Memory Efficient Kernel Approximation

Si Si
Department of Computer Science
University of Texas at Austin

ICML
Beijing, China
June 23, 2014

Joint work with Cho-Jui Hsieh and Inderjit S. Dhillon

Outline

- Background
- Motivation—Low-Rank vs. Block Structure
- Memory Efficient Kernel Approximation(MEKA)
  - Captures both block and low-rank structures
  - Main algorithm
  - Analysis
- Experimental Results
  - Kernel approximation
  - Kernel ridge regression

Background

- Kernel machines: Kernel SVM, Kernel regression, Kernel PCA, etc.
- Kernel functions: $K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^T \varphi(\mathbf{y})$.
- Shift-invariant kernels: $K(\mathbf{x}, \mathbf{y}) = f(\eta(\mathbf{x} - \mathbf{y}))$.
- Examples:
  - Gaussian kernel: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|_2^2}$;
  - Laplacian Kernel: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|_1}$.
- Challenge for computing and storing the kernel matrix:
  - Space: $O(n^2)$;
  - Time: $O(dn^2)$.
- MNIST2M dataset(containing 2 million data points):
  - Space for storing Gaussian kernel: 16TBytes;
  - Time for computing Gaussian kernel: more than 10 hours.

Low-rank Approximation

- Popular Solution: low-rank approximation to G with $G_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$,

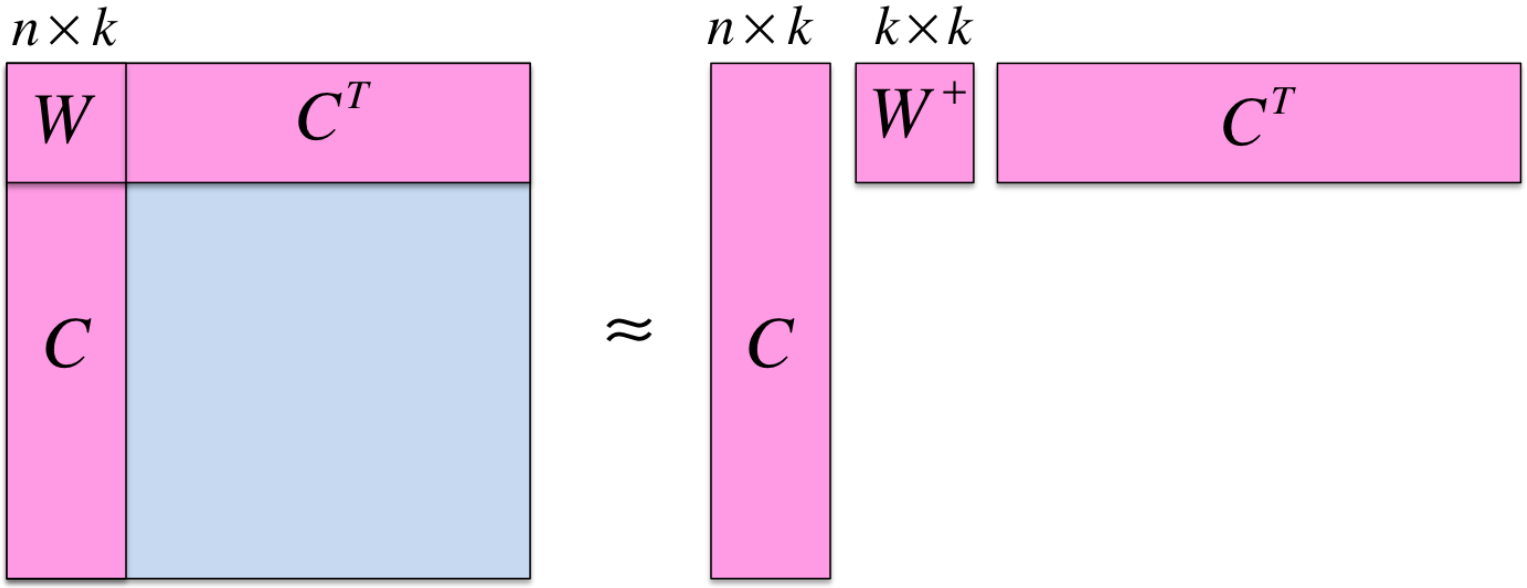
$$G \approx \tilde{G} = XX^T,$$

  where G is $n \times n$ and X is $n \times k$.
- Benefits of using X:
  - Memory: $O(nk)$;
  - Kernel machines change to linear problems.
- State-of-the-art approaches:
  - Various Nyström kernel approximation methods(Drineas and Mahoney, 2005) (Zhang et al. 2008) (Kumar et al. 2009);
  - Random Kitchen Sinks (Rahimi and Recht, 2007);
  - Fastfood with Hadamard features(Le et al. 2013), etc.

Standard Nyström Kernel Approximation

- Goal: rank-k approximation $\tilde{G}$ to G.

