Deconstructing Data Science

David Bamman, UC Berkeley

Info 290
Lecture 15: Support Vector Machines

Mar 14, 2016
classification, so far

- Logistic regression
- Probabilistic graphical models
- Perceptron
- Decision trees
- Random forests
- Naive Bayes
Recall the perceptron

\[
\hat{y}_i = \begin{cases} 
1 & \text{if } \sum_{i}^{F} x_i \beta_i \geq 0 \\
-1 & \text{otherwise}
\end{cases}
\]

**Algorithm 4** Perceptron stochastic gradient descent

1: Data: training data \( x \in \mathbb{R}^F, y \in \{-1, 1\} \)
2: \( \beta = 0^F \)
3: \( \eta = 1 \)  \( \triangleright \) step size
4: while not converged do
5:      for \( i = 1 \) to \( N \) do
6:          \( \beta_{t+1} = \beta_t + \eta y_i x_i \)
7:      end for
8: end while
Recall the perceptron

- At the end of training, the coefficients $\beta$ are a linear combination of the inputs $x$

$$ \hat{\beta} = \sum_{i=1}^{N} a_i y_i x_i $$

- $a_i =$ the number of times data point $i$ was misclassified
Recall the perceptron

\[ \hat{y}_i = \begin{cases} 
1 & \text{if } \beta^\top x_i \geq 0 \\
-1 & \text{otherwise}
\end{cases} \]

\[ \hat{y}_i = \begin{cases} 
1 & \text{if } \left( \sum_{j=1}^{N} a_j y_j x_j \right)^\top x_i \geq 0 \\
-1 & \text{otherwise}
\end{cases} \]

\[ \hat{y}_i = \begin{cases} 
1 & \text{if } \sum_{j=1}^{N} a_j y_j \left( x_j^\top x_i \right) \geq 0 \\
-1 & \text{otherwise}
\end{cases} \]
Recall the perceptron

\[ \hat{y}_i = \begin{cases} 
1 & \text{if } \sum_{j=1}^{N} a_j y_j (x_j^T x_i) \geq 0 \\
-1 & \text{otherwise} 
\end{cases} \]

We can replace this inner product with a kernel.
Kernels

\[ k(x, x') \in \mathbb{R} \]

• Often symmetric — \( K(x', x) = K(x, x') \)
• And non-negative — \( K(x, x') \geq 0 \) (but need not be)
• Often thought of as a measure of “similarity”
Kernels

dot product = linear kernel

\[ \kappa(x, x') = x^\top x' = \sum_{i=1}^{F} x_i x_i' \]

cosine similarity kernel

\[ \kappa(x, x') = \frac{\sum_{i=1}^{F} x_i x_i'}{\sqrt{\sum_{i=1}^{F} x_i^2} \sqrt{\sum_{i=1}^{F} x_i'^2}} \]
Kernels

Gaussian kernel/RBF kernel

\[ \kappa(x, x') = \exp \left( -\frac{1}{2} \sum_{i=1}^{F} \frac{1}{\sigma^2_i} (x_i - x'_i)^2 \right) \]
Higher dimensions

\[ K(x, x') = (x^\top x')^2 \]

\[
= \left( \sum_{i=1}^{F} (x_i x'_i) \right)^2 \\
= \sum_{i=1}^{F} (x_i x'_i) \sum_{i=j}^{F} (x_j x'_j) 
\]
Higher dimensions

\[
\begin{align*}
&= \sum_{i=1}^{F} (x_i x'_i) \sum_{i=j}^{F} (x_j x'_j) \\
&= \sum_{i=1}^{F} \sum_{j=1}^{F} x_i x_j x'_i x'_j \\
&= \sum_{i,j=1}^{F} (x_i x_j)(x'_i x'_j)
\end{align*}
\]
Higher dimensions

\[ F = \sum_{i,j=1}^{F} (x_i x_j)(x'_i x'_j) \]

\[ \Phi(x) = \phi(x)^\top \phi(x') \]

Non-linear kernels imply a higher-dimensional feature representation for \( x \)
“Implicit” feature space

\[(x^\top x')^2 = \phi(x)^\top \phi(x')\]

<table>
<thead>
<tr>
<th>(x)</th>
<th>(x')</th>
<th>(\phi(x))</th>
<th>(\phi(x'))</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>X'1</td>
<td>X1X1</td>
<td>X'1X'1</td>
</tr>
<tr>
<td>X2</td>
<td>X'2</td>
<td>X1X2</td>
<td>X'1X'2</td>
</tr>
<tr>
<td>X3</td>
<td>X'3</td>
<td>X1X3</td>
<td>X'1X'3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X2X1</td>
<td>X'2X'1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X2X2</td>
<td>X'2X'2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X2X3</td>
<td>X'2X'3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X3X1</td>
<td>X'3X'1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X3X2</td>
<td>X'3X'2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X3X3</td>
<td>X'3X'3</td>
</tr>
</tbody>
</table>

original feature space

implied feature space
### Original Feature Space

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>1</td>
</tr>
<tr>
<td>not</td>
<td>1</td>
</tr>
<tr>
<td>movie</td>
<td>0</td>
</tr>
</tbody>
</table>

