Coreset Stochastic Variance-Reduced Gradient with Application to Optimal Margin Distribution Machine
AAAI'19

Zhi-Hao Tan, Teng Zhang, Wei Wang

National Key Laboratory for Novel Software Technology
Nanjing University, China

{tanzh, zhangt, wangw}@lamda.nju.edu.cn

Jun 29, 2019
Motivation

• Kernel-based predictor

\[ f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x) \]

• Prohibitive computation
  - generating the kernel matrix needs \( O(m^2) \)
  - model complexity grows linearly with \( m \)
Accelerating methods

• Random Fourier Feature [Rahimi & Recht, 2007]

\[ f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x) \]

\[ k(x_i, x) \approx \sum_{j=1}^{D} z(x_i; w_j) z(x; w_j) = \langle z(x_i), z(x) \rangle \]

\[ f(x) \approx \sum_{j=1}^{D} \beta_j z(x; w_j) \]

But \( z(x) \) is data independent, the generalization can be improved [Yang et al. 2012]
Accelerating methods

• Nyström method [Williams & Seeger, 2001]

\[ K \approx K_{m,s}K_{s,s}^{-1}K_{s,m} \]

\[ \phi(x)^\top = [k(x, x_1), \ldots, k(x, x_s)]K_{s,s}^{-1/2} \]

\[ K_{ij} \approx \phi(x_i)^\top \phi(x_j) \]

\[ f(x) = w^\top \phi(x) = \sum_{i=1}^{s} \alpha_i k(x_i, x) \]

But basis \( \{x_1, \ldots, x_s\} \) is uniformly sampled, the landmarks can be better.
Some thought

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x) \quad \Rightarrow \quad f(x) = \sum_{i=1}^{s} \alpha_i k(x_i, x)$$

Can we learn the coefficients $\alpha$ directly?

Is there a better way to get $s$ landmarks from $m$ instances?

This paper gives an answer.
Framework

• **Objective**

\[
\min_w \frac{1}{m} \sum_{i=1}^{m} \psi_i(w) = \frac{1}{2} \|w\|_H^2 + \frac{\lambda}{m} \sum_{i=1}^{m} l(w; x_i, y_i)
\]

• **Gradient**

\[
\nabla \psi_i(w) = w + \lambda l'(w; x_i, y_i) \phi(x_i)
\]

• **Solution**

\[
w = \sum_{i=1}^{m} \alpha_i \phi(x_i) \quad f(x) = w^\top \phi(x)
\]
Choose landmarks

• Coreset

Core point:
Blue center point

Coreset:
Set of core points
\{c_1, \ldots, c_s\}
Coreset optimization

Gradient using coreset

\[ \nabla \psi_i(w) = w + \lambda l'(w; x_i, y_i) \phi(x_i) \]
\[ \nabla \psi_i^c(w) = w + \lambda l'(w; x_i, y_i) \phi(c_i) \]

Solution

\[ w = \sum_{i=1}^{m} \alpha_i \phi(x_i) \]
\[ w^c = \sum_{i=1}^{s} \alpha_i \phi(c_i) \]

Only the coefficients \( \alpha = [\alpha_1, \ldots, \alpha_s] \) will be optimized, where \( s \) satisfies \( s \ll m \).
Coreset SGD

\[ w_t = w_{t-1} - \eta \nabla \psi_t^c(w_{t-1}) \]

Coreset SVRG

\[ w_t = w_{t-1} - \eta \left[ \nabla \psi_t^c(w_{t-1}) - \nabla \psi_t^c(\bar{w}) + \mu^c \right] \]

\[ \mu^c = \frac{1}{m} \sum_{i=1}^{m} l'(\bar{w}; x_i, y_i) \phi(c_i) \]
Convergence

- Nyström
  \[ \| K - \hat{K} \|_2 \text{ or } F \]

- Coreset SGD
  \[ E[f(w_T) - f(w^*]) \leq \frac{H(\log(T) + 1)}{T} + \Omega \]

- We bound the gap between optimal solution and approximated solution [Theorem 8]
  \[ E[f(\tilde{w}_s) - f(w^*)] \leq \rho^S E[f(\tilde{w}_0) - f(w^*)] + \Omega \]

\( \Omega \) approaches 0 when diameter of coreset \( \rightarrow 0 \)
Empirically

- Optimal Margin Distribution Machine (ODM)

\[
\min_{\omega, \xi, \varepsilon} \frac{1}{2} \|\omega\|^2 + \frac{\lambda}{m} \sum_{i=1}^{m} \frac{\xi_i^2 + \mu \varepsilon_i^2}{(1 - \theta)^2}
\]

s. t. \[y_i \omega^T \phi(x_i) \geq 1 - \theta - \xi_i, \forall i\]
\[y_i \omega^T \phi(x_i) \geq 1 + \theta + \varepsilon_i, \forall i\]

Compared to SVM, it enjoys better statistical property by optimizing the margin distribution. [Zhang et al. 2016]
# Datasets