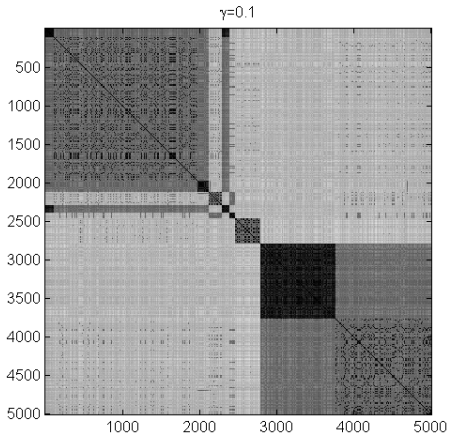
$$G \approx \tilde{G} = CW^{\dagger}C^{T} = XX^{T}.$$



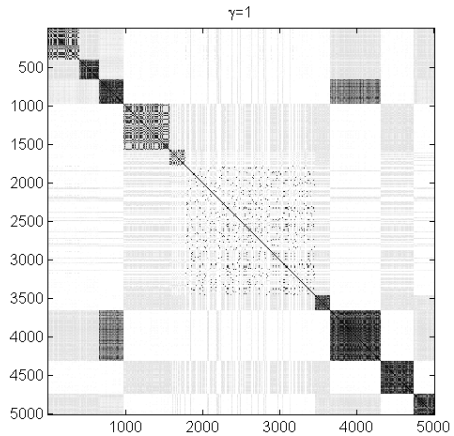
- In practice, oversample a few more columns.
- Running time: $O(nkd + k^3)$.
- Memory: $O(nk)$.

Motivations

- Low rank or block Structure?
- Take the Gaussian kernel as an example: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}$.
  - $\gamma \to 0$, $K(\mathbf{x}, \mathbf{y}) \to 1$, low-rank structure.
  - $\gamma \to \infty$, $K(\mathbf{x}, \mathbf{y}) \to 0$, $K(\mathbf{x}, \mathbf{x}) \to 1$, full-rank, block structure.



(a) $\gamma = 0.1$

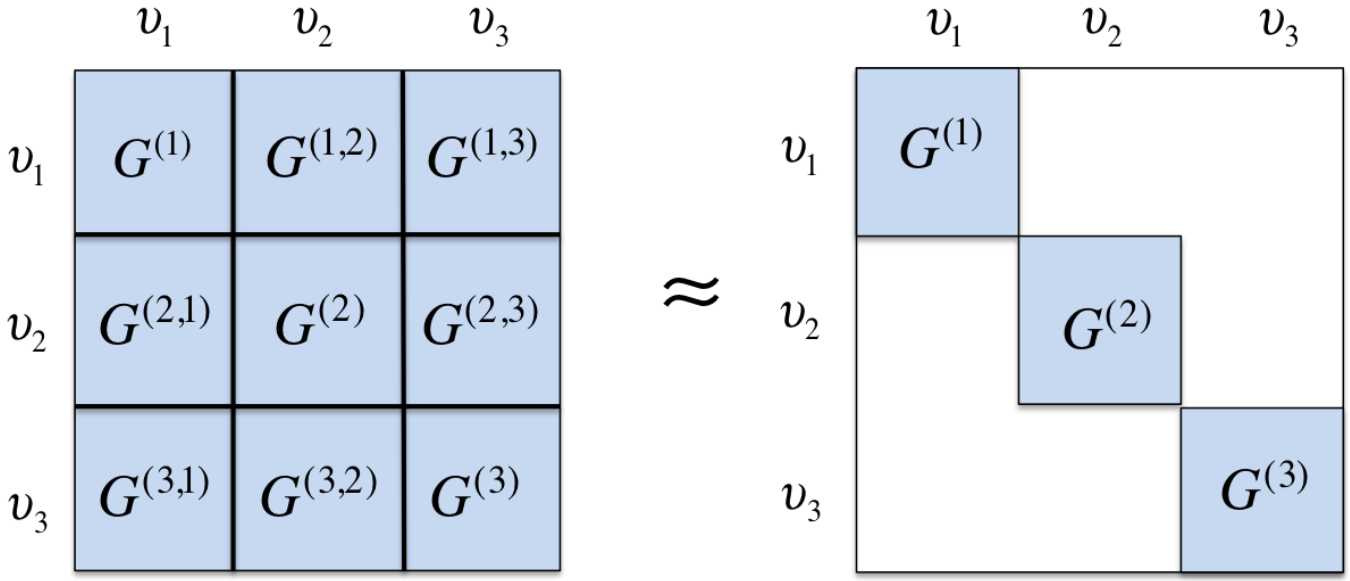


(b) $\gamma = 1$

- With appropriate partitions, each block shows low-rank structure.

Block Kernel Approximation(BKA)

- Block Kernel Approximation (BKA):
  1. Partition n data points into c clusters, $\{\mathcal{V}_s\}_{s=1}^{c}$.
  2. Compute diagonal blocks formed by points in $\{\mathcal{V}_s\}_{s=1}^{c}$.



- The error comes from the off-diagonal blocks $G^{(s,t)}(s \neq t)$:

$$\|G - \tilde{G}\|_F^2 = \sum_{i,j} K(\mathbf{x}_i, \mathbf{x}_j)^2 - \sum_{s=1}^{c} \sum_{i,j \in \mathcal{V}_s} K(\mathbf{x}_i, \mathbf{x}_j)^2.$$

Clustering— Two Objectives

- Minimize the approximation error is to maximize:

$$D(\{\mathcal{V}_s\}_{s=1}^c) = \sum_{s=1}^{c} \sum_{i,j \in \mathcal{V}_s} K(\mathbf{x}_i, \mathbf{x}_j)^2.$$

  (1) needs to compute all G_{ij}; (2) all data points go to one cluster.

