . . . , **a**n ∈ Rd costs O(n2 d) time. [The objective in (3) can then be (approximately) minimized using](#bookmark16) Lloyd’s algorithm at a cost of O(n2 k) time per iteration. This requires the n-dimensional non-linear feature vectors obtained from the full EVD of **K**; computing these feature vectors takes O(n3 ) time, because **K** is, in general, full-rank. Thus, approximately solving the kernel [k-means clustering problem by optimizing (3) costs](#bookmark16) O(n3 + n2 d + Tn2k) time, where T is the number of iterations of Lloyd’s algorithm.

Kernel approximation techniques, including the Nystro…[m method (Nystro…m,](#bookmark17) [1930;](#bookmark17) [Williams and Seeger,2001;](#bookmark18)[Gittens and Mahoney,](#bookmark19) [2016) and random feature maps (Rahimi](#bookmark19) [and Recht,](#bookmark20) [2007), have been applied to decrease the cost of solving kernelized machine](#bookmark20) learning problems: the idea is to replace **K** with a low-rank approximation, which allows for more e伍cient computations. [Chitta et al.](#bookmark21) [(2011,](#bookmark21) [2012) proposed to apply kernel](#bookmark22) approximations to e伍ciently approximate kernel k-means clustering. Although kernel approximations mitigate the computational challenges of kernel k-means clustering, the aforementioned works do not provide guarantees on the clustering performance: how

Wang, Gittens, and Mahoney

accurate must the low-rank approximation of **K** be to ensure near optimality of the approximate clustering obtained via this method?

We propose a provable approximate solution to the kernel k-means problem based on the Nystro…m approximation. Our method has three steps: ﬁrst, extract c (c 冬 n) features using the Nystro…m method; second, reduce the features to s dimensions (k ≤ s < c) using [the truncated SVD;1](#bookmark23)third, apply any of-the-shelf lineark-means clustering algorithm upon the s-dimensional features to obtain the ﬁnal clusters. The total time complexity of the ﬁrst two steps is O(ndc + nc2 ). The time complexity of the third step depends on the speciﬁc linear k-means algorithm; for example, using Lloyd’s algorithm, the per-iteration complexity is O(nsk), and the number of iterations may depend on s[.2](#bookmark24)

Our me˜thod comes with a strong approximation ratio guarantee. Suppose we sets = k/∈

and c = O(μs/∈) for any ∈ ∈ (0, 1), where μ ∈ [1, ] is the coherence parameter of the

dominant s-dimensional singular space of the kernel matrix **K**. Also suppose the standard

kernel k-means and our approximate method use the same linear k-means clustering

algorithm,e.g., Lloyd’s algorithm or some other algorithm that comes with diferent provable

approximation guarantees. As guaranteed by Theorem [2, when the quality of the clustering](#bookmark25)

[is measured by the cost function deﬁned in (2), with probability at least 0](#bookmark14).9,our algorithm

returns a clustering that is at most 1 + O(∈) times worse than the standard kernel k-means

clustering. Our theory makes explicit the trade-of between accuracy and computation:

larger s and c lead to high accuracy and also high computational cost.

[Spectral clustering (Shi and Malik,](#bookmark26) [2000;](#bookmark26) [Ng et al.,](#bookmark27) [2002) is a popular alternative to](#bookmark27) kernel k-means clustering that can also partition non-linearly separable data such as those in Figure [1(b).](#bookmark12) Unfortunately, because it requires computing ann ×n a伍nitymatrix and the top k eigenvectors of the corresponding graph Laplacian, spectral clustering is ine伍cient for large n. [Fowlkes et al.](#bookmark28) [(2004) applied the Nystr](#bookmark28)o…m approximation to increase the scalability of spectral clustering. Since then, spectral clustering with Nystro…m approximation has [been used in many works, e.g., (Arikan,](#bookmark29) [2006;](#bookmark29) [Berg et al.,](#bookmark30) [2004;](#bookmark30) [Chen et al.,](#bookmark31) [2011;](#bookmark31) [Wang](#bookmark32) [et al.,](#bookmark32) [2016b;](#bookmark32) [Weiss et al.,](#bookmark33) [2009;](#bookmark33) [Zhang and Kwok,](#bookmark34) [2010)](#bookmark34). Despite its popularity in practice, this approach does not come with guarantees on the approximation ratio for the obtained clustering. Our algorithm, which combines kernel k-means with Nystro…m approximation, is an equally computationally e伍cient alternative that comes with strong bounds on the

approximation ratio, and can be used wherever spectral clustering is applied.

**1.1. Contributions**

[Using tools developed in (Boutsidis et al.,](#bookmark35) [2015;](#bookmark35) [Cohen et al.,](#bookmark36) [2015;](#bookmark36) [Feldman et al.,](#bookmark37) [2013),](#bookmark37) we rigorously analyze the performance of approximate kernel k-means clustering with the Nystro…m approximation, and show that a rank- 

1. This is why our variant is called the rank-restricted Nystro…m approximation. The rank-restriction serves two purposes. First, although we do not know whether the rank-restriction is necessary for the 1 + ∈ bound, we are unable to establish the bound without it. Second, the rank-restriction makes the third step, linear k-means clustering, much less costly. [For the computational beneﬁt, previous works (Boutsidis](#bookmark38) [et al.,2009,](#bookmark38)[2010,](#bookmark39)[2015;](#bookmark35)[Cohen et al.,2015;](#bookmark36)[Feldman et al.,2013) have considered dimensionality reduction](#bookmark37) for linear k-means clustering.

2. Without the rank-restriction, the per-iteration cost would be O(nck), and the number of iterations may depend on c.

Scalable Kernel K-Means Clustering with Nystrm Approximation

approximation ratio guarantee, relative to the guarantee provided by the same algorithm without the use of the Nystro…m method.

As part of the analysis of kernel k-means with Nystro…m approximation, we establish the ﬁrst relative-error bound for rank-restricted Nystro…[m approximation,3](#bookmark40) which has independent interest.

Kernel k-means clustering and spectral clustering are competing solutions to the nonlinear clustering problem, neither of which scales well with n. [Fowlkes et al.](#bookmark28) [(2004)](#bookmark28) introduced the use of Nystro…m approximations to make spectral clustering scalable; this approach has become popular in machine learning. We identify fundamental mathematical problems with this heuristic. These concerns and an empirical comparison establish that our proposed combination of kernel k-means with rank-restricted Nystro…m approximation is a theoretically well-founded and empirically competitive alternative to spectral clustering with Nystro…m approximation.

Finally, we demonstrate the scalability of this approach by measuring the performance of an Apache Spark implementation of a distributed version of our approximate kernel k- means clustering algorithm using the MNIST8M data set, which has 8.1 million instances and 10 classes.

**1.2. Relation to Prior Work**

The key to our analysis of the proposed approximate kernel k-means clustering algorithm is

a novel relative-error trace norm bound for a rank-restricted Nystro…m approximation. We restrict the rank of the Nystro…m approximation in anon-standard manner (see Remark [1)](#bookmark41). Our relative-error trace norm bound is not a simple consequence of the existing bounds for non-rank-restricted Nystro…m approximation such as the ones provided by [Gittens and](#bookmark19) [Mahoney](#bookmark19) [(2016)](#bookmark19). The relative-error bound which we provide for the rank-restricted Nystro…m approximation is potentially useful in other applications involving the Nystro…m method.

[The projection-cost preservation (PCP) property (Cohen et al.,](#bookmark36) [2015;](#bookmark36) [Feldman et al.,](#bookmark37) [2013) is an important tool for analyzing approximate linear](#bookmark37) k-means clustering. We apply [our novel relative-error trace norm bound as well as existing tools in (Cohen et al.,](#bookmark36) [2015)](#bookmark36) to prove that the rank-restricted Nystro…m approximation enjoys the PCP property. We do not rule out the possibility that the non-rank-restricted (rank-c) Nystro…m approximation satisﬁes the PCP property and/or also enjoys a 1 + ∈ approximation ratio guarantee when applied to kernel k-means clustering. However, the cost of the linear k-means clustering step in the algorithm is proportional to the dimensionality of the feature vectors, so the rank- restricted Nystro…m approximation, which produces s-dimensional feature vectors, where s < c, is more computationally desirable.

[Musco and Musco](#bookmark42) [(2017) similarly establishes a 1 +](#bookmark42) ∈ approximation ratio for the kernel k-means objective when a Nystro…m approximation is used in place of the full kernel matrix. Speciﬁcally, [Musco and Musco](#bookmark42) [(2017) shows that when](#bookmark42) c =   **K** are sampled using ridge leverage score (RLS) sampling [(Alaoui and Mahoney,](#bookmark43) [2015;](#bookmark43) [Cohen](#bookmark44) [et al.,](#bookmark44) [2017;](#bookmark44) [Musco and Musco,](#bookmark42) [2017) and are used to form a Nystr](#bookmark42)o…m approximation,

3. Similar relative-error bounds were independently developed by contemporaneous work of [Tropp et al.](#bookmark45) [(2017), in service of the analysis of a novel streaming algorithm for ﬁxed-rank approximation of positive](#bookmark45) semideﬁnite matrices. Preliminary versions of this work and theirs were simultaneously submitted to arXiv in June 2017.

Wang, Gittens, and Mahoney

Table 1: Commonly used parameters. It holds that k ≤ s ≤ c ≤ n.

|  |  |
| --- | --- |
| **Notation** | **Deﬁnition** |
| n | number of samples |
| d | number of features (attributes) |
| k | number of clusters |
| s | target rank of the Nystro…m approximation |
| c | sketch size of the Nystro…m approximation |

then applying linear k-means clustering to the c-dimensional Nystro…m features returns a clustering that has objective value at most 1 + ∈ times as large as the objective value of the best clustering. Our theory is independent of that in [Musco and Musco](#bookmark42) [(2017), and](#bookmark42) difers in that (1) [Musco and Musco](#bookmark42) [(2017) applies speciﬁcally to Nystr](#bookmark42)o…m approximations formed using RLS sampling, whereas our guarantees apply to any sketching method that satisﬁes the “subspace embedding” and “matrix multiplication” properties (see Lemma [10](#bookmark47) for deﬁnitions of these two properties); (2) [Musco and Musco](#bookmark42) [(2017) establishes a 1 +](#bookmark42) ∈ approximation ratio for the non-rank-restricted RLS-Nystro…m approximation, whereas we establish a 1+ ∈ approximation ratio for the (more computationally e伍cient) rank-restricted Nystro…m approximation.

**1.3. Paper Organization**

In Section [2, we start with a deﬁnition of the notation used throughout this paper as](#bookmark48) well as a background on matrix sketching methods. Then, in Section [3, we present our](#bookmark49) main theoretical results: Section [3.1](#bookmark50) presents an improved relative-error rank-restricted Nystro…m approximation; Section [3.2](#bookmark51) presents the main theoretical results on kernel k- means with Nystro…m approximation; and Section [3.3](#bookmark52) studies kernel k-means with kernel principal component analysis. Section [4](#bookmark53) discusses and evaluates the theoretical and empirical merits of kernel k-means clustering versus spectral clustering, when each is approximated using Nystro…m approximation. Section [5](#bookmark54) empirically compares the Nystro…m method and the random feature maps for the kernel k-means clustering on medium-scale data. Section [6](#bookmark55) presents a large-scale distributed implementation in Apache Spark and its empirical evaluation on a data set with 8.1 million points. Section [7](#bookmark56) provides a brief conclusion. Proofs are provided in the Appendices.

**2. Notation**

This section deﬁnes the notation used throughout this paper. A set of commonly used

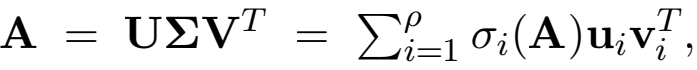
parameters is summarized in Table [1.](#bookmark46)

**Matrices and vectors.** We take **I**n to be the n × n identity matrix, **0** to be a vector or matrix of all zeros of the appropriate size, and **1**n to be the n-dimensional vector of all ones.

Scalable Kernel K-Means Clustering with Nystrm Approximation

**Sets.** The set {1, 2, · · · , n} is written as [n]. We call {J1 , · · · , Jk} a k-partition of [n] if J1 ∪ · · · ∪ Jk = [n] and Jp ∩ Jq = ∅ when p  q. Let |J| denote the cardinality of the set J.

**Singular value decomposition (SVD).** Let **A** ∈ Rn ×d and P = rank(**A**). A (compact) singular value decomposition (SVD) is deﬁned by



where **U**, **Σ** , **V** are an n × P column-orthogonal matrix, a P × P diagonal matrix with nonnegative entries, and a d × P column-orthogonal matrix, respectively. If **A** is symmetric positive semi-deﬁnite (SPSD), then **U** = **V**, and this decomposition is also called the (reduced) eigenvalue decomposition (EVD). By convention, we take σ 1 (**A**) ≥ · · · ≥ σρ(**A**). **Truncated SVD.** The matrix **A**s = Σ=1 σi is a rank-s truncated SVD of **A**, and is an optimal rank-s approximation to **A** when the approximation error is measured in a unitarily invariant norm.

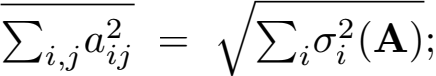
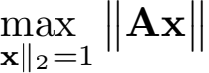
**The Moore-Penrose inverse** of **A** is deﬁned by **A**t = **VΣ**—1 **U**T .

**Leverage score and coherence.** Let **U** ∈ Rn ×ρ be deﬁned in the above and **u**i be the i-th row of **U**. The row leverage scores of **A** are li = Ⅱ**u**iⅡ for i ∈ [n]. The row coherence

of **A** is μ(**A**) =  maxiⅡ**u**iⅡ. The leverage scores for a matrix **A** can be computed exactly

in the time it takes to compute the matrix **U**; and the leverage scores can be approximated [(in theory (Drineas et al.,](#bookmark57) [2012) and in practice (Gittens and Mahoney,](#bookmark57) [2016)) in roughly](#bookmark19) the time it takes to apply a random projection matrix to the matrix **A**.

**Matrix norms.** We use three matrix norms in this paper:

Frobenius Norm: Ⅱ**A**ⅡF = √ Spectral Norm: Ⅱ**A**Ⅱ2 = Ⅱ2 = σ 1 

Trace Norm: Ⅱ**A**Ⅱ\* = Σi σi (**A**).

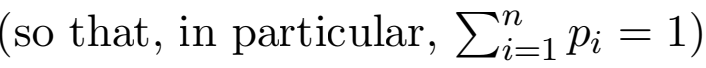
Any square matrix satisﬁes tr(**A**) ≤ Ⅱ**A**Ⅱ\* . If additionally **A** is SPSD, then tr(**A**) = Ⅱ**A**Ⅱ\* .

**Matrix sketching** Here, we brie丑y review matrix sketching methods that are commonly [used within randomized linear algebra (RLA) (Mahoney,](#bookmark58) [2011)](#bookmark58).

Given a matrix **A** ∈ Rm ×n , we call **C** = **AP** ∈ Rm ×c (typically c 冬 n) a sketch of **A** and **P** ∈ Rn ×c a sketching matrix. Within RLA, sketching has emerged as a powerful primitive, where one is primarily interested in using random projections and random sampling to [construct randomzed sketches (Mahoney,](#bookmark58) [2011;](#bookmark58) [Drineas and Mahoney,](#bookmark59) [2016)](#bookmark59). In particular, sketching is useful as it allows large matrices to be replaced with smaller matrices which are more amenable to e伍cient computation, but provably retain almost optimal accuracy in [many computations (Mahoney,2011;](#bookmark58)[Woodruf,2014)](#bookmark60). The columns of **C** typically comprise a rescaled subset of the columns of **A**, or random linear combinations of the columns of **A**; the former type of sketching is called column selection or random sampling, and the latter is referred to as random projection.

Column selection forms **C** ∈ Rm ×c using a randomly sampled and rescaled subset of the columns of **A** ∈ Rm ×n. Let p1 , · · · , pn ∈ (0, 1) be the sampling probabilities associated

Wang, Gittens, and Mahoney

with the columns of **A** . The columns of the sketch are selected identically and independently as follows: each column of **C** is randomly sampled from the columns of **A** according to the sampling probabilities andrescaled by  , where i is the index of the column of **A** that was selected. In our matrix multiplication formulation

for sketching, columnselection corresponds to a sketching matrix **P** ∈ Rn ×c that has exactly

one non-zero entry in each column, whose position and magnitude correspond to the indexof the columnselected from **A**. Uniform sampling is column sampling with p1 = · · · = pn =  , and leverage score sampling takes pi =  for i ∈ , where li is the i-th leverage score of some matrix (typically **A**, [**A** k, or an randomized approximation thereto) (Drineas et al.,](#bookmark57) [2012)](#bookmark57).

Gaussian projection is a type of random projection where the sketching matrix is taken to be **P** =  **G** ∈ Rn ×c ; here the entries of **G** are i.i.d. N(0, 1) random variables. Gaussian projection is ine伍cient relative to column sampling: the formation of a Gaussian sketch of a dense m × n matrix requires O(mnc) time. The Subsampled Randomized Hadamard Transform (SRHT) is a more e伍cient alternative that enjoys similar properties to the [Gaussian projection (Drineas et al.,](#bookmark61) [2011;](#bookmark61) [Lu et al.,](#bookmark62) [2013;](#bookmark62) [Tropp,](#bookmark63) [2011), and can be applied](#bookmark63) to a dense m × n matrix in only O(mn log c) time. The CountSketch is even more e伍cient: [it can be applied to any matrix **A** in O(nnz(**A**)) time (Clarkson and Woodruf,](#bookmark64) [2013;](#bookmark64) [Meng](#bookmark65) [and Mahoney,2013;](#bookmark65)[Nelson and Nguyˆen,2013), where nnz(·) denotes the number of nonzero](#bookmark66) entries in a matrix.

**3. Our Main Results: Improved SPSD Matrix Approximation and Kernel** k**-means Approximation**

In this section, we present our main theoretical results. We start, in Section [3.1, by](#bookmark50) presenting Theorem [1, a novel result on SPSD matrix approximation with the rank](#bookmark67)- restricted Nystro…m method. This result is of independent interest, and so we present it in detail, but in this paper we will use it to establish our main result. Then, in Section [3.2,](#bookmark51) we present Theorem [2, which is our main result for approximate kernel](#bookmark25) k-means with the Nystro…m approximation. In Section [3.3, we establish novel guarantees on kernel](#bookmark52) k-means with dimensionality reduction.

**3.1. The Nystr”om Method**

[The Nystro…m method (Nystro…m,](#bookmark17) [1930) is the most popular kernel approximation method](#bookmark17) in the machine learning community. Let **K** ∈ Rn ×n be an SPSD matrix and **P** ∈ Rn ×c be a sketching matrix. The Nystro…m method approximates **K** with **CW**t**C**T, where **C** = **KP** and **W** = **P**T **KP**. The Nystro…m method was introduced to the machine learning community by [Williams and Seeger](#bookmark18) [(2001); since then, numerous works have studied its theoretical](#bookmark18) [properties, e.g., (Drineas and Mahoney,](#bookmark68) [2005;](#bookmark68) [Gittens and Mahoney,](#bookmark19) [2016;](#bookmark19) [Jin et al.,](#bookmark69) [2013;](#bookmark69) [Kumar et al.,](#bookmark70) [2012;](#bookmark70) [Wang and Zhang,](#bookmark71) [2013;](#bookmark71) [Yang et al.,](#bookmark72) [2012)](#bookmark72).

[Empirical results in (Gittens and Mahoney,](#bookmark19) [2016;](#bookmark19) [Wang et al.,](#bookmark32) [2016b;](#bookmark32) [Yang et al.,](#bookmark72) [2012)](#bookmark72) demonstrated that the accuracy of the Nystro…m method signiﬁcantly increases when the spectrum of **K** decays fast. This suggests that the Nystro…m approximation captures the dominant eigenspaces of **K**, and that error bounds comparing the accuracy of the Nystro…m

Scalable Kernel K-Means Clustering with Nystrm Approximation

Table 2: Su伍cient sketch sizes for several sketching methods used to construct Nystro…m

approximations to the matrix **K**. Let **K**s = **V**s**Σ**s**V** be the truncated SVD of **K**,

then μ denotes the row coherence of **V**s ∈ Rn ×s and the leverage score sampling is done using the row leverage scores of **V**s.

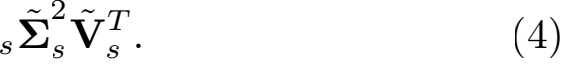
|  |  |  |
| --- | --- | --- |
| **sketching** | **sketch size (**c**)** | **time complexity (**T**)** |
| uniform sampling  leverage sampling  Gaussian projection  SRHT  CountSketch | O(μs/∈ + μslogs)  O(s/∈ + slogs)  O(s/∈)  O((s + log n)(∈-1 + logs))  O(s/∈ + s2 ) | O~(nc) O(n2 s) O(n2 c)  O(n2 log c) O(n2 ) |

approximation of **K** to that of the best rank-s approximation **K**s (for s < c) would provide a meaningful measure of the performance of the Nystro…m kernel approximation. [Gittens](#bookmark19) [and Mahoney](#bookmark19) [(2016) established the ﬁrst relative-error bounds showing that for su伍ciently](#bookmark19) large c, the trace norm error Ⅱ**K** — **CW**t**C**TⅡ\* is comparable to Ⅱ**K** — **K**sⅡ\* . Such results quantify the beneﬁts of spectral decay to the performance of the Nystro…m method, and are su伍cient to analyze the performance of Nystro…m approximations in applications such as [kernel ridge regression (Alaoui and Mahoney,](#bookmark43) [2015;](#bookmark43) [Bach,](#bookmark74) [2013) and kernel support vector](#bookmark74) [machines (Cortes et al.,](#bookmark75) [2010)](#bookmark75).

