Faster Support Vector Machines

ALENEX’19 · January 8, 2019
Sebastian Schlag†, Matthias Schmitt†, Christian Schulz‡

†KIT, ‡University of Vienna
Binary Classification Problem
Binary Classification Problem

Send any comments regarding submissions directly to submitter.

Archives at http://arxiv.org/
To unsubscribe, e-mail To: cs@arXiv.org, Subject: cancel

Submissions to: Distributed, Parallel, and Cluster Computing
Data Structures and Algorithms
Social and Information Networks
received from Fri 31 Aug 18 18:00:00 GMT to Tue 4 Sep 18
Binary Classification Problem

Congratulations!
You have been selected winner of the recent Publishers Clearing House major prize awards sponsored by the EU funding!
Get your exclusive chance to claim (USD $850,000.00) in cash!
Please confirm the following information today:
First: Last name: Telephone #: Gender: Age: Nationality:
Confirm and reply to this email (eupchint@collector.org)
We look forward to seeing you claim your prize!
Best regards, your winning team PCH EU

*© 2018 European Union*
Binary Classification Problem

Congratulations!
You have been selected winner of the recent Publishers Clearing House major prize awards sponsored by the EU funding!
Get your exclusive chance to claim USD $850,000.00 in cash!
Please confirm the following information today:
First : Last name : Telephone #: Gender: Age: Nationality:
Confirm and reply to this e-mail (eupchint@collector.org )
We look forward to seeing you claim your prize!
Best regards, your winning team PCH EU

* © 2018European Union*

Send any comments regarding submissions directly to submitter.

Archives at http://arxiv.org/
To unsubscribe, e-mail To: cs@arXiv.org, Subject: cancel

Submissions to: Distributed, Parallel, and Cluster Computing
Data Structures and Algorithms
Social and Information Networks
received from Fri 31 Aug 18 18:00:00 GMT to Tue 4 Sep 18
Binary Classification Problem

Train classifier on \( n \) labeled data points

\[
(x_i, y_i)
\]

Data point \( x \in \mathbb{R}^d \)
Features:
words, text patterns, sender, ...

Label \( y \in \{-1, +1\} \)
Result:
+1: spam
−1: no spam

Goal: Assign label \( y_{n+1} \) to new data points \( x_{n+1} \)
Support Vector Machines [CV’97]

Find **hyperplane** with **maximum margin** between classes $C^{-}$ & $C^{+}$

Linear SVM
Support Vector Machines [CV’97]

Find hyperplane with maximum margin between classes $C^{-}$ & $C^{+}$

Linear SVM

Soft Margin SVM
Support Vector Machines [CV’97]

Find **hyperplane** with **maximum margin** between classes $C^-$ & $C^+$

- **Linear SVM**
  - Input: Linearly separable
  - Output: Maximum margin hyperplane

- **Soft Margin SVM**
  - Input: Nonlinearly separable
  - Output: Maximum margin hyperplane with some misclassifications

- **Nonlinear SVM**
  - Input: Nonlinearly separable
  - Output: Maximum margin hyperplane after feature mapping $\phi$
Support Vector Machines [CV’97]

SVM Optimization Problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
\end{align*}
\]
Support Vector Machines [CV’97]

SVM Optimization Problem:

\[
\text{minimize} \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
\]

Slack penalty
Slack variables
Support Vector Machines [CV’97]

SVM Optimization Problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
\end{align*}
\]

Mapping to higher dimensional space

\[\phi : \mathbb{R}^d \rightarrow \mathbb{R}^p (d \leq p)\]
Support Vector Machines [CV’97]

SVM Optimization Problem:

\[
\begin{align*}
&\text{minimize} & & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\
&\text{subject to} & & y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
\end{align*}
\]

Mapping to higher dimensional space
\[\phi : \mathbb{R}^d \to \mathbb{R}^p(d \leq p)\]