- Spectral clustering on kernel, maximize:

$$D^{\text{kernel}}(\{\mathcal{V}_s\}_{s=1}^c) = \sum_{s=1}^{c} \frac{1}{|\mathcal{V}_s|} \sum_{i,j \in \mathcal{V}_s} K(\mathbf{x}_i, \mathbf{x}_j)^2.$$

  (1) needs the computation of all G_{ij}; (2) time $O(n^2)$; memory $O(n^2)$.

- kmeans in the input space, minimize:

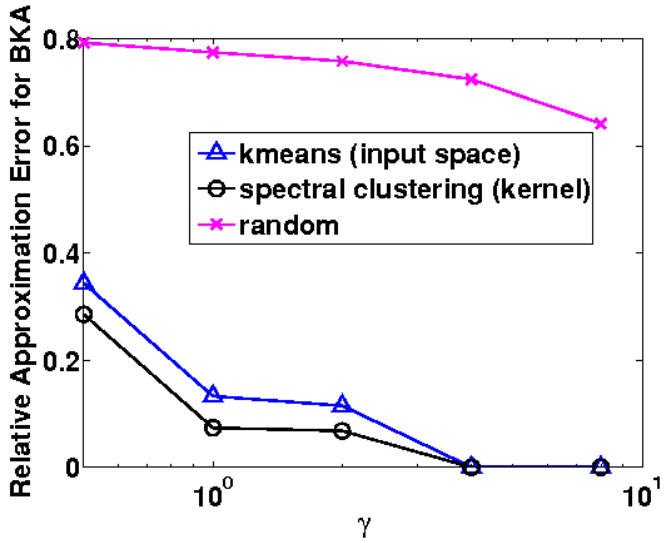
$$D^{\text{kmeans}}(\{\mathcal{V}_s\}_{s=1}^c) = \sum_{s=1}^{c} \sum_{i \in \mathcal{V}_s} \|\mathbf{x}_i - \mathbf{m}_s\|_2^2, \mathbf{m}_s = (\sum_{i \in \mathcal{V}_s} \mathbf{x}_i)/|\mathcal{V}_s|.$$

  (1) no need to compute any G_{ij}; (2) time $O(nd)$.

Error Analysis for Clustering

- Theorem 1: k-means and spectral clustering on kernel:
  For any $n \times n$ shift-invariant kernel that satisfies certain assumptions,

$$D^{\text{kernel}}(\{\mathcal{V}_s\}_{s=1}^c) \geq \bar{C} - \eta^2 R^2 D^{\text{kmeans}}(\{\mathcal{V}_s\}_{s=1}^c).$$



- k-means in the input space performs similar to spectral clustering.
- Much more efficient, time complexity is $O(nd)$.

Low-rank Structure of Each Block

- Drawbacks of BKA:
  - Ignores all off-diagonal blocks $G^{(s,t)}(s \neq t)$;
  - Expensive to compute and store diagonal blocks $G^{(s)}$.
- Theorem 2: Low-rank structure of each block $G^{(s,t)}$:

$$\|G^{(s,t)} - G_k^{(s,t)}\|_F \leq 4Ck^{-1/d}\sqrt{|\mathcal{V}_s||\mathcal{V}_t|}\min(r_s, r_t),$$

where $G_k^{(s,t)}$ is the best rank-k approximation to $G^{(s,t)}$; r_s and $|\mathcal{V}_s|$ is the radius and the size of the s-th cluster respectively.

Table : k-means(avg. rank 13).

16	14	13	7	7
14	29	13	9	9
13	13	20	10	10
7	9	10	29	11
7	9	10	11	28

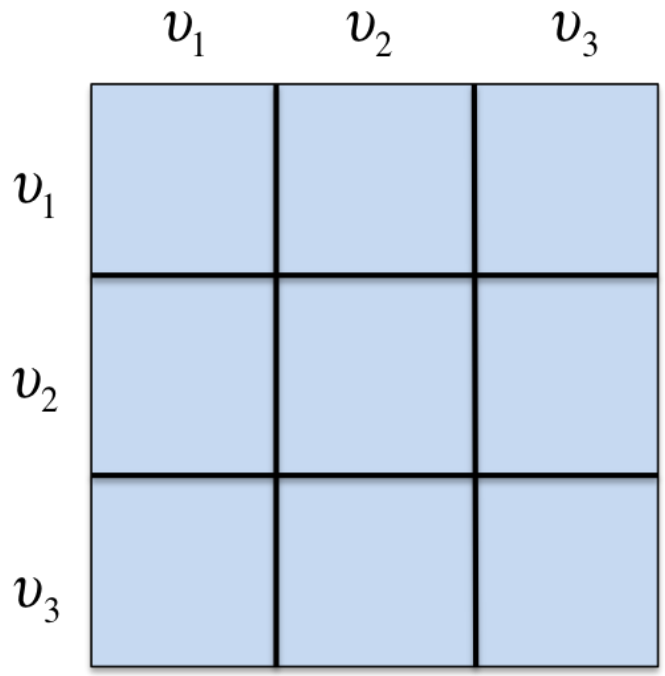
Table : random partition(avg. rank 67).