However, [Gittens and Mahoney](#bookmark19) [(2016) did not analyze the performance of rank](#bookmark19)- restricted Nystro…m approximations; they compared the approximation accuracies of the rank-c matrix **CW**t**C**T and the rank-s matrix **K**s (recall s < c). In our application to approximate kernel k-means clustering, it is the rank-s matrix ~(**C**~**W**~t **C**T)s that is of

relevance. Given **C** and **W**, the truncated SVD ((**W**t )1/2 **C**T)s = **U**s**Σ**s**V** can be found

using O(nc2 ) time. Then the rank-s Nystro…m approximation can be written as

(**CW**t**C**T )s = (**C**(**W**t )1/2)s (**C**(**W**t )1/2) = 

Theorem [1](#bookmark67) [provides a relative-error trace norm approximation guarantee for the sketch (4)](#bookmark76) and is novel; a proof is provided in Appendix [A.](#bookmark77)

**Theorem 1 (Relative-Error Rank-Restricted Nystr”om Approximation)** Let **K** ∈ Rn ×n be an SPSD matrix, s be the target rank, and ∈ ∈ (0, 1) be an error parameter. Let **P** ∈ Rn ×c be a sketching matrix corresponding to one of the sketching methods listed in Table [2](#bookmark73). Let **C** = **KP** and **W** = **P**T **KP**. Then

**"K** — (**CW**t**C**T)s **"**\* ≤ (1 + ∈) **"K** — **K**s**"**\*

holds with probability at least 0.9. In addition, there exists an n × s column orthogonal matrix **Q** such that (**CW**t**C**T)s = **K**1/2**QQ**T**K**1/2 .

**Remark 1 (Rank Restrictions)** The traditional rank-restricted Nystro…m approximation, **C**(**W**s)t**C**T , [(Drineas and Mahoney,](#bookmark68) [2005;](#bookmark68) [Fowlkes et al.,](#bookmark28) [2004;](#bookmark28) [Gittens and Mahoney,](#bookmark19) [2016;](#bookmark19) [Li et al.,](#bookmark78) [2015) is not known to satisfy a relative-error bound of the form](#bookmark78)

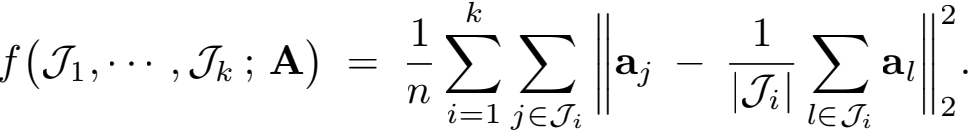
Wang, Gittens, and Mahoney

guaranteed in Theorem [1](#bookmark67). [Pourkamali-Anaraki and Becker](#bookmark79) [(2016) pointed out the](#bookmark79) drawbacks of the traditional rank-restricted Nystro…m approximation and proposed the use of the rank-restricted Nystro…m approximation (**CW**t**C**T)s in applications requiring kernel approximations, but provided only empirical evidence of its performance. This work provides guarantees on the approximation error of the rank-restricted Nystro…m approximation (**CW**t**C**T)s, and applies this approximation to the kernel k-means clustering problem. The contemporaneous work [Tropp et al.](#bookmark45) [(2017) provides similar guarantees on the approximation](#bookmark45) error of (**CW**t**C**T)s, and uses this Nystro…m approximation as the basis of a streaming algorithm for ﬁxed-rank approximation of positive-semideﬁnite matrices.

**3.2. Main Result for Approximate Kernel** k**-means**

In this section we establish the approximation ratio guarantees for the objective function of kernel k-means clustering. We ﬁrst deﬁne √-approximate k-means algorithms (where √ ≥ 1), then present our main result in Theorem [2.](#bookmark25)

Let **A** be a matrix with n rows **a**1 , · · · , **a**n. The objective function for linear k-means clustering over the rows of **A** is



The minimization of f w.r.t. the k-partition {J1 , · · · , Jk} [is NP-hard (Garey et al.,](#bookmark80) [1982;](#bookmark80) [Aloise et al.,](#bookmark81) [2009;](#bookmark81) [Dasgupta and Freund,](#bookmark82) [2009;](#bookmark82) [Mahajan et al.,](#bookmark83) [2009;](#bookmark83) [Awasthi et al.,](#bookmark84) [2015),](#bookmark84) but approximate solutions can be obtained in polynomial time. √-approximate algorithms capture one useful notion of approximation.

**Deﬁnition 1 (**√**-Approximate Algorithms)** A linear k-means clustering algorithm A√

takes as input a matrix **Z** with n rows and outputs {J , · · · , J}. We call A√ a √ -

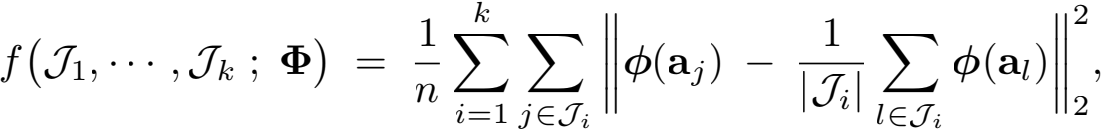
approximate algorithm if, for any such matrix **Z**,

f (J , · · · , J ; **Z**) ≤ √ · J1m,··n,Jk f (J1 , · · · , Jk ; **Z**).

Here {J1 , · · · , Jk} and {J , · · · , J} are k-partitions of [n].

Many (1 + ∈)-approximation algorithms have been proposed, but they are computation- [ally expensive (Chen,](#bookmark86) [2009;](#bookmark86) [Har-Peled and Mazumdar,](#bookmark87) [2004;](#bookmark87) [Kumar et al.,](#bookmark88) [2004;](#bookmark88) [Matousek,](#bookmark89) [2000)](#bookmark89). There are also relatively e伍cient constant factor approximate algorithms, e.g., [(Arthur and Vassilvitskii,](#bookmark90) [2007;](#bookmark90) [Kanungo et al.,](#bookmark91) [2002;](#bookmark91) [Song and Rajasekaran,](#bookmark92) [2010)](#bookmark92).

Let **φ** be a feature map, **Φ** be the matrix with rows φ(**a**1 ), . . . , φ(**a**n), and **K** = **ΦΦ**T ∈ **R**n ×n be the associated kernel matrix. Analogously, we denote the objective function for kernel k-means clustering by



where {J1 , · · · , Jk} is a k-partition of [n].

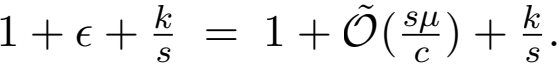
Scalable Kernel K-Means Clustering with Nystrm Approximation

**Theorem 2 (Kernel** k**-Means with Nystr”om Approximation)** Choose a sketching ma- trix **P** ∈ Rn ×c and sketch size c consistent with Table [2](#bookmark73). Let **CW**t**C**T be the previously deﬁned Nystro…m approximation of~**K**. Let~ **B** ∈ Rn ×s be any matrix satisfying **BB**T = (**CW**t**C**T)s . Let the k-partition {J1 , · · · , Jk} be the output of a √ -approximate algorithm applied to the rows of **B**. With probability at least 0.9,

f (1 , · · · , k ; **Φ**) ≤ √ (1 + ∈ +  · J1m;··n;Jk f (J1 , · · · , Jk ; **Φ**).

**Remark 2** Kernel k-means clustering is an NP-hard problem. Therefore, instead of com- paring with min f, we compare with clusterings obtained using √ -approximate algorithms. Theorem [2](#bookmark25) shows that, when uniform sampling to form the Nystro…m approximation, if  and c = , then the returned clustering has an objective value that is at most a factor of ∈ larger than the objective value of the kernel k-means clustering returned by the √ -approximate algorithm.

**Remark 3** Assume we are in apractical setting where c, the budget of column samples one can use to form a Nystro…m approximation, and k, the number of desired cluster centers, are ﬁxed. The pertinent question is how to choose s to produce a high-quality approximate clustering. Theorem [2](#bookmark25) shows that for uniform sampling, the error ratio is



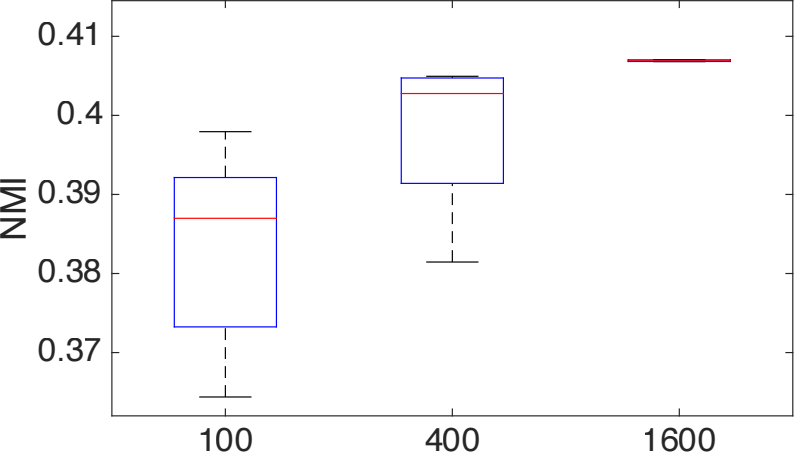
To balance the two sources of error, s must be larger than k, but not too large a fraction of c. To minimize the above error ratio, s should be selected on the order of √kc/μ . Since the matrix coherence μ (≥ 1) is unknown, it can be heuristically treated as a constant.

We empirically study the efect of the values of c and s using a data set comprising 8.1 million samples. Note that computing the kernel k-means clustering objective function requires the formation of the entire kernel matrix **K**, which is infeasible for a data set [of this size; instead, we use normalized mutual information (NMI) (Strehl and Ghosh,](#bookmark94) [2002)|a standard measure of the performance of clustering algorithms](#bookmark94)| to measure the quality of the clustering obtained by approximating kernel k-means clustering using Nystro…m approximations formed through uniform sampling. NMI scores range from zero to one, with a larger score indicating better performance. We report the results in Figure [2.](#bookmark95) The complete details of the experiments, including the experimental setting and time costs, are given in Section [6.](#bookmark55)

From Figure [2(a)](#bookmark96) we observe that larger values of c lead to better and more stable clusterings: the mean of the NMI increases and its standard deviation decreases. This is reasonable and in accordance with our theory. However, larger values of c incur more computations, so one should choose c to trade of computation and accuracy.

Figure [2(b)](#bookmark97) shows that for ﬁxed k and c, the clustering performance is not monotonic in s, which matches Theorem [2](#bookmark25) (see the discussion in Remark [3)](#bookmark93). Setting s as small as k results in poor performance. Setting s over-large not only incurs more computations, but also negatively afects clustering performance; this may suggest the necessity of rank- restriction. Furthermore, in this example, √kc = 126.5, which corroborates the suggestion made in Remark [3](#bookmark93) that setting s around √kc/μ (where μ is unknown but can be treated as a constant larger than 1) can be a good choice.

Wang, Gittens, and Mahoney



0.44 0.43 0.42 0.41 0.4 0.39

NMI

Sketch Size c

10 20 40 80 160 320 Target Dimension s

(b) Fix c = 1600 and vary s.

(a) Fix s = 20 and vary c.

Figure 2: Performance of approximate kernel k-means clustering on the MNIST8M data set, for which n = 8.1 × 106 , d = 784, and k = 10. We use Nystro…m approximations formed using uniform sampling, and the built-in k-means algorithm of Spark MLlib (a parallelized variant of the k-means++ method) with the setting MaxIter = 100.

**Remark 4** [Musco and Musco](#bookmark42) [(2017) established a](#bookmark42) 1 + ∈ approximation ratio for the kernel k-means objective value when a non-rank-restricted Nystro…m approximation is formed using ridge leverage scores (RLS) sampling; their analysis is speciﬁc to RLS sampling and does not extend to other sketching methods. By way of comparison, our analysis covers several popular sampling schemes and applies to rank-restricted Nystro…m approximations, but does not extend to RLS sampling.

**3.3. Approximate Kernel** k**-Means with KPCA and Power Method**

The use of dimensionality reduction to increase the computational e伍ciency of k-means clustering has been widely studied,e.g. [in (Boutsidis et al.,](#bookmark39) [2010,](#bookmark39) [2015;](#bookmark35) [Cohen et al.,](#bookmark36) [2015;](#bookmark36) [Feldman et al.,](#bookmark37) [2013;](#bookmark37) [Zha et al.,](#bookmark98) [2002)](#bookmark98). Kernel principal component analysis (KPCA) is [particularly well-suited to this application (Dhillon et al.,](#bookmark99) [2004;](#bookmark99) [Ding et al.,](#bookmark100) [2005)](#bookmark100). Applying Lloyd’s algorithm on the rows of **Φ** or **K**-1/2 has an O(n2 k) per-iteration complexity; if s features are extracted using KPCA and Lloyd’s algorithm is applied to the resulting s-dimensional feature map, then the per-iteration cost reduces to O(nsk). Proposition [3](#bookmark101) states that, to obtain a 1 + ∈ approximation ratio in terms of the kernel k-means objective function, it su伍ces to use s =  KPCA features. This proposition is a simple consequence [of (Cohen et al.,](#bookmark36) [2015)](#bookmark36).

**Proposition 3 (KPCA)** Let **Φ** be a matrix with n rows, and **K** = **ΦΦ**T ∈ **R**n ×n be the

corresponding kernel matrix. Let **K**s = **V**s**Λ**˜ s**V** b˜e the truncated SVD of **K** and take

**B** = **V**s**Λ**/2 ∈ Rn ×s . Let the k-partition {J1 , · · · , Jk} be the output of a √ -approximate

algorithm applied to the rows of **B**. Then

f (1 , · · · , k ; **Φ**) ≤ √ (1 +  · J1m;··n;Jk f (J1 , · · · , Jk ; **Φ**).

12

Scalable Kernel K-Means Clustering with Nystrm Approximation

**Algorithm 1** Approximate Kernel k-Means using the Power Method.

1: **Input**: kernel matrix **K** ∈ Rn ×n , number of clusters k, target dimensions (≥ k), sketch size c (≥ s), number of iterations t (≥ 1)

2: Draw a Gaussian projection matrix **P** ∈ Rn ×c ;

3: **for** all j ∈ [t] **do** 4: **P** ←− **KP**;

5: **end for**

6: Orthogonalize **P** to obtain **U** ∈ Rn ×c ; 7: Compute **C** = **KU** and **W** = **U**T **KU**;

8: Compute a **B** ∈ Rn ×s satisfying (**CW**t**C**T )s = **BB**T ; 9: **return B**.

In practice, the truncated SVD (equivalently EVD) of **K** is computed using the power

method or Krylov subspace methods. These numerical methods do not compute the

exact decomposition **K**s = **V**s**Λ**s**V**, so Proposition [3](#bookmark101) is not directly applicable. It is

useful ˜to ˜ha˜ve a theory that captures the efect of realisticly inaccurate estimates like

**K**s ≈ **V**s**Λ**s**V** on the clustering process. As one particular example, consider that general-

purpose implementations of the truncated SVD attempt to mitigate the fact that the

computed decompositions are inacc˜urate by returning very high-precision solutions, e.g.

solutions that satisfy Ⅱ(**I**n −**V**s**V**)**V**sⅡ2 ≤ 10-10 . Understanding the trade-of between the

precision of the truncated SVD solution and the impact on the approximation ratio of the

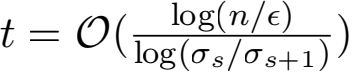
approximate kernel k-means solution allows us to more precisely manage the computational

complexity of our algorithms. Are such high-precision solutions necessary for kernel k-means

clustering?

Theorem [4](#bookmark103)answers this question by establishing that highly accurate eigenspaces are not signiﬁcantly more useful in approximate kernel k-means clustering than eigenspace estimates with lower accuracy. A low-precision solution obtained by running the power iteration for a few rounds su伍ces for kernel k-means clustering applications. We prove Theorem [4](#bookmark103) in Appendix [C.](#bookmark104)

**Theorem 4 (The Power Method)** Let **Φ** be a matrix with n rows, **K** = **ΦΦ**T ∈ Rn ×n be the corresponding kernel matrix, and σi be the i-th singular value of **K**. Fix an error

parameter ∈ ∈˜(0, 1). R˜ un Algorithm [1](#bookmark102) with  to obtain **B** ∈ Rn ×s . Let the

k-partition {J1 , · · · , Jk} be the output of a √ -approximate algorithm applied to the rows of

**B**. If c = s + O(log ), then

f (1 , · · · , k ; **Φ**) ≤ √ (1 + ∈ +  · J1m,··n,Jk f (J1 , · · · , Jk ; **Φ**).

holds with probability at least 1 − δ . If c = s, then the above inequality holds with probability 0.9 − O(s-τ ), where τ is a positive constant [(Tao and Vu,](#bookmark105) [2010)](#bookmark105) .

Note that the power method requires forming the entire kernel matrix **K** ∈ Rn ×n , which may not ﬁt in memory even in a distributed setting. Therefore, in practice, the power method may not be as e伍cient as the Nystro…m approximation with uniform sampling, which avoids forming **K**.

Theorem [2, Proposition](#bookmark25) [3, and Theorem](#bookmark101) [4](#bookmark103) are highly interesting from a theoretical perspective. These results demonstrate that s = 1 + o(1)) features are su伍cient to

Wang, Gittens, and Mahoney

ensure a (1 + ∈) approximation ratio. [Prior work (Dhillon et al.,](#bookmark99) [2004;](#bookmark99) [Ding et al.,](#bookmark100) [2005) set](#bookmark100) s = k and did not provide approximation ratio guarantees. Indeed, a lower bound in the linear [k-means clustering case due to (Cohen et al.,](#bookmark36) [2015) shows that](#bookmark36)  to obtain a 1 + ∈ approximation ratio.

**4. Comparison to Spectral Clustering with Nystro…m Approximation**

In this section, we provide a brief discussion and empirical comparison of our clustering algorithm, which uses the Nystro…m method to approximate kernel k-means clustering, with the popular alternative algorithm that uses the Nystro…m method to approximate spectral clustering.

**4.1. Background**

[Spectral clustering is a method with a long history (Cheeger,](#bookmark106) [1969;](#bookmark106) [Donath and Hofman,](#bookmark107) [1972,](#bookmark107) [1973;](#bookmark108) [Fiedler,](#bookmark109) [1973;](#bookmark109) [Guattery and Miller,](#bookmark110) [1995;](#bookmark110) [Spielman and Teng,](#bookmark111) [1996)](#bookmark111). Within [machine learning, spectral clustering is more widely used than kernel k-means clustering (Ng](#bookmark27) [et al.,](#bookmark27) [2002;](#bookmark27) [Shi and Malik,](#bookmark26) [2000), and the use of the Nystro…m method to speed up spectral](#bookmark26) clustering has been popular since [Fowlkes et al.](#bookmark28) [(2004)](#bookmark28). Both spectral clustering and kernel k-means clustering can be approximated in time linear in n by using the Nystro…m method with uniform sampling. Practitioners reading this paper may ask:

How does the approximate kernel k-means clustering algorithm presented here, which uses Nystro…m approximation, compare to the popular heuristic of combin- ing spectral clustering with Nystro…m approximation?

Based on our theoretical results and empirical observations, our answer to this reader is:

Although they have equivalent computational costs, kernel k-means clustering with Nystro…m approximation is both more theoretically sound and more e ective in practice than spectral clustering with Nystro…m approximation.

We ﬁrst formally describe spectral clustering, and then substantiate our claim regarding the theoretical advantage of our approximate kernel k-means method. Our discussion is limited to the normalized and symmetric graph Laplacians used in [Fowlkes et al.](#bookmark28) [(2004),](#bookmark28) but spectral clustering using asymmetric graph Laplacians encounters similar issues.

**4.2. Spectral Clustering with Nystr”om Approximation**

The input to the spectral clustering algorithm is an a伍nity matrix **K** ∈ R×n that measures the pairwise similarities between the points being clustered; typically **K** is a kernel matrix or the adjacency matrix of a weighted graph constructed using the data points as vertices.

Let **D** = diag(**K1**n) be the diagonal degree matrix associated with **K**, and **L** = **I**n - **D**-1/2 **KD**-1/2 be the associated normalized graph Laplacian matrix. Let **V** k ∈ Rn ×k denote the bottom k eigenvectors of **L**, or equivalently, the top k eigenvectors of **D**-1/2 **KD**-1/2 . Spectral clustering groups the data points by performing linear k-means clustering on the normalized rows of **V**k . [Fowlkes et al.](#bookmark28) [(2004) popularized the application of the Nystr](#bookmark28)o…m approximation to spectral clustering. This algorithm computes an approximate spectral

Scalable Kernel K-Means Clustering with Nystrm Approximation

clustering by: (1) fo˜rming a N˜ystro…m a˜pproximation to **K**, denoted by ; (2) comp˜uting

t˜he degree matrix **D** = diag(**K1**n) of **K**; (3) computing the top k ˜singular ˜vectors **V**k of

**D**-1/2 **KD**-1/2, which are equivalent to the bottom keigenvect˜ ors of **L** = **I**n-**D**-1/2 **KD**-1/2;

(4) performing linear k-means over the normalized rows of **V** k ∈ Rn ×k .

To the best of our knowledge, spectral clustering with Nystro…m approximation does not have a bounded approximation ratio relative to exact spectral clustering. In fact, it seems unlikely that the approximation ratio could be bounded, as there are fundamental problems with the application of the Nystro…m approximation to the a伍nity matrix.

• The a伍nity matrix **K** used in spectral clustering must be elementwise nonnegative.

However, th˜e Nystro…m approximation of such a matrix can have numerous negative

entries, so **K** is, in general, not proper input for˜ the spect˜ral clustering algorithm.

