Greedy Nyström Approximation Supplementary Material

Ahmed K. Farahat **Ali Ghodsi** **Mohamed S. Kamel**
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{afarahat,aghodsib,mkamel}@uwaterloo.ca

1 Projection for rank-1 Nyström approximation

This section provides a proof for the claim that rank-1 Nyström approximation based on the i-th column implicitly projects all data points onto a line which contains data point i and the origin in the high-dimensional feature space defined by the kernel.

The rank-1 Nyström approximation of matrix K based on column i is calculated as:

$$\tilde{K}_{\{i\}} = \frac{1}{K_{ii}} K_{:i} K_{:i}^T, \tag{1}$$

where $K_{:i}$ denotes the i-th column of K, and K_{ii} denotes the i-th diagonal element of K.

Assume that a linear kernel is used. The kernel matrix is calculated as $K = X^T X$, where X is an $m \times n$ data matrix, and m is the number of features. Let $X_{:i}$ be the i-th column of X. $X_{:i}$ also represents a vector that connects data point i and the origin. $K_{:i}$ and K_{ii} can be written as: $K_{:i} = X^T X_{:i}$, and $K_{ii} = \|X_{:i}\|^2$, where $\|.\|$ is the ℓ_2 norm. Based on this, $\tilde{K}_{\{i\}}$ can be expressed as:

$$\tilde{K}_{\{i\}} = X^T \frac{X_{:i}}{\|X_{:i}\|} \frac{X_{:i}^T}{\|X_{:i}\|} X, \tag{2}$$

where $X^T X_{:i}/\|X_{:i}\|$ is a column vector whose j-th element is $X_{:j}^T X_{:i}/\|X_{:i}\|$. This value is the scalar projection of data point j onto $X_{:i}$ (as illustrated in Figure 1). This means that $\tilde{K}_{\{i\}}$ implicitly projects all data points into a 1-dimensional subspace (i.e., a line) which contains data point i and the origin, and then calculates the inner-products between the projected points.

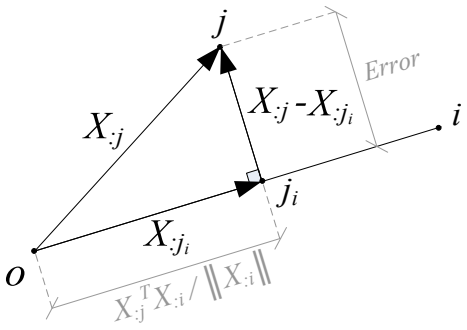

Figure 1: Projection for rank-1 Nyström approximation.

The same proof applies to other kernel types, as any kernel matrix implicitly maps data vectors to a high-dimensional linear space. In this case, the columns of X represent data vectors in the high-dimensional feature space defined by the kernel.

2 Updates formulas for $\boldsymbol{f}$ and $\boldsymbol{g}$

In this section, the recursive formulas for $\boldsymbol{f}$ and $\boldsymbol{g}$ are derived. The greedy selection criterion at iteration t is:

$$l = arg\ \underset{i}{max}\ \|\frac{1}{\sqrt{E_{ii}}}E_{:i}\|^2, \tag{3}$$

where E is the residual matrix at iteration t, $E_{:i}$ denotes the i-th column of E, and E_{ii} denotes the i-th diagonal element of E.

As the Nyström approximation is calculated in a recursive manner based on the residual matrix at the previous iteration, E, $E_{:i}$, and E_{ii} for a candidate column i can be recursively calculated as follows:

$$\begin{aligned} E^{(t)} &= (E - \frac{1}{\alpha}\boldsymbol{\delta}\boldsymbol{\delta}^T)^{(t-1)} = (E - \boldsymbol{\omega}\boldsymbol{\omega}^T)^{(t-1)}, \\ E_{:i}^{(t)} &= (E_{:i} - \frac{\boldsymbol{\delta}_i}{\alpha}\boldsymbol{\delta})^{(t-1)} = (E_{:i} - \boldsymbol{\omega}_i\boldsymbol{\omega})^{(t-1)}, \\ E_{ii}^{(t)} &= (E_{ii} - \frac{\boldsymbol{\delta}_i^2}{\alpha})^{(t-1)} = (E_{ii} - \boldsymbol{\omega}_i^2)^{(t-1)}. \end{aligned} \tag{4}$$