139	99	101	44	45
99	116	86	43	44
101	86	131	46	47
44	43	46	47	45
45	44	47	45	49

Memory Efficient Kernel Approximation(MEKA)

Three steps:

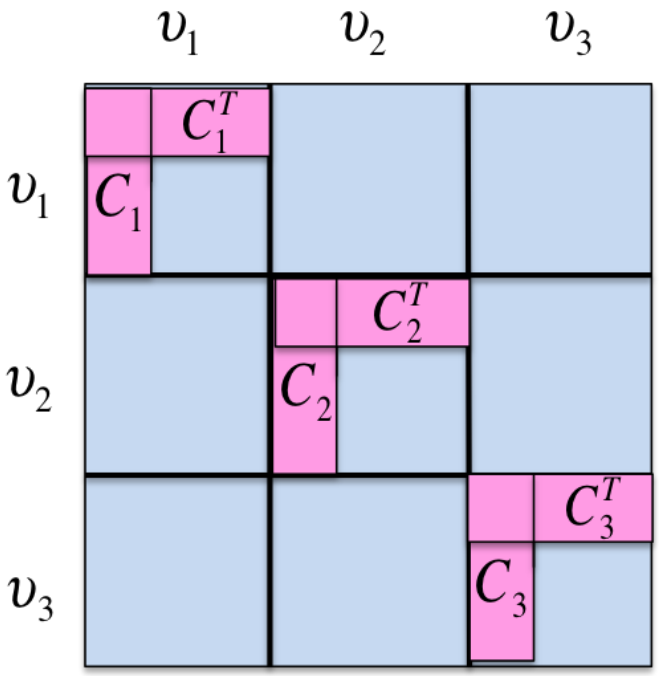
1. k-means on n data points to generate c clusters $\{\mathcal{V}_s\}_{s=1}^{c}$ and c^2 blocks in G.
2. Form rank-k approximation for each of the c diagonal blocks.
3. Form low-rank 'basis' X_s for $G^{(s)}$ and use X_s and X_t and to approximate $G^{(s,t)}(s \neq t)$.



Memory Efficient Kernel Approximation(MEKA)

Three steps:

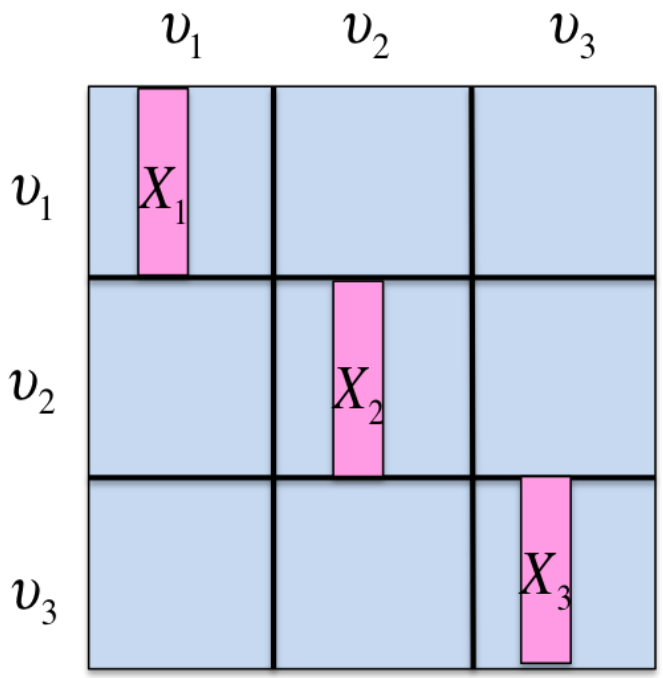
1. k-means on n data points to generate c clusters $\{\mathcal{V}_s\}_{s=1}^{c}$ and c^2 blocks in G.
2. Form rank-k approximation for each of the c diagonal blocks.
3. Form low-rank 'basis' X_s for $G^{(s)}$ and use X_s and X_t to approximate $G^{(s,t)}(s \neq t)$.



Memory Efficient Kernel Approximation(MEKA)

Three steps:

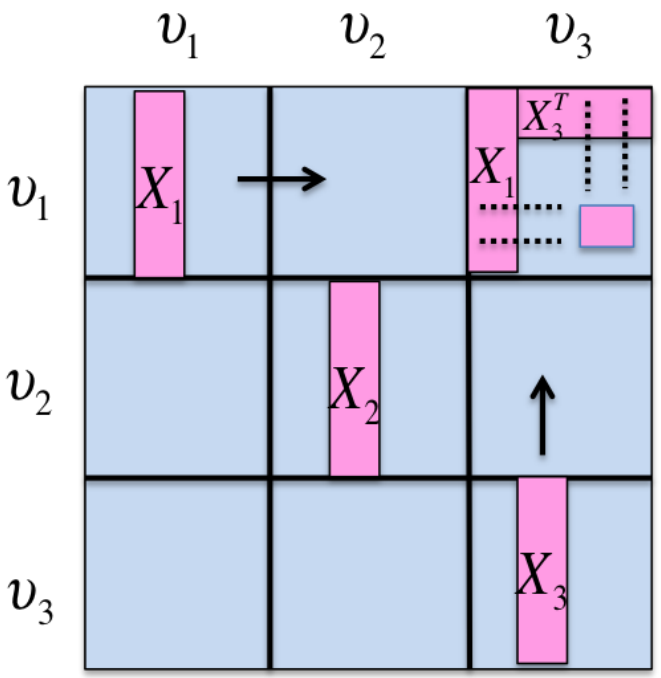
1. k-means on n data points to generate c clusters $\{\mathcal{V}_s\}_{s=1}^{c}$ and c^2 blocks in G.
2. Form rank-k approximation for each of the c diagonal blocks.
3. Form low-rank 'basis' X_s for $G^{(s)}$ and use X_s and X_t to approximate $G^{(s,t)}(s \neq t)$.