Kernel Trick:
- Replace inner product by kernel fct.
- \[k(x_i, x_j) = \phi(x_i)^T \phi(x_j)\]
- Here: \(k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)\)

Gaussian kernel (RBF)
Support Vector Machines – Training

**Model Selection:** instance-specific tuning of several parameters
- Slack penalty $C$
- Kernel parameters (here: $\gamma$)

**Complexity:**
- Solver running time between $O(n^2)$ & $O(n^3)$ \[GCBDV’04\]
- Model selection $\rightsquigarrow$ train many models
Support Vector Machines – Training

Model Selection: instance-specific tuning of several parameters
- Slack penalty $C$
- Kernel parameters (here: $\gamma$

Complexity:
- Solver running time between $O(n^2)$ & $O(n^3)$ [GCBDV’04]
- Model selection $\leadsto$ train many models

Performance Improvements:
- Sampling [SC’00]
- Parallelization [CZWBLQC’07, ZCWZC’09, CWLTP’17]
- Hierarchical techniques
  - Input space [YYH’03, HSCKCLP’11, HSD’14]
  - Graph representation [RS’15, SJKLLRS’17]

Training on large data sets becomes infeasible!
From Feature Vectors to Graphs
From Feature Vectors to Graphs

Approx. $k$-nearest neighbors

\[ \omega(e) = \frac{1}{\text{dist}(p, q)} \Rightarrow \text{encode proximity information into edge weights} \]

[RS'15]
Multilevel Support Vector Machines [RS’15]
Multilevel Support Vector Machines [RS’15]

Coarsening

Initial SVM training
Multilevel Support Vector Machines [RS’15]

Coarsening

Initial SVM training

Refinement

Uncoarsening
Multilevel Support Vector Machines [RS’15]

Coarsening

Uncoarsening

Initial SVM training

Refinement
Multilevel Support Vector Machines [RS’15]

- mlsvm-IIS: independent sets [RS’15]
- mlsvm-AMG: algebraic distances [SJKLRS’17]
- KaSVM: label propagation [this presentation]
Algorithm 1: KaSVM

preprocess data
build $k$-nearest neighbor graphs for $C^+$ and $C^-$
contract graphs recursively, build hierarchy
train initially on coarsest problem

while levels in the hierarchy do
  train model on uncontracted support vectors of prev. level

return best model of all levels
Algorithm 1: KaSVM

preprocess data
build k-nearest neighbor graphs for $C^+$ and $C^-$
contract graphs recursively, build hierarchy
train initially on coarsest problem
while levels in the hierarchy do
  train model on uncontracted support vectors of prev. level
return best model of all levels
Label Propagation [RAK’07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster $V_i$ having **strongest** connection

\[ c[v] = \arg\max_{V_i} \omega(\{(v, u) \mid u \in N(v) \cap V_i\}) \]
Label Propagation \[\text{[RAK’07]}\]

**Cut-based, linear time clustering algorithm**

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster \(V_i\) having **strongest** connection
  \[c[v] = \arg\max_{V_i} \omega(\{(v, u) | u \in N(v) \cap V_i\})\]

Cluster close nodes
Label Propagation [RAK’07]

Cut-based, linear time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster $V_i$ having strongest connection

$$\Rightarrow c[v] = \arg\max_{V_i} \omega(\{(v, u) | u \in N(v) \cap V_i\})$$
Label Propagation [RAK’07]

**Cut**-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster $V_i$ having **strongest** connection

$$\Rightarrow c[v] = \arg\max_{V_i} \omega(\{(v, u) | u \in N(v) \cap V_i\})$$

Cluster close nodes
Label Propagation [RAK’07]

**Cut**-based, **linear** time clustering algorithm
- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster $V_i$ having **strongest** connection
  \[ c[v] = \arg\max_{V_i} \omega(\{(v, u) | u \in N(v) \cap V_i\}) \]
Label Propagation [RAK’07]

Cut-based, linear time clustering algorithm
- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster $V_i$ having strongest connection
  ⇒ $c[v] = \arg\max_{V_i} \omega(\{(v, u) | u \in N(v) \cap V_i\})$