Let $\boldsymbol{f}_i = \|E_{:i}\|^2$ and $\boldsymbol{g}_i = E_{ii}$ be the numerator and denominator of the criterion function for data point i respectively. Based on (4), $\boldsymbol{f}_i^{(t)}$ can be calculated as:

$$\begin{aligned} \boldsymbol{f}_i^{(t)} &= \left(\|E_{:i} - \boldsymbol{\omega}_i\boldsymbol{\omega}\|^2\right)^{(t-1)} = \left((E_{:i} - \boldsymbol{\omega}_i\boldsymbol{\omega})^T(E_{:i} - \boldsymbol{\omega}_i\boldsymbol{\omega})\right)^{(t-1)} \\ &= \left(E_{:i}^TE_{:i} - 2\boldsymbol{\omega}_iE_{:i}^T\boldsymbol{\omega} + \boldsymbol{\omega}_i^2\|\boldsymbol{\omega}\|^2\right)^{(t-1)} = \left(\boldsymbol{f}_i - 2\boldsymbol{\omega}_iE_{:i}^T\boldsymbol{\omega} + \boldsymbol{\omega}_i^2\|\boldsymbol{\omega}\|^2\right)^{(t-1)}. \end{aligned} \tag{5}$$

Similarly, $\boldsymbol{g}_i^{(t)}$ can be calculated as:

$$\boldsymbol{g}_i^{(t)} = E_{ii}^{(t)} = \left(E_{ii} - \boldsymbol{\omega}_i^2\right)^{(t-1)} = \left(\boldsymbol{g}_i - \boldsymbol{\omega}_i^2\right)^{(t-1)}. \tag{6}$$

Let $\boldsymbol{f} = [\boldsymbol{f}_i]_{i=1..n}$and $\boldsymbol{g} = [\boldsymbol{g}_i]_{i=1..n}$, $\boldsymbol{f}^{(t)}$ and $\boldsymbol{g}^{(t)}$ can be expressed as:

$$\begin{aligned} \boldsymbol{f}^{(t)} &= \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ E\boldsymbol{\omega}\right) + \|\boldsymbol{\omega}\|^2\left(\boldsymbol{\omega} \circ \boldsymbol{\omega}\right)\right)^{(t-1)}, \\ \boldsymbol{g}^{(t)} &= \left(\boldsymbol{g} - \left(\boldsymbol{\omega} \circ \boldsymbol{\omega}\right)\right)^{(t-1)}, \end{aligned} \tag{7}$$

where $\circ$ represents the Hadamard product operator, and $\|.\|$ is the ℓ_2 norm.

Based on the recursive formula of E, the term $E\boldsymbol{\omega}$ at iteration $(t-1)$ can be expressed as:

$$E\boldsymbol{\omega} = \left(K - \Sigma_{r=1}^{t-2}\left(\boldsymbol{\omega}\boldsymbol{\omega}^T\right)^{(r)}\right)\boldsymbol{\omega} = K\boldsymbol{\omega} - \Sigma_{r=1}^{t-2}\left(\boldsymbol{\omega}^{(r)^T}\boldsymbol{\omega}\right)\boldsymbol{\omega}^{(r)} \tag{8}$$

Substitute with $E\boldsymbol{\omega}$ in Equation (7), the update formulas for $\boldsymbol{f}$ and $\boldsymbol{g}$ are given as:

$$\begin{aligned} \boldsymbol{f}^{(t)} &= \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ \left(K\boldsymbol{\omega} - \Sigma_{r=1}^{t-2}\left(\boldsymbol{\omega}^{(r)^T}\boldsymbol{\omega}\right)\boldsymbol{\omega}^{(r)}\right)\right) + \|\boldsymbol{\omega}\|^2\left(\boldsymbol{\omega} \circ \boldsymbol{\omega}\right)\right)^{(t-1)}, \\ \boldsymbol{g}^{(t)} &= \left(\boldsymbol{g} - \left(\boldsymbol{\omega} \circ \boldsymbol{\omega}\right)\right)^{(t-1)}. \end{aligned} \tag{9}$$

3 Partition-based greedy Nyström algorithm

This section describes the partition-based algorithm in details and derives recursive formulas for $\boldsymbol{f}$ and $\boldsymbol{g}$. The partition-based algorithm first divides data points into $c \ll n$ random groups, and then selects the column of K which best represents the centroids of these groups in the high-dimensional feature space.

