Greedy Nyström Approximation

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

Abstract

The Nyström method is an efficient technique which obtains a low-rank approximation of a large kernel matrix based on a subset of its columns. The quality of the Nyström approximation highly depends on the subset of columns used, which are usually selected using random sampling. This paper presents a novel recursive algorithm for calculating the Nyström approximation, and an effective greedy criterion for column selection[1].

1 The Nyström method

The Nyström method obtains a low-rank approximation of a kernel matrix using a subset of its columns. This method has been used in many large-scale applications in machine learning and data mining, including efficient learning of kernel-based models, fast dimension reduction, and efficient spectral clustering. Let K be an $n \times n$ kernel matrix defined over n data instances. The Nyström method starts by sampling $k \ll n$ columns of K. Let $\mathcal{S}$ be the set of the indices of the sampled columns, D be an $n \times k$ matrix which consists of the sampled columns, and A be a $k \times k$ matrix whose elements are $\{K_{ij} : i, j \in \mathcal{S}\}$, where K_{ij} denotes the element of K at row i and column j. The Nyström method calculates a rank-k approximation of K based on $\mathcal{S}$ as [1]:

$$\tilde{K}_{\mathcal{S}} = DA^{-1}D^T, \tag{1}$$

The Nyström method can also be used to approximate the $d \leq k$ leading singular values and vectors of K using those of A [1]. In addition, the approximate singular values and vectors of K can be used to map data points to a d-dimensional space, where the kernel over the data points represents a rank-d approximation of K, which is referred to as $\tilde{K}_{\mathcal{S},d}$ throughout the rest of the paper.

The quality of the Nyström approximation highly depends on the subset of selected columns. Different sampling schemes have been used with the Nyström method. These schemes include: uniform sampling [1], which has been the most common technique for column selection; non-uniform sampling [2, 3], using probabilities calculated based on the kernel matrix; adaptive sampling [4, 5], in which probabilities are updated based on intermediate Nyström approximations; and deterministic sampling [6, 7], where columns are selected such that some criterion function is optimized.

2 Recursive Nyström approximation

This section proposes a novel recursive algorithm for calculating Nystrom̈ approximation. Let $l \in \mathcal{S}$ be the index of one of the sampled columns, α be the l-th diagonal element of K, $\boldsymbol{\delta}$ be the l-th column of K, and $\boldsymbol{\beta}$ be a column vector of length $k-1$ whose elements are $\{K_{il} : i \in \mathcal{S} \setminus \{l\}\}$. Without loss of generality, the rows and columns of K, A and D can be rearranged such that the first row and column correspond to l:

$$A = \begin{bmatrix} \alpha & \boldsymbol{\beta}^T \\ \boldsymbol{\beta} & \Gamma \end{bmatrix}, \quad D = \begin{bmatrix} \boldsymbol{\delta} & \Delta^T \end{bmatrix} \tag{2}$$

[1] An extended version of this paper is currently under review at the Journal of Machine Learning Research.

where Γ is a $(k-1) \times (k-1)$ sub-matrix of A whose elements are $\{K_{ij} : i, j \in \mathcal{S} \setminus \{l\}\}$, and Δ is a $(k-1) \times (n)$ sub-matrix of D whose elements are $\{K_{ij} : i \in \mathcal{S} \setminus \{l\}, j \in \{1, .., n\}\}$.

Let $S = \Gamma - \frac{1}{\alpha}\boldsymbol{\beta}\boldsymbol{\beta}^T$ be the Schur complement of α in A. Use the block-wise inversion formula of A^{-1} and substitute with D and A^{-1} in Equation (1):

$$\begin{aligned}\tilde{K}_{\mathcal{S}} &= \begin{bmatrix} \boldsymbol{\delta} & \Delta^T \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha} + \frac{1}{\alpha^2}\boldsymbol{\beta}^T S^{-1}\boldsymbol{\beta} & -\frac{1}{\alpha}\boldsymbol{\beta}^T S^{-1} \\ -\frac{1}{\alpha} S^{-1}\boldsymbol{\beta} & S^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^T \\ \Delta \end{bmatrix} \\ &= \frac{1}{\alpha}\boldsymbol{\delta}\boldsymbol{\delta}^T + \left(\Delta - \frac{1}{\alpha}\boldsymbol{\beta}\boldsymbol{\delta}^T\right)^T S^{-1} \left(\Delta - \frac{1}{\alpha}\boldsymbol{\beta}\boldsymbol{\delta}^T\right)\end{aligned} \tag{3}$$