In particular, the ap˜proximated degree matrix **D** = diag(**K1**n) may have negative

diagonal entries, so **D**-1/2 is not guaranteed to be a real matrix; such˜exceptions must

be handled heuristically. The approximate asymmetric Laplacian **L** = **I**n - **KD**˜ -1

does avoid the introductio˜n of complex values; however, the negative entries in **D**-1

a Laplacian matrix. Thus the bottom eigenvectors of **L** cannot be viewed as useful coordinates for linear k-means clustering in the same way that the eigenvectors of **L** can be.

negate whole columns of **K**, leading to less meaningful negative similarities/distances.

• Even if -1/2 is real, the matrix = **I**n - -1/2-1/2˜ may not be SPSD, much less

• Such approxim˜ ation is also problematic in terms of matrix approximation accuracy.

Even when **K** approxim˜ ates **K** ˜well, which can be theoretically guaranteed, the

approximate Laplacian˜**L** = **I**n - **D**-1/2 **KD**-1/2 can be far from **L**. This is beca˜use a

small perturbation in **D** can have an out-sized in丑uence on the eigenvectors of **L**.

• One may propose˜ to approximate **N** = **D**-1/2 **KD**-1/2, rather than **K**, with aNystr˜o…m

appro˜ximation **N**; this ensures that the approximate normalized graph Laplacian **L** =

costs. Furthermore, although the resulting approximation, **L**, is SPSD, it is not a graph Laplacian: its of-diagonal entries are not guaranteed to be non-positive, and its smallest eigenvalue may be nonzero.

**I**n - **N** is SPSD. However, this approach requires forming the entirety of **K** in order

to compute the degree matrix **D**, and thus has quadratic (wit˜h n) time and memory

[In summary, spectral clustering using the Nystro…m approximation (Fowlkes et al.,](#bookmark28) [2004),](#bookmark28) which has proven to be a useful heuristic, and which is composed of theoretically principled parts, is less principled when viewed in its entirety. Approximate kernel k-means clustering using Nystro…m approximation is an equivalently e伍cient, but theoretically more principled alternative.

**4.3. Empirical Comparison with Approximate Spectral Clustering using**

**Nystr”om Approximation**

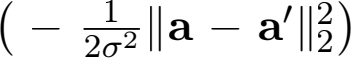
To complement our discussion of the relative merits of the two methods, we empirically compared the performance of our novel method of approximate kernel k-means clustering

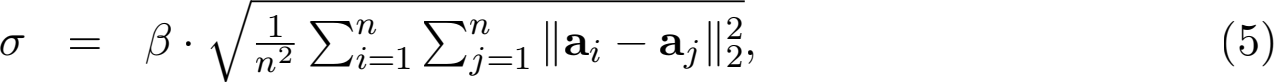
Wang, Gittens, and Mahoney

Table 3: Summary of the data sets used in our comparisons.

|  |  |  |  |
| --- | --- | --- | --- |
| **dataset** | #**instances** (n) | #**features** (d) | #**clusters** (k) |
| [MNIST (LeCun et al.,](#bookmark114) [1998)](#bookmark114)  [Mushrooms (Frank and Asuncion,](#bookmark115) [2010)](#bookmark115) [PenDigits (Frank and Asuncion,](#bookmark115) [2010)](#bookmark115) | 60,000 | 780 | 10 |
| 8,124 | 112 | 2 |
| 7,494 | 16 | 10 |

using the Nystro…m method with the popular method of approximate spectral clustering using the Nystro…m method. We used three classiﬁcation data sets, described in Table [3.](#bookmark113) The data sets used are available at [http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/](http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/) [datasets/.](http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/)

Let **a**1 , . . . , **a**n ∈ Rd be the input vectors. We take both the a伍nity matrix for spectral clustering and the kernel matrix for kernel k-means to be the RBF kernel matrix **K** = ij ∈ Rn ×n , where h  = exp  and σ is the kernel width parameter. We choose σ based on the average interpoint distance in the data sets as



where we take β = 0.2, 1, or 5.

The algorithms under comparison are all implemented in Python 3.5.2. Our imple- [mentation of approximate spectral clustering follows the code in (Fowlkes et al.,](#bookmark28) [2004)](#bookmark28). To compute linear k-means clusterings, we use the function sklearn.cluster.KMeans present in the scikit-learn package. Our algorithm for approximate kernel k-means clustering is described in more detail in Section [5.1.](#bookmark117) We ran the computations on a MacBook Pro with a 2.5GHz Intel Core i7 CPU and 16GB of RAM.

We compare approximate spectral clustering (SC) with approximate kernel k-means clustering (KK), with both using the rank-restricted Nystro…m method with uniform sampling[.4](#bookmark118) [We used normalized mutual information (NMI) (Strehl and Ghosh,](#bookmark94) [2002)](#bookmark94) to evaluate clustering performance: the NMI falls between 0 (representing no mutual information between the true and approximate clusterings) and 1 (perfect correlation of the two clusterings), so larger NMI indicates better performance. The target dimension s is taken to be k; and, for each method, the sketch size c is varied from 5s to 50s. We record the time cost of the two methods, excluding the time spent on the k-means clustering required in both algorithms[.5](#bookmark119) We repeat this procedure 100 times and report the averaged NMI and average elapsed time.

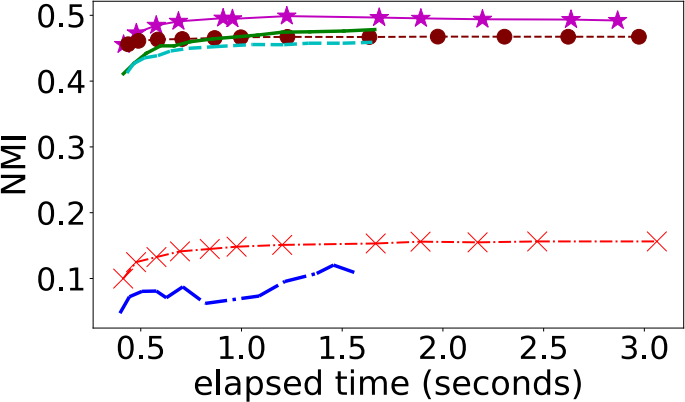
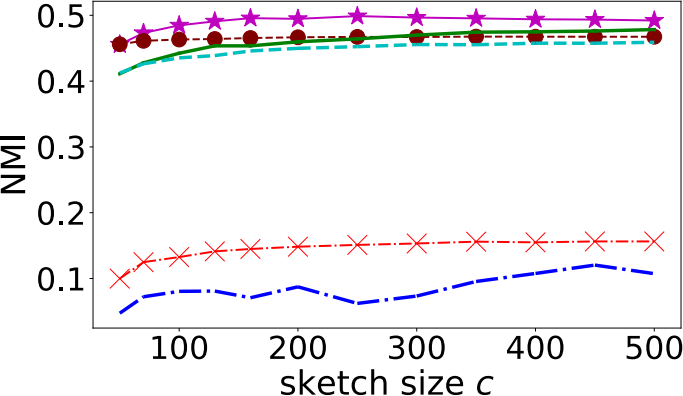
We note that, at small sketch sizes c, exceptions often arise during approximate spectral clustering due to negative entries in the degree matrix. (This is an example, as discussed in Section [4.2, of when approximate spectral clustering heuristics do not perform well.) We](#bookmark112) discard the trials where such exceptions occur.

Our results are summarized in Figure [3.](#bookmark120) Figure [3](#bookmark120) illustrates the NMI of SC and KK as a function of the sketch size c and as a function of elapsed time for both algorithms. While

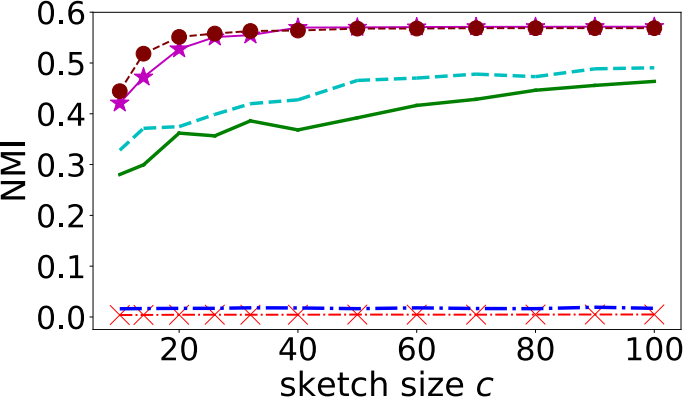
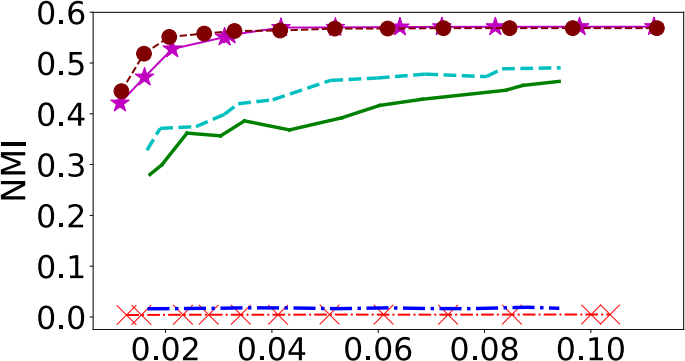
4. Uniform sampling is appropriate for the value of σ used in Eqn. [(5); see](#bookmark116) [Gittens and Mahoney](#bookmark19) [(2016)](#bookmark19) for a detailed discussion of the efect of varying σ .

5. For both SC and KK with Nystro…m approximation, the extracted feature matrices have dimension n × k, so the k-means clusterings required by both SC and KK have identical cost.

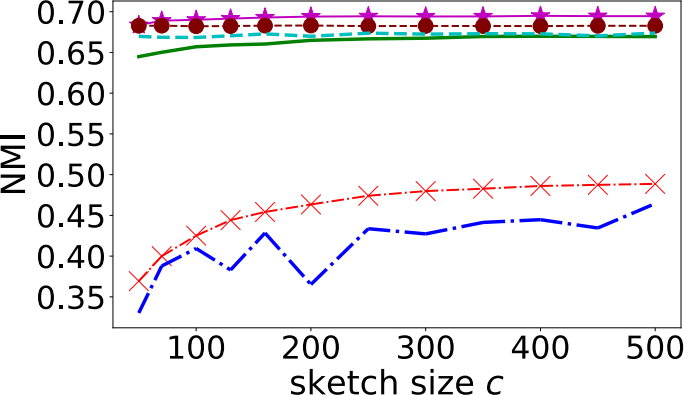
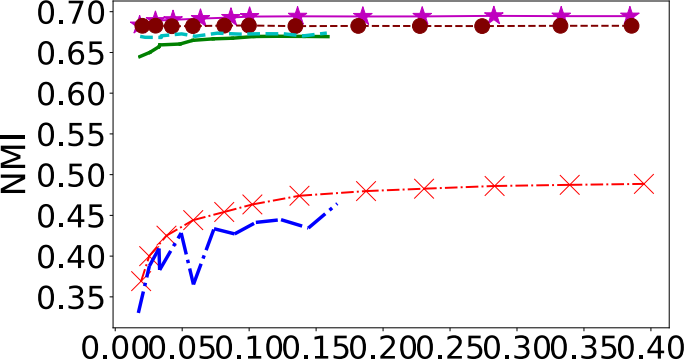
Scalable Kernel K-Means Clustering with Nystrm Approximation



(a) MNIST (n = 60, 000, d = 780, k = 10).



(b) Mushroom (n = 8, 124, d = 112, k = 2).



(c) PenDigits (n = 7, 494, d = 16, k = 10).

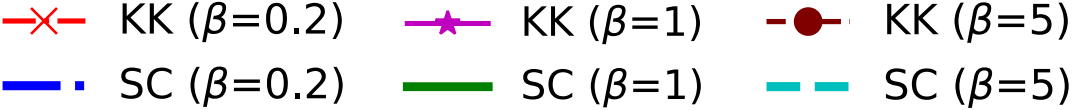


Figure 3: Comparisons between the runtimes and accuracies of approximate spectral clustering (SC) and approximate kernel k-means clustering (KK), both using uniform column sampling Nystro…m approximation.

17

Wang, Gittens, and Mahoney

|  |  |
| --- | --- |
| **Algorithm 2** Approximate Kernel k-Means Clustering using Nystro…m Approximation. | |
| 1: | **Input**: data set **a**1 , . . . , **a**n ∈ Rd , number of clusters k, target dimension s (≥ k), arbitrary integer l (> s), sketch size c (> l), kernel function h. |
| 2: | **// Step 1: The Nystr”om Method** |
| 3: | Form sketches **C** = **KP** ∈ Rn ×c and **W** = **P**T **C** ∈ Rc×c , where **K** = [h(**a**i,**a**j )]ij ∈ Rn ×n is the |
| 4: | kernel matrix and **P** ∈ Rn ×c is some sketching matrix, e.g., uniform sampling; Compute a matrix **R** ∈ Rn×l such that **RR**T = **CW**l-1**C**T ; |
| 5: | **// Step 2: Dimensionality Reduction** |
| 6: | Compute the rank-s ~truncated SVD **R**s = ; |
| 7: | Let **B** = **U**s **Σ**s (= **RV**s ) ∈ Rn ×s ; |
| 8: | **// Step 3: Linear** k**-Means Clustering** |
| 9: | Perform k-means clustering over the rows of **B**; |
| 10: | **return** the clustering results. |
| there are quantitative diferences between the results on the three data sets, the plots all show that KK is more accurate as a function of the sketch size or elapsed time than SC.  **5. Single-Machine Medium-Scale Experiments**  In this section, we empirically compare the Nystro…m method and random feature maps [(Rahimi and Recht,](#bookmark20) [2007) for kernel](#bookmark20) k-means clustering. We conduct experiments on the data listed in Table [3.](#bookmark113) For the Mushrooms and PenDigits data, we are able to evaluate the objective function value of kernel k-means clustering.  **5.1. Single-Machine Implementation of Approximate Kernel** k**-Means**  Our algorithm for approximate kernel k-means clustering comprises three steps: **Nystr”om approximation**, **dimensionality reduction**, and **linear** k**-means clustering**. Both the single-machine as well as the distributed variants of the algorithm are governed by three parameters: s, the number of features used in the clustering; l, a regularization parameter; and c, the sketch size. These parameters satisfy k ≤ s < l ≤ c 冬 n.  1. **Nystr”om approximation.** Let c be the sketch size and **P** ∈ Rn ×c be a sketching matrix. Let **C** = **KP** and **W** = **P**T **KP** = **P**T **C**. The standard Nystro…m approximation is **CW**t**C**T; small singular values in **W** can lead to instability in the Moore-Penrose inverse, so a widely used heuristic is to choose l < c and use **CW**l-1 **C**T instead of the standard Nystro…m approximation[.6](#bookmark122) We set l = 「c/2l  (arbitrarily). Let **W** l = **UW**,l**ΛW**,l**U**,l be the truncated SVD of **W** and return  **R** = **CUW**,l**ΛW**-1,2 ∈ Rn ×l as the output of the Nystro…m method. | |

6. The Nystro…m approximation **CW**t**C**T is correct in theory, but the Moore-Penrose inverse often causes numerical errors in practice. The Moore-Penrose inverse drops all the zero singular values, however, due to the ﬁnite numerical precision, it is di伍cult to determine whether a singular value, say 10-12 , should be zero or not, and this makes the computation unstable: if such a small singular value is believed to be zero, it will be dropped; otherwise, the Moore-Penrose inverse will invert it to obtain a singular value of 1012 . Dropping some portion of the smallest singular values is a simple heuristic that avoids this instability. This is why we heuristically use **CW**l-1**C**T instead of **CW**t**C**T . Currently we do not have theory for this heuristic. [Chiu and Demanet](#bookmark123) [(2013) considers the theoretical implications of this](#bookmark123) regularization heuristic, but their results do not apply to our problem.

Scalable Kernel K-Means Clustering with Nystrm Approximation

2. **Dimensionality reductio**˜**n.** Let s ∈ Rl ×s contain the dominant s right singular

vectors of **R**. Let **B** = **RV** s ∈ Rn ×s. It can be veriﬁed that **BB**T = (**CW** l-1 **C**T)s ,

which is our desired rank-restricted Nystro…m approximation.

3. **Linear** k**-means clustering.** With **B** ∈ Rn ×s at hand, use an arbitrary of-the-shelf linear k-means clustering algorithm to cluster the rows of **B**.

See Algorithm [2](#bookmark121) for the single-machine version of this approximate kernel k-means clustering algorithm. Observe that we can use uniform sampling to form **C** and **W**, and thereby avoid computing most of **K**.

Let **R** ∈ Rn ×c be the feature matrix computed by random feature maps (RFM). To

make the comparison fair, we perform **dimensionality reduction** for RFM˜[(Rahimi and](#bookmark20)

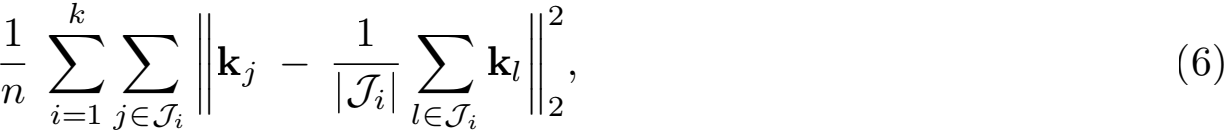
[Recht,2007) in the same way as described in Algorithm](#bookmark20) [2](#bookmark121)to compute **B** = **RV** s, and apply linear k-means clustering on the rows of **B**.

**5.2. Comparing Nystr”om, Random Feature Maps, and Two-Step Method**

We empirically compare the clustering performances of kernel approximations formed using Nystro…[m, random feature map (RFM) (Rahimi and Recht,](#bookmark20) [2007), and the two-step](#bookmark20) [method (Chitta et al.,](#bookmark21) [2011) on the data sets detailed in Table](#bookmark21) [3.](#bookmark113)

[We use the RBF kernel with width parameter given by (5); Figure](#bookmark116)[3](#bookmark120)indicates that β = 1 is a good choice for these data sets. We conduct dimensionality reduction for both Nystro…m and RFM to obtain s-dimensional features, and consider three choices: s = k, s =「√ck1, and without dimensionality reduction (equivalently, s = c).

The quality of the clusterings is quantiﬁed using both normalized mutual information [(NMI) (Strehl and Ghosh,](#bookmark94) [2002) and the objective function value:](#bookmark94)



where **k**1 , · · · , **k**n ∈ Rn are the columns of the kernel matrix **K**, and the disjoint sets J1 , · · · , Jk re丑ect the clustering.

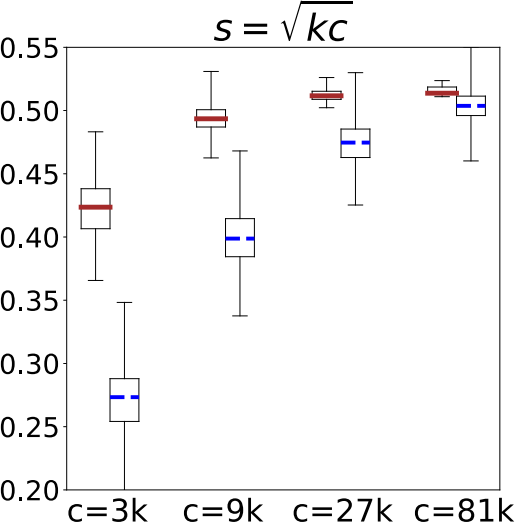
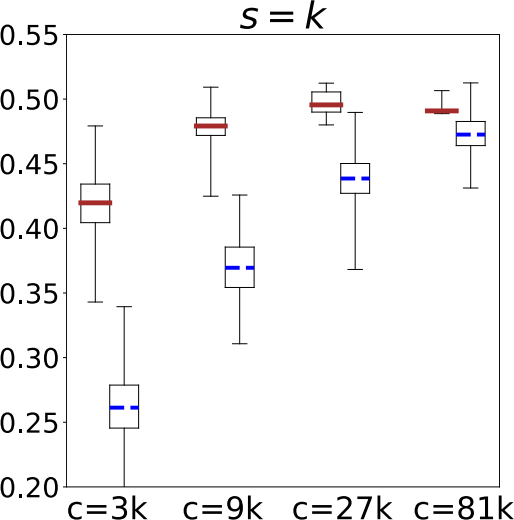
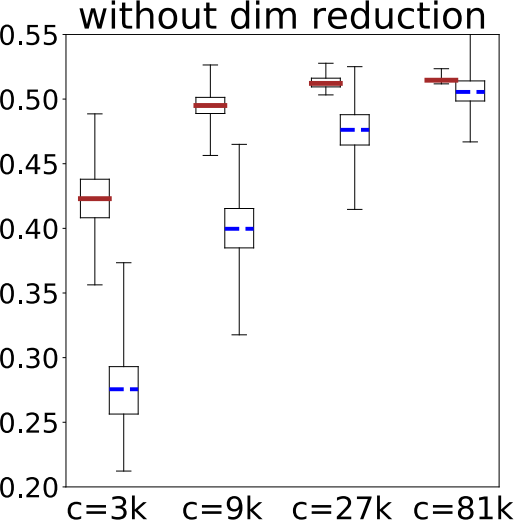
We repeat the experiments 500 times and report the results in Figures [4](#bookmark124) and [5.](#bookmark125) The experiments show that as measured by both NMIs and objective values, the Nystro…m method outperforms RFM in most cases. Both the Nystro…m method and RFM are consistently [superior to the two-step method of (Chitta et al.,](#bookmark21) [2011), which requires a large sketch size](#bookmark21). All the compared methods improve as the sketch size c increases.

Judging from these medium-scale experiments, the target rank s has little impact on the NMI and clustering objective value. This phenomenon is not general; in the large-scale experiments of the next section we see that setting s properly allows one to obtain a better NMI than an over-small or over-large s.