Let $\mathcal{P}_j$ be the set of data points that belong to the j-th partition, $P = \{\mathcal{P}_1, \mathcal{P}_2, ...\mathcal{P}_c\}$ be a random partitioning of data points into c groups, and G be an $c \times n$ matrix whose element G_{ji} is the inner-product of the centroid of the j-th group and the i-th data point, weighted with the size of the j-th group. The use of weighted inner-products avoids any bias towards larger groups when calculating the sum of scalar projections. As the scalar projections are implicitly calculated in a high-dimensional linear space defined by the kernel matrix K, G_{ji} can be calculated in this linear space as: $G_{ji} = \sum_{r \in \mathcal{P}_j} K_{ir}$. In general, it requires $\mathcal{O}\left(n^2\right)$ to calculate G given K. However, G needs to be calculated only once during the calculation of K. In the case of a linear kernel, this complexity could be significantly reduced by calculating the centroids of each group in the feature space, and then the inner-products between each centroid and all data points. This computational complexity could be reduced further if the data matrix is very sparse. In addition, there is no need to calculate and store the whole kernel matrix in order to calculate G.

Let $E^{(t)}$ and $H^{(t)}$ be the residual matrices of K and G at iteration t respectively. The efficient sampling criterion based on centroids can be expressed as follows:

$$l = arg\ \underset{i}{max}\ \| \frac{1}{\sqrt{E_{ii}}} H_{:i} \|^2, \tag{10}$$

where $H_{:i}$ denotes the i-th column of H, and E_{ii} denotes the i-th diagonal element of E. The term $H_{ji}/\sqrt{E_{ii}}$ is the scalar projection of the j-th centroid onto $X_{:i}$. Let $\boldsymbol{\delta}^{(t)}$ be column of E selected at iteration t, $\alpha^{(t)}$ be the corresponding diagonal element of E, and $\boldsymbol{\gamma}^{(t)}$ be the corresponding column of H. Define $\boldsymbol{\omega}^{(t)} = \boldsymbol{\delta}^{(t)}/\sqrt{\alpha^{(t)}}$, and $\boldsymbol{\upsilon}^{(t)} = \boldsymbol{\gamma}^{(t)}/\sqrt{\alpha^{(t)}}$. The rank-1 approximation of $H^{(t)}$ can be calculated as:

$$\tilde{H}^{(t)}_{\{l\}} = \frac{1}{\alpha^{(t)}} \boldsymbol{\gamma}^{(t)} \boldsymbol{\delta}^{T(t)} = \boldsymbol{\upsilon}^{(t)} \boldsymbol{\omega}^{T(t)}, \tag{11}$$

and the new residual matrix H can be calculated as:

$$H^{(t+1)} = H^{(t)} - \boldsymbol{\upsilon}^{(t)} \boldsymbol{\omega}^{T(t)} \tag{12}$$

Based on this recursive formula, the greedy sampling criterion (Equation (10)) can be calculated in a recursive manner as follows. Similar to Equation (4), H, $H_{:i}$, and E_{ii} can be recursively calculated as:

$$\begin{aligned} H^{(t)} &= (H - \boldsymbol{\upsilon}\boldsymbol{\omega}^T)^{(t-1)}, \\ H^{(t)}_{:i} &= (H_{:i} - \boldsymbol{\omega}_i \boldsymbol{\upsilon})^{(t-1)}, \\ E^{(t)}_{ii} &= (E_{ii} - \boldsymbol{\omega}_i^2)^{(t-1)}. \end{aligned} \tag{13}$$

Similar to Section 2, let $\boldsymbol{f}_i = \|H_{:i}\|^2$ and $\boldsymbol{g}_i = E_{ii}$ be the numerator and denominator of the criterion function for data point i. $\boldsymbol{f}_i^{(t)}$ and $\boldsymbol{g}_i^{(t)}$ can be calculated as follows:

$$\begin{aligned} \boldsymbol{f}_i^{(t)} &= \left(\|H_{:i} - \boldsymbol{\omega}_i \boldsymbol{\upsilon}\|^2\right)^{(t-1)} = \left(\boldsymbol{f}_i - 2\boldsymbol{\omega}_i H_{:i}^T \boldsymbol{\upsilon} + \boldsymbol{\omega}_i^2 \|\boldsymbol{\upsilon}\|^2\right)^{(t-1)}, \\ \boldsymbol{g}_i^{(t)} &= E_{ii}^{(t)} = \left(E_{ii} - \boldsymbol{\omega}_i^2\right)^{(t-1)} = \left(\boldsymbol{g}_i - \boldsymbol{\omega}_i^2\right)^{(t-1)}. \end{aligned} \tag{14}$$

Let $\boldsymbol{f} = [\boldsymbol{f}_i]_{i=1..n}$ and $\boldsymbol{g} = [\boldsymbol{g}_i]_{i=1..n}$, $\boldsymbol{f}^{(t)}$ and $\boldsymbol{g}^{(t)}$ can be expressed as:

$$\begin{aligned} \boldsymbol{f}^{(t)} &= \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ H^T \boldsymbol{\upsilon}\right) + \|\boldsymbol{\upsilon}\|^2 (\boldsymbol{\omega} \circ \boldsymbol{\omega})\right)^{(t-1)}, \\ \boldsymbol{g}^{(t)} &= (\boldsymbol{g} - (\boldsymbol{\omega} \circ \boldsymbol{\omega}))^{(t-1)}, \end{aligned} \tag{15}$$

where $\circ$ represents the Hadamard product operator, and $\|.\|$ is the ℓ_2 norm.

