Multi-scale Nyström Method

Woosang Lim[1] woosang.lim@cc.gatech.edu
Rundong Du[1] rdu@gatech.edu
Bo Dai[1] bodai@gatech.edu
Kyomin Jung[2] kjung@snu.ac.kr
Le Song[1] lsong@cc.gatech.edu
Haesun Park[1] hpark@cc.gatech.edu
[1]*Georgia Institute of Technology, Atlanta, GA, USA*
[2]*Seoul National University, Seoul, South Korea*

Abstract

Kernel methods are powerful tools for modeling nonlinear data. However, the amount of computation and memory required for kernel methods becomes the bottleneck when dealing with large-scale problems. In this paper, we propose Nested Nyström Method (NNM) which achieves a delicate balance between the approximation accuracy and computational efficiency by exploiting the multilayer structure and multiple compressions. Even when the size of the kernel matrix is very large, NNM consistently decomposes very small matrices to update the eigen-decomposition of the kernel matrix. We theoretically show that NNM implicitly updates the principal subspace through the multiple layers, and also prove that its corresponding errors of rank-k PSD matrix approximation and kernel PCA (KPCA) are decreased by using additional sublayers before the final layer. Finally, we empirically demonstrate the decreasing property of errors of NNM with the additional sublayers through the experiments on the constructed kernel matrices of real data sets, and show that NNM effectively controls the efficiency both for rank-k PSD matrix approximation and KPCA.

1 Introduction

The scalability of kernel methods is the major bottleneck for applying them to large-scale problems due to the computational and memory cost caused by the large dense kernel matrices. Nyström method is one of the effective methods for accelerating the kernel methods by low-rank approximation of the kernel matrix, $\mathbf{K} \in \mathbb{R}^{n \times n}$. There has been a large body of work that further improves the approximation quality and computational efficiency via adopting various sampling methods [5, 14, 4, 8, 12, 3, 10, 21, 9] and refining approximation formula [7, 5, 11, 19, 12, 17]. Especially, for rank-k spectral decomposition of $\mathbf{K}$, there are two basic rank-k Nyström methods which are *rank-k Standard Nyström Method* (SNM) [5] and *orthogonal Nyström method* (ONM) [7]. Recently, their efficient versions which are *SNM using Randomized SVD* (SNM+Rand.SVD) [11] and *Double Nyström Method* (DNM) [12] were proposed. All these four methods implicitly approximate the first k principal directions $\mathbf{U}_{\mathbf{Y},k}$ of n mapped data points $\mathbf{Y}$ in the feature space to compute the rank-k spectral decomposition of $\mathbf{K} = \mathbf{Y}^\top \mathbf{Y}$ with distinct schemes based on different motivations [12]. Rank-k SNM [5] actually computes the first k principal directions $\mathbf{U}_{\mathbf{S},k}$ of s sample mapped points $\mathbf{S}$ in the feature space, and SNM+Rand.SVD [11] uses randomized SVD to improve efficiency for computing the principal directions of sample mapped points. That is, rank-k SNM and SNM+Rand.SVD approximate $\mathbf{U}_{\mathbf{Y},k}$ via $\mathbf{U}_{\mathbf{S},k}$, which is computed by a particular form. However, it is known that both these two approximations are biased to the sample subspace which is $\mathrm{range}(\mathbf{S})$. On the other hand, the ONM computes the best k approximate principal *orthogonal* direction in the sample subspace $\mathrm{range}(\mathbf{S})$ in the sense to minimize the KPCA reconstruction error [12]. However, such approximation requires extra computation, resulting higher time complexity $O(s^2 n)$ compared to the time complexity of rank-k SNM which is $O(ksn + k^3)$. To further accelerate ONM, DNM [12] uses ONM twice in different scales, so that to compress the sample subspace $\mathrm{range}(\mathbf{S})$ for reducing the dimension of possible solution space for efficient computing of $\mathbf{U}_{\mathbf{Y},k}$. Although the algorithm performs well in practice, there is no analysis about how its rank-k approximation error varies after compression of sample subspace, and it is not clear whether the double scales are enough in terms of the balance between approximation accuracy and computation efficiency.