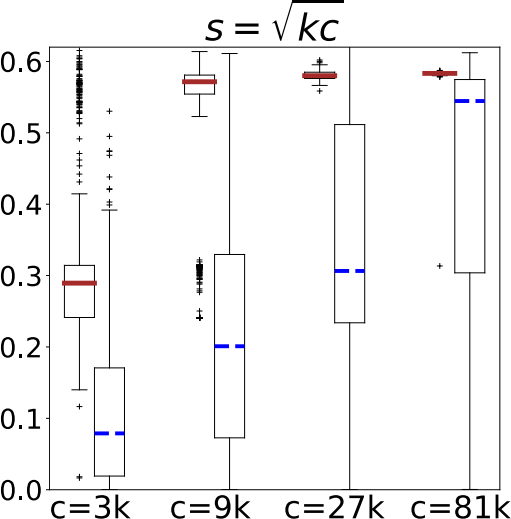
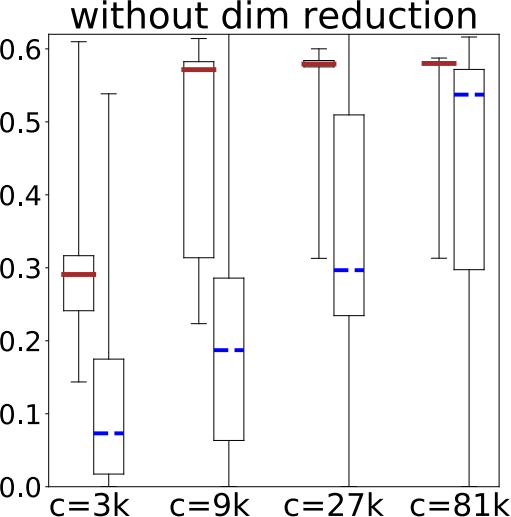
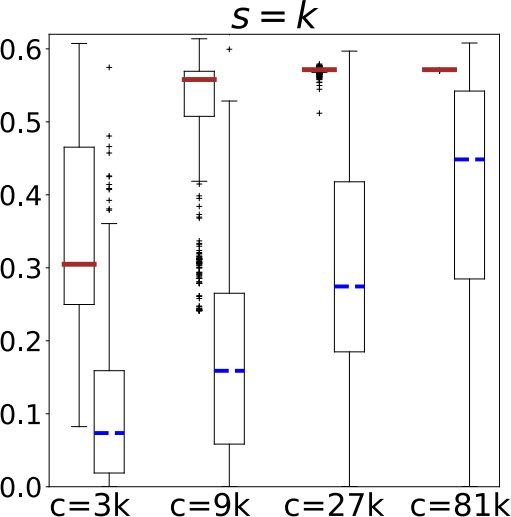
**6. Large-Scale Experiments using Distributed Computing**

In this section, we empirically study our approximate kernel k-means clustering algorithm on large-scale data. We state a distributed version of the algorithm, implement it in Apache

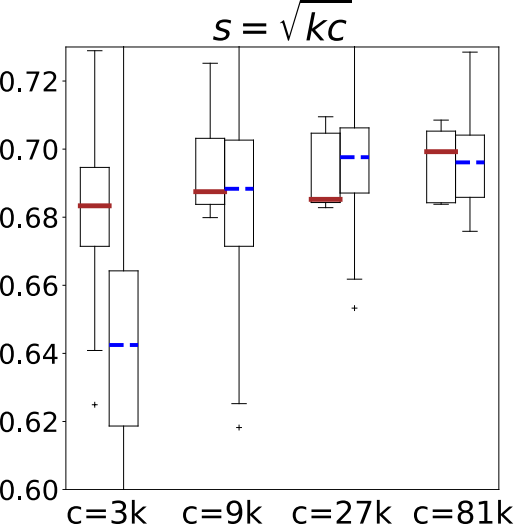
Wang, Gittens, and Mahoney

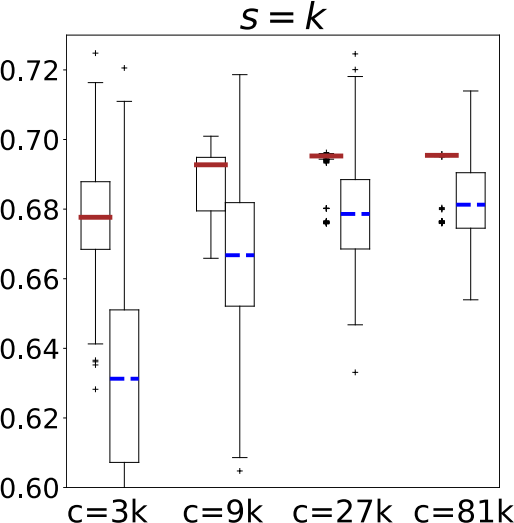
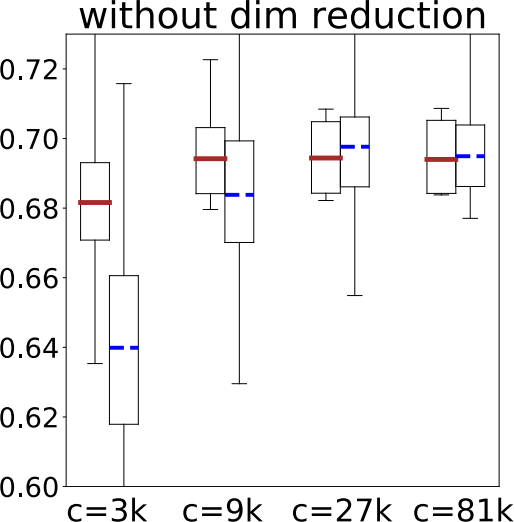


(a) MNIST (n = 60, 000, d = 780, k = 10). The results of Nystr¨om and RFM are plotted in the ﬁgures. The medians of the two-step method with c = 3k, 9k, 27k, and 81k are 0.072, 0.012, 0.017, and 0.032, respectively.



(b) Mushroom (n = 8, 124, d = 112, k = 2). The results of Nystr¨om and RFM are plotted in the ﬁgures. The medians of the two-step method with c = 3k, 9k, 27k, and 81k are 0.123, 0.224, 0.263, and 0.494, respectively.





(c) PenDigits (n = 7, 494, d = 16, k = 10). The results of Nystr¨om and RFM are plotted in the ﬁgures. The medians of the two-step method with c = 3k, 9k, 27k, and 81k are 0.399, 0.413, 0.422, and 0.421,

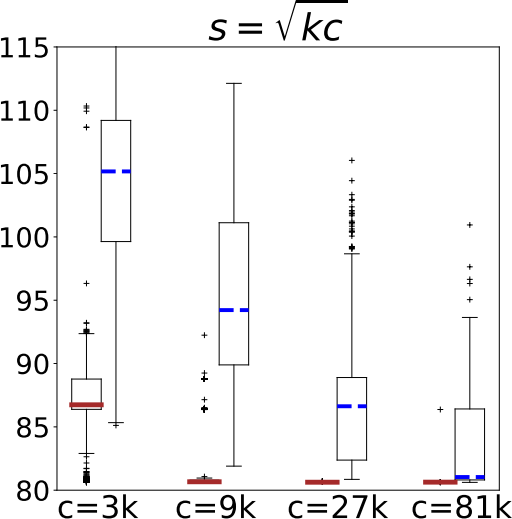
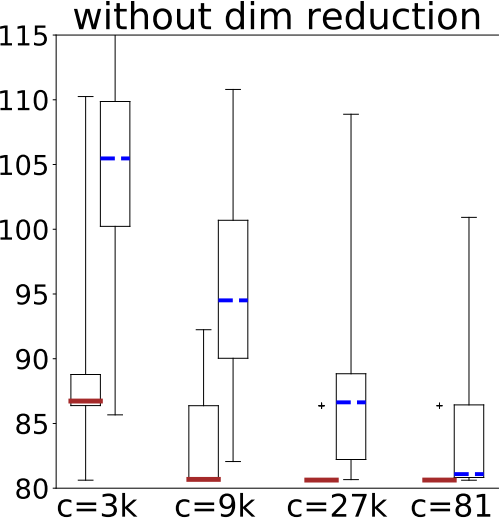
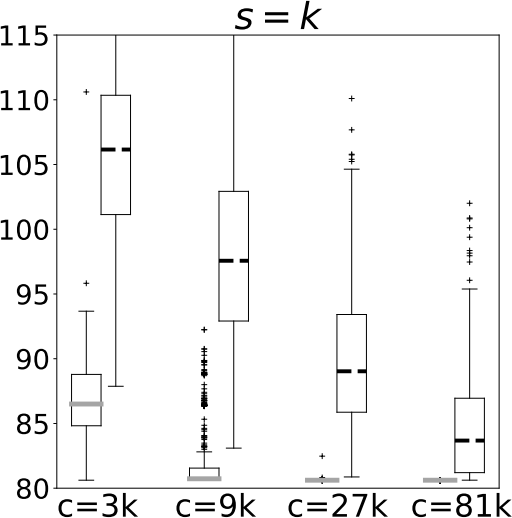
respectively.



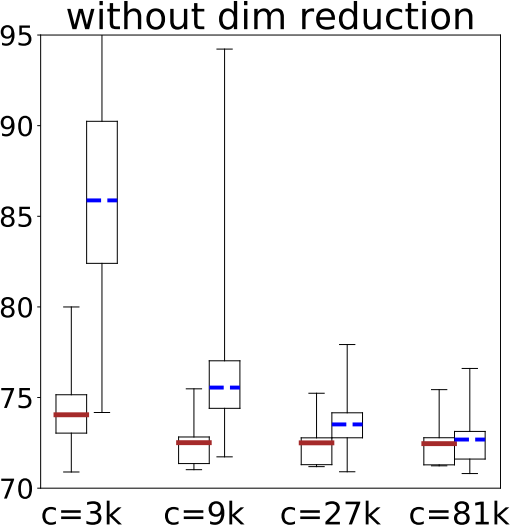
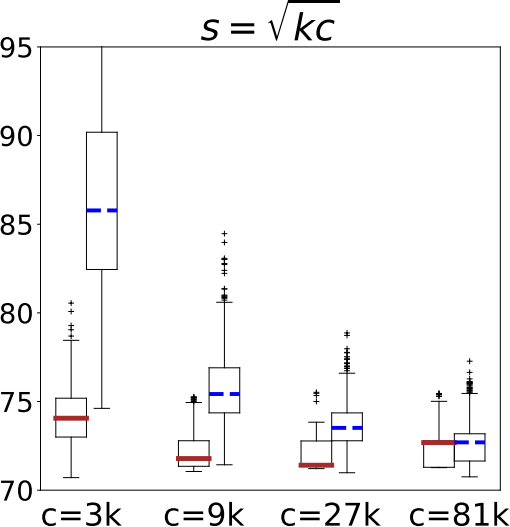
Figure 4: Quality of approximate kernel k-means clusterings using Nystro…m,random feature [maps (Rahimi and Recht,](#bookmark20) [2007), and the two-step method (Chitta et al.,](#bookmark20) [2011)](#bookmark21). (Dimensionality reduction is not applicable to the two-step method). The y-axis reports the normalized mutual information (NMI).

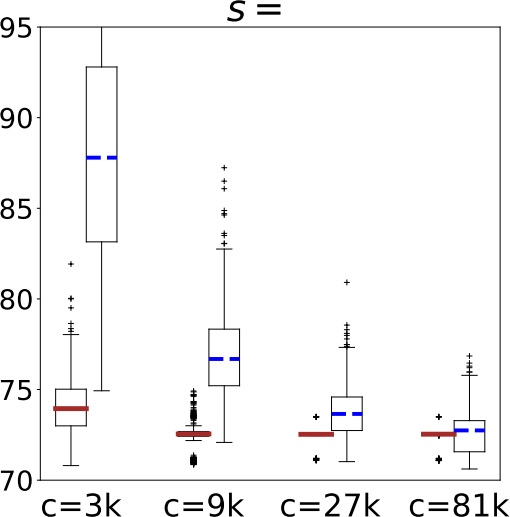
20

Scalable Kernel K-Means Clustering with Nystrm Approximation



(a) Mushroom (n = 8, 124, d = 112, k = 2). The results of Nystr¨om and RFM are plotted in the ﬁgures. The medians of the objective values of the two-step method with c = 3k, 9k, 27k, and 81k are 101.5, 92.8, 89.5, and 85.3, respectively. A random partition results in a median objective function value of 119.5.





(b) PenDigits (n = 7, 494, d = 16, k = 10). The results of Nystr¨om and RFM are plotted in the ﬁgures. The medians of the two-step method with c = 3k, 9k, 27k, and 81k are 210.5, 198.2, 193.5, and 193.4, respectively. A random partition results in a median objective function value of 395.4.



Figure 5: Quality of approximate kernel k-means clusterings obtained using rank-restricted Nystro…[m, RFMs (Rahimi and Recht,](#bookmark20) [2007), and the two-step method (Chitta](#bookmark20) [et al.,2011)](#bookmark21). (Dimensionality reduction is not applicable to the two-step method). The y-axis reports the kernel k-means clustering objective function value.

21

Wang, Gittens, and Mahoney

|  |
| --- |
| **Algorithm 3** Distributed Approximate Kernel k-Means Clustering using Nystro…m Approximation. |
| 1: **Input**: data set **a**1 , . . . , **a**n ∈ Rd distributed among m machines, number of clusters k, target dimension s (≥ k), arbitrary integer l > s, sketch size c (> l), kernel function h.  2: **// Step 1: The Nystr”om Method**  3: Sample c vectors from **a**1 , . . . , **a**n to form **a** , . . . , **a** and send to the driver;  4: Driver computes **W** = [h(a, a)]ij ∈ Rc×c and a matrix **Z** ∈ Rc×l satisfying **ZZ**T = **W**l—1; 5: Broadcast **Z** and **a** , . . . , **a** to all executors;  6: **for** all j ∈ [n] **do**  7: Each executor locally computes **c**j = [h(**a**j , **a**); . . . ; h(**a**j , **a**)] ∈ Rc and **r**j = **Z**T**c**j ∈ Rl ;  8: **end for**  9: **// St**˜**ep 2: Dimensionality Reduction**  10: Let **V**s contain the dominant s right singular vectors of the distributed row matrix **R** =  [**r**1, . . . , **r**n]T˜ ∈ Rn×l (computed using a distributed truncated SVD algorithm);  11: Broadcast **V**s ∈ Rl ×s to all executors; 12: **for** all j ∈ [n] **do**  13: Each executor locally computes **b**j = **r**j ∈ Rs ;  14: **end for**  15: **// Step 3: Linear** k**-Means Clustering**  16: Perform k-means clustering over the Nystro…m features **b**1, . . . , **b**n ∈ Rs (using a distributed linear k-means clustering algorithm).  17: **return** the clustering results. |
| [Spark7 , and evaluate its performance on NERSC’s Cori supercomputer](#bookmark127). We investigate the efect of increased parallelism, sketch size c, and target dimension s.  Algorithm [3](#bookmark126) is a distributed version of our method described in Section [5.1.](#bookmark117) Again, we use uniform sampling to form **C** and **W** to avoid computing most of **K**. We mainly focus on the **Nystr”om approximation** step, as the other two steps are well supported by distributed computing systems such as Apache Spark.  **6.1. Experimental Setup**  We implemented Algorithm [3](#bookmark126) [in the Apache Spark framework (Zaharia et al.,](#bookmark128) [2010,](#bookmark128) [2012),](#bookmark129) using the Scala API. We computed the Nystro…m approximation using the matrix operations [provided by Spark, and invoked the MLlib library for machine learning in Spark (Meng et al.,](#bookmark130) [2016) to perform the dimensionality reduction and linear](#bookmark130) k-means clustering steps. For the linear k-means clustering, we set the maximum number of iterations to 100.  We ran our experiments on Cori Phase I, a NERSC supercomputer, located at Lawrence Berkeley National Laboratory. Cori Phase I is a Cray XC40 system with 1632 compute nodes, each of which has two 2.3GHz 16-core Haswell processors and 128GB of DRAM. The Cray Aries high-speed interconnect linking the compute nodes is conﬁgured in a dragon丑y topology.  We used the MNIST8M data set to conduct our empirical evaluations; this data set has n = 8.1 × 106 instances, d = 784 features, and k = 10 clusters. We vary c and s and set l =  . [We chose the RBF kernel width parameter according to (5) with](#bookmark116) β = 1.0. We use |

7. This implementation is available at [https://github.com/wangshusen/SparkKernelKMeans.git.](https://github.com/wangshusen/SparkKernelKMeans.git)

Scalable Kernel K-Means Clustering with Nystrm Approximation

Table 4: The mean and standard deviation of the runtimes of Algorithm [3, in seconds, as](#bookmark126) a function of the number of compute nodes. We report the total runtime as well as the runtimes of the three stages: **Nystr”om approximation**, **dimensionality reduction (DR)**, and **linear** k**-means clustering**. The total time is the sum of the three stages and all the Spark overheads, e.g., Spark initialization. We ﬁx k = 10 and c = 400. In the upper table, we set s = 20, and the NMI is 0.400 ± 0.009; in the lower table, we set s = 80, and the NMI is 0.410 ± 0.010.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 8 Nodes | 16 Nodes | 32 Nodes | 64 Nodes | 128 Nodes |
| **Nystr“om** | 2998.6 ± 263.9 | 1268.8 ± 106.4 | 665.2 ± 39.6 | 369.7 ± 126.1 | 181.4 ± 22.2 |
| **DR** | 37.4 ± 58.3 | 63.8 ± 72.5 | 38.9 ± 18.2 | 87.3 ± 37.8 | 183.9 ± 89.1 |
| k**-means** | 117.2 ± 132.3 | 153.1 ± 98.8 | 119.9 ± 76.9 | 223.2 ± 88.4 | 391.0 ± 155.9 |
| **Total** | 3201.5 ± 344.0 | 1532.0 ± 146.9 | 867.9 ± 97.5 | 734.7 ± 152.0 | 828.2 ± 210.4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 8 Nodes | 16 Nodes | 32 Nodes | 64 Nodes | 128 Nodes |
| **Nystr“om** | 3008.4 ± 385.6 | 1312.9 ± 141.2 | 696.0 ± 111.8 | 342.5 ± 16.9 | 197.0 ± 24.1 |
| **DR** | 53.0 ± 41.9 | 58.7 ± 22.7 | 94.6 ± 31.8 | 179.4 ± 55.8 | 470.6 ± 118.5 |
| k**-means** | 58.2 ± 24.7 | 80.9 ± 31.8 | 104.6 ± 28.7 | 211.3 ± 103.1 | 501.7 ± 162.9 |
| **Total** | 3168.1 ± 434.1 | 1492.6 ± 135.3 | 940.6 ± 139.3 | 775.9 ± 151.6 | 1232.4 ± 232.1 |

Cori’s default setting of Spark conﬁgurations. For each setting of parameters, we repeated

the experiments 10 times and recorded the NMIs and elapsed time.

**6.2. Efect of Increased Parallelism**

We varied the number of nodes to test the impact of increased parallelism on each of the three steps in Algorithm [3.](#bookmark126) We set the sketch size to c = 400 and the target dimension to s = 20 or 80. Table [4](#bookmark131) reports the mean and standard deviation of the elapsed times. As a reference point, using 32 nodes, our algorithm takes 15 minutes on average to group the 8.1 million input instances into 10 clusters. We also plot the elapsed times in Figure [6.](#bookmark132)

The **Nystr”om method** scales very well with the increase of nodes: its elapsed time is inversely proportional to the number of nodes. This is because the **Nystr”om method** requires only 2 rounds of communications; the elapsed time is spent mostly on computation on the executors.

**Dimensionality reduction (DR)** and **linear** k**-means clustering** are highly iterative and thus have high latency, incur straggler delays, and have large scheduler and communication overheads. [See (Gittens et al.,](#bookmark133) [2016) for an in-depth discussion of the](#bookmark133) performance concerns when using Spark for distributed matrix computations. As the number of nodes increases from 32 to 64 and higher, **DR** and k**-means** exhibit anti- scaling behavior: as the number of nodes goes up, their runtimes increase rather than decrease. For **DR**, the input dimension is l =  = 200, and the target dimension is s = 20 or 80; while for k**-means**, the input dimension is s = 20 or 80. Clearly, the issue here is not that **DR** and k**-means** are computationally intensive. Instead, as the number of nodes increase, although the per-node computational time decreases, the Spark communications overheads are increasing, making **DR** and k**-means** less performant. Such

T ime (s)

T ime (s)

Wang, Gittens, and Mahoney

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Other  K-Means  DR  Nystrom | | |
|  |  |  |

8 16 32 64 128

Nodes

(a) k = 10, s = 20, c = 400.



|  |  |  |  |
| --- | --- | --- | --- |
|  | Other  K-Means  DR  Nystrom | | |
|  |  |  |

8 16 32 64 128

Nodes

(b) k = 10, s = 80, c = 400.

Figure 6: The mean of the runtimes of Algorithm [3, in seconds, as a function of the number](#bookmark126) of compute nodes.

3500

3000

2500

2000

1500

1000

500

0

3500

3000

2500

2000

1500

1000

500

0

behavior has been characterized and studied in other Spark implementations of linear [algebra algorithms (Gittens et al.,](#bookmark133) [2016)](#bookmark133).

Table 5: The NMI and elapsed time (seconds) for varying sketch sizes c. Here “T” denotes the elapsed time. We use 32 nodes. In the upper table, we set s = 20; in the lower table, we set s = 80.

|  |  |  |  |
| --- | --- | --- | --- |
|  | c = 100 | c = 400 | c = 1, 600 |
| NMI  T(Nystro…m)  T(DR)  T(k-Means)  T(Total) | 0.3833 ± 0.0117  138.5 ± 62.1  1.2 ± 0.5  60.9 ± 28.7  227.4 ± 101.3 | 0.3975 ± 0.0112  665.2 ± 39.5  38.9 ± 18.2  119.9 ± 76.9  867.9 ± 97.5 | 0.4069 ± 0.0001  2634.5 ± 94.4  31.3 ± 6.6  86.9 ± 27.3  2785.4 ± 83.2 |

|  |  |  |
| --- | --- | --- |
|  | c = 400 | c = 1600 |
| NMI  T(Nystro…m)  T(DR)  T(k-Means)  T(Total) | 0.4101 ± 0.0101  696.0 ± 111.8  94.6 ± 31.8  104.6 ± 28.7  940.6 ± 139.3 | 0.4233 ± 0.0131  2728.1 ± 190.3  97.9 ± 20.6  88.0 ± 23.9  2952.2 ± 174.5 |

**6.3. Efect of Sketch Size** c

We executed our Spark implementation using 32 compute nodes, setting s = 20 or 80, and varying the target dimension c. Table [5](#bookmark134)reports the observed normalized mutual information

T ime (s)

T ime (s)

Scalable Kernel K-Means Clustering with Nystrm Approximation

|  |  |  |
| --- | --- | --- |
| Other  K-Means  DR  Nystrom | |  |
|  |  |

100 400 1600

Sketch Size c (a) k = 10 and s = 20.

|  |  |
| --- | --- |
|  |  |

400 1600

Sketch Size c (b) k = 10 and s = 80.

