Locally Linear Support Vector Machines

Ľubor Ladický

Philip H.S. Torr
Given: \{ [x_1, y_1], [x_2, y_2], \ldots [x_n, y_n] \}

\begin{align*}
\text{x}_i & \text{ - feature vector} \\
\text{y}_i & \in \{-1, 1\} \text{ - label}
\end{align*}

The task is:

To design \( y = H(x) \) that predicts a label \( y \) for a new \( x \).
Given : \{ \[x_1, y_1\], \[x_2, y_2\], \ldots \[x_n, y_n\] \} \\
\[x_i\] - feature vector \quad \[y_i\] \in \{-1, 1\} - label

The task is : 
To design \( y = H(x) \) that predicts a label \( y \) for a new \( x \)

Several approaches proposed : 
Support Vector Machines, Boosting, Random Forests, Neural networks, Nearest Neighbour, ..
Linear vs. Kernel SVMs

Linear SVMs
• Fast training and evaluation
• Applicable to large scale data sets
• Low discriminative power

Kernel SVMs
• Slow training and evaluation
• Not feasible to too large data sets
• Much better performance for hard problems
The goal is to design an SVM with:

- Good trade-off between the performance and speed
- Scalability to large scale data sets (solvable using SGD)
Local codings

Points approximated as a weighted sum of anchor points

\[ x \approx \sum_{v \in C} \gamma_v(x) v \]

\( v \) – anchor points
\( \gamma_v(x) \) - coordinates
Local codings

Points approximated as a weighted sum of anchor points

\[ x \approx \sum_{v \in C} \gamma_v(x) v \]

\( v \) – anchor points \( \gamma_v(x) \) - coordinates

Coordinates obtained using:

- Distance based methods (Gemert et al. 2008, Zhou et al. 2009)
- Reconstruction methods
Local codings

For a normalised codings and any Lipschitz function $f$:
(Yu et al. 2009)

$$x \approx \sum_{v \in C} \gamma_v(x)v \quad \Rightarrow \quad f(x) \approx \sum_{v \in C} \gamma_v(x)f(v)$$

$v$ – anchor points

$\gamma_v(x)$ - coordinates
Linear SVMs

The form of the classifier

\[ H(x) = w^T x + b = \sum_{i=1}^{n} w_i x_i + b. \]

Weights \( w \) and bias \( b \) obtained as:

\[
\arg \min_{w,b} \frac{\lambda}{2} \|w\|^2 + \frac{1}{|S|} \sum_{k \in S} \max(0, 1 - y_k (w^T x_k + b))
\]

- \([x_k, y_k]\) – training samples
- \(S\) – number of samples
- \(\lambda\) – regularisation weight

(Vapnik & Learner 1963, Cortes & Vapnik 1995)
Locally Linear SVMs

The decision boundary should be smooth

- Approximately linear is sufficiently small region
Locally Linear SVMs

The decision boundary should be smooth

- Approximately linear is sufficiently small region
The decision boundary should be smooth

- Approximately linear is sufficiently small region

\[ H(x) = w(x)^T x + b(x) = \sum_{i=1}^{n} w_i(x)x_i + b(x) \]
The decision boundary should be smooth

- Approximately linear is sufficiently small region

\[ H(x) = w(x)^T x + b(x) = \sum_{i=1}^{n} w_i(x)x_i + b(x) \]

- Curvature is bounded
  - Functions \( w_i(x) \) and \( b(x) \) are Lipschitz
  - \( w_i(x) \) and \( b(x) \) can be approximated using local coding
Locally Linear SVMs

The decision boundary should be smooth

- Approximately linear is sufficiently small region

\[ H(x) = w(x)^T x + b(x) = \sum_{i=1}^{n} w_i(x)x_i + b(x) \]