To achieve a better trade-off between these two factors, we extend DNM to a multi-scale Nyström method. Accelerating the algorithms by exploiting multi-scale structures has been studied for the various methods

OPTML 2017: 10th NIPS Workshop on *Optimization for Machine Learning* (NIPS 2017).

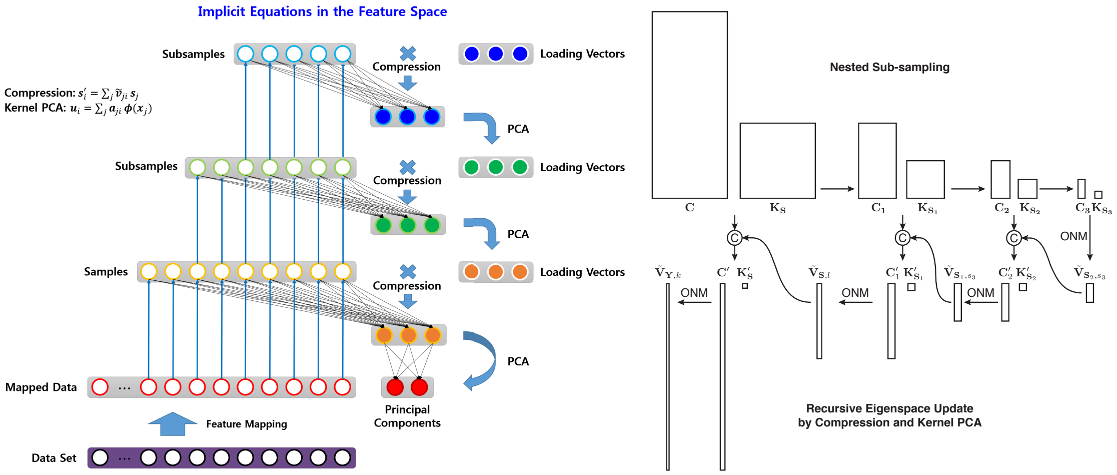

Figure 1: An example of muti-scale structure of NNM with four layers which are three sublayers and the final layer. **Left figure**: Implicitly, NNM with four layers consists of three Fully Connected (FC) neural networks to compute the rank-k spectral decomposition of $\mathbf{K}$. The output of each FC neural network can be considered as approximate principal directions of n mapped data points, and NNM uses them to compute loading vectors of the subsamples/samples on the upper layer. By using the computed loading vectors, we can update the next FC neural network. **Right figure**: Explicitly, NNM samples submatrices of PSD matrix $\mathbf{K}$ according to a nested set of subsamples, and reconstructs eigen-decomposition of square matrix on the upper layer. Right arrow denotes subsampling, and ONM with the arrow denotes eigen-decomposition using ONM. Circle C means a compression of sample matrices with approximate eigenvectors.