The term $H^T\boldsymbol{\upsilon}$ at iteration $(t-1)$ can be calculated recursively as:

$$H^T\boldsymbol{\upsilon} = \left(G^T - \Sigma_{r=1}^{t-2} \left(\boldsymbol{\omega}\boldsymbol{\upsilon}^T\right)^{(r)}\right)\boldsymbol{\upsilon} = G^T\boldsymbol{\upsilon} - \Sigma_{r=1}^{t-2} \left(\boldsymbol{\upsilon}^{(r)T}\boldsymbol{\upsilon}\right)\boldsymbol{\omega}^{(r)}$$

Substitute with $H^T\boldsymbol{\upsilon}$ in Equation (15), the update formulas for $\boldsymbol{f}$ and $\boldsymbol{g}$ are given as:

$$\begin{aligned} \boldsymbol{f}^{(t)} &= \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ \left(G^T\boldsymbol{\upsilon} - \Sigma_{r=1}^{t-2}\left(\boldsymbol{\upsilon}^{(r)T}\boldsymbol{\upsilon}\right)\boldsymbol{\omega}^{(r)}\right)\right) + \|\boldsymbol{\upsilon}\|^2 (\boldsymbol{\omega} \circ \boldsymbol{\omega})\right)^{(t-1)}, \\ \boldsymbol{g}^{(t)} &= (\boldsymbol{g} - (\boldsymbol{\omega} \circ \boldsymbol{\omega}))^{(t-1)}. \end{aligned} \tag{16}$$

Table 1: Properties of data sets used to evaluate different Nyström methods. n and m are the number of instances and features respectively.

Data set	Type	n	m
Reuters-21578	Documents	5946	18933
Reviews	Documents	4069	36746
LA1	Documents	3204	29714
MNIST-4K	Digit Images	4000	784
PIE-20	Face Images	3400	1024
Yale-B-38	Face Images	2414	1024

4 The complete partition-based greedy Nyström algorithm

Inputs: K, k, d,

Outputs: $\mathcal{S}$, $\tilde{K}_{\mathcal{S}}$,$\tilde{K}_{\mathcal{S},d}$, Y

Steps:

1. Initialize $\mathcal{S} = \{\ \}$
2. Generate a random partitioning P, Calculate G: $G_{ji} = \sum_{r \in \mathcal{P}_j} K_{ir}$
3. Initialize $\boldsymbol{f}_i^{(0)} = \|G_{:i}\|^2$, $\boldsymbol{g}_i^{(0)} = K_{ii}$, $\boldsymbol{f}^{(0)} = \left[\boldsymbol{f}_i^{(0)}\right]_{i=1..n}$ and $\boldsymbol{g}^{(0)} = \left[\boldsymbol{g}_i^{(0)}\right]_{i=1..n}$
4. Repeat $t = 1 \rightarrow k$:
   (a) $l = arg\ \underset{i}{max}\ \boldsymbol{f}_i^{(t)} / \boldsymbol{g}_i^{(t)}$
   (b) $\mathcal{S} = \mathcal{S} \cup \{l\}$
   (c) Update $\boldsymbol{\delta}^{(t)} = K_{:l} - \sum_{r=1}^{t-1} \boldsymbol{\omega}_l^{(r)} \boldsymbol{\omega}^{(r)}$, $\quad \boldsymbol{\gamma}^{(t)} = G_{:l} - \sum_{r=1}^{t-1} \boldsymbol{\omega}_l^{(r)} \boldsymbol{\upsilon}^{(r)}$, $\quad \alpha^{(t)} = \boldsymbol{\delta}_l^{(r)}$
   (d) Calculate $\boldsymbol{\omega}^{(t)} = \boldsymbol{\delta}^{(t)} / \sqrt{\alpha^{(t)}}$, $\boldsymbol{\upsilon}^{(t)} = \boldsymbol{\gamma}^{(t)} / \sqrt{\alpha^{(t)}}$
   (e) Update $\boldsymbol{f}$, $\boldsymbol{g}$:

$$\boldsymbol{f}^{(t)} = \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ \left(G^T \boldsymbol{\upsilon} - \Sigma_{r=1}^{t-2} \left(\boldsymbol{\upsilon}^{(r)T} \boldsymbol{\upsilon}\right) \boldsymbol{\omega}^{(r)}\right)\right) + \|\boldsymbol{\upsilon}\|^2 (\boldsymbol{\omega} \circ \boldsymbol{\omega})\right)^{(t-1)},$$

$$\boldsymbol{g}^{(t)} = \left(\boldsymbol{g} - (\boldsymbol{\omega} \circ \boldsymbol{\omega})\right)^{(t-1)}.$$