Figure 7: The mean of the runtimes of Algorithm [3, in seconds, as a function of the sketch](#bookmark126) size c.

3000

2500

2000

1500

1000

500

0

3000

2500

2000

1500

1000

500

0

[(NMI) (Strehl and Ghosh,](#bookmark94) [2002) and elapsed times](#bookmark94). As predicted by our theory, for ﬁxed k and s, larger c always leads to better performance.

In Figure [7](#bookmark135) we plot the runtime as a function of the sketch size c. The plots shows the signiﬁcant in丑uence of c on the running time. In Figure [7, the runtime of](#bookmark135) Nystro…m grows superlinearly with c. According to our analysis, the computational and communication costs of Nystro…m are both superlinear in c. Therefore, the user should keep in mind this trade-of between the clustering performance and the computational cost.

**6.4. Efect of Target Dimension** s

We executed our Spark implementation using 32 computational nodes, ﬁxing k = 10 and c = 1600 and varying the target dimension s. Table [6](#bookmark136) reports the observed NMIs and elapsed times. For ﬁxed k and c, a moderately large s leads to better clustering performance. However, as s grows,  decreases but  increases, so as predicted in Remark [3, the clustering](#bookmark93) performance is nonmonotonic in s. Indeed, as s grows from 80 to 160, the NMI deteriorates. In practice, for ﬁxed k and c, one should set s moderately large, but not over-large.

In Figure [8](#bookmark137) we plot the runtime as a function of the target dimension s. Note that s does not afect the time cost of the Nystro…m method. As s grows, the time costs of dimensionality reduction and linear k-means clustering both increase. This implies that a moderate s is good for computational purpose. Previously, in Figure [2, we plot the NMI](#bookmark95) as a function of s while ﬁxing k and c. Figure [2](#bookmark95) shows that a proper s leads to better NMI than an over-large or over-small s. In sum, setting s properly has both computational and accuracy beneﬁts, which corroborates our theory.

Wang, Gittens, and Mahoney

Table 6: The NMI and elapsed time (seconds) for varying target dimensions s. Here “T” denotes the elapsed time. We use 32 nodes and ﬁx c = 1600.

T ime (s)

3600

3400

3200

3000

2800

2600

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | s = 10 | s = 20 | s = 40 | s = 80 | s = 160 | s = 320 |
| NMI | 0.3852 ± 0.0003 | 0.4069 ± 0.0001 | 0.4130 ± 0.0062 | 0.4233 ± 0.0131 | 0.4086 ± 0.0125 | 0.4099 ± 0.0098 |
| T(Nystro…m) | 2702.2 ± 131.6 | 2634.5 ± 94.4 | 2815.3 ± 403.6 | 2728.1 ± 190.3 | 2849.9 ± 511.2 | 2757.1 ± 271.9 |
| T(DR) | 21.1 ± 3.2 | 31.3 ± 6.6 | 57.5 ± 9.2 | 97.9 ± 20.6 | 168.2 ± 29.0 | 570.5 ± 151.0 |
| T(k-Means) | 84.0 ± 6.4 | 86.9 ± 27.3 | 90.6 ± 9.0 | 88.0 ± 23.9 | 99.8 ± 12.2 | 169.3 ± 64.2 |
| T(Total) | 2840.9 ± 132.0 | 2785.4 ± 83.2 | 3003.4 ± 413.7 | 2952.2 ± 174.5 | 3157.5 ± 549.2 | 3537.5 ± 256.3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Other  K-Means  DR  Nystrom | |  |  |  |  |
|  |  |

10 20 40 80 160 320

Target Dimension s

Figure 8: The mean of the runtimes of Algorithm [3, in seconds, as a function of the target](#bookmark126) dimension s. We use 32 nodes and ﬁx k = 10 and c = 1600.

**7. Conclusion**

We provided principled algorithms for computing approximate kernel k-means clusterings. In particular, we showed that the combination of linear k-means with rank-restricted Nystro…m approximation is theoretically sound, practically useful, and scalable to large data sets. This should be contrasted with approximate spectral clustering using Nystro…m approx- imation. Although the latter is a widely-used approach to scalable non-linear clustering, it has theoretical deﬁciencies and practical limitations. Experiments demonstrated that approximate kernel k-means clustering using the rank-restricted Nystro…m approximation consistently outperforms approximate spectral clustering.

Our analysis uses the concept of projection-cost preservation and builds upon the existing theory of randomized linear algebra. Our main result is a 1 + ∈ approximation ratio guarantee for kernel k-means clustering: when s =  approximation ratio is guaranteed with high probability. As an intermediate theoretical result of independent interest, we introduced a novel rank-restricted Nystro…m approximation and proved that it gives a 1 + ∈ relative-error low-rank approximation guarantee in the trace norm.

Scalable Kernel K-Means Clustering with Nystrm Approximation

To complement our theory, we investigated the performance of the rank-restricted Nystro…m approximation when applied to kernel k-means clustering. We produced an Apache Spark implementation of a distributed version of our approximate kernel k-means algorithm, and applied it to cluster 8.1 million vectors; the results demonstrate the usefulness and simplicity of kernel k-means with rank-restricted Nystro…m approximation for clustering moderately large-scale data sets. Combined with recent work on user-friendly [frameworks for distributed computing systems (Zaharia et al.,](#bookmark128) [2010,](#bookmark128) [2012), large-scale](#bookmark129) [machine learning (Meng et al.,](#bookmark130) [2016), and large-scale randomized linear algebra (Gittens](#bookmark130) [et al.,](#bookmark133) [2016), our results suggest the use of kernel](#bookmark133) k-means with rank-restricted Nystro…m approximations for simple, computationally e伍cient, and theoretically principled clustering of large-scale data sets.

**Acknowledgments**

We thank the authors of [Musco and Musco](#bookmark42) [(2017) for bringing to our attention their results](#bookmark42) on approximate kernel k-means clustering through the use of ridge leverage score Nystro…m approximations, the authors of [Tropp et al.](#bookmark45) [(2017) for informing us of their contemporaneous](#bookmark45) guarantees on the approximation error of rank-restricted Nystro…m approximations, and the anonymous reviewers for their helpful suggestions. We would like to acknowledge ARO, DARPA, NSF, and ONR for providing partial support of this work.

**Appendix A. Proof of Theorem** [**1**](#bookmark67)

In this section, we will provide a proof of Theorem [1, our main quality-of-approximation](#bookmark67) result for rank-restricted Nystro…m approximation. We will start, in Section [A.1, by](#bookmark138)

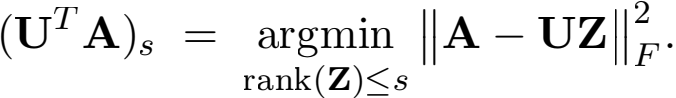
establishing three technical lemmas; then, in Section [A.2, we will establish an important](#bookmark139)

structural result on the rank-restricted Nystro…m approximation as well as key approximate matrix multiplication properties for low-rank matrix approximation; and ﬁnally, in Section [A.3, we will use these results to prove Theorem](#bookmark140) [1.](#bookmark67)

**A.1. Technical Lemmas**

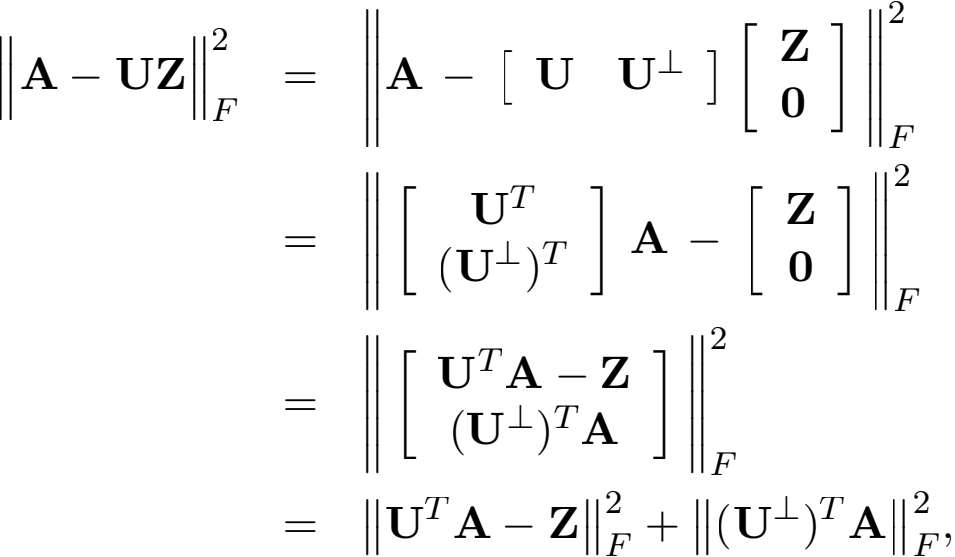
Lemma [5](#bookmark141) is a very well known result. [See (Boutsidis et al.,](#bookmark142) [2014;](#bookmark142) [Zhang,](#bookmark143) [2015)](#bookmark143). Here we ofer a simpliﬁed proof.

**Lemma 5** Let **A** ∈ Rn ×m be any matrix and **U** ∈ Rn ×c have orthonormal columns. Let s be any positive integer no greater than c. Then

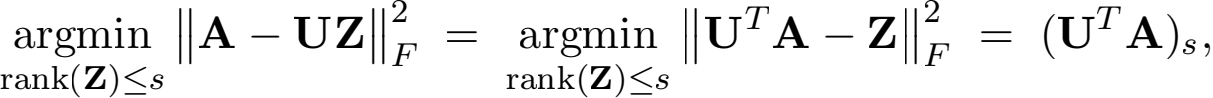


Wang, Gittens, and Mahoney

**Proof** Let **U**丄 ∈ Rn × (n—c) be the orthogonal complement of **U**. It holds that



where the second equality follows from the unitary invariance of the Frobenius norm. Then



by which the lemma follows. 

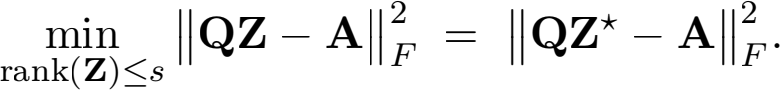
Lemma [6](#bookmark144) is a new result established in this work. The lemma plays an important role in analyzing the rank-restricted Nystro…m method.

**Lemma 6** Let **A** ∈ Rn ×m be any matrix and **U** ∈ Rn ×c has orthonormal columns. Let s ≤ c be arbitrary integer. Let **Q** ∈ Rn ×s be the orthonormal bases of the rank s matrix **U**(**U**T**A**)s ∈ Rn ×m . Then **QQ**T**A** = **U**(**U**T**A**)s .

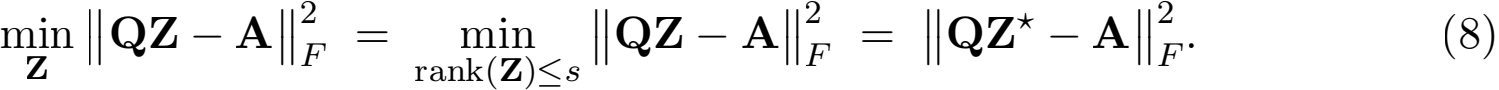
**Proof** Let **Z**\* = **Q**T **U**(**U**T**A**)s ∈ Rs ×m. By the deﬁnition of **Q**, we have

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **QQ**T | **U**(**U**T**A**)s | = **U**(**U**T**A**)s. (7) | | |
| It holds that |  |  |  |  |
| **"U**(**U**T**A**)s — **A "** = min  rank(**X**)≤s | **"UX** — **A "** | 2 F | ≤ min  rank(**Z**)≤s | **"QZ** — **A "** ≤ **"QZ**\* — **A "** , |

where the equality follows from Lemma [5; the former inequality follows from that](#bookmark141) range(**Q**) C range(**U**). Because **U**(**U**T**A**)s = **QZ**\* [by (7), the lefthand and righthand](#bookmark145) sides are equal, and thereby

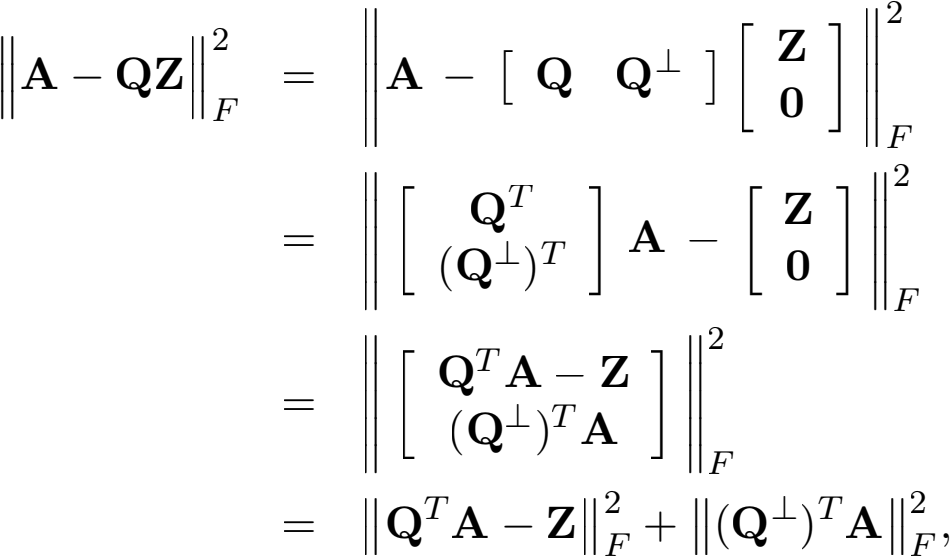


Since **Z** is s × m, it follows that



Scalable Kernel K-Means Clustering with Nystrm Approximation

Let **Q**丄 ∈ Rn × (n—s) be the orthogonal complement of **Q**. It holds that



where the second equality follows from the unitary invariance of the Frobenius norm.

Obviously, **Z** = **Q**T **A** is the unique minimizer of min**Z** Ⅱ**QZ** - **A**Ⅱ; otherwise **Q**T **A** - **Z** in

the righthand side is non-zero, making the objective function increase.

On the one hand, Eqn. [(8) shows that](#bookmark146) **Z**\* = **Q**T **U**(**U**T**A**)s ∈ Rs ×m is one minimizer of min**Z** Ⅱ**QZ** - **A**Ⅱ . On the other hand, we have that **Z** = **Q**T **A** is the unique minimizer of min**Z** Ⅱ**QZ** - **A**Ⅱ . By the uniqueness, we have

**Z**\* = **Q**T **U**(**U**T**A**)s = **Q**T **A**.

It follows that

**QQ**T**U**(**U**T**A**)s = **QQ**T**A**. [The lemma follows from (7) and the above equality.](#bookmark145)

Lemma [7](#bookmark147) is a new result established by this work. Let **A** be any symmetric matrix. Lemma [7](#bookmark147) analyzes the power scheme: using **A**t**P**, instead of **AP**, as a sketch of **A**. The lemma shows that the power scheme leads to an improvement of ( )t—1, where σi is the i-th biggest singular value of **A**. Lemma [7](#bookmark147) with t [= 1 is identical to (Boutsidis et al.,](#bookmark142) [2014,](#bookmark142) Lemma 9). The power scheme has been studied by [Gittens and Mahoney](#bookmark19) [(2016);](#bookmark19) [Halko](#bookmark148) [et al.](#bookmark148) [(2011);](#bookmark148) [Woodruf](#bookmark60) [(2014); their results do not allow the rank restriction rank(**X**)](#bookmark60) ≤ s. To prove Theorem [1, we only need](#bookmark67) t = 1; we will use Lemma [7](#bookmark147) to extend Theorem [1](#bookmark67) to the power method.

**Lemma 7** Let **A** ∈ Rn ×n be any SPSD matrix and **A**s = **V**s**Σ**s**V** be the truncated SVD.

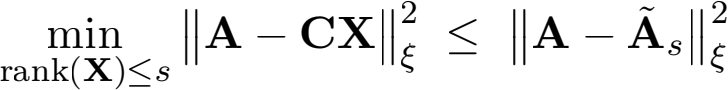
Let **P** ∈ Rn ×c satisfy that **P**T**V**s ∈ Rc ×s has full column rank. Let t ≥ 1 be any integer and **C** = **A**t**P**. Then for ξ = 2 or F,

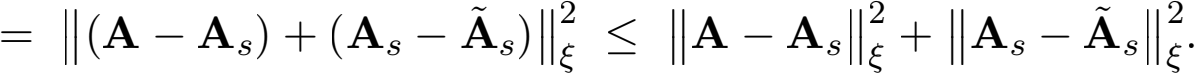
ran)≤s **"A** - **CX "** ≤ **"A** - **A**s **"** +()t—1 **"**(**A** - **A**s)**P**(**VP**)t **"** ,

where σi is the i-th largest singular value of **A**.

**Proof** We co~nstruct the rank s matrix s = **C**(**VP**)t **Σ**—t**V** and use it to facilitate our

proof. Since **A**s has rank s, and its column space is in range(**C**), it holds that





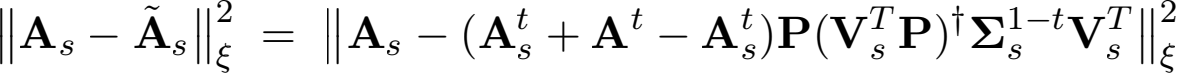
Wang, Gittens, and Mahoney

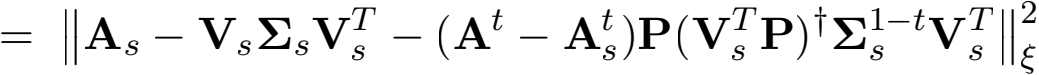
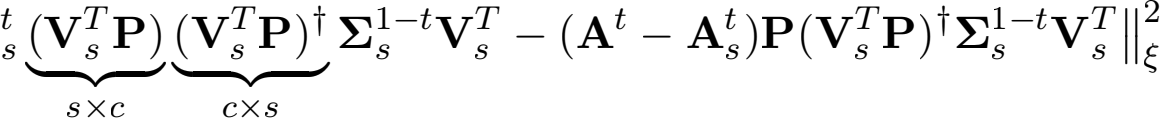
~

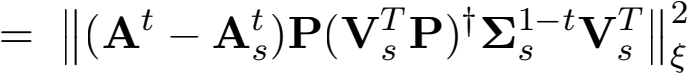
The latter inequality follows from that the row space of **A**s - **A**s is in range(**V**s), that

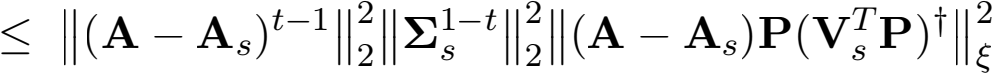
(**A**-**A**s)**V**s = **0**, and the matrix Pythagorean theorem. By the assumption rank(**VP**) = s,

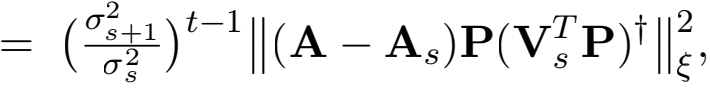
it holds that (**VP**)(**VP**)t = **I**s. We have



**A**s - **V**s**Σ**







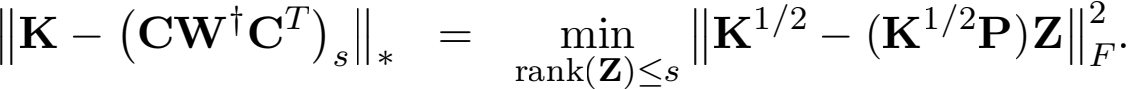
by which the lemma follows. 

**A.2. Structural Result and Approximate Matrix Multiplication Properties**

Lemma [8](#bookmark149) is a structural result of independent interest that is important for establishing Theorem [1.](#bookmark67) By structural result, we mean that it is a linear algebraic result that holds for any matrix **P**, and in particular for any matrix **P** that is a randomized sketching matrix. In particular, given this structural result, randomness in an algorithm enters only via **P**. The importance of establishing such structural results within randomized linear algebra has [been highlighted previously (Drineas et al.,](#bookmark61) [2011;](#bookmark61) [Mahoney,](#bookmark58) [2011;](#bookmark58) [Gittens and Mahoney,](#bookmark19) [2016;](#bookmark19) [Mahoney and Drineas,](#bookmark150) [2016;](#bookmark150) [Wang et al.,](#bookmark151) [2017)](#bookmark151).

**Lemma 8** Let **K** ∈ Rn ×n be any SPSD matrix, **P** ∈ Rn ×c be any matrix, and s ≤ c be any positive integer. Let **D** = **K**1/2 **P** ∈ Rn ×c , P = rank(**D**) ≤ c, and **U** ∈ Rn ×ρ be the left singular vectors of **D**. Let **Q** ∈ Rn ×s be any orthonormal bases of **U**(**U**T **K**1/2)s . It holds that

(**CW**t**C**T)s = **K**1/2**QQ**T**K**1/2 ,



**Proof** Let **D** = **K**1/2 **P** = **UΣV**T ∈ Rn ×c be the SVD of **D**. It holds that **C** = **KP** = **K**1/2 **D** and **W** = **P**T **KP** = **D**T**D**. It follows the SVD of **D** that

**C**(**W**t)1/2 = **K**1/2 **D**((**D**T**D**)t)1/2 = **K**1/2(**UΣV**T) (**VΣ**—2**V**T)1/2 = **K**1/2 **UV**T .

We need the following lemma to prove Lemma [8.](#bookmark149) The following lemma is not hard to prove.