Algorithm 1 Nested Nyström Method (NNM)

```
Require: $n \times s$ matrix $\mathbf{C}$ and $s \times s$ matrix $\mathbf{K_S}$, where $\mathbf{C} = \mathbf{Y}^\top \mathbf{S}$ and $\mathbf{K_S} = \mathbf{S}^\top \mathbf{S}$, where $s \ll n$
Ensure: rank-$k$ spectral decomposition of $\mathbf{K}$
1: Subsampling part:
     Subsampling indices from the index set $\mathcal{J}$ of $\mathbf{S}$ s.t. $\mathcal{J} \supseteq \mathcal{J}_1 \supseteq ... \supseteq \mathcal{J}_t$, and corresponding $\mathbf{C} \supseteq \mathbf{K_S} \supseteq \mathbf{C}_1 \supseteq \mathbf{K}_{\mathbf{S}_1} \cdots \supseteq \mathbf{C}_t \supseteq \mathbf{K}_{\mathbf{S}_t}$, where $|\mathcal{J}_i| = s_i$, $s \gg s_1 \gg ... \gg s_t$
2: For $i$-th sublayer from the 1st to the $t$-th sublayer:
     Rank-$s_t$ Nyström method: Compute $\tilde{\mathbf{V}}_{\mathbf{S}_{t-i},s_t}$ of $\mathbf{K}_{\mathbf{S}_{t-i}}$ with $\mathbf{C}'_{t-(i-1)}$ and $\mathbf{K}'_{\mathbf{S}_{t-(i-1)}}$ (optional use ONM)
     Compression: Compress sample matrices $\mathbf{C}_{t-i}$ and $\mathbf{K}_{\mathbf{S}_{t-i}}$ as $\mathbf{C}'_{t-i}$ and $\mathbf{K}'_{\mathbf{S}_{t-i}}$ (Eqn (2))
3: Final layer:
     Run ONM [7] with $\mathbf{C}'$ and $\mathbf{K}'_{\mathbf{S}}$
```

including FEM [6], Bayesian optimization [20] and neural network [1] to solve the nonlinear problems, and there are also a number of applications such as multi-scale stable kernel construction [16], manifold learning [18], dictionary learning [15], and object detection [2, 13]. Among them, feature pyramid networks [13] successfully achieves both efficient and accurate object detection.

Inspired by the multi-scale approximation, we propose a multi-scale Nyström method, Nested Nyström Method (NNM), for both efficient and accurate eigen-decomposition of PSD matrices. NNM has a multilayer structure which consists of t sublayers and the final layer to efficiently and accurately updates the first k principal direction $\mathbf{U}_{\mathbf{Y},k}$ for computing a rank-k spectral decomposition of $\mathbf{K}$. We note that NNM is a general multi-scale framework which can be combined with any other column sampling, and our contribution is orthogonal to the column samplings. Interestingly, it contains t fully connected neural networks in the structure of NNM for the compressions of sample subspaces as described in Fig 1.

2 Nested Nyström Method

The multilayer architecture of NNM is described in Alg 1 and Fig 1, and it consists of the following three parts: subsampling part, rank-s_t Nyström method part, and compression part.

Subsampling part: Given index set $\mathcal{J}$ of s samples and the corresponding sample matrices $\mathbf{S}$ and $\mathbf{K}_\mathbf{S}$, we construct a nested index sets $\mathcal{J} \supseteq \mathcal{J}_1 \supseteq ... \supseteq \mathcal{J}_t$ and the corresponding nested sequence of submatrices as Eqn (1).

$$\mathbf{S} \supseteq \mathbf{S}_1 \supseteq \mathbf{S}_2 \supseteq \cdots \supseteq \mathbf{S}_t, \;\; \mathbf{C} \supseteq \mathbf{K}_\mathbf{S} \supseteq \mathbf{C}_1 \supseteq \mathbf{K}_{\mathbf{S}_1} \supseteq \mathbf{C}_2 \supseteq \mathbf{K}_{\mathbf{S}_2} \supseteq \cdots \supseteq \mathbf{C}_t \supseteq \mathbf{K}_{\mathbf{S}_t}, \tag{1}$$

where $|\mathcal{J}_i| = s_i$, and $s \gg s_1 \gg ... \gg s_t$. Especially, we can understand $(s_{i-1}) \times s_i$ matrix $\mathbf{C}_i$ and $s_i \times s_i$ matrix $\mathbf{K}_{\mathbf{S}_i}$ with implicit equations as $\mathbf{C}_i = \mathbf{S}_{i-1}^\top \mathbf{S}_i$ and $\mathbf{K}_{\mathbf{S}_i} = \mathbf{S}_i^\top \mathbf{S}_i$ for $1 \leq i \leq t$, where $\mathbf{S}_i$ is $d \times s_i$ and $\mathbf{S}_0 = \mathbf{S}$, and $\mathbf{C} = \mathbf{Y}^\top \mathbf{S}$. We will compress $\mathbf{S}_i$, $\mathbf{C}_i$ and $\mathbf{K}_{\mathbf{S}_i}$ as $\mathbf{S}'_i$, $\mathbf{C}'_i$ and $\mathbf{K}'_{\mathbf{S}_i}$ later.