- Curvature is bounded
  - Functions \( w_i(x) \) and \( b(x) \) are Lipschitz
  - \( w_i(x) \) and \( b(x) \) can be approximated using local coding

\[
H(x) = \sum_{i=1}^{n} \sum_{v \in C} \gamma_v(x)w_i(v)x_i + \sum_{v \in C} \gamma_v(x)b(v)
\]

\[ = \gamma(x)^T W x + \gamma(x)^T b \]
The classifier takes the form:

\[ H(x) = \gamma(x)^T W x + \gamma(x)^T b \]

Weights \( W \) and biases \( b \) are obtained as:

\[
\arg\min_{W,b} \frac{\lambda}{2} \|W\|^2 + \frac{1}{|S|} \sum_{k \in S} \max(0, 1 - y_k H_{W,b}(x_k))
\]

where

\[ \|W\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} W_{ij}^2 \]
Locally Linear SVMs

Optimised using stochastic gradient descent (Bordes et al. 2005):

\[
\begin{align*}
    t &= 0, \text{ count } = \text{skip}, \quad W = W_0, \quad b = b_0 \\
    \text{while } t \leq T \text{ do} \\
    \gamma_t &= \text{LocalCoding}(x_t, C) \\
    H_t &= 1 - y_t (\gamma_t^T W x_t + \gamma_t^T b) \\
    \text{if } H_t > 0 \text{ then} \\
    \quad W &= W + \frac{1}{\lambda(t+t_0)} y_t (x_t \gamma_t^T) \\
    \quad b &= b + \frac{1}{\lambda(t+t_0)} y_t \gamma_t \\
    \text{end if} \\
    \text{count } &= \text{count } - 1 \\
    \text{if } \text{count } \leq 0 \text{ then} \\
    \quad W &= W (1 - \frac{\text{skip}}{t+t_0}) \\
    \quad \text{count } &= \text{skip} \\
    \text{end if} \\
    t &= t + 1 \\
    \text{end while}
\]
Relation to other models

- Generalisation of Linear SVM on $x$:
  \[ H(x) = w^T x + b' \]
  - representable by $W = (w \ w \ ..)^\top$ and $b = (b' \ b' \ ..)^\top$

- Generalisation of Linear SVM on $\gamma$:
  \[ H(x) = \sum_{v \in C} \gamma_v(x) w_v \]
  - representable by $W = 0$ and $b = w$

- Generalisation of model selecting Latent(MI)-SVM:
  - representable by $\gamma = (0, 0, .., 1, .. 0)$
The finite kernel classifier takes the form

\[ H(x) = \gamma(x)^T W \Phi(x) + \gamma(x)^T b \]

where \( K(x_1, x_2) = \Phi(x_1) \Phi(x_2) \) is the kernel function

Weights \( W \) and \( b \) obtained as:

\[
\arg\min_{W,b} \frac{\lambda}{2} \|W\|^2 + \frac{1}{|S|} \sum_{k \in S} \max(0, 1 - y_k H_{w,b}(x_k))
\]
Experiments

MNIST, LETTER & USPS datasets

- Anchor points obtained using K-means clustering
- Coordinates evaluated on kNN (k = 8) (slow part)
- Coordinates obtained using weighted inverse distance
- Raw data used

CALTECH-101 (15 training samples per class)

- Coordinates evaluated on kNN (k = 5)
- Approximated Intersection kernel used (Vedaldi & Zisserman 2010)
- Spatial pyramid of BOW features
- Coordinates evaluated based on histogram over whole image
## Experiments