Memory Efficient Kernel Approximation(MEKA)

Three steps:

1. k-means on n data points to generate c clusters $\{\mathcal{V}_s\}_{s=1}^{c}$ and c^2 blocks in G.
2. Form rank-k approximation for each of the c diagonal blocks.
3. Form low-rank 'basis' X_s for $G^{(s)}$ and use X_s and X_t to approximate $G^{(s,t)}(s \neq t)$. $\underset{L_{st}}{\text{minimize}} \| \hat{G}^{(s,t)} - \hat{X}_s L_{st} \hat{X}_t^T \|_F$.



Memory Efficient Kernel Approximation(MEKA)

Three steps:

1. k-means on n data points to generate c clusters $\{\mathcal{V}_s\}_{s=1}^{c}$ and c^2 blocks in G.
2. Form rank-k approximation for each of the c diagonal blocks.
3. Form low-rank 'basis' X_s for $G^{(s)}$ and use X_s and X_t to approximate $G^{(s,t)}(s \neq t)$.

$$G \approx \tilde{G} = \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & X_c \end{bmatrix} \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1c} \\ L_{21} & L_{22} & \cdots & L_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ L_{11} & L_{12} & \cdots & L_{cc} \end{bmatrix} \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & X_c \end{bmatrix}^{\mathsf{T}}.$$

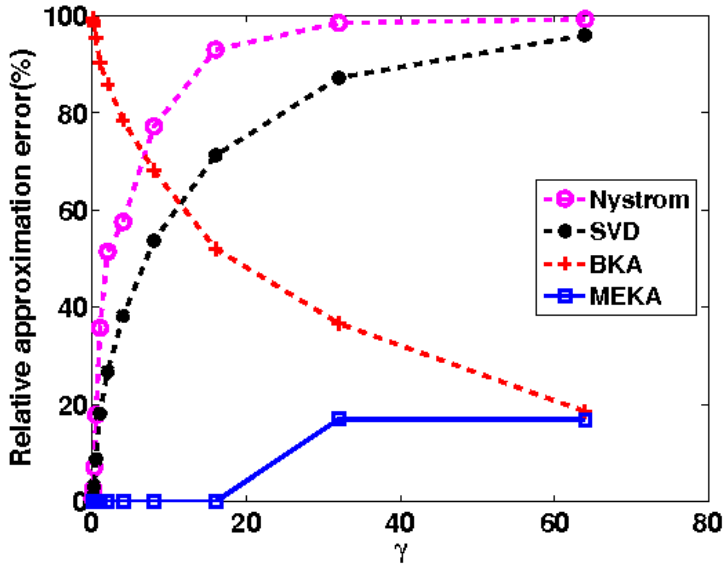
$$\text{Rank-}k \text{ approximation: } G^{(s,t)} \approx X_s L_{st} X_t^T.$$

$$\text{Rank-}ck \text{ approximation: } G \approx XLX^T.$$

O(nk) space to generate rank-ck approximation

Comparison of MEKA with Other Methods

- Comparison of different kernel approximation methods on various γ:



- Memory and time analysis of MEKA:

Method	Storage	Rank	Time Complexity
SVD	$O(cnk)$	ck	$O(n^3 + n^2 d)$
RKS	$O(cnk)$	ck	$O(cnkd)$
Nyström	$O(cnk)$	ck	$O(cnkd + (ck)^3)$
MEKA	$O(nk)$	ck	$O(nkd + ck^3) + T_L + T_C$

Analysis for MEKA

- Theorem 3: Approximation error bound for $\|\tilde{G} - G\|_2$ and $\|\tilde{G} - G\|_F$

$$\|G - \tilde{G}\|_2 \leq \|G - G_{ck}\|_2 + \frac{1}{\sqrt{c}} \frac{2n}{\sqrt{m}} G_{max}(1+\theta) + 2\|\Delta\|_2,$$

$$\|G - \tilde{G}\|_F \leq \|G - G_{ck}\|_F + \left(\frac{64k}{m}\right)^{\frac{1}{4}} n G_{max}(1+\theta)^{\frac{1}{2}} + 2\|\Delta\|_F.$$

- If $\|G - G_k\|_2 - \|G - G_{ck}\|_2 \geq 2\|\Delta\|_2$, then

$$\|G - \tilde{G}\|_2 \leq \|G - G_k\|_2 + \frac{1}{\sqrt{c}} \frac{2n}{\sqrt{m}} G_{max}(1+\theta).$$

The second term is only $\frac{1}{\sqrt{c}}$ of that in the spectral norm error bound for standard Nyström to obtain the rank-k approximation(Kumar et al.2009).