Rank-s_t Nyström method part: In this part, we compute the approximate eigenvectors $\tilde{\mathbf{V}}_{\mathbf{S}_i}$ of $\mathbf{K}_{\mathbf{S}_i}$ by using compressed submatrices $\mathbf{C}'_{\mathbf{S}_{i+1}}$ and $\mathbf{K}'_{\mathbf{S}_{i+1}}$, where $\mathbf{A}_k = \mathbf{U}_{\mathbf{A},k} \mathbf{\Sigma}_{\mathbf{A},k} \mathbf{V}_{\mathbf{A},k}^\top$ denotes the rank-k SVD of a general matrix A, and tilde means their approximations. From the 1st to the $(t-1)$-th sublayer: We compute the first s_t approximate eigenvectors $\tilde{\mathbf{V}}_{\mathbf{S}_i, s_t}$ of $\mathbf{K}_{\mathbf{S}_i}$ by using compressed submatrices $\mathbf{C}'_{i+1}$ and $\mathbf{K}'_{\mathbf{S}_{i+1}}$ on the $(t-i)$-th layer, where $i \in \{1, 2, ..., (t-1)\}$ and $\mathbf{C}'_t = \mathbf{C}_t$ and $\mathbf{K}'_{\mathbf{S}_t} = \mathbf{K}_{\mathbf{S}_t}$. On the t-th sublayer: We compute the first s_t approximate eigenvectors $\tilde{\mathbf{V}}_{\mathbf{S}, s_t}$ of $\mathbf{K}_\mathbf{S}$ by using $\mathbf{C}'_1$ and $\mathbf{K}'_{\mathbf{S}_1}$, and select $\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ from $\tilde{\mathbf{V}}_{\mathbf{S},s_t}$, where $s_t \geq \ell \geq k$.

Compression part: In this part, we compress sample matrices by using the approximate eigenvectors. We compress sample matrices $\mathbf{C}_i$ and $\mathbf{K}_{\mathbf{S}_i}$ by using $\tilde{\mathbf{V}}_{\mathbf{S}_i, s_t}$ as

$$\mathbf{C}'_i = \mathbf{C}_i \tilde{\mathbf{V}}_{\mathbf{S}_i, s_t}, \;\; \mathbf{K}'_{\mathbf{S}_i} = (\tilde{\mathbf{V}}_{\mathbf{S}_i, s_t})^\top \mathbf{K}_{\mathbf{S}_i} \tilde{\mathbf{V}}_{\mathbf{S}_i, s_t}, \;\; \mathbf{C}' = \mathbf{C} \tilde{\mathbf{V}}_{\mathbf{S},\ell}, \;\; \mathbf{K}'_\mathbf{S} = (\tilde{\mathbf{V}}_{\mathbf{S},\ell})^\top \mathbf{K}_\mathbf{S} \tilde{\mathbf{V}}_{\mathbf{S},\ell}, \tag{2}$$

where $\tilde{\mathbf{V}}_{\mathbf{S}_i, s_t}$ is computed at $(t-i)$-th layer with $i \in \{1, 2, ..., (t-1)\}$, and we compress sample matrices $\mathbf{C}$ and $\mathbf{K}_\mathbf{S}$ by using $\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ with $k \leq \ell \leq s_t$. We can connect the compression of sample matrices to the compression of sample subspace with implicit equations