Let $\tilde{K}_{\{l\}} = \frac{1}{\alpha}\boldsymbol{\delta}\boldsymbol{\delta}^T$ be the rank-1 Nyström approximation of K obtained using the column corresponding to l [2], and $E^{(K)}_{\{l\}}$ be an $n \times n$ residual matrix which is calculated as: $E^{(K)}_{\{l\}} = K - \tilde{K}_{\{l\}}$. It can be shown that $E^{(\Gamma)}_{\{l\}} = S$ and $E^{(\Delta)}_{\{l\}} = \Delta - \frac{1}{\alpha}\boldsymbol{\beta}\boldsymbol{\delta}^T$ are the sub-matrices of $E^{(K)}_{\{l\}}$ corresponding to Γ and Δ respectively. $\tilde{K}_{\mathcal{S}}$ can be written in terms of $E^{(\Gamma)}_{\{l\}}$ and $E^{(\Delta)}_{\{l\}}$ as:

$$\tilde{K}_{\mathcal{S}} = \tilde{K}_{\{l\}} + {E^{(\Delta)}_{\{l\}}}^T {E^{(\Gamma)}_{\{l\}}}^{-1} E^{(\Delta)}_{\{l\}}. \tag{4}$$

The second term is the Nyström approximation of the residual matrix $E^{(K)}_{\{l\}} = K - \tilde{K}_{\{l\}}$ based on $\mathcal{S} \setminus \{l\}$. This means that rank-$k$ Nyström approximation of matrix K can be constructed in a recursive manner by first calculating a rank-1 Nyström approximation of K based on one column, and then calculating the rank-$(k-1)$ Nyström approximation of the residual matrix.

Let $\boldsymbol{\delta}^{(t)}$ be the column sampled at iteration t of the recursive algorithm, $\alpha^{(t)}$ be the corresponding diagonal element, and $\boldsymbol{\omega}^{(t)} = \boldsymbol{\delta}^{(t)}/\sqrt{\alpha^{(t)}}$. The rank-$k$ Nyström approximation of K can be calculated as: $\tilde{K}_{\mathcal{S}} = \sum_{t=1}^{k} \boldsymbol{\omega}^{(t)}\boldsymbol{\omega}^{(t)T}$, while $\boldsymbol{\delta}^{(t)}$ and $\alpha^{(t)}$ can be efficiently calculated as: $\boldsymbol{\delta}^{(t)} = K_{:l} - \sum_{r=1}^{t-1} \boldsymbol{\omega}^{(r)}_l \boldsymbol{\omega}^{(r)}$ and $\alpha^{(t)} = \boldsymbol{\delta}^{(t)}_l$, where l is the index of the column selected at iteration t, $K_{:l}$ denotes the l-th column of K, and $\boldsymbol{\delta}_l$ denotes the l-th element of $\boldsymbol{\delta}$. $\tilde{K}_{\mathcal{S}}$ can also be expressed in a matrix form as: $W^T W$, where W is an $k \times n$ matrix whose t-th row is $\boldsymbol{\omega}^{(t)T}$. The columns of W can be used to represent data instances in a k-dimensional space. However, as the rows of W are non-orthogonal, the proposed algorithm calculates the singular decomposition of W, and then uses the d leading eigenvectors to represent data instances in a low-dimension space.

Although the recursive Nyström algorithm calculates the same rank-k Nyström approximation $\tilde{K}_{\mathcal{S}}$ as the traditional Nyström formula (Equation 1), it calculates different estimates of the low-dimension basis and $\tilde{K}_{\mathcal{S},d}$. The advantage of the recursive algorithm is that the basis of low-dimension representation is orthogonal, and that $\tilde{K}_{\mathcal{S},d}$ is the best rank-d approximation of $\tilde{K}_{\mathcal{S}}$.

3 Greedy sampling criterion

The recursive nature of the Nyström method can be used to develop an efficient greedy algorithm for sampling columns while calculating the low-rank approximation. The basic idea here is to select, at each iteration, the column that constructs the best rank-1 Nyström approximation of the current residual matrix. The proposed sampling criterion defines the best approximation based on the key observation that the rank-1 Nyström approximation based on the i-th column implicitly projects all data points onto a vector which connects data point i and the origin in the high-dimensional feature space defined by the kernel. Let $X_{:i}$ be the vector of the data point i in the high-dimensional space defined by the kernel. The sampling criterion selects the column $\boldsymbol{\delta} = K_{:l}$ which achieves the least squared error between data points in the feature space and their projections onto $X_{:l}$ (i.e., the reconstruction error). The intuition behind this criterion is that greedily minimizing reconstruction error in the high-dimensional feature space leads to minimizing the difference between kernel matrices in the original and reconstructed spaces.