**Lemma 9** Let the integers n,c,P,s satisfy n ≥ c ≥ P ≥ s. Let **A** ∈ Rn ×ρ be any matrix with rank at least s and **V** ∈ Rc ×ρ has orthonormal columns. Then (**AV**T)s = **A**s**V**T .

Scalable Kernel K-Means Clustering with Nystrm Approximation

If rank(**K**1/2 **U**) < s, then obviously **K**1/2 **UV**T = (**K**1/2 **UV**T)s = (**K**1/2 **U**)s**V**T . We thus assume rank(**K**1/2 **UV**T) ≥ s and can apply the above lemma to show (**K**1/2 **UV**T)s = (**K**1/2 **U**)s**V**T . In either case, we have

(**K**1/2 **UV**T)s = (**K**1/2 **U**)s**V**T . It follows from (9) that (**C**(**W**t )1/2)s= (**K**1/2 **U**)s**V**T . Thus

(**CW**t**C**T)s = (**K**1/2 **U**)s(**U**T **K**1/2)s = (**K**1/2 **U**)s**U**T**U**(**U**T **K**1/2)s

Let **Q** ∈ Rn ×s be the orthonormal bases of the rank s matrix **U**(**U**T **K**1/2)s. Lemma [6](#bookmark144) ensures that

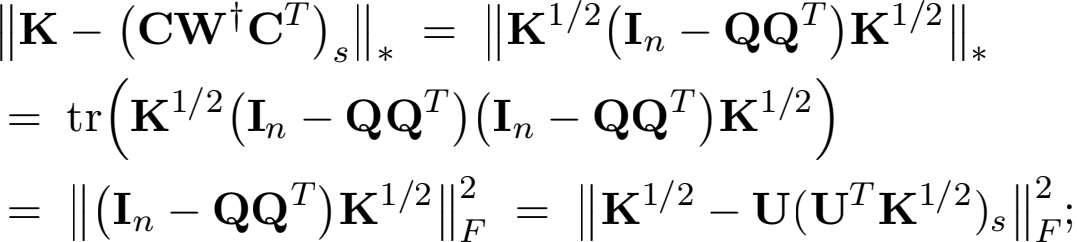
**QQ**T**K**1/2 = **U**(**U**T **K**1/2)s.

It follows that

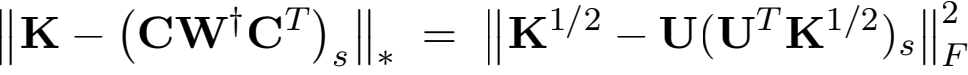
(**CW**t**C**T)s = **K**1/2**QQ**T**K**1/2 ; (9)

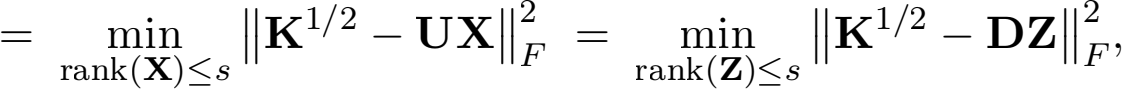
by which the former conclusion of the lemma follows.

It is well known that Ⅱ**Y**Ⅱ\* = tr(**Y**) for any SPSD matrix **Y** and that tr(**A**T **A**) = Ⅱ**A**Ⅱ for any matrix **A**. [It follows from (9) that](#bookmark152)



here the second equality holds because **K**1/2(**I**n - **QQ**T )**K**1/2 is SPSD and (**I**n - **QQ**T ) is orthogonal projection matrix; the last equality follows from Lemma [6.](#bookmark144) It follows from Lemma [5](#bookmark141) that





by which the latter conclusion of the lemma follows. 

Lemma [10](#bookmark47) formally deﬁnes the key approximate matrix multiplication properties (subspace embedding property and approximate orthogonality property) which—when sketching dimensions are chosen appropriately (see Table [7)—are shared by the uniform](#bookmark153) sampling, leverage score sampling, Gaussian projection, SRHT, and CountSketch sketching methods. Establishing approximate matrix multiplication bounds is the key step in proving [many approximate regression and low-rank approximation results (Mahoney,](#bookmark58) [2011), and](#bookmark58) this lemma will be used crucually in the proof of Theorem [1.](#bookmark67)

**Lemma 10 (Key Approximate Matrix Multiplication Properties for a Sketch)** Let η; ∈; δ1 ; δ2 ∈ (0; 1) be ﬁxed parameters. Fix **Y** ∈ Rn ×d and let **V** ∈ Rn ×s have orthonormal

Wang, Gittens, and Mahoney

columns. Let **P** ∈ Rn ×c be one of the sketching methods listed in Table [7.](#bookmark153) When c is larger than the quantity in the middle column of the corresponding row of Table [7, the subspace](#bookmark153) embedding property

**ⅡV**T **PP**T**V** - **I**s**Ⅱ**2 ≤ η

holds with probability at least 1 - δ1 . When c is larger than the quantity in the right-hand column of the corresponding row of Table [7, the matrix multiplication property](#bookmark153)

**ⅡV**T **PP**T**Y** - **V**T**YⅡ** ≤ ∈Ⅱ**Y**Ⅱ

holds with probability at least 1 - δ2 .

Table 7: Su伍cient sketching dimensions for the sketching properties of Lemma [10](#bookmark47) to hold. The leverage score sampling referred to here is with respect to the leverage scores of the matrix **V** described in Lemma [10.](#bookmark47) Similarly, μ is the row coherence of **V**.

|  |  |  |
| --- | --- | --- |
| **Sketching** | **Subspace Embedding** | **Matrix Multiplication** |
| Leverage Score Sampling  Uniform Sampling  SRHT  Gaussian Projection  CountSketch | log  log   log  s+log(1/δ1 )  η2 s2 δ1 η2 | s  ∈δ2  μs  ∈δ2  s+log n  ∈δ2  s  ∈δ2 s ∈δ2 |

**Proof** [This lemma is a reproduction of (Wang et al.,](#bookmark32) [2016b, Lemma 2), and is a collation](#bookmark32) of heterogenously stated results from the literature. The random sampling estimates were established by [Drineas et al.](#bookmark154) [(2008);](#bookmark154) [Wang et al.](#bookmark155) [(2016a);](#bookmark155) [Woodruf](#bookmark60) [(2014)](#bookmark60). The Gaussian projection was ﬁrstly analyzed by [Johnson and Lindenstrauss](#bookmark156) [(1984); see](#bookmark156)[Woodruf](#bookmark60) [(2014) for a proof of the stated su伍cient sketching dimensions](#bookmark60). The SRHT estimates are from [Drineas et al.](#bookmark61) [(2011);](#bookmark61) [Lu et al.](#bookmark62) [(2013);](#bookmark62) [Tropp](#bookmark63) [(2011)](#bookmark63). The CountSketch estimates are from [Meng and Mahoney](#bookmark65) [(2013);](#bookmark65) [Nelson and Nguyˆen](#bookmark66) [(2013)](#bookmark66). 

**A.3. Completing the Proof of Theorem** [**1**](#bookmark67)

**Proof** Let **K**s = **V**s**Λ**s**V** ∈ Rn ×n be the truncated SVD of **K**. It follows from Lemma [10](#bookmark47)

and the sketch sizes deﬁned in Table [2](#bookmark73) that

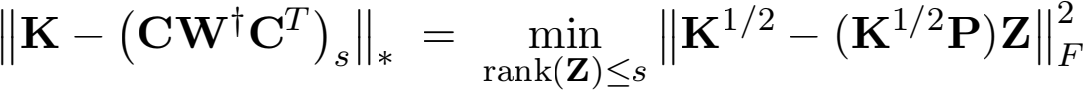
**ⅡVPP**T **V**s - **I**s**Ⅱ**2 ≤ η;

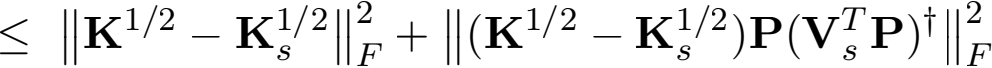
**Ⅱ**(**K**1/2 - **K**/2)**PP**T **V**s **Ⅱ** ≤ ∈ **ⅡK**1/2 - **K**/2Ⅱ

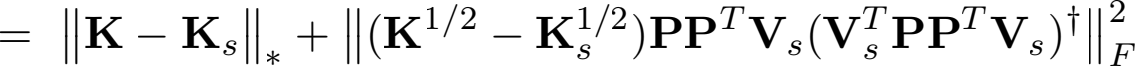
hold simultaneously with probability at least 0.9.

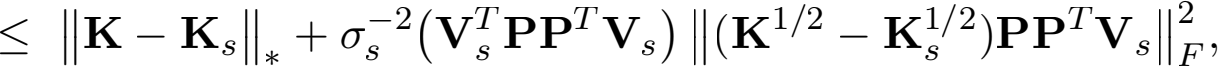
Scalable Kernel K-Means Clustering with Nystrm Approximation

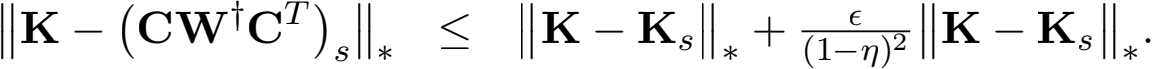
It follows from Lemma [8](#bookmark149) that







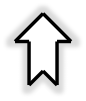


where the former inequality follows from Lemma [7](#bookmark147) (with t = 1). It follows that 

We let η be a constant and obtain the former claim of Theorem [1.](#bookmark67) The latter claim of Theorem [1](#bookmark67) follows from Lemma [8.](#bookmark149) 

**Appendix B. Proof of Theorem** [**2**](#bookmark25)

In this section, we will provide a proof of Theorem [2, our main result for kernel](#bookmark25) k-means approximation. We will start, in Sections [B.1,](#bookmark157) [B.2, and](#bookmark158) [B.3, by presenting several technical](#bookmark15) tools of independent interest (Lemmas [11,](#bookmark159) [12, and](#bookmark160) [13, respectively)](#bookmark161). Then, in Section [B.4,](#bookmark162) we will use use these tools to prove Theorem [2.](#bookmark25)

The proof of Theorem [2](#bookmark25) will proceed by combining Theorem [1](#bookmark67) with Lemmas [11,](#bookmark159) [12,](#bookmark160) and [13.](#bookmark161) For convenience, the structure of proof is given in Figure [9.](#bookmark163)

|  |
| --- |
| **Lemma 12**  projection-cost preservation for linear k-means |

|  |
| --- |
| **Lemma 13**  kernel trick  for kernel  k-means |

|  |
| --- |
| **Lemma 11**  projection-cost  preservation  via low-rank  approximation |

|  |
| --- |
| **Theorem 2**  main result |

|  |
| --- |
| **Theorem 1**  relative-error  rank-restricted  Nystrom  approximation |

Figure 9: The structure of the proof of Theorem [2.](#bookmark25)

We use the notation of orthogonal projection matrix throughout this section. A matrix **M** ∈ Rn ×n is an orthogonal projection matrix if and only if there exists a matrix **Q** with orthonormal columns such that **M** = **QQ**T . It follows that **M**T = **M**, **MM** = **M**, **M** 二 **I**n and that **I**n - **M** is also an orthogonal projection matrix.

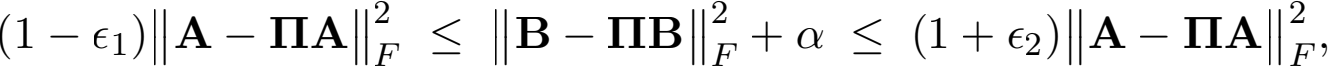
**B.1. Projection-Cost Preservation via Low-Rank Approximation**

We ﬁrst deﬁne projection-cost preservation, then show that certain low-rank approximations are projection-cost preserving. Projection-cost preservation was named and systematized

Wang, Gittens, and Mahoney

by [Cohen et al.](#bookmark36) [(2015), but the idea has been used in earlier works (Boutsidis et al.,](#bookmark36) [2010,](#bookmark39) [2015;](#bookmark35) [Feldman et al.,](#bookmark37) [2013;](#bookmark37) [Liang et al.,](#bookmark165) [2014)](#bookmark165).

**Deﬁnition 2 (Rank** k **Projection-Cost Preservation)** Fix a matrix **A** ∈ Rn ×d and error parameters ∈ 1 , ∈2 > 0. The matrix **B** ∈ Rn ×d has the rank k projection-cost preserving property with respect to **A** if for any rank k orthogonal projection matrices **Π** ∈ Rn ×n ,



for some ﬁxed non-negative Q that can depend on **A** and **B** but is independent of **Π** .

The next lemma provides a˜ way to construct a projection-cost preserv˜ in˜g sketch of **A**. It

states that if a rank-s matrix **A**s in the span of **A** is con˜structed so that **A**s**A** approximates

**AA**T su伍ciently well in terms of the trace norm, then **A**s has the projection-cost preserving property. Observe that this lemma explicitly relates projection-cost preserving sketches to [approximate matrix multiplication (Drineas et al.,](#bookmark168) [2006;](#bookmark168) [Mahoney,](#bookmark58) [2011), where the latter](#bookmark58) [is measures with respect to the trace norm (Gittens and Mahoney,](#bookmark19) [2016)](#bookmark19). We provide a proof of Lemma [11](#bookmark159) in Appendix [D.](#bookmark169)

**Lemma 11 (Projection-Cost Preservation via Low-Rank Appr**˜**oximation)** Fix an

error parameter ∈ ∈ (0, 1). Let **A** ∈ Rn ×d and choose a rank-s matrix **A**s that satisﬁes the following conditions:

(i) there is an orthogonal projection matrix **M** such that s = **AMA**T, and

(ii) Ⅱ**AA**T — **A**s**A**Ⅱ\* ≤ (1 + ∈)**"AA**T — **A**s**A"**\* .

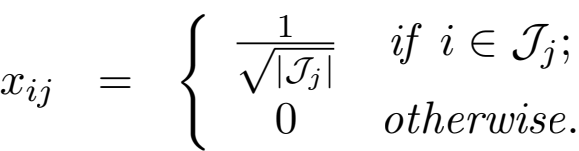
Then there exists a ﬁxed Q ≥ 0 such that, for any rank k projection matrices **Π** ∈ Rn ×n ,

**"**(**I**n — **Π**)**A"** ≤ **"**(**I**n — **Π**)s **"** + Q ≤ (1 + ∈ + **"**(**I**n — **Π**)**A"** .

**B.2. Linear** k**-Means Clustering and Projection-Cost Preservation**

Linear [k-means clustering is formally deﬁned by the optimization problem (1)](#bookmark9). To relate the linear k-means clustering problem to the projection-cost preservation property, we [use an equivalent formulation of (1) as an optimization over the set of rank-k](#bookmark9) projection matrices of the form **XX**T , where **X** is a cluster indicator matrix. This approach was adopted by [Boutsidis et al.](#bookmark38) [(2009,](#bookmark38) [2010,](#bookmark39) [2015);](#bookmark35) [Cohen et al.](#bookmark36) [(2015);](#bookmark36) [Ding et al.](#bookmark100) [(2005)](#bookmark100). Following them, we deﬁne the cluster indicator matrix in the following and give an example in Figure [10.](#bookmark167)

**Deﬁnition 3 (Cluster Indicator Matrices)** Let n and k be given, and let J1 , . . . , Jk be a k-partition of the set [n]. The cluster indicator matrix corresponding to J1 , . . . , Jk is the n × k matrix **X** deﬁned by



We take Xn,k to be the collection of cluster indicator matrices corresponding to all the k- partitions of [n].

Scalable Kernel K-Means Clustering with Nystrm Approximation

XT =

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1    4 | 0 | 1 | 0 | 1    4 | 0 | 1    4 | 0 | 0 | 0 | 0 | 0 |
| 4 |
| 0 | 0 | 0 | 1    3 | 0 | 0 | 0 | 0 | 1    3 | 0 | 1    3 | 0 |
| 0 | 1 V 5 | 0 | 0 | 0 | 1 V 5 | 0 | 1 V 5 | 0 | 1 V 5 | 0 | 1 V 5 |

*J1*={1, 3, 5, 7}

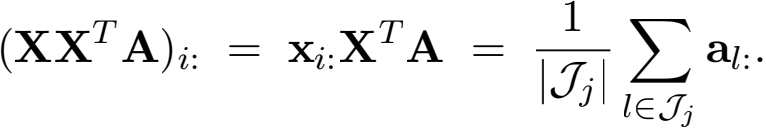
*J2*={4, 9, 11}

*J3*={2, 6, 8, 10, 12}

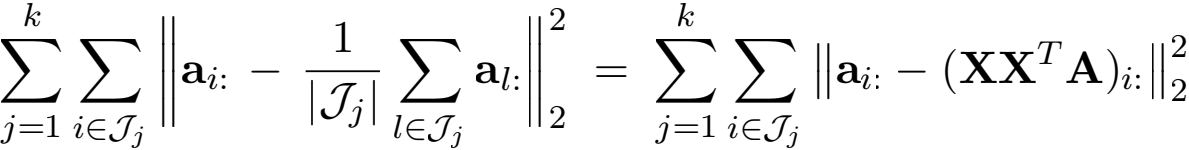
Figure 10: An example of a cluster indicator matrix **X** ∈ Xn,k . In this example, n = 12, k = 3, and **X** corresponds to the indicated 3-partition J1 , J2 , J3 .

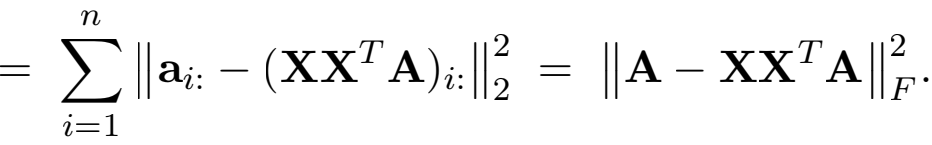
A cluster indicator matrix **X** has exactly one non-zero entry in each row and has orthonormal columns. That is, because of the normalization of the non-zero entries,a cluster indicator matrix **X** is an orthogonal matrix, but one with the additional constraint that there is only one non-zero entry in each row. (Indeed, the insight of [Boutsidis et al.](#bookmark38) [(2009,](#bookmark38) [2010,](#bookmark39) [2015) was that the rank-constrained optimization over all orthonormal matrices,](#bookmark35) which provides the usual PCA/SVD-based low-rank approximation, is a relaxation of the optimization over this smaller set of orthonormal matrices.) Among other things, it follows that **XX**T is a rank-k projection matrix.

Given this, assume i ∈ Jj; it can be veriﬁed that the i-th row of **XX**T**A** ∈ Rn ×d is the centroid of Jj:

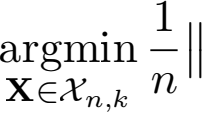


[Consider the objective function of (1)](#bookmark9). When **X** is the cluster indicator matrix corresponding to the input k-partition, this objective can be rewritten in terms of **XX**T as





Therefore, the linear [k-means clustering problem (1) is equivalent to the optimization](#bookmark9) problem

**A** - **XX**T**A**(10)

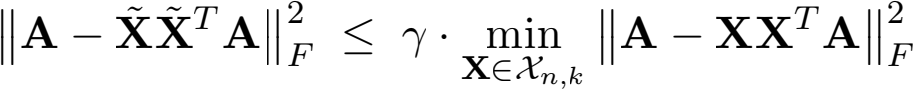
In the remainder of this section, we use this equivalence between k-partitionings of [n] and cluster indicator matrices without comment; e.g., we refer to the outputs of linear k-means clustering algorithms as cluster indicator matrices. Because the linear k-means clustering [problem (10) is NP-hard, in practice it is solved by variants of Lloyd’s algorithm](#bookmark170).

Deﬁnition [4](#bookmark171)deﬁnes an appropriate notion of approximation for linear k-means clustering algorithms. Deﬁnition [4](#bookmark171) is an equivalent statement of Deﬁnition [1.](#bookmark85)

Wang, Gittens, and Mahoney

**Deﬁnition 4 (**√**-Approximate Algorithms)** We call an algorithm A√ (√ ~≥ 1) a √ -

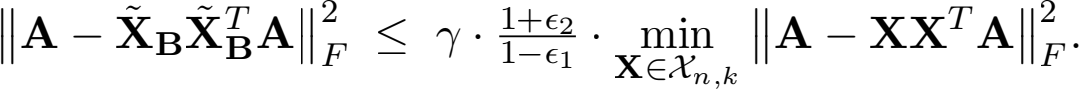
approximate linear k-means algorithm if it produces a cluster indicator matrix **X** such that



for any conformal matrix **A**.