$$\mathbf{C}'_i = \mathbf{S}_{i-1}^\top \mathbf{S}'_i, \;\; \mathbf{C}' = \mathbf{Y}^\top \mathbf{S}', \;\; \mathbf{K}'_{\mathbf{S}_i} = \mathbf{S}_i'^\top \mathbf{S}'_i, \;\; \mathbf{K}'_\mathbf{S} = \mathbf{S}'^\top \mathbf{S}', \tag{3}$$

where $\mathbf{S}'_i = \mathbf{S}_i \tilde{\mathbf{V}}_{\mathbf{S}_i, s_t}$, $\mathbf{S}' = \mathbf{S} \tilde{\mathbf{V}}_{\mathbf{S},\ell}$, $i \in \{1, 2, ..., (t-1)\}$, and $\mathbf{S}_0 = \mathbf{S}$.

Based on Eqn (3), it can be interpreted that the sample subspace $\mathrm{range}(\mathbf{S}_i)$ is compressed into a smaller dimensional subspace $\mathrm{range}(\mathbf{S}'_i)$, where $i \in 0, 1, ..., (t-1)$ and $\mathbf{S}_0 = \mathbf{S}$. If we use ONM for rank-s_t Nyström method part in NNM, and set the nested sequence of subsamples with $\sum_{j=1}^t s_j = O(s)$, where $s \gg s_1 \gg ... \gg s_t \geq \ell \geq k$. Then, the total time and space complexities of NNM are $O(\ell s n + s_t s_1 s)$ and $O(sn)$, respectively. A large portion of the total time complexity $O(\ell s n + s_t s_1 s)$ is $O(\ell s n)$ which corresponds to the simple matrix multiplications in the compression parts. Furthermore, by extending the multilayer structure of NNM, we can efficiently update the spectral decomposition with additional samples and data points.

2.1 Error Analysis of NNM

NNM efficiently and accurately updates the compressed sample matrix $\mathbf{S}'_i$ so that $\mathrm{range}(\mathbf{S}'_i)$ closely approximates the true principal subspace based on Eqn (3) until the final layer. That is, we want to compute $\mathbf{S}'$ s.t. $\mathrm{range}(\mathbf{U}_k) \subset \mathrm{range}(\mathbf{S}')$, and we can give the implicit representation of the principal subspace as $\mathrm{range}(\mathbf{U}_k) = \mathrm{range}(\mathbf{U}_k \mathbf{\Sigma}_{\mathbf{Y},k}) = \mathrm{range}(\mathbf{Y}\mathbf{V}_{\mathbf{Y},k})$. Similarly, we can give the implicit representations of compressed sample subspaces, and Lem 1 formally provide them.

Lemma 1 *Given the mutilayer Nyström structure of NNM with t sublayers, NNM computes* $\mathbf{S}'_i = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s_t}$ *on the* $(t-i)$*-th layer, and* $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ *with* $(\tilde{\mathbf{V}}_{\mathbf{Y},s_t})^\top \tilde{\mathbf{V}}_{\mathbf{Y},s_t} = \mathbf{I}$ *and* $(\tilde{\mathbf{V}}_{\mathbf{Y},\ell})^\top \tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{I}$.

Now we provide Lem 2 which provides the differences between the optimal error and the error of NNM both for KPCA and rank-k PSD kernel matrix approximation.