Experimental Results

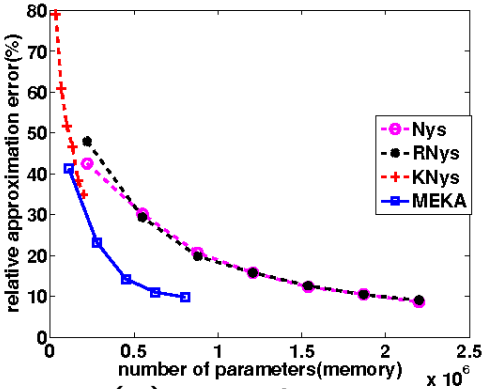
- Methods compared in the experiments:
  1. Standard Nyström (Nys)(Drineas and Mahoney, 2005);
  2. Kmeans Nyström (KNys)(Zhang et al. 2008);
  3. Random Kitchen Sinks(RKS)(Rahimi and Recht, 2007);
  4. Fastfood with "Hadamard features"(fastfood)(Le et al. 2013);
  5. Ensemble Nyström (ENys)(Kumar et al. 2009);
  6. Nyström using randomized SVD (RNys)(Li et al. 2010).
- Data set statistics (n: number of samples):

Dataset	n	d	Dataset	n	d
wine	6,497	11	census	22,784	137
cpusmall	8,192	12	ijcnn	49,990	22
pendigit	10,992	16	covtype	581,012	54
cadata	20,640	8	MNIST2M	2,000,000	784

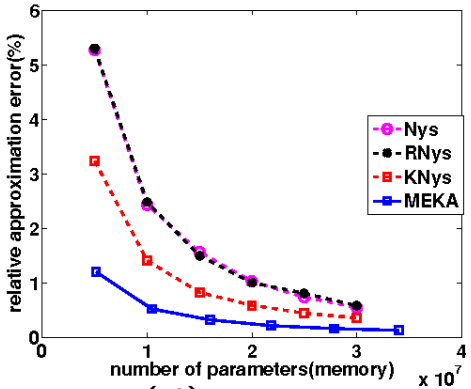
Kernel Approximation Results

Relative kernel approximation error $\|G - \tilde{G}\|_F / \|G\|_F$.

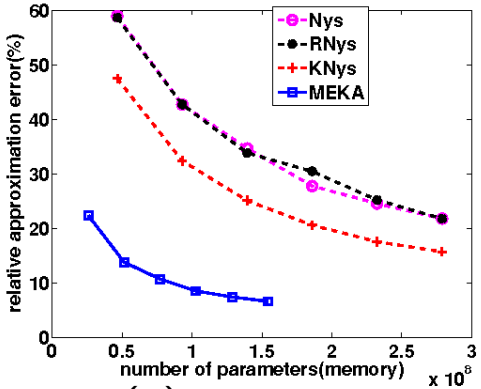
- Memory vs. kernel approximation error:



(c) pendigit

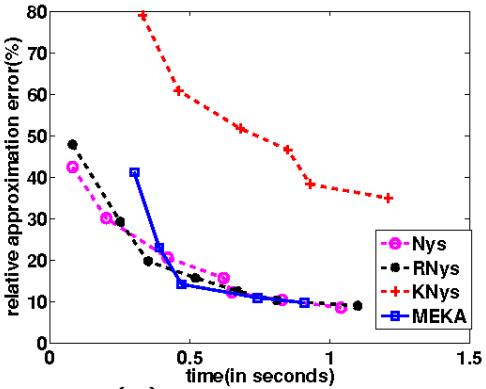


(d) ijcnn

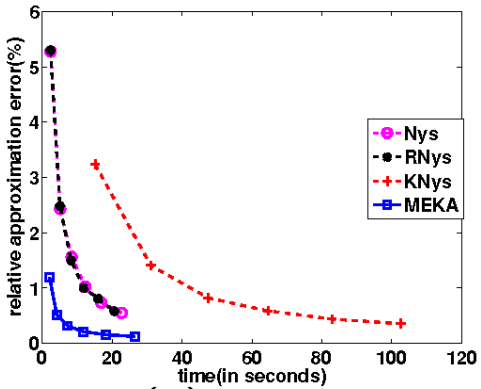


(e) covtype

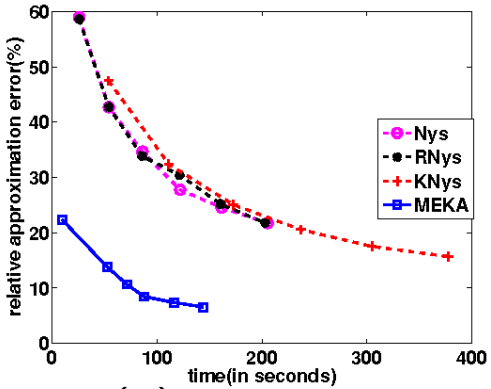
- Time vs. kernel approximation error:



(f) pendigit



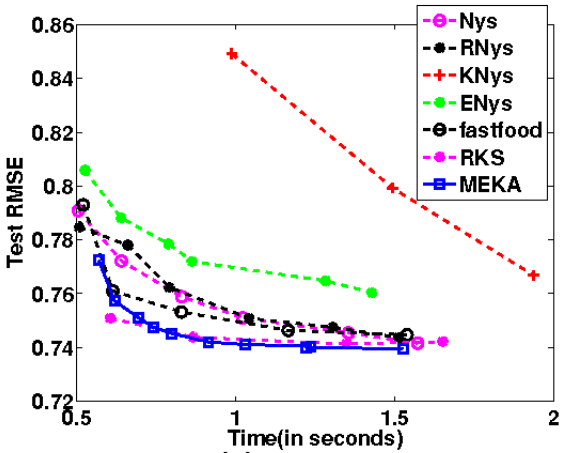
(g) ijcnn



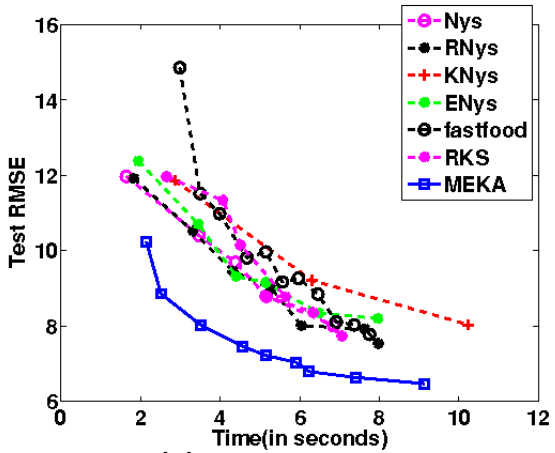
(h) covtype

Kernel Ridge Regression Results

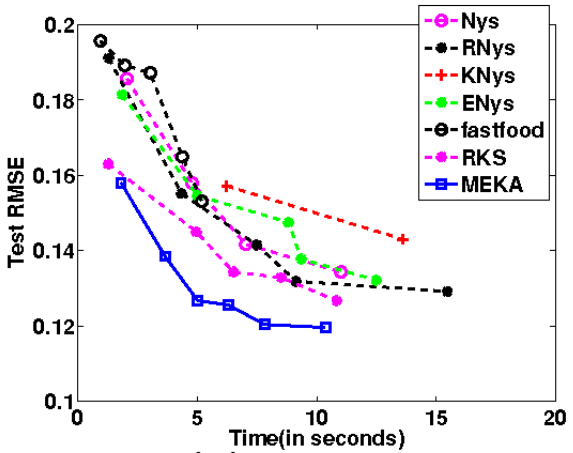
Time vs. kernel ridge regression error(test RMSE):



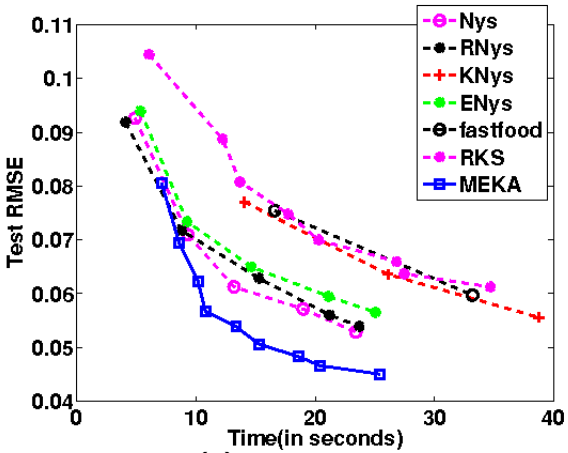
(i) wine



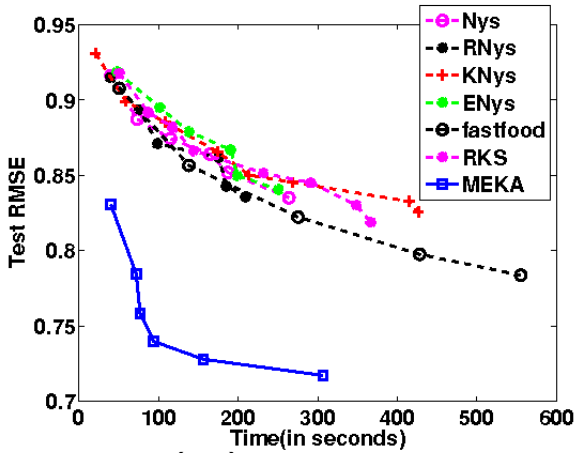
(j) cpusmall



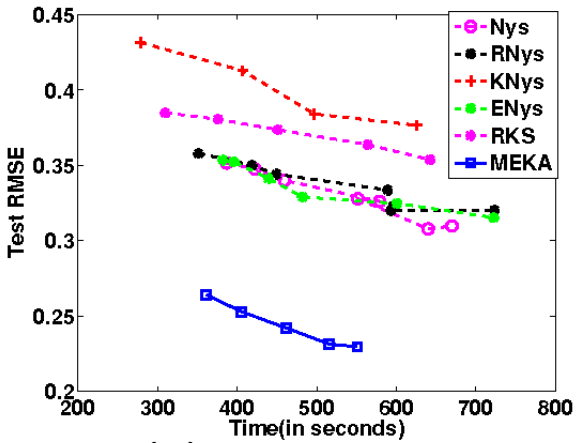
(k) cadata



(l) census



(m) covtype



(n) MNIST2M

Conclusions

- Observation: kernel matrices have block as well as low-rank structure.
- MEKA: a memory efficient and fast kernel approximation approach.
  - k-means to capture block structure.
  - Low-rank approximation in each block to exploit low-rank structure.
- Theoretical guarantees.
- Experimental results on real-world datasets.
- Code is available at: www.cs.utexas.edu/~ssi/meka/

References

[1] Si, S., Hsieh, C.-J., and Dhillon, I. S. *Memory Efficient Kernel Approximation.* ICML, 2014.

