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
Kernel machines: Kernel SVM, Kernel regression, Kernel PCA, etc.

Kernel functions: $K(x, y) = \varphi(x)^T \varphi(y)$.

Shift-invariant kernels: $K(x, y) = f(\eta(x - y))$.

Examples:
- Gaussian kernel: $K(x, y) = e^{-\gamma \|x - y\|^2_2}$;
- Laplacian Kernel: $K(x, y) = e^{-\gamma \|x - y\|_1}$.

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.
Popular Solution: low-rank approximation to $G$ with $G_{ij} = K(x_i, x_j)$,

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

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: $K(x, y) = e^{-\gamma \|x-y\|^2}$.
  - $\gamma \to 0$, $K(x, y) \to 1$, low-rank structure.
  - $\gamma \to \infty$, $K(x, y) \to 0$, $K(x, x) \to 1$, full-rank, block structure.

With appropriate partitions, each block shows low-rank structure.
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 \( G^{(s,t)}(s \neq t) \):  

\[
\| G - \tilde{G} \|_F^2 = \sum_{i,j} K(x_i, x_j)^2 - \sum_{s=1}^c \sum_{i,j \in \mathcal{V}_s} K(x_i, x_j)^2.
\]
Clustering— Two Objectives

- Minimize the approximation error is to maximize:

\[ D(\{V_s\}_{s=1}^c) = \sum_{s=1}^c \sum_{i,j \in V_s} K(x_i, x_j)^2. \]

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

- Spectral clustering on kernel, maximize:

\[ D_{\text{kernel}}(\{V_s\}_{s=1}^c) = \sum_{s=1}^c \frac{1}{|V_s|} \sum_{i,j \in V_s} K(x_i, x_j)^2. \]

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

- kmeans in the input space, minimize:

\[ D_{\text{kmeans}}(\{V_s\}_{s=1}^c) = \sum_{s=1}^c \sum_{i \in V_s} \|x_i - m_s\|_2^2, m_s = \frac{\sum_{i \in V_s} x_i}{|V_s|}. \]

(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,

  \[ D_{\text{kernel}}(\{V_s\}_{s=1}^c) \geq \bar{C} - \eta^2 R^2 D_{\text{kmeans}}(\{V_s\}_{s=1}^c). \]

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

---

Si Si, Cho-Jui Hsieh, and Inderjit S. Dhillon

Memory Efficient Kernel Approximation
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 $G^{(s,t)}$:

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

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

<table>
<thead>
<tr>
<th>16</th>
<th>14</th>
<th>13</th>
<th>7</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>29</td>
<td>13</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>10</td>
<td>29</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>139</th>
<th>99</th>
<th>101</th>
<th>44</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>116</td>
<td>86</td>
<td>43</td>
<td>44</td>
</tr>
<tr>
<td>101</td>
<td>86</td>
<td>131</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>44</td>
<td>43</td>
<td>46</td>
<td>47</td>
<td>45</td>
</tr>
<tr>
<td>45</td>
<td>44</td>
<td>47</td>
<td>45</td>
<td>49</td>
</tr>
</tbody>
</table>
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 $\{V_s\}_{s=1}^c$ 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 $\{V_s\}_{s=1}^c$ 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 $\{V_s\}_{s=1}^c$ 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 $G(s,t) (s \neq t)$. minimize $\| \hat{G}(s,t) - \hat{X}_s L_{st} \hat{X}_t^T \|_F$. 

Si Si, Cho-Jui Hsieh, and Inderjit S. Dhillon
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 \( G^{(s,t)}(s \neq t) \).

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

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

Rank-\( ck \) 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 $\gamma$:

- Memory and time analysis of MEKA:

<table>
<thead>
<tr>
<th>Method</th>
<th>Storage</th>
<th>Rank</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD</td>
<td>$O(cn_k)$</td>
<td>$ck$</td>
<td>$O(n^3 + n^2d)$</td>
</tr>
<tr>
<td>RKS</td>
<td>$O(cn_k)$</td>
<td>$ck$</td>
<td>$O(cnkd)$</td>
</tr>
<tr>
<td>Nyström</td>
<td>$O(cn_k)$</td>
<td>$ck$</td>
<td>$O(cnkd + (ck)^3)$</td>
</tr>
<tr>
<td>MEKA</td>
<td>$O(nk)$</td>
<td>$ck$</td>
<td>$O(nkd + ck^3) + T_L + T_C$</td>
</tr>
</tbody>
</table>
Theorem 3: Approximation error bound for $\| \tilde{G} - G \|_2$ and $\| \tilde{G} - G \|_F$

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

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

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

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

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);

Data set statistics ($n$: number of samples):

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$n$</th>
<th>$d$</th>
<th>Dataset</th>
<th>$n$</th>
<th>$d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>wine</td>
<td>6,497</td>
<td>11</td>
<td>census</td>
<td>22,784</td>
<td>137</td>
</tr>
<tr>
<td>cpusmall</td>
<td>8,192</td>
<td>12</td>
<td>ijcnn</td>
<td>49,990</td>
<td>22</td>
</tr>
<tr>
<td>pendigit</td>
<td>10,992</td>
<td>16</td>
<td>covtype</td>
<td>581,012</td>
<td>54</td>
</tr>
<tr>
<td>cadata</td>
<td>20,640</td>
<td>8</td>
<td>MNIST2M</td>
<td>2,000,000</td>
<td>784</td>
</tr>
</tbody>
</table>
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

Si Si, Cho-Jui Hsieh, and Inderjit S. Dhillon
Memory Efficient Kernel Approximation
Kernel Ridge Regression Results

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

(i) wine

(j) cpusmall

(k) cadata

(l) census

(m) covtype

(n) MNIST2M

Si Si, Cho-Jui Hsieh, and Inderjit S. Dhillon
Memory Efficient Kernel Approximation
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

Robustness of MEKA

- Robust to the number of clusters $c$ and various $\gamma$:

  (o) different $\gamma$.

  (p) different $c$.

- Laplacian kernel:

  (q) memory vs approx. error.

  (r) time vs approx. error.