Given this, we now state Lemma [12, which relates projection-cost preservation to linear](#bookmark160) k-means clustering. Let **A** ∈ Rn ×d be the input matrix and **B** ∈ Rn ×s be a smaller matrix (s < d) satisfying the projection-cost preservation property. The projection-cost preserving property ensures that the k-partition obtained by applying linear k-means clustering to the rows of **B**, instead of **A**, is agood k-partition for the rows of **A**. This lemma was established by [Cohen et al.](#bookmark36) [(2015)](#bookmark36). To make this paper self-contained, we provide a proof of Lemma [12](#bookmark160) in Appendix [E.](#bookmark172)

**Lemma 12 (Projection-Cost Preservation for Linear** k**-Means)** Fix **A** ∈ Rn ×d and assume **B** ∈ Rn ×s satisﬁes the rank k projection-cost preservation property in Deﬁnition [2](#bookmark164) wit~h probability at least 1 — δ . Let the error parameters ∈ 1 , ∈2 ∈ (0, 1) be as in Deﬁnition [2](#bookmark164). If **XB** is the result of applying a √ -approximate linear k-means clustering algorithm to the rows of **B**, then with probability at least 1 — 2δ ,



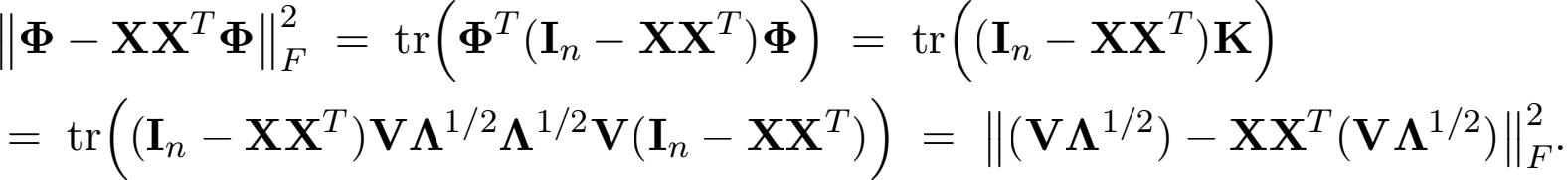
**B.3. Kernel** k**-Means Clustering**

Given input vectors **a**1 , . . . , **a**n, the kernel k-means clustering algorithm applies lin- ear k-means clustering to feature vectors **φ**(**a**1 ), . . . , **φ**(**a**n). For convenience, we let **φ**(**a**1 )T , . . . , **φ**(**a**n)T constitute the rows of the matrix **Φ** . Lemma [13](#bookmark161) argues that all the information in **Φ** relevant to the kernel k-means clustering problem is present in the kernel matrix **K** = **ΦΦ**T . [Variants of this lemma are well-known (Scho…lkopf and Smola,](#bookmark173) [2002)](#bookmark173).

**Lemma 13 (Kernel Trick for Kernel** k**-Means)** Let **Φ** be a matrix with n rows and let **K** = **ΦΦ**T ∈ Rn ×n . Let **K** = **VΛV**T be the EVD of **K**. Then for any matrix **X** with orthonormal columns,

**"Φ** — **XX**T **Φ"** = **"**(**VΛ**1/2) — **XX**T(**VΛ**1/2) **"** .

**Proof** Because **X** has orthonormal columns, **XX**T is an orthogonal projection matrix, and (**I**n — **XX**T) is the orthogonal projection onto the complementary space. It follows that, as claimed,



The equalities are justiﬁed by, respectively, the deﬁnition of the squared Frobenius normand the idempotence of projections; the cyclicity of the trace and the fact that **K** = **ΦΦ**T; the cylicity of the trace and the EVD of **K**; and the deﬁnition of the squared Frobenius norm. 

Scalable Kernel K-Means Clustering with Nystrm Approximation

**B.4. Completing the Proof of Theorem** [**2**](#bookmark25)

To complete the proof of Theorem [2, we ﬁrst establish Lemma](#bookmark25) [14, and then use it to prove](#bookmark162) Theorem [15, which is equivalent to Theorem](#bookmark174) [2.](#bookmark25) Let **Φ** be a matrix with n rows, and **K** = **ΦΦ**T ∈ Rn ×n be the corresponding kernel matrix.

**Lemma 14** Assume that **B** ∈ Rn ×s satisﬁes

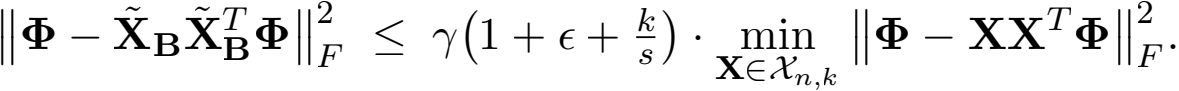
tr (**K** − **BB**T) ≤ (1 + ∈)**"K** − **K**s**"**\* (11)

and that there is an orthogonal projection matrix **M** such that

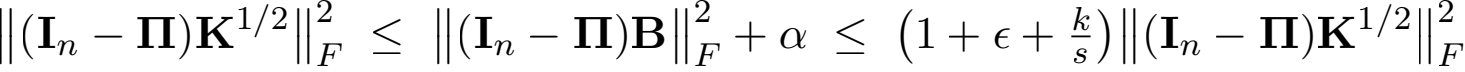
**BB**T = **K**1/2 **MK**1/2 . (12)

Use the rows of **B** as input to a √ -approximation algorithm (see Deﬁnition [4) and let](#bookmark171) **B**

be the output cluster indicator matrix (see Deﬁnition [3), then](#bookmark166)



**Proof** [Because assumptions (11) and (12) are satisﬁed, Lemma](#bookmark175) [11](#bookmark159)ensures that **B** is a rank k projection-cost preserving sketch of **K**. That is, there exists a constant Q ≥ 0 independent of **Π** such that

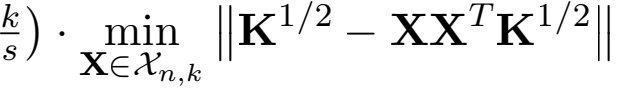


holds for any rank-k orthogonal projection matrix **Π** . It follows by Lemma [12](#bookmark160) that **B**

gives an almost optimal linear k-means clustering over the rows of **K**1/2:

**"K**1/2 − **B K**1/2**"**

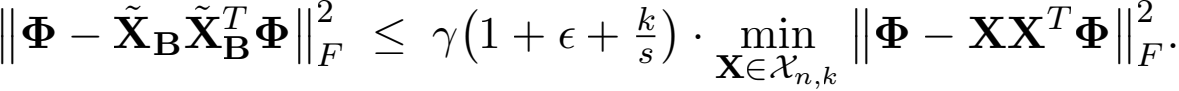
2 F

≤ √ (1 + ∈ + 

2

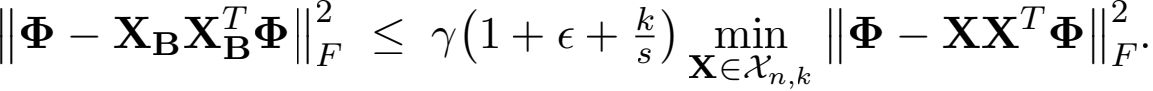
F .

Finally, Lemma [13](#bookmark161) states that clustering the rows of **K**1/2 is equivalent to clustering the rows of **Φ**, so we reach the desired conclusion that



Finally, we state and prove Theorem [15.](#bookmark174) Theorem [15](#bookmark174) shows that approximate kernel k-means clustering using the Nystro…m method exhibits a 1 + ∈ +  approximation ratio. Observe that Theorem [15](#bookmark174)is equivalent to Theorem [2, and thus establishing it also establishes](#bookmark25) Theorem [2.](#bookmark25) Recall the Nystro…m approximation is **K** ≈ **CW**t**C**T, where **C** = **KP**, **W** = **P**T **KP**, and **P** is a sketching matrix.

**Theorem 15** Choose a sketching sketching matrix **P** ∈ Rn ×c and sketch size c consistent with Table [2](#bookmark73). Let **B** ∈ Rn ×s be any matrix satisfying **BB**T = (**CW**t**C**T)s . Let the cluster indicator matrix **XB** be the output of any √ -approximate k-means clustering algorithm applied to the rows of **B**. With probability at least 0.9,



Wang, Gittens, and Mahoney

**Proof** Theorem [1](#bookmark67) [shows that (11) holds with probability at least 0](#bookmark175).9. Theorem [1](#bookmark67) also shows that **BB**T = **K**1/2**QQ**T**K**1/2 where **Q** is a n × s matrix with orthonormal columns, [so (12) holds surely.](#bookmark176) The theorem now follows from Lemma [14.](#bookmark162) 

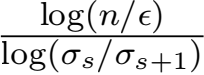
**Appendix C. Proof of Theorem** [**4**](#bookmark103)

Section [C.1](#bookmark177) analyzes rank-restricted Nysto…m with power method. Section [C.2](#bookmark178)completes the proof of Theorem [4.](#bookmark103) Here we re-state the algorithm of Theorem [4: ﬁrst, set any](#bookmark103) c ≥ s and draw a Gaussian projection matrix **P** ∈ Rn ×c ; second, run the power iteration **P** ←— **KP** for t times; third, orthogonalize **P** ∈ Rn ×c to obtain **U** ∈ Rn ×c ; fourth, compute **C** = **KU** and **W** = **U**T **KU**; last, ﬁnd any **B** ∈ Rn ×s satisfying **BB**T = (**CW**t**C**T)s.

**C.1. Rank-Restricted Nystr”om with Power Method**

Theorem [16](#bookmark177) analyzes the rank-restricted Nystro…m with power method. The theorem will be used to prove Theorem [4.](#bookmark103)

**Theorem 16** Let **K** ∈ Rn ×n be an SPSD matrix, s be the target rank, c (≥ s) be arbitrary, and ∈ ∈ (0, 1) be an approximation parameter. Let **C** ∈ Rn ×c and **W** ∈ Rc ×c be computed by

the above algorithm with t = O() power iterations, where σi is the i-th singular value of **K**. If c = s + O(log ), then

**"K** — (**CW**t**C**T)s **"**\* ≤ (1 + ∈) **"K** — **K**s**"**\*

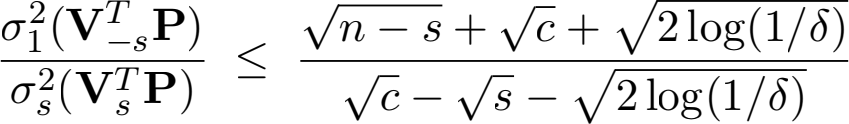
with probability at least 1 — δ . If c = s, then this bound holds with a constant probability (that depends on s) .

**Proof** Let **V** — s ∈ Rn × (n—s) be the orthogonal complement of **V**s. Let **P** be an n × c

standard Gaussian matrix. Then **V**T— s**P** is (n — s) × c standard Gaussian matrix, and **VP**

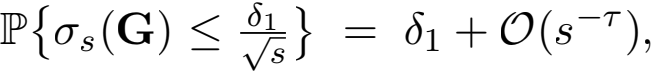
is s × c standard Gaussian matrix.

It is well known that the largest singular value of an (n — s) × c standard Gaussian matrix is at most √n — s + √c + ζ with probability 1 — exp(—ζ2 /2). [See (Vershynin,](#bookmark179) [2010)](#bookmark179). Consider c > s. The least singular value of an s × c standard Gaussian matrix is at least √c — √s — ζ with probability 1 — exp(—ζ2 /2). Combining the above results, we have that

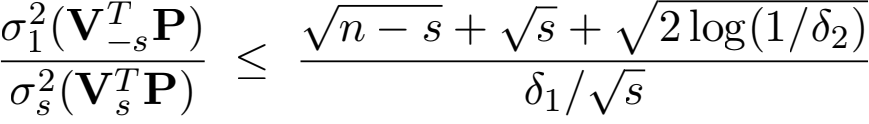


holds with probability 1 — 2δ .

Consider c = s. [Rudelson and Vershynin](#bookmark180) [(2008);](#bookmark180) [Tao and Vu](#bookmark105) [(2010) showed that the](#bookmark105) least singular value of an s × s standard Gaussian matrix **G** satisfy



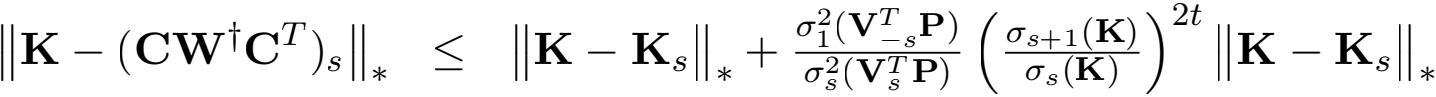
where τ > 0 is a constant. It follows that

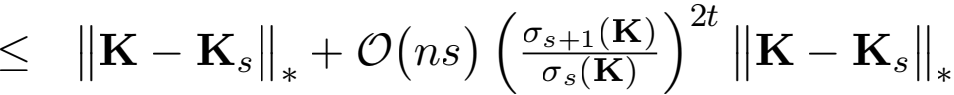


Scalable Kernel K-Means Clustering with Nystrm Approximation

holds with probability 1 — δ1 — δ2 — O(s—τ ).

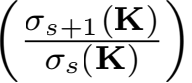
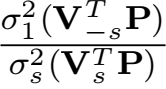
For either c > s or c = s, it follows from Lemma [17, which will be proved subsequently,](#bookmark181) that



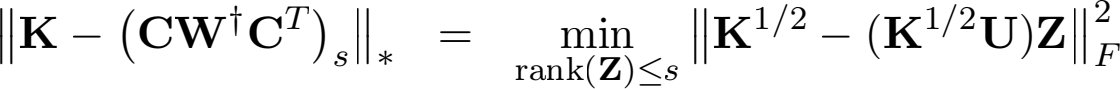


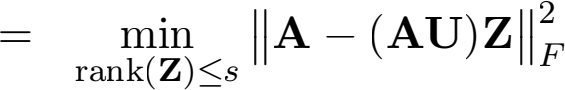
holds with constant probability, by which the theorem follows. 

**Lemma 17** Let **K** ∈ Rn ×n be any ﬁxed SPSD matrix, s and t be any positive integer, and **V**s ∈ Rn ×s be the top s singular vectors of **K**. Let **P** ∈ Rn ×c satisfy that **P**T**V**s ∈ Rc ×s has full column rank. Let **U** ∈ Rn ×c be the orthonormal bases of **K**t**P**. Let **C** = **KU** and **W** = **U**T **KU**. Then

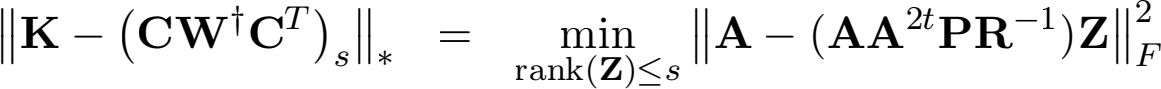
**"K** — (**CW**t**C**T)s **"**\* ≤ **"K** — **K**s **"**\* + 2t  **"K** — **K**s **"**\* .

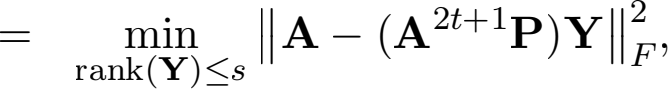
Here **V** — s ∈ Rn × (n—s) is the orthogonal complement of **V**s . **Proof** It follows from Lemma [8](#bookmark149) that



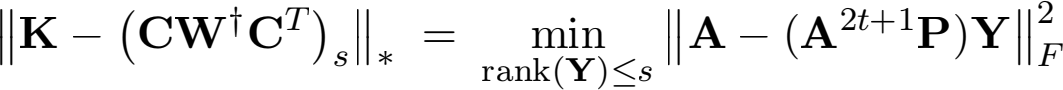


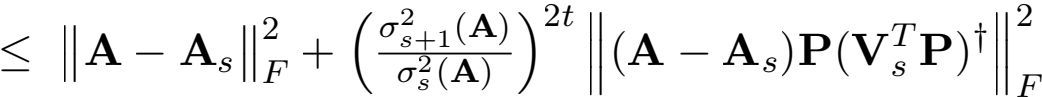
where we let **A** = **K**1/2 . By deﬁnition, **U** contains the orthonormal bases of **K**t**P** = **A**2t**P**, and thus there exists an s × s non-singular matrix **R** such that **A**2t**P** = **UR**. It follows that

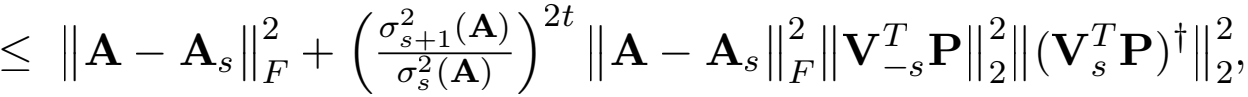




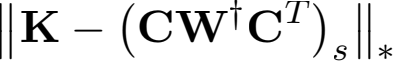
where the latter equality follows from that the column spaces of **A**2t+1 **P** and **A**2t+1 **PR**—1 are the same. It follows from Lemma [7](#bookmark147) that

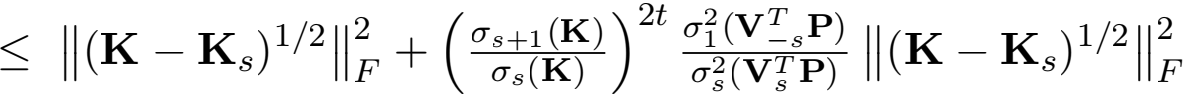


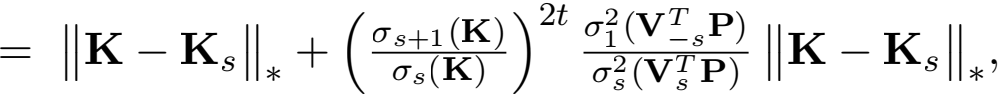




where we deﬁne **V** — s ∈ Rn × (n—s) as the orthogonal complement of **V**s. By our deﬁnition **A** = **K**1/2, it follows that







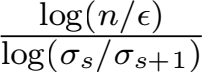
Wang, Gittens, and Mahoney

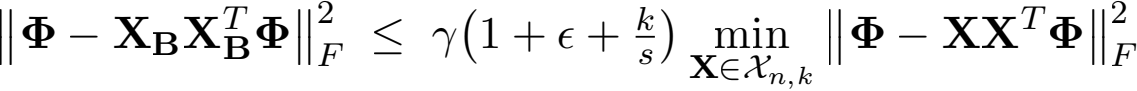
by which the lemma follows. 

**C.2. Completing the Proof of Theorem** [**4**](#bookmark103)

Finally, we state and prove Theorem [18.](#bookmark178) Observe that Theorem [18](#bookmark178) is equivalent to Theorem [4, and thus establishing it also establishes Theorem](#bookmark103) [4.](#bookmark103)

**Theorem 18** Let c, t, and **B** ∈ Rn ×s be deﬁned in the begining of this section. Set t =

O(). Let the cluster indicator matrix **XB** be the output of any √ -approximate k-means clustering algorithm applied to the rows of **B**. If c = s + O(log ), then



holds with probability at least 1 — δ . If c = s, then the above inequality holds with probability 0.9 — O(s—τ ), where τ is a constant.

**Proof** Since **BB**T = (**CW**t**C**T)s by deﬁnition, Theorem [16](#bookmark177) [ensures that (11) holds with](#bookmark175) some constant probability. Theorem [1](#bookmark67) shows that **BB**T = **K**1/2**QQ**T**K**1/2 where **Q** is a n × s [matrix with orthonormal columns, so (12) holds surely.](#bookmark176) The theorem now follows from Lemma [14.](#bookmark162) 

**Appendix D. Proof of Lemma** [**11**](#bookmark159)

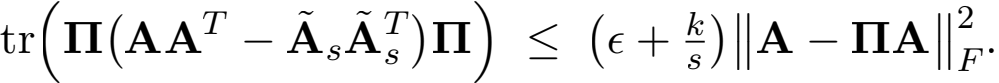
To prove Lemma [11, we ﬁrst establish a key lemma, Lemma](#bookmark159) [19.](#bookmark183) This lemma will make use of the following two assumptions.