Lemma 2 *Suppose that* $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ *is the compressed samples as an input of the final layer of NNM, where* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}^\top \tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{I}$ *and* $k \leq \ell \leq s_t$*. Then, the differences between approximation error of NNM and the optimal errors for KPCA and rank-k PSD matrix approximation are bounded by constant times of* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$*, where* $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ *is any submatrix consisting of* k *columns of* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$*, and* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k}) = \mathrm{tr}(\mathbf{V}_{\mathbf{Y},k}^\top \mathbf{K} \mathbf{V}_{\mathbf{Y},k}) - \mathrm{tr}(\tilde{\mathbf{V}}_{\mathbf{Y},k}^\top \mathbf{K} \tilde{\mathbf{V}}_{\mathbf{Y},k})$ *is the sum of errors of eigenvalues from* $\tilde{\mathbf{V}}_{\mathbf{Y},k}$*.*

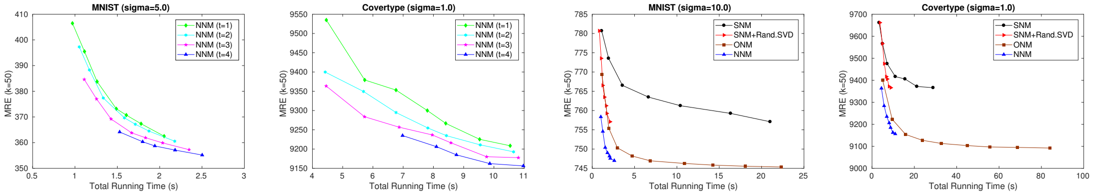

Figure 2: Two left figures: Performance comparison for rank-k kernel matrix approximation among the NNM with 1, 2, 3, 4 sublayers. NNM $(t = 4)$ is more efficient than NNM $(t = 1, 2, 3)$. Two right figures: Comparison of $\mathrm{MRE}(\tilde{\mathbf{K}}_k)$ for rank-k kernel matrix approximation among the four representative methods with: SNM [5], SNM+Rand.SVD [11], ONM [7], NNM (ours). NNM is more efficient than other state-of-the art Nyström methods given the short time.

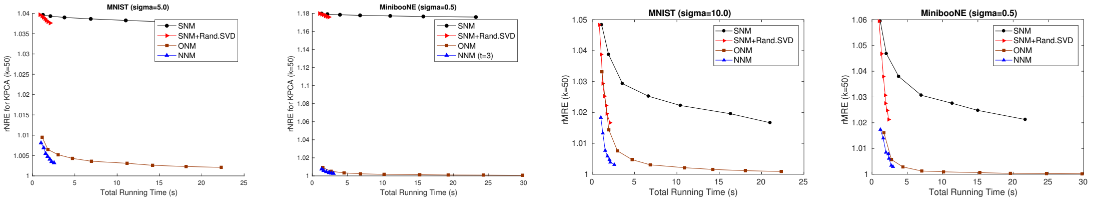

Figure 3: Comparison of convergence to the optimal error with $\mathrm{rNRE}(\tilde{\mathbf{K}}_k)$ for KPCA (two left figures) and the corresponding rank-k kernel matrix approximation (two right figures). It shows that errors of both KPCA and rank-k kernel matrix approximation rapidly decrease compared to errors of other Nyström methods.

Lem 2 states that if $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ goes to 0, then the approximation errors of NNM go to the optimal errors both for KPCA and rank-k PSD matrix approximation. Since reducing $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ is important, we need to show how $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ varies through the sublayers.

Now, we provide our main theoretical result Thm 1 which states that the quality of compressed input at the final layer is important, and we can increase accuracy by using more sublayers.

Theorem 1 *Suppose that we use ONM for the kernel PCA parts in the sublayers. As we use additional sublayers,*$\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *decreases, and the upper error bounds of NNM in Lem 2 decrease.*