Cluster close nodes
Label Propagation [RAK’07]

Cut-based, linear time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster $V_i$ having strongest connection

\[ c[v] = \arg\max_{V_i} \omega(\{(v, u) | u \in N(v) \cap V_i\}) \]

⇒ average feature values of clustered nodes:

\[ x_{\text{coarse}} = \frac{1}{c(V_i)} \sum_{v_i \in V_i} c(v_i) x_i \]
Initial Training

Train on **coarsest** problem

- Model selection \((C, \gamma)\) via uniform design (UD) [HLLH’07]
- Solver: LibSVM
- Validation on random 10% of training set
Uncoarsening/Refinement

Try to improve model
- Uncontract support vectors
- Ensure similar size for $C^-$ & $C^+$
- **Reuse** $C$, $\gamma$ from prev. level $\Rightarrow$ 2nd UD sweep around old params.
- Validation on random 10% of training set
Experimental Setup

**Machine:** AMD Opteron 6168 with 1.9 GHz, 256 GB of RAM

**Implementation:**
- approx. $k$-nearest neighbors: FLANN 1.8.4 [ML’09]
- SVM training: LibSVM 3.22

**Configuration:**
- $k = 10$ nearest neighbors
- $\ell = 10$ label propagation iterations
- stop coarsening $|C^+/−|$ \approx 500 nodes

**Algorithms:**
- KaSVM / KaSVM$_{fast}$
- mlsvm-AMG (outperforms DC-SVM & EnsembleSVM) [SJKLRS’17]
- LibSVM
Experimental Setup

**Machine:** AMD Opteron 6168 with 1.9 GHz, 256 GB of RAM

**Implementation:**
- approx. $k$-nearest neighbors: FLANN 1.8.4 [ML’09]
- SVM training: LibSVM 3.22

**Configuration:**
- $k = 10$ nearest neighbors
- $\ell = 10$ label propagation iterations
- stop coarsening $|C^+/−| ≈ 500$ nodes

**Algorithms:**
- KaSVM / KaSVM\textsubscript{fast} (initially trained model / no refinement)
- mlsvm-AMG (outperforms DC-SVM & EnsembleSVM) [SJKLRS’17]
- LibSVM
### Instances