**Assumption 1** Assume that s satisﬁes tr (**AA**T — s ) ≤ (1 + ∈)**"AA**T — **A**s**A"**\* .

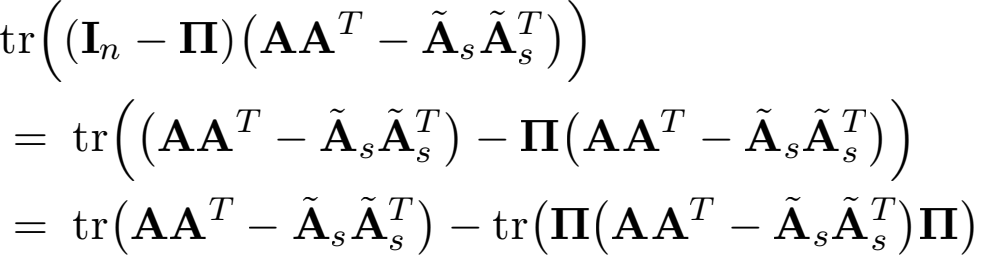
**Assumption 2** Assume there exists such an orthogonal projection matrix **M** that s =

**AMA**T .

**Lemma 19** Let~ **A** ∈ Rn ×d be any ﬁxed matrix. Fix an error parameter ∈ ∈ (0, 1). Let the rank s matrix **A**s satisfy Assumptions [1](#bookmark182) and [2](#bookmark184). Then for any rank k orthogonal projection matrix **Π** ∈ Rn ×n ,



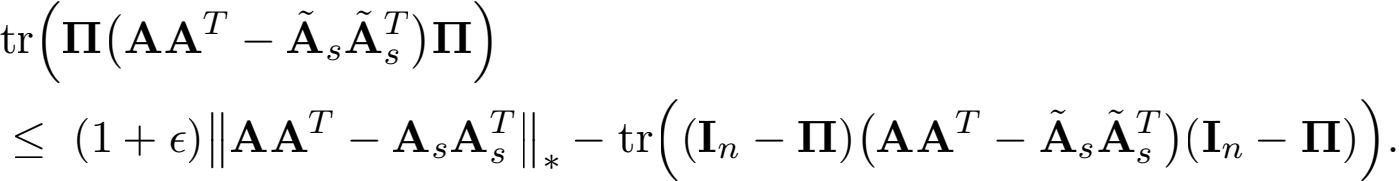
**Proof** It holds that





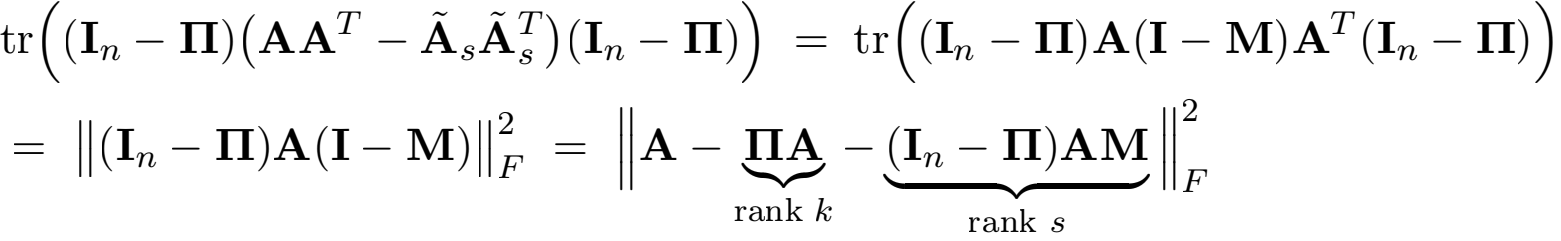
Scalable Kernel K-Means Clustering with Nystrm Approximation

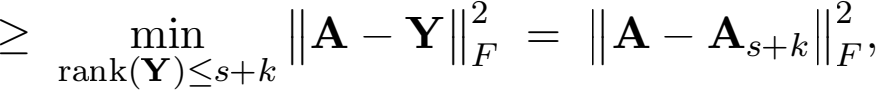
where the inequality follows from Assumption [1.](#bookmark182) It can be equivalently written as

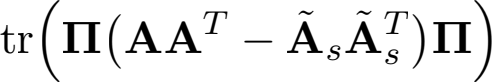


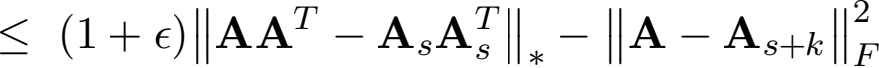
Let **M** be a˜no˜rthogonal projection matrix deﬁned in Assumption [2; by this assumption, it](#bookmark184)

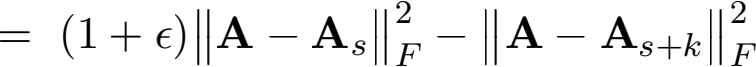
holds that **A**s**A** = **AMA**T and rank(**MA**) = s. It follows that

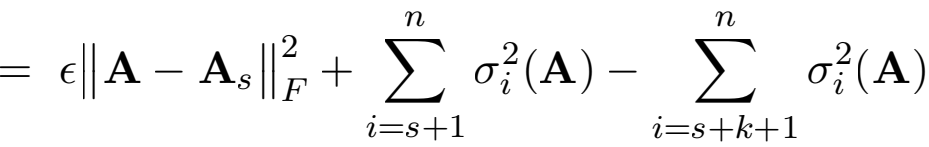


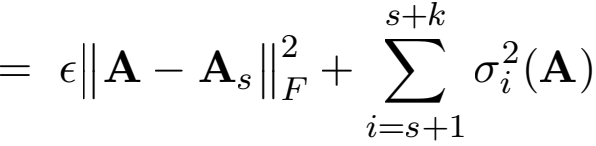


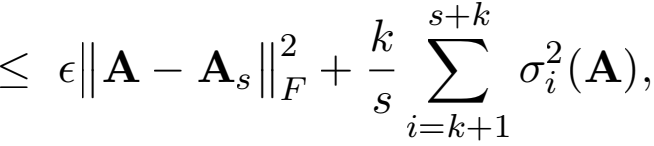
where the inequality follows from that matrix rank is subadditive function. It follows that 











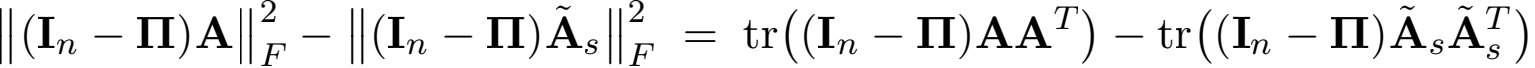
where the last inequality follows by the decrease of the singular values. Finally, we obtain that

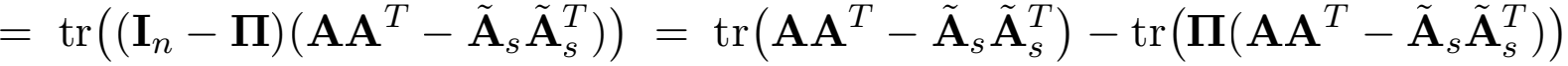
tr (**Π** (**AA**T - s )**Π**) ≤ ∈ **"A** - **A**k**"** + **A** - **A**k**"** ≤ (∈ + **"A** - **ΠA "** ;

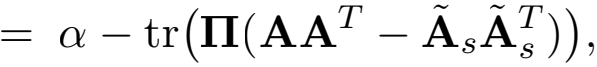
by which the lemma follows. 

Given Lemma [19, we now provide the proof of Lemma](#bookmark183) [11.](#bookmark159)

**Proof** For any orthogonal projection matrix **M**, it holds that **M** = **MM**. It follows that







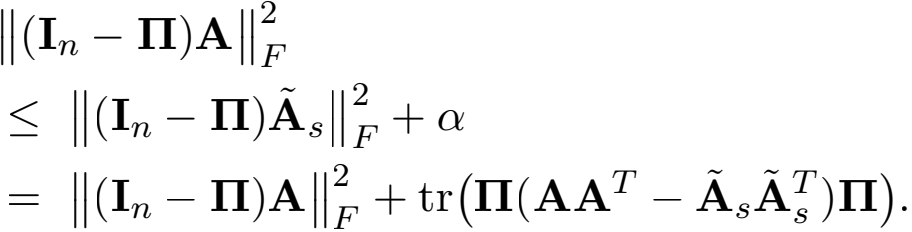
Wang, Gittens, and Mahoney

where the last equality follows by letting Q = tr (**AA**T — s ) ≥ 0 which is independent

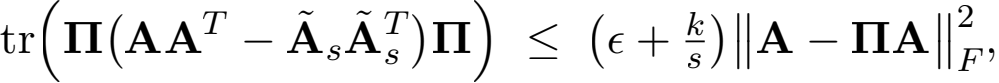
of **Π** . The above equality can be equivalently written as

**"**(**I**n — **Π**)s **"** + Q = **"**(**I**n — **Π**)**A"** + tr (**Π**(**AA**T — s )**Π**). (13)

Under Assumption [2](#bookmark184) that s 二 **AA**T[, it follows from (13) that](#bookmark185)



Under Assumptions [1](#bookmark182) and [2, we can apply Lemma](#bookmark184) [19](#bookmark183) to bound the right-hand side:



by which the lemma follows. 

**Appendix E. Proof of Lemma** [**12**](#bookmark160)

Because **B** enjoys the projection-cost preservation property, there exists a constant Q ≥ 0 such that for any rank k orthogonal projection matrices **Π** 1 and **Π**2, the two inequalities

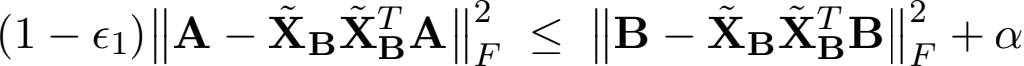
[(1 — ∈1 ) **"A** — **Π** 1**A"** ≤ **"B** — **Π** 1**B"** + Q; (14)](#bookmark187)

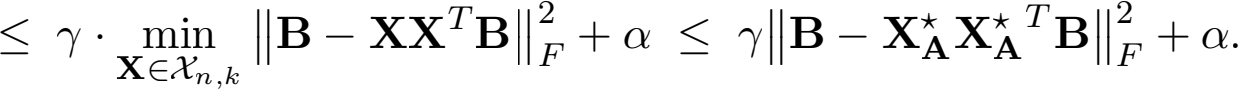
**"B** — **Π**2**B"** + Q ≤ (1 + ∈2 ) **"A** — **Π**2**A"** ([15)](#bookmark188)

hold simultaneously with probability at least 1 — 2δ . Let **Π** 1 = **B** and **X** =

argmin**X**∈Xn;k Ⅱ**A** — **XX**T**A**Ⅱ . [It follows from (14) and the deﬁnition of](#bookmark186) √-approximate

k-means algorithm that

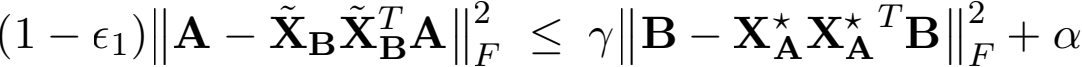


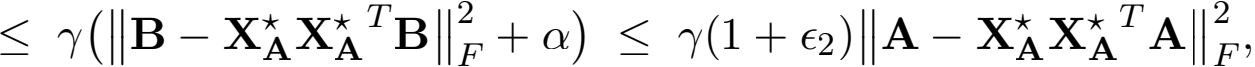


Let **Π**2 = **XX**T . [It follows from (15) that](#bookmark187)

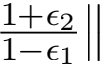
**"B** — **XX**T**B "** + Q ≤ (1 + ∈2 ) **"A** — **XX**T**A "** .

Combining the above results, we have that





where the second inequality follows from that √ ≥ 1 and Q ≥ 0. It follows that

**"A** — **B A "** ≤ √ **A** — **XX**T**A "** = √  **X** **"A** — **XX**T**A"**

This concludes our proof.

Scalable Kernel K-Means Clustering with Nystrm Approximation

**References**

Ahmed Alaoui and Michael W. Mahoney. Fast Randomized Kernel Ridge Regression with Statistical Guarantees. In Advances in Neural Information Processing Systems (NIPS) . 2015.

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. Machine Learning, 75(2):245–248, 2009.

Okan Arikan. Compression of motion capture databases. In ACM Transactions on Graphics (TOG), volume 25, pages 890–897. ACM, 2006.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In Annual ACM-SIAM Symposium on Discrete Algorithms, 2007.

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of Euclidean k-means. arXiv preprint arXiv:1502.03316, 2015.

Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In International Conference on Learning Theory (COLT), 2013.

Tamara L. Berg, Alexander C. Berg, Jaety Edwards, and David A. Forsyth. Who’s in the picture. Advances in Neural Information Processing Systems (NIPS), 2004.

Christos Boutsidis, Petros Drineas, and Michael W. Mahoney. Unsupervised Feature Selection for the k-means Clustering Problem. In Advances in Neural Information Processing Systems (NIPS) . 2009.

Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k- means clustering. In Advances in Neural Information Processing Systems (NIPS), 2010.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-Optimal Column-Based Matrix Reconstruction. SIAM Journal on Computing, 43(2):687–717, 2014.

Christos Boutsidis, Anastasios Zouzias, Michael W. Mahoney, and Petros Drineas. Randomized Dimensionality Reduction for k-Means Clustering. IEEE Transactions on Information Theory, 61(2):1045–1062, 2015.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM Computing Surveys, 41(3):1–58, 2009.

Anil Chaturvedi, J. Douglas Carroll, Paul E. Green, and John A. Rotondo. A feature- based approach to market segmentation via overlapping k-centroids clustering. Journal of Marketing Research, 34(3):370–377, 1997.

Jef Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In Problems in Analysis, Papers dedicated to Salomon Bochner, pages 195–199. Princeton University Press, 1969.

Ke Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. SIAM Journal on Computing, 39(3):923–947, 2009.

Wang, Gittens, and Mahoney

Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel Spectral Clustering in Distributed Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(3):568–586, 2011.

Radha Chitta, Rong Jin, Timothy C. Havens, and Anil K. Jain. Approximate Kernel k-means: Solution to Large Scale Kernel Clustering. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2011.

Radha Chitta, Rong Jin, and Anil K. Jain. E伍cient Kernel Clustering Using Random Fourier Features. In IEEE International Conference on Data Mining (ICDM), 2012.

Jiawei Chiu and Laurent Demanet. Sublinear Randomized Algorithms for Skeleton Decompositions. SIAM Journal on Matrix Analysis and Applications, 34(3):1361–1383, 2013.

Kenneth L. Clarkson and David P. Woodruf. Low rank approximation and regression in input sparsity time. In Annual ACM Symposium on Theory of Computing (STOC), 2013.

Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In Annual ACM Symposium on Theory of Computing (STOC), 2015.

Michael B Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2017.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In Conference on Artiﬁcial Intelligence and Statistics (AISTATS), 2010.

Sanjoy Dasgupta and Yoav Freund. Random projection trees for vector quantization. IEEE Transactions on Information Theory, 55(7):3229–3242, 2009.

Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2004.

Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In SIAM International Conference on Data Mining (SDM), 2005.

William E Donath and Alan J Hofman. Algorithms for partitioning graphs and computer logic based on eigenvectors of connection matrices. IBM Technical Disclosure Bulletin, 15(3):938–944, 1972.

William E Donath and Alan J Hofman. Lower bounds for the partitioning of graphs. IBM Journal of Research and Development, 17(5):420–425, 1973.

Petros Drineas and Michael W. Mahoney. On the Nystro…m method for approximating a Gram matrix for improved kernel-based learning. Journal of Machine Learning Research, 6:2153–2175, 2005.

Scalable Kernel K-Means Clustering with Nystrm Approximation

Petros Drineas and Michael W Mahoney. RandNLA: randomized numerical linear algebra. Communications of the ACM, 59(6):80–90, 2016.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. SIAM Journal on Computing, 36(1): 132–157, 2006.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. SIAM Journal on Matrix Analysis and Applications, 30(2):844–881, September 2008.

Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tam/as Sarl/os. Faster least squares approximation. Numerische Mathematik, 117(2):219–249, 2011.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruf. Fast approximation of matrix coherence and statistical leverage. Journal of Machine Learning Research, 13:3441–3472, 2012.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: constant-size coresets fork-means, PCA and projective clustering. In Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2013.

Miroslav Fiedler. Algebraic connectivity of graphs. Czechoslovak mathematical journal, 23

(2):298–305, 1973.

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nystro…m method. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2):214–225, 2004.

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL [http://archive.](http://archive.ics.uci.edu/ml) [ics.uci.edu/ml.](http://archive.ics.uci.edu/ml)

Michael R. Garey, D. S. Johnson, and Hans S. Witsenhausen. The complexity of the generalized Lloyd-Max problem. IEEE Transactions on Information Theory, 28(2):255– 256, 1982.

Alex Gittens and Michael W. Mahoney. Revisiting the Nystro…m method for improved large-scale machine learning. Journal of Machine Learning Research, 17(1):3977–4041, 2016.

Alex Gittens, Jey Kottalam, Jiyan Yang, Michael F. Ringenburg, Jatin Chhugani, Evan Racah, Mohitdeep Singh, Yushu Yao, Curt Fischer, Oliver Ruebel, Benjamin Bowen, Norman G. Lewis, Michael W. Mahoney, Venkat Krishnamurthy, and Prabhat. A Multi- Platform Evaluation of the Randomized CX Low-Rank Matrix Factorization in Spark. In IEEE International Parallel and Distributed Processing Symposium Workshops, 2016.

Stephen Guattery and Gary L. Miller. On the performance of spectral graph partitioning methods. In Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1995.

Wang, Gittens, and Mahoney

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review, 53(2):217–288, 2011.

Mark S. Handcock, Adrian E. Raftery, and Jeremy M. Tantrum. Model-based clustering for social networks. Journal of the Royal Statistical Society: Series A (Statistics in Society) , 170(2):301–354, 2007.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In Annual ACM Symposium on Theory of Computing (STOC), 2004.

Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. Computer

Vision, Graphics, and Image Processing, 29(1):100–132, 1985.

Rong Jin, Tianbao Yang, Mehrdad Mahdavi, Y Li, and Z Zhou. Improved bounds for the Nystro…m method with application to kernel classiﬁcation. IEEE Transactions on Information Theory, 59(10):6939–6949, 2013.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. Contemporary mathematics, 26(189-206), 1984.

Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. In Annual Symposium on Computational Geometry, 2002.

Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time (1 + ∈)- approximation algorithm for k-means clustering in any dimensions. In Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2004.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nystro…m method. Journal of Machine Learning Research, 13:981–1006, 2012.

Yann LeCun, L/eon Bottou, Yoshua Bengio, and Patrick Hafner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Mu Li, Wei Bi, J.T. Kwok, and Bao-Liang Lu. Large-scale Nystrm… kernel matrix approximation using randomized SVD. IEEE Transactions on Neural Networks and Learning Systems, 26(1):152–164, Jan 2015.

Yingyu Liang, Maria-Florina F Balcan, Vandana Kanchanapally, and David Woodruf. Improved distributed principal component analysis. In Advances in Neural Information Processing Systems (NIPS), 2014.

Stuart Lloyd. Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2):129–137, 1982.

Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized Hadamard transform. In Advances in Neural Information Processing Systems (NIPS), 2013.

Scalable Kernel K-Means Clustering with Nystrm Approximation

Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is NP-hard. In International Workshop on Algorithms and Computation, pages 274–285. Springer, 2009.

Michael W. Mahoney. Randomized algorithms for matrices and data. Foundations and Trends in Machine Learning, 3(2):123–224, 2011.

Michael W. Mahoney and Petros Drineas. Structural properties underlying high-quality randomized numerical linear algebra algorithms. In P. Bu…hlmann, P. Drineas, M. Kane, and M. van de Laan, editors, Handbook of Big Data, pages 137–154. CRC Press, 2016.

Jiri Matousek. On approximate geometric k-clustering. Discrete & Computational

Geometry, 24(1):61–84, 2000.

Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input- sparsity time and applications to robust linear regression. In Annual ACM Symposium on Theory of Computing (STOC), 2013.

Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Mateh Zaharia, and Ameet Talwalkar. MLlib: Machine Learning in Apache Spark. Journal of Machine Learning Research, 17(34):1–7, 2016.

Cameron Musco and Christopher Musco. Recursive sampling for the Nystrom method. In Advances in Neural Information Processing Systems (NIPS), 2017.

John Nelson and Huy L Nguyˆen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2013.

Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. Advances in Neural Information Processing Systems (NIPS), 2002.

…

Evert J. Nystro…m. Uber die praktische au丑o…sung von integralgleichungen mit anwendungen auf randwertaufgaben. Acta Mathematica, 54(1):185–204, 1930.

Farhad Pourkamali-Anaraki and Stephen Becker. Randomized Clustered Nystro…mfor Large- Scale Kernel Machines. arXiv preprint arXiv:1612.06470, 2016.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems (NIPS), 2007.

Mark Rudelson and Roman Vershynin. The Littlewood–Oford problem and invertibility of random matrices. Advances in Mathematics, 218(2):600–633, 2008.

Bernhard Scho…lkopf and Alexander J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2002.

Bernhard Scho…lkopf, Alexander Smola, and Klaus-Robert Mu…ller. Nonlinear component analysis as a kernel eigenvalue problem. Neural computation, 10(5):1299–1319, 1998.

Wang, Gittens, and Mahoney

Roded Sharan, Rani Elkon, and Ron Shamir. Cluster analysis and its applications to gene expression data. In Bioinformatics and Genome Analysis, pages 83–108. Springer, 2002.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

Si Si, Cho-Jui Hsieh, and Inderjit Dhillon. Memory e伍cient kernel approximation. In International Conference on Machine Learning (ICML), 2014.

Mingjun Song and Sanguthevar Rajasekaran. Fast algorithms for constant approximation k-means clustering. Transactions on Machine Learning and Data Mining, 3(2):67–79, 2010.

Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and ﬁnite element meshes. In Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1996.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 3(Dec):583–617, 2002.

Terence Tao and Van Vu. Random matrices: The distribution of the smallest singular values. Geometric And Functional Analysis, 20(1):260–297, 2010.

Joel A Tropp. Improved analysis of the subsampled randomized hadamard transform. Advances in Adaptive Data Analysis, 3(01n02):115–126, 2011.

Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Fixed-rank approximation of a positive-semideﬁnite matrix from streaming data. In Advances in Neural Information Processing Systems (NIPS), 2017.

L. H. Ungar and D. P. Foster. Clustering Methods for Collaborative Filtering. In Proc. Recommender Systems, Papers from 1998 Workshop, 1998. Technical Report WS-98-08.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027, 2010.

Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nystro…m approximation via adaptive sampling. Journal of Machine Learning Research, 14:2729– 2769, 2013.

Shusen Wang, Luo Luo, and Zhihua Zhang. SPSD matrix approximation vis column selection: Theories, algorithms, and extensions. Journal of Machine Learning Research, 17(49):1–49, 2016a.

Shusen Wang, Zhihua Zhang, and Tong Zhang. Towards more e伍cient SPSD matrix approximation and CUR matrix decomposition. Journal of Machine Learning Research, 17(210):1–49, 2016b.

Scalable Kernel K-Means Clustering with Nystrm Approximation

Shusen Wang, Alex Gittens, and Michael W Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. In International Conference on Machine Learning (ICML), 2017.

Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In Advances in Neural Information Processing Systems (NIPS), 2009.

Christopher Williams and Matthias Seeger. Using the Nystro…m method to speed up kernel machines. In Advances in Neural Information Processing Systems (NIPS), 2001.

David P Woodruf. Sketching as a tool for numerical linear algebra. Foundations and Trends in Theoretical Computer Science, 10(1–2):1–157, 2014.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nystro…m method vs random Fourier features: A theoretical and empirical comparison. In Advances in Neural Information Processing Systems (NIPS), 2012.

Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. HotCloud, 10(10-10):95, 2010.

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In USENIX Conference on Networked Systems Design and Implementation (NSDI), 2012.

Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. Spectral relaxation for k-means clustering. In Advances in Neural Information Processing Systems (NIPS) , 2002.

Kai Zhang and James T. Kwok. Clustered Nystro…m method for large scale manifold learning and dimension reduction. IEEE Transactions on Neural Networks, 21(10):1576–1587, 2010.

Zhihua Zhang. The singular value decomposition, applications and beyond. arXiv:1510.08532, 2015.