5. $W = \left[\ w^{(1)} \quad w^{(2)} \quad ... \quad w^{(k)}\ \right]^T$, $\tilde{K}_{\mathcal{S}} = W^T W$
6. $\Omega = eigvec(WW^T)$
7. $Y = \Omega_d^T W$, $\tilde{K}_{\mathcal{S},d} = Y^T Y$

5 Properties of data sets

Experiments have been conducted on six benchmark data sets, whose properties are summarized in Table 1[1]. The *Reuters-21578* is the training set of the Reuters-21578 collection [1], which has been used in the evaluation of many document clustering and classification techniques. The *Reviews* and *LA1* are documents data sets from TREC collections[2]. The pre-processed versions of *Reviews* and *LA1* that are distributed with the CLUTO Toolkit[3] [2] were used. The *MNIST-4K* is a subset of the MNIST data set of handwritten digits, which has been used to evaluate different digit recognition

[1]The data sets *Reuters-21578*, *MNIST-4K*, *PIE-20* and *YaleB-38* are available in MAT format at: `http://www.zjucadcg.cn/dengcai/Data/data.html`. *PIE-20* is a subset of *PIE-32x32* with the images of the first 20 persons.

[2]`http://trec.nist.gov`

[3]`http://www.cs.umn.edu/~karypis/cluto`

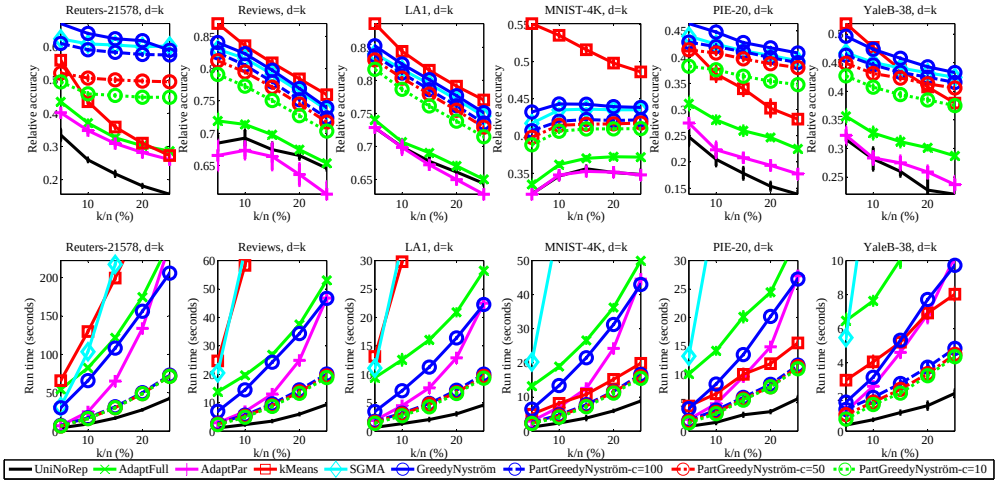

Figure 2: The relative accuracy and run time for the calculation of rank-k approximations $\tilde{K}_{\mathcal{S}}$.

algorithms[4]. The *PIE-20* and *YaleB-38* are subsets of the CMU PIE [3] and Extended Yale Face [4] data sets respectively. These two data sets contain face images and have been used in the evaluation of many face recognition algorithms.

6 Results for rank-k Nyström approximation

Figure 2 shows the relative accuracy and run times for different methods when calculating the rank-k Nyström approximations $\tilde{K}_{\mathcal{S}}$. It can be observed from these results that the proposed algorithms obtain very accurate low-rank approximations, with minimum overhead in run time.

References

[1] D.D. Lewis. Reuters-21578 text categorization test collection distribution 1.0, 1999.

[2] G. Karypis. CLUTO - a clustering toolkit. Technical Report #02-017, University of Minnesota, Department of Computer Science, 2003.

[3] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1615–1618, 2003.

[4] K.C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):684–698, 2005.

[4] `http://yann.lecun.com/exdb/mnist`