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Feat.</th>
<th>$C^+$</th>
<th>$C^-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertisement</td>
<td>3 279</td>
<td>1 558</td>
<td>459</td>
<td>2 820</td>
</tr>
<tr>
<td>Buzz</td>
<td>140 707</td>
<td>77</td>
<td>27 775</td>
<td>112 932</td>
</tr>
<tr>
<td>Clean (Musk)</td>
<td>6 598</td>
<td>166</td>
<td>1 017</td>
<td>5 581</td>
</tr>
<tr>
<td>Cod-rna</td>
<td>59 535</td>
<td>8</td>
<td>19 845</td>
<td>39 690</td>
</tr>
<tr>
<td>EEG Eye State</td>
<td>14 980</td>
<td>14</td>
<td>6 723</td>
<td>8 257</td>
</tr>
<tr>
<td>Forest (Class 3)</td>
<td>581 012</td>
<td>54</td>
<td>35 754</td>
<td>369 172</td>
</tr>
<tr>
<td>Forest (Class 5)</td>
<td>581 012</td>
<td>54</td>
<td>9 493</td>
<td>571 519</td>
</tr>
<tr>
<td>Forest (Class 7)</td>
<td>581 012</td>
<td>54</td>
<td>20 510</td>
<td>560 502</td>
</tr>
<tr>
<td>Hypothyroid</td>
<td>3 919</td>
<td>21</td>
<td>240</td>
<td>3 679</td>
</tr>
<tr>
<td>Isolet (Class A)</td>
<td>6 919</td>
<td>617</td>
<td>240</td>
<td>5 998</td>
</tr>
<tr>
<td>Letter (Class Z)</td>
<td>20 000</td>
<td>16</td>
<td>734</td>
<td>19 266</td>
</tr>
<tr>
<td>Nursery</td>
<td>12 960</td>
<td>8</td>
<td>4 320</td>
<td>8 640</td>
</tr>
<tr>
<td>Protein</td>
<td>145 751</td>
<td>74</td>
<td>1 296</td>
<td>144 455</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>7 400</td>
<td>20</td>
<td>3 664</td>
<td>3 736</td>
</tr>
<tr>
<td>Twonorm</td>
<td>7 400</td>
<td>20</td>
<td>3 703</td>
<td>3 697</td>
</tr>
<tr>
<td>APS failure</td>
<td>76 000</td>
<td>170</td>
<td>1 375</td>
<td>74 625</td>
</tr>
<tr>
<td>Census</td>
<td>299 285</td>
<td>41</td>
<td>18 568</td>
<td>280 717</td>
</tr>
<tr>
<td>Letter (Class A)</td>
<td>20 000</td>
<td>16</td>
<td>786</td>
<td>19 266</td>
</tr>
<tr>
<td>Letter (Class B)</td>
<td>20 000</td>
<td>16</td>
<td>766</td>
<td>19 266</td>
</tr>
<tr>
<td>Letter (Class H)</td>
<td>20 000</td>
<td>16</td>
<td>734</td>
<td>19 266</td>
</tr>
<tr>
<td>Skin</td>
<td>245 057</td>
<td>3</td>
<td>50 859</td>
<td>194 198</td>
</tr>
<tr>
<td>Sleep (Class 1)</td>
<td>105 908</td>
<td>13</td>
<td>9 052</td>
<td>96 856</td>
</tr>
</tbody>
</table>

**mlsvm-AMG**

*mlsvm-AMG [SJKLRS’17]*

**UCI Machine Learning Repository**
Experimental Methodology

k-fold cross validation:

- Shuffle data set \(\rightarrow\) split into \(k = 5\) parts
- \(k\) training repetitions:
  - \(\Rightarrow\) **Training** set: \(k-1\) parts
  - \(\Rightarrow\) **Test** set: \(1\) part
- 5 \(k\)-folds per instance

Performance Measures:

- **Accuracy** \((ACC)\) = \(\frac{TP + TN}{FP + TN + TP + FN}\)
- **Sensitivity** \((SN)\) = \(\frac{TP}{TP + FN}\)
- **Specificity** \((SP)\) = \(\frac{TN}{TN + FP}\)
- Geometric mean = \(\sqrt{SP \cdot SN}\)
## Running Time