Table 1: Characteristics of 9 large-scale datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#instance</th>
<th>#feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>magic04</td>
<td>19020</td>
<td>10</td>
</tr>
<tr>
<td>adult-a</td>
<td>32561</td>
<td>123</td>
</tr>
<tr>
<td>a9a</td>
<td>48842</td>
<td>123</td>
</tr>
<tr>
<td>w8a</td>
<td>49749</td>
<td>300</td>
</tr>
<tr>
<td>cod-rna</td>
<td>59535</td>
<td>8</td>
</tr>
<tr>
<td>mini-boo-ne</td>
<td>130064</td>
<td>50</td>
</tr>
<tr>
<td>ijcnnl</td>
<td>141691</td>
<td>22</td>
</tr>
<tr>
<td>webspam</td>
<td>350000</td>
<td>254</td>
</tr>
<tr>
<td>covtype</td>
<td>581012</td>
<td>54</td>
</tr>
</tbody>
</table>
Diameter of coreset

• Control the **trade-off between performance and efficiency** according to the demand.

![Graphs showing classification error and model size vs. square of radius of coverage for cod-rna and adult-a datasets.](http://lamda.nju.edu.cn)
### Performance

- Our method has better efficiency and accuracy

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Algorithm</th>
<th>Train</th>
<th>Test</th>
<th>Accuracy</th>
<th>Train</th>
<th>Test</th>
<th>Accuracy</th>
<th>Train</th>
<th>Test</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>magic04</td>
<td>LIBSVM</td>
<td>50.88</td>
<td>1.11</td>
<td>87.00 ± 0.28</td>
<td>13.12</td>
<td>24.92</td>
<td>84.50 ± 0.28</td>
<td>108.61</td>
<td>13.11</td>
<td>84.79 ± 0.00</td>
</tr>
<tr>
<td></td>
<td>ODM</td>
<td>5.52</td>
<td>0.25</td>
<td>86.55 ± 0.27</td>
<td>32.88</td>
<td>0.68</td>
<td>84.61 ± 0.24</td>
<td>105.75</td>
<td>1.69</td>
<td>85.29 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>AVM</td>
<td>1.12</td>
<td>0.64</td>
<td>82.05 ± 0.13</td>
<td>2.47</td>
<td>1.39</td>
<td>83.21 ± 0.32</td>
<td>5.88</td>
<td>1.71</td>
<td>81.46 ± 0.35</td>
</tr>
<tr>
<td></td>
<td>CSVRG+ODM</td>
<td>1.32</td>
<td>0.09</td>
<td>84.43 ± 0.40</td>
<td>1.68</td>
<td>0.33</td>
<td>84.08 ± 0.22</td>
<td>1.70</td>
<td>0.18</td>
<td>84.26 ± 0.20</td>
</tr>
<tr>
<td>w8a</td>
<td>LIBSVM</td>
<td>14.50</td>
<td>29.22</td>
<td>98.67 ± 0.05</td>
<td>37.96</td>
<td>3.78</td>
<td>95.52 ± 0.10</td>
<td>243.03</td>
<td>84.36</td>
<td>92.22 ± 0.07</td>
</tr>
<tr>
<td></td>
<td>ODM</td>
<td>111.34</td>
<td>1.93</td>
<td>98.57 ± 0.08</td>
<td>50.79</td>
<td>2.79</td>
<td>95.57 ± 0.09</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AVM</td>
<td>19.79</td>
<td>29.15</td>
<td>97.09 ± 0.08</td>
<td>2.94</td>
<td>1.61</td>
<td>91.32 ± 1.39</td>
<td>9.03</td>
<td>5.52</td>
<td>83.50 ± 0.25</td>
</tr>
<tr>
<td></td>
<td>CSVRG+ODM</td>
<td>2.44</td>
<td>0.81</td>
<td>97.14 ± 0.07</td>
<td>1.78</td>
<td>0.15</td>
<td>94.22 ± 0.14</td>
<td>8.40</td>
<td>1.47</td>
<td>85.21 ± 0.53</td>
</tr>
<tr>
<td>ijcnn1</td>
<td>LIBSVM</td>
<td>80.00</td>
<td>16.32</td>
<td>99.31 ± 0.03</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ODM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AVM</td>
<td>7.03</td>
<td>3.70</td>
<td>90.45 ± 0.01</td>
<td>157.40</td>
<td>196.11</td>
<td>78.30 ± 1.05</td>
<td>22.79</td>
<td>9.57</td>
<td>70.25 ± 0.28</td>
</tr>
<tr>
<td></td>
<td>CSVRG+ODM</td>
<td>9.32</td>
<td>0.95</td>
<td>91.33 ± 0.07</td>
<td>31.32</td>
<td>24.17</td>
<td>83.90 ± 0.21</td>
<td>14.44</td>
<td>1.68</td>
<td>73.22 ± 0.17</td>
</tr>
</tbody>
</table>
Thanks