3 Experiments

In this section, we present experimental results that demonstrate our theoretical work. We use 3 real data sets which are MNIST, MiniBooNE, and Covertype. We compare rank-k Nyström methods to the rank-k kernel matrix approximation and KPCA. The three error measures which we used are *matrix reconstruction error* $(\mathrm{MRE}(\tilde{\mathbf{K}}_k) = \|\mathbf{K} - \tilde{\mathbf{K}}_k\|_{\mathrm{F}})$, *relative matrix reconstruction error* $(\mathrm{rMRE}(\tilde{\mathbf{K}}_k) = \frac{\|\mathbf{K}-\tilde{\mathbf{K}}_k\|_{\mathrm{F}}}{\|\mathbf{K}-\mathbf{K}_k\|_{\mathrm{F}}} \in [1, \infty))$, and *relative KPCA reconstruction error* $(\mathrm{rNRE}(\tilde{\mathbf{U}}_k) = \frac{\mathrm{NRE}(\tilde{\mathbf{U}}_k)}{\mathrm{NRE}(\mathbf{U}_k)} \in [1, \infty))$, where $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ consists of the first k approximate principal directions which are implicitly computed by KPCA, $\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) = \|\mathbf{Y} - \tilde{\mathbf{U}}_{\mathbf{Y},k}\tilde{\mathbf{U}}_{\mathbf{Y},k}^{\top}\mathbf{Y}\|_{\mathrm{F}}/\|\mathbf{Y}\|_{\mathrm{F}}$ is the normalized reconstruction error (NRE) of KPCA, and $\|\mathbf{K} - \mathbf{K}_k\|_{\mathrm{F}}$ and $\mathrm{NRE}(\mathbf{U}_k)$ are the optimal error which comes from SVD. The optimum of rMRE and rNRE is 1. To construct PSD matrix $\mathbf{K}$, we use RBF kernel which is defined as $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{2\sigma^2}\right)$, where σ is a kernel parameter. We abbreviate NNM with i sublayers to NNM $(t = i)$ for convenience, and DNM [12] is the same with NNM $(t = 1)$.

Fig 2 demonstrates that the error of NNM decreases and its efficiency can be improved as we use additional sublayers regardless of data sets, and also shows that the errors of NNM is smaller than errors of other state-of-the art Nyström methods within the same short time. Fig 3 shows that the errors of NNM both for KPCA and rank-k kernel matrix approximation rapidly decrease compared to other rank-k Nyström methods.

References

[1] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[2] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Proceedings of ECCV*, pages 354–370. Springer, 2016.

[3] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2:225–247, 2006.

[4] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13:3475–3506, 2012.

[5] Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

[6] Yalchin Efendiev, Thomas Y Hou, Victor Ginting, et al. Multiscale finite element methods for nonlinear problems and their applications. *Communications in Mathematical Sciences*, 2(4):553–589, 2004.

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

[8] Alex Gittens and Michael W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *Proceedings of ICML*, 2013.

[9] Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Fast prediction for large-scale kernel machines. In *Proceeding of NIPS*, pages 3689–3697, 2014.

[10] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*, 98888:981–1006, 2012.

[11] Mu Li, James Tin-Yau Kwok, and Baoliang Lü. Making large-scale nyström approximation possible. In *ICML 2010-Proceedings, 27th International Conference on Machine Learning*, page 631, 2010.

[12] Woosang Lim, Minhwan Kim, Haesun Park, and Kyomin Jung. Double Nyström method: An efficient and accurate Nyström scheme for large-scale data sets. In *Proceedings of ICML*, pages 1367–1375, 2015.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of CVPR*, 2017.

[14] Michael W. Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[15] Francesca Petralia, Joshua T Vogelstein, and David B Dunson. Multiscale dictionary learning for estimating conditional distributions. In *Proceedings of NIPS*, pages 1797–1805, 2013.

[16] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of CVPR*, pages 4741–4748, 2015.

[17] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Proceedings of NIPS*, pages 1657–1665, 2015.

[18] Chang Wang and Sridhar Mahadevan. Multiscale manifold learning. In *Proceedings of AAAI*, 2013.

[19] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14(1):2729–2769, 2013.

[20] Ziyu Wang, Babak Shakibi, Lin Jin, and Nando Freitas. Bayesian multi-scale optimistic optimization. In *Proceedings of AISTATS*, pages 1005–1014, 2014.

[21] Kai Zhang and James T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, pages 1576–1587, 2010.