<table>
<thead>
<tr>
<th>Dataset</th>
<th>mlsvm-AMG</th>
<th>KaSVM</th>
<th>KaSVM_{fast}</th>
<th>LibSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertisement</td>
<td>343</td>
<td>192</td>
<td>70</td>
<td>557</td>
</tr>
<tr>
<td>APS failure</td>
<td>1 473</td>
<td>109</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>Buzz</td>
<td>110</td>
<td>121</td>
<td>19</td>
<td>-</td>
</tr>
<tr>
<td>Census</td>
<td>3 047</td>
<td>657</td>
<td>38</td>
<td>-</td>
</tr>
<tr>
<td>Clean (Musk)</td>
<td>14</td>
<td>8</td>
<td>4</td>
<td>320</td>
</tr>
<tr>
<td>Cod-rna</td>
<td>80</td>
<td>43</td>
<td>7</td>
<td>15 700</td>
</tr>
<tr>
<td>EEG Eye State</td>
<td>123</td>
<td>1 320</td>
<td>0.9</td>
<td>2 700</td>
</tr>
<tr>
<td>Forest (Class 3)</td>
<td>10 156</td>
<td>744</td>
<td>99</td>
<td>-</td>
</tr>
<tr>
<td>Forest (Class 5)</td>
<td>6 986</td>
<td>1 090</td>
<td>158</td>
<td>-</td>
</tr>
<tr>
<td>Forest (Class 7)</td>
<td>5 393</td>
<td>1 990</td>
<td>114</td>
<td>-</td>
</tr>
<tr>
<td>Hypothyroid</td>
<td>2</td>
<td>3</td>
<td>0.9</td>
<td>13</td>
</tr>
<tr>
<td>Isolet (Class A)</td>
<td>1 627</td>
<td>23</td>
<td>7</td>
<td>856</td>
</tr>
<tr>
<td>Letter (Class A)</td>
<td>17</td>
<td>4</td>
<td>2</td>
<td>1 930</td>
</tr>
<tr>
<td>Letter (Class B)</td>
<td>55</td>
<td>4</td>
<td>2</td>
<td>1 590</td>
</tr>
<tr>
<td>Letter (Class H)</td>
<td>74</td>
<td>9</td>
<td>2</td>
<td>1 970</td>
</tr>
<tr>
<td>Letter (Class Z)</td>
<td>31</td>
<td>3</td>
<td>2</td>
<td>1 710</td>
</tr>
<tr>
<td>Nursery</td>
<td>7</td>
<td>2</td>
<td>0.7</td>
<td>998</td>
</tr>
<tr>
<td>Protein</td>
<td>3 654</td>
<td>41</td>
<td>17</td>
<td>-</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>10</td>
<td>13</td>
<td>0.6</td>
<td>161</td>
</tr>
<tr>
<td>Skin</td>
<td>81</td>
<td>18</td>
<td>12</td>
<td>38 200</td>
</tr>
<tr>
<td>Sleep (Class 1)</td>
<td>1 594</td>
<td>3 080</td>
<td>12</td>
<td>-</td>
</tr>
<tr>
<td>Twonorm</td>
<td>7</td>
<td>0.7</td>
<td>0.5</td>
<td>109</td>
</tr>
</tbody>
</table>
# Running Time

<table>
<thead>
<tr>
<th></th>
<th>mlsvm-AMG</th>
<th>KaSVM</th>
<th>KaSVM\textsubscript{fast}</th>
<th>LibSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running time [s]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertisement</td>
<td>343</td>
<td>192</td>
<td>70</td>
<td>557</td>
</tr>
<tr>
<td>APS failure</td>
<td>1 473</td>
<td>109</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Buzz</td>
<td>110</td>
<td>121</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Census</td>
<td>3 047</td>
<td>657</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Clean (Musk)</td>
<td>14</td>
<td>8</td>
<td>4</td>
<td>320</td>
</tr>
<tr>
<td>Cod-rna</td>
<td>80</td>
<td>43</td>
<td>7</td>
<td>15 700</td>
</tr>
<tr>
<td>EEG Eye State</td>
<td>123</td>
<td>1 320</td>
<td>0.9</td>
<td>2 700</td>
</tr>
<tr>
<td>Forest (Class 3)</td>
<td>10 156</td>
<td>744</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>Forest (Class 5)</td>
<td>6 986</td>
<td>1 090</td>
<td>158</td>
<td></td>
</tr>
<tr>
<td>Forest (Class 7)</td>
<td>5 393</td>
<td>1 990</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>Hypothyroid</td>
<td>2</td>
<td>3</td>
<td>0.9</td>
<td>13</td>
</tr>
<tr>
<td>Isolet (Class A)</td>
<td>1 627</td>
<td>23</td>
<td>7</td>
<td>856</td>
</tr>
<tr>
<td>Letter (Class A)</td>
<td>17</td>
<td>4</td>
<td>2</td>
<td>1 930</td>
</tr>
<tr>
<td>Letter (Class B)</td>
<td>55</td>
<td>4</td>
<td>2</td>
<td>1 590</td>
</tr>
<tr>
<td>Letter (Class H)</td>
<td>74</td>
<td>9</td>
<td>2</td>
<td>1 970</td>
</tr>
<tr>
<td>Letter (Class Z)</td>
<td>31</td>
<td>3</td>
<td>2</td>
<td>1 710</td>
</tr>
<tr>
<td>Nursery</td>
<td>7</td>
<td>2</td>
<td>0.7</td>
<td>998</td>
</tr>
<tr>
<td>Protein</td>
<td>3 654</td>
<td>41</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Ringnorm</td>
<td>10</td>
<td>13</td>
<td>0.6</td>
<td>161</td>
</tr>
<tr>
<td>Skin</td>
<td>81</td>
<td>18</td>
<td>12</td>
<td>38 200</td>
</tr>
<tr>
<td>Sleep (Class 1)</td>
<td>1 594</td>
<td>3 080</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Twonorm</td>
<td>7</td>
<td>0.7</td>
<td>0.5</td>
<td>109</td>
</tr>
</tbody>
</table>