The sampling criterion at the first iteration can be expressed as the following optimization problem:

$$l = \underset{i}{arg\,min}\ \Sigma_{j=1}^{n} \|X_{:j} - X_{:j_i}\|^2, \tag{5}$$

[2] This can be obtained using Equation (1) when A is a scalar and D is a column vector.

where $X_{:j_i}$ represents a vector in the direction of $X_{:i}$ whose length is the scalar projection of data point j onto $X_{:i}$. Since vector $(X_{:j} - X_{:j_i})$ is orthogonal to $X_{:j_i}$, $\|X_{:j} - X_{:j_i}\|^2 = \|X_{:j}\|^2 - \|X_{:j_i}\|^2$, and the objective function of the sampling criterion is: $\Sigma_{j=1}^{n}\|X_{:j} - X_{:j_i}\|^2 = \Sigma_{j=1}^{n}\|X_{:j}\|^2 - \Sigma_{j=1}^{n}\|X_{:j_i}\|^2$. The term $\Sigma_{j=1}^{n}\|X_{:j}\|^2$ is the sum of the lengths of all data vectors which is a constant for different values of i, and the term $\Sigma_{j=1}^{n}\|X_{:j_i}\|^2$ can be written as: $\Sigma_{j=1}^{n}(X_{:j}^T \frac{X_{:i}}{\|X_{:i}\|})^2 = \|\frac{1}{\sqrt{K_{ii}}}K_{:i}\|^2$. Accordingly, the optimization problem (5) is equivalent to:

$$l = \underset{i}{arg\,max}\, \|\frac{1}{\sqrt{K_{ii}}}K_{:i}\|^2. \tag{6}$$

This means that to obtain the best rank-1 approximation according to the squared error criterion, the proposed algorithm first computes $\|K_{:i}/\sqrt{K_{ii}}\|^2$ for all the columns of K, and then selects the column with the maximum criterion function. The same selection procedure is then applied during the next iterations of the recursive algorithm on the new residual matrices (i.e., $\|E_{:i}/\sqrt{E_{ii}}\|^2$). The computational complexity of the selection criterion is $\mathcal{O}\left(n^2 + n\right)$ per iteration, and it requires $\mathcal{O}\left(n^2\right)$ memory to store the residual of the whole kernel matrix after each iteration.

To reduce the memory requirements of the greedy criterion, the sampling score for each data instance can be calculated in a recursive manner with no need to store and update the whole residual matrix. 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, $\boldsymbol{f} = [\boldsymbol{f}_i]_{i=1..n}$, and $\boldsymbol{g} = [\boldsymbol{g}_i]_{i=1..n}$. It can be shown that $\boldsymbol{f}$ and $\boldsymbol{g}$ can be calculated recursively as follows:

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

where $\circ$ represents the Hadamard product operator, and $\|.\|$ is the ℓ_2 norm. This means that the greedy sampling criterion can be memory-efficient by only maintaining two score variables for each data point, $\boldsymbol{f}_i$ and $\boldsymbol{g}_i$, and updating them at each iteration based on their previous values and the selected columns so far.

In order to reduce the computational complexity, a novel partition-based criterion is proposed, which reduces the number of scalar projections to be calculated at each iteration. The partition-based criterion divides data points into c random groups, and selects the column of K which best represents the centroids of these groups in the high-dimensional feature space. In this case, more efficient update formulas can be developed for $\boldsymbol{f}$, and $\boldsymbol{g}$. The computational complexity of the partition-based criterion is $\mathcal{O}\left(nc + n\right)$ per iteration, It has been empirically shown that using few random groups ($c \ll n$) achieves a very good approximation accuracy.

4 Experiments and results

Experiments have been conducted on six benchmark data sets, where the proposed greedy Nyström methods are compared to five well-known sampling methods: uniform sampling without replacement, adaptive sampling based on the full kernel matrix (**AdaptFull**) [4], adaptive sampling based on a part of the kernel matrix (**AdaptPart**) [5], k-means [7], and the sparse greedy matrix approximation (**SGMA**) algorithm with probabilistic speedup [6]. For each method, the parameters recommended in the corresponding paper were used. Similar to previous work [3, 7], the low-rank approximations obtained by the greedy Nyström algorithm are compared to those obtained by other Nyström methods relative to the best low-rank approximation obtained by singular value decomposition. Linear kernels were used for document data sets (*Reuters-21578*, *Reviews*, and *LA1*), and Gaussian kernels with $\sigma = 10$ for image data sets (*MNIST-4K*, *PIE-20*, and *YaleB-38*). Figure 1 shows the relative accuracy and run times for different methods when calculating the rank-d Nyström approximations $\tilde{K}_{\mathcal{S},d}$.