- **MNIST**

<table>
<thead>
<tr>
<th>Method</th>
<th>error</th>
<th>training time</th>
<th>test time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM (Bordes et al., 2009) (10 passes)</td>
<td>12.00%</td>
<td>1.5 s</td>
<td>8.75 µs</td>
</tr>
<tr>
<td>Linear SVM on LCC (Yu et al., 2009) (512 a.p.)</td>
<td>2.64%</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Linear SVM on LCC (Yu et al., 2009) (4096 a.p.)</td>
<td>1.90%</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>LibSVM (Chang &amp; Lin, 2001)</td>
<td>1.36%</td>
<td>17500 s</td>
<td>46 ms</td>
</tr>
<tr>
<td>LA-SVM (Bordes et al., 2005) (1 pass)</td>
<td>1.42%</td>
<td>4900 s</td>
<td>40.6 ms</td>
</tr>
<tr>
<td>LA-SVM (Bordes et al., 2005) (2 passes)</td>
<td>1.36%</td>
<td>12200 s</td>
<td>42.8 ms</td>
</tr>
<tr>
<td>MC-SVM (Crammer &amp; Singer, 2002)</td>
<td>1.44%</td>
<td>25000 s</td>
<td>N/A</td>
</tr>
<tr>
<td>SVMstruct (Tsochantaridis et al., 2005)</td>
<td>1.40%</td>
<td>265000 s</td>
<td>N/A</td>
</tr>
<tr>
<td>LA-RANK (Bordes et al., 2007) (1 pass)</td>
<td>1.41%</td>
<td>30000 s</td>
<td>N/A</td>
</tr>
<tr>
<td>LL-SVM (100 a.p., 10 passes)</td>
<td>1.85%</td>
<td>81.7 s</td>
<td>470 µs</td>
</tr>
</tbody>
</table>
Experiments

- **UIUC / LETTER**

<table>
<thead>
<tr>
<th>Method</th>
<th>error</th>
<th>training time</th>
<th>error</th>
<th>training time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM (Bordes et al., 2009)</td>
<td>9.57%</td>
<td>0.26 s</td>
<td>41.77%</td>
<td>0.18 s</td>
</tr>
<tr>
<td>MCSVM (Crammer &amp; Singer, 2002)</td>
<td>4.24%</td>
<td>60 s</td>
<td>2.42%</td>
<td>1200 s</td>
</tr>
<tr>
<td>SVM$_{struct}$ (Tsochantaridis et al., 2005)</td>
<td>4.38%</td>
<td>6300 s</td>
<td>2.40%</td>
<td>24000 s</td>
</tr>
<tr>
<td>LA-RANK (Bordes et al., 2007) (1 pass)</td>
<td>4.25%</td>
<td>85 s</td>
<td>2.80%</td>
<td>940 s</td>
</tr>
<tr>
<td>LL-SVM (10 passes)</td>
<td>5.78%</td>
<td>6.2 s</td>
<td>5.32%</td>
<td>4.2 s</td>
</tr>
</tbody>
</table>

- **Caltech-101**

<table>
<thead>
<tr>
<th>Method</th>
<th>accuracy</th>
<th>training time</th>
<th>test time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM (Bordes et al., 2009) (30 passes)</td>
<td>63.2%</td>
<td>605 s</td>
<td>3.1 ms</td>
</tr>
<tr>
<td>Intersection kernel SVM (Vedaldi &amp; Zisserman, 2010) (30 passes)</td>
<td>68.8%</td>
<td>3680 s</td>
<td>33 ms</td>
</tr>
<tr>
<td>SVM-KNN (Zhang et al., 2006)</td>
<td>59.1%</td>
<td>0 s</td>
<td>N/A</td>
</tr>
<tr>
<td>LLC (Wang et al., 2010)</td>
<td>65.4%</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MKL (Vedaldi et al., 2009)</td>
<td>71.1%</td>
<td>150000 s</td>
<td>1300 ms</td>
</tr>
<tr>
<td>NN (Boiman et al., 2008)</td>
<td>72.8%</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Locally linear SVM (30 passes)</td>
<td>66.9%</td>
<td>3400 s</td>
<td>25 ms</td>
</tr>
<tr>
<td>Locally additive SVM (30 passes)</td>
<td>70.1%</td>
<td>18200 s</td>
<td>190 ms</td>
</tr>
</tbody>
</table>
We propose novel Locally Linear SVM formulation with

- Good trade-off between speed and performance
- Scalability to large scale data sets
- Easy to implement

Optimal way of learning anchor points is an open question

Questions?