Running time > 24h
## Running Time

<table>
<thead>
<tr>
<th>Data Set</th>
<th>mlsvm-AMG</th>
<th>KaSVM</th>
<th>KaSVM_{fast}</th>
<th>LibSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>APS failure</td>
<td>1</td>
<td>473</td>
<td>109</td>
<td>-</td>
</tr>
<tr>
<td>Buzz</td>
<td>110</td>
<td>121</td>
<td>19</td>
<td>-</td>
</tr>
<tr>
<td>Census</td>
<td>3047</td>
<td>657</td>
<td>38</td>
<td>-</td>
</tr>
<tr>
<td>Clean (Musk)</td>
<td>14</td>
<td>8</td>
<td>4</td>
<td>320</td>
</tr>
<tr>
<td>Cod-rna</td>
<td>80</td>
<td>43</td>
<td>7</td>
<td>15700</td>
</tr>
<tr>
<td>EEG Eye S</td>
<td>123</td>
<td>1</td>
<td>320</td>
<td>2700</td>
</tr>
<tr>
<td>Forest (Class 3)</td>
<td>10</td>
<td>156</td>
<td>744</td>
<td>-</td>
</tr>
<tr>
<td>Forest (Class 5)</td>
<td>6</td>
<td>986</td>
<td>1090</td>
<td>-</td>
</tr>
<tr>
<td>Forest (Class 7)</td>
<td>5</td>
<td>393</td>
<td>1990</td>
<td>-</td>
</tr>
<tr>
<td>Hypothyroid</td>
<td>2</td>
<td>3</td>
<td>0.9</td>
<td>13</td>
</tr>
<tr>
<td>Isolet (Class A)</td>
<td>1627</td>
<td>23</td>
<td>7</td>
<td>856</td>
</tr>
<tr>
<td>Letter (Class A)</td>
<td>17</td>
<td>4</td>
<td>2</td>
<td>1930</td>
</tr>
<tr>
<td>Letter (Class B)</td>
<td>55</td>
<td>4</td>
<td>2</td>
<td>1590</td>
</tr>
<tr>
<td>Letter (Class H)</td>
<td>74</td>
<td>9</td>
<td>2</td>
<td>1970</td>
</tr>
<tr>
<td>Letter (Class Z)</td>
<td>31</td>
<td>3</td>
<td>2</td>
<td>970</td>
</tr>
<tr>
<td>Nursery</td>
<td>7</td>
<td>2</td>
<td>0.7</td>
<td>998</td>
</tr>
<tr>
<td>Protein</td>
<td>394</td>
<td>41</td>
<td>17</td>
<td>-</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>10</td>
<td>13</td>
<td>0.6</td>
<td>161</td>
</tr>
<tr>
<td>Skin</td>
<td>81</td>
<td>18</td>
<td>12</td>
<td>200</td>
</tr>
<tr>
<td>Sleep (Class 1)</td>
<td>1594</td>
<td>5080</td>
<td>12</td>
<td>38200</td>
</tr>
<tr>
<td>Twonorm</td>
<td>7</td>
<td>0.7</td>
<td>0.5</td>
<td>109</td>
</tr>
</tbody>
</table>
## Classification Quality