It can be observed from the results that the greedy Nyström algorithm (**GreedyNyström**) achieves significant improvement in estimating low-rank approximations of a kernel matrix, compared to other sampling-based methods. It also achieves better accuracy than **SGMA** and k**-means** for most data sets. Although the k**-means** achieves better accuracy for some data sets, it obtains much worse

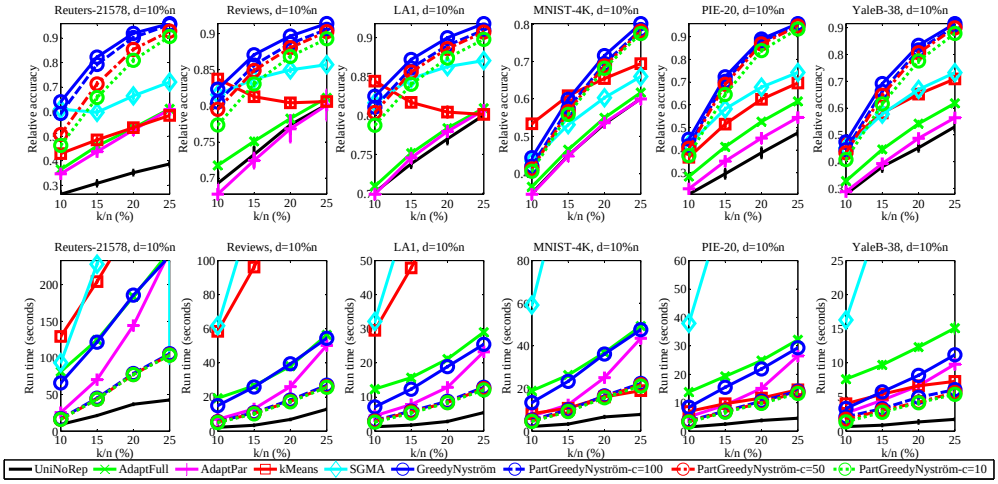

Figure 1: The relative accuracy and run time for the calculation of rank-d approximations $\tilde{K}_{S,d}$.

accuracy for others. This inconsistency could be due to the nature of the k-means algorithm, which might obtain a poor local minimum. **GreedyNyströmn** is more efficient than **SGMA** and **AdaptFull**, but is less efficient than uniform sampling and **AdaptPart**. The latter two methods, however, obtain inferior accuracies. The greedy Nyström algorithm (**GreedyNyström**) is computationally less complex than k**-means** for data sets with large number of features. On the other hand, the alterative partition-based algorithm (**PartGreedyNyström**) for greedy Nyström outperforms all other adaptive and deterministic sampling method in obtaining low-rank approximations, and it requires small overhead in run time compared to uniform sampling. In addition, it is not sensitive to the number of random partitions used.

5 Conclusion

This paper presents a novel recursive algorithm for Nyström approximation and an effective greedy criterion for column selection, which minimizes the reconstruction error in the high-dimensional feature space implicitly defined by the kernel. It has been empirically shown that the proposed algorithm consistently achieves a significant improvement in obtaining low-rank approximations, with minimum overhead in run time.

References

[1] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Proc. of NIPS'01*, pages 682–688. MIT Press, 2001.

[2] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM J. on Computing*, 36(1):158–183, 2007.

[3] S. Kumar, M. Mohri, and A. Talwalkar. Sampling techniques for the Nyström method. In *Proc. of AISTATS'09*, pages 304–311, 2009.

[4] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proc. of ACM-SIAM SODA*, pages 1117–1126. ACM, 2006.

[5] S. Kumar, M. Mohri, and A. Talwalkar. On sampling-based approximate spectral decomposition. In *Proc. of ICML'09*, pages 553–560. ACM, 2009.

[6] A.J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proc. of ICML'00*, pages 911–918. ACM, 2000.

[7] K. Zhang, I.W. Tsang, and J.T. Kwok. Improved Nyström low-rank approximation and error analysis. In *Proc. of ICML'08*, pages 1232–1239. ACM, 2008.