[2] Drineas, P. and Mahoney, M. W. *On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning.* JMLR, 2005.

[3] Kumar, S., Mohri, M., and Talwalkar, A. *Ensemble Nyström Methods.* NIPS, 2009.

[4] Le, Q. V., Sarlos, T., and Smola, A. J. *Fastfood – Approximating Kernel Expansions in Loglinear Time.* ICML, 2013.

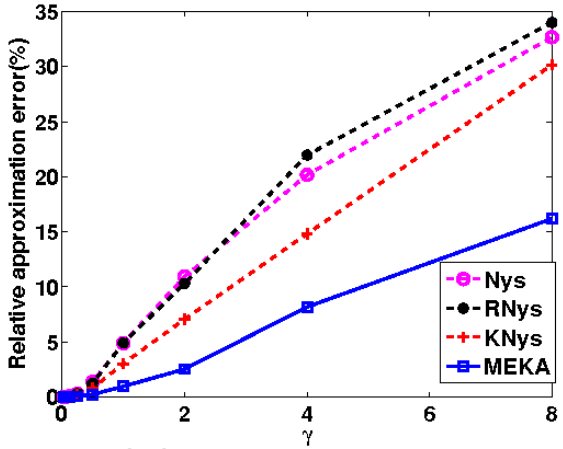
[5] Li, Mu, Kwok, J. T., and Lu, Bao-Liang. *Making Large-Scale Nyström Approximation Possible.* ICML, 2010.

[6] Rahimi, A. and Recht, Benjamin. *Random Features for Large-scale Kernel Machines.* NIPS, 2007.

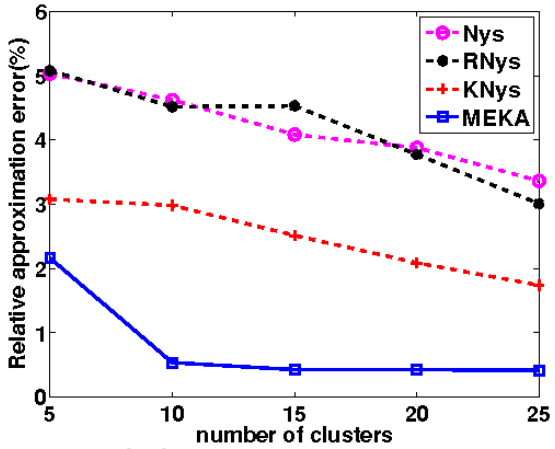
[7] Zhang, K, Tsang, I. W., and Kwok, J. T. *Improved Nyström Low Rank Approximation and Error Analysis.* ICML, 2008.

Robustness of MEKA

- Robust to the number of clusters c and various γ:

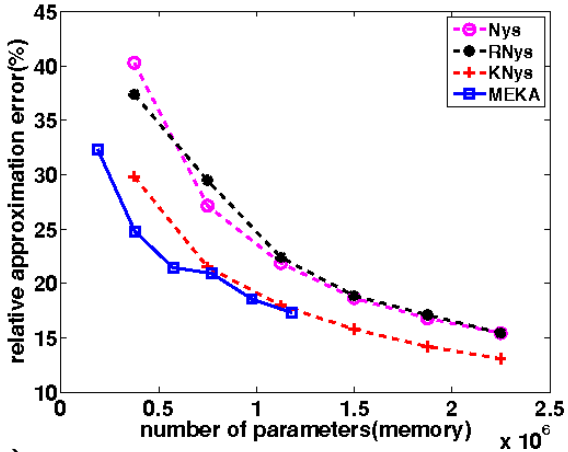


(o) different γ.

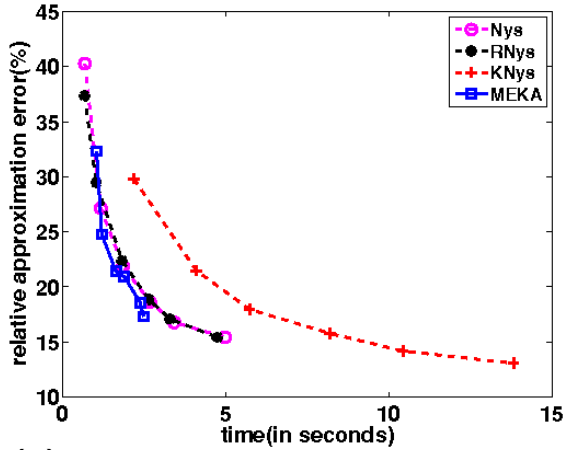


(p) different c.

- Laplacian kernel:



(q) memory vs approx. error.



(r) time vs approx. error.