<table>
<thead>
<tr>
<th>Instance</th>
<th>mlsvm-AMG G-mean</th>
<th>KaSVM G-mean</th>
<th>KaSVM&lt;sub&gt;fast&lt;/sub&gt; G-mean</th>
<th>LibSVM G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertisement</td>
<td>0.91</td>
<td>0.91</td>
<td>0.87</td>
<td>0.941</td>
</tr>
<tr>
<td>APS failure</td>
<td>0.94</td>
<td>0.92</td>
<td>0.94</td>
<td>-</td>
</tr>
<tr>
<td>Buzz</td>
<td>0.95</td>
<td>0.94</td>
<td>0.93</td>
<td>-</td>
</tr>
<tr>
<td>Census</td>
<td>0.81</td>
<td>0.83</td>
<td>0.81</td>
<td>-</td>
</tr>
<tr>
<td>Clean (Musk)</td>
<td>0.88</td>
<td>0.96</td>
<td>0.88</td>
<td>0.996</td>
</tr>
<tr>
<td>Cod-rna</td>
<td>0.94</td>
<td>0.94</td>
<td>0.93</td>
<td>0.855</td>
</tr>
<tr>
<td>EEG Eye State</td>
<td>0.75</td>
<td>0.83</td>
<td>0.63</td>
<td>0.929</td>
</tr>
<tr>
<td>Forest (Class 3)</td>
<td>0.94</td>
<td>0.95</td>
<td>0.94</td>
<td>-</td>
</tr>
<tr>
<td>Forest (Class 5)</td>
<td>0.79</td>
<td>0.90</td>
<td>0.90</td>
<td>-</td>
</tr>
<tr>
<td>Forest (Class 7)</td>
<td>0.86</td>
<td>0.93</td>
<td>0.91</td>
<td>-</td>
</tr>
<tr>
<td>Hypothyroid</td>
<td>0.94</td>
<td>0.90</td>
<td>0.94</td>
<td>0.927</td>
</tr>
<tr>
<td>Isolet (Class A)</td>
<td>0.00</td>
<td>0.99</td>
<td>0.88</td>
<td>0.990</td>
</tr>
<tr>
<td>Letter (Class A)</td>
<td>0.94</td>
<td>0.96</td>
<td>0.95</td>
<td>0.995</td>
</tr>
<tr>
<td>Letter (Class B)</td>
<td>0.88</td>
<td>0.92</td>
<td>0.92</td>
<td>0.979</td>
</tr>
<tr>
<td>Letter (Class H)</td>
<td>0.76</td>
<td>0.89</td>
<td>0.86</td>
<td>0.970</td>
</tr>
<tr>
<td>Letter (Class Z)</td>
<td>0.92</td>
<td>0.95</td>
<td>0.95</td>
<td>0.991</td>
</tr>
<tr>
<td>Nursery</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.000</td>
</tr>
<tr>
<td>Protein</td>
<td>0.92</td>
<td>0.93</td>
<td>0.91</td>
<td>-</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>0.98</td>
<td>0.97</td>
<td>0.82</td>
<td>0.987</td>
</tr>
<tr>
<td>Skin</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.000</td>
</tr>
<tr>
<td>Sleep (Class 1)</td>
<td>0.69</td>
<td>0.70</td>
<td>0.41</td>
<td>-</td>
</tr>
<tr>
<td>Twonorm</td>
<td>0.97</td>
<td>0.96</td>
<td>0.96</td>
<td>0.981</td>
</tr>
</tbody>
</table>
Conclusion & Discussion

KaSVM – multilevel SVM using label propagation
- Comparable classification quality
- Training: up to two orders of magnitude faster

Future Work:
- shared/distributed-memory parallelization
- small-diameter clustering
- different solvers for training

KaSVM - Open Source: https://algo2.itl.kit.edu/kasvm

KaSVM vs. mlsvm-AMG