### Implied Feature Space

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>good good</td>
<td>1</td>
</tr>
<tr>
<td>good not</td>
<td>1</td>
</tr>
<tr>
<td>good movie</td>
<td>0</td>
</tr>
<tr>
<td>not good</td>
<td>1</td>
</tr>
<tr>
<td>not not</td>
<td>1</td>
</tr>
<tr>
<td>not movie</td>
<td>0</td>
</tr>
<tr>
<td>movie good</td>
<td>0</td>
</tr>
<tr>
<td>movie not</td>
<td>0</td>
</tr>
<tr>
<td>movie movie</td>
<td>0</td>
</tr>
</tbody>
</table>

\( \phi(x) \)

- \( x \)
Kernels

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29

so much depends upon

a red wheel barrow

glazed with rain water

beside the white chickens.
Code
Support vector machines

Two principles:

1. Kernel trick
2. Margin maximization
Margin

- Distance from the closest point to the decision boundary
Support vector machines

• For all of the training examples, we want to:
  • Maximize the margin
  • Subject to all of the training examples being on the correct side.
Loss functions

log loss
(logistic regression)

\[- \sum_{i=1}^{N} \log P(y \mid x, \beta) - \sum_{j=1}^{F} \beta_j^2\]

hinge loss
(SVM)

\[\sum_{i=1}^{N} \max(0, 1 - y \eta) - \sum_{j=1}^{F} \beta_j^2\]

No loss is suffered if the prediction is outside the margin on the correct side.
Hinge loss

$$\max(0, 1 - y\eta)$$

$$\eta = \text{score}$$
$$y = \{1, -1\}$$
Support vector machines

"slack variable"  \[ \xi_i = \max(0, 1 - y_i n_i) \]

\[
\arg \min_{\beta} \quad \text{loss} \quad C \frac{1}{n} \sum_{i=1}^{N} \xi_i + \sum_{j=1}^{F} \beta_j^2 \\
\text{regularization}
\]

s.t.: \[ y n_i \geq 1 - \xi \]
\[ \xi \geq 0 \]
Support vector machines

\[ \hat{\beta} = \sum_{i=1}^{N} \alpha_i y_i x_i \]

where \( \alpha_i = 0 \) for all \( x_i \) not on the margin

all \( x_i \) where \( \alpha_i \neq 0 \) are the support vectors

Same form as perceptron (with different semantics for \( \alpha \))
Support vectors

$$\hat{\beta} = \sum_{i=1}^{N} a_i y_i x_i$$

- The support vectors are the small set of training data points that are most important for determining the decision boundary
Support vector machines

\[ \hat{y} = \hat{\beta}^\top x \]

\[ \hat{y} = \sum_{i=1}^{N} a_i y_i x_i^\top x \]

\[ \hat{y} = \sum_{i=1}^{N} a_i y_i K(x_i, x) \]
Multiclass SVM

SVMs are inherently binary

One-versus-rest: K classifiers, one for each class versus all other classes

One-versus-one: K(K-1)/2 classifiers, one for each pair of classes
classification, so far
Genre classification

**[TABLE 1] Typical Features Used to Characterize Music Content.**

<table>
<thead>
<tr>
<th>TIMBRE</th>
<th>MELODY/HARMONY</th>
<th>RHYTHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture model: model of features over texture window: 1) Simple modeling with low-order statistics 2) Modeling with autoregressive model 3) Modeling with distribution estimation algorithms (for example, EM estimation of a GMM of frames)</td>
<td>Pitch function: measure of the energy in function of music notes 1) Unfolded function: describes pitch content and pitch range 2) Folded function: describes harmonic content</td>
<td>Periodicity function: measure of the periodicities of features 1) Tempo: periodicities typically in the range 0.3–1.5 s (i.e., 200–40 BPM) 2) Musical pattern: periodicities between 2 and 6 s (corresponding to the length of one or more measure bar)</td>
</tr>
</tbody>
</table>

**[TABLE 3] Confusion Matrix for the Dataset I and for the Algorithm Submitted by the Authors to MIREX 2005.**

<table>
<thead>
<tr>
<th>Truth Prediction</th>
<th>Ambient</th>
<th>Blues</th>
<th>Classic</th>
<th>Electronic</th>
<th>Ethnic</th>
<th>Folk</th>
<th>Jazz</th>
<th>New-Age</th>
<th>Punk</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient</td>
<td>52.94%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>7.32%</td>
<td>4.82%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>26.47%</td>
<td>0.00%</td>
<td>5.95%</td>
</tr>
<tr>
<td>Blues</td>
<td>0.00%</td>
<td>76.47%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>3.57%</td>
</tr>
<tr>
<td>Classic</td>
<td>2.94%</td>
<td>0.00%</td>
<td>100.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Electronic</td>
<td>5.88%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>53.66%</td>
<td>6.02%</td>
<td>4.17%</td>
<td>4.55%</td>
<td>5.88%</td>
<td>0.00%</td>
<td>19.05%</td>
</tr>
<tr>
<td>Ethnic</td>
<td>2.94%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>7.32%</td>
<td>12.50%</td>
<td>4.55%</td>
<td>4.55%</td>
<td>20.59%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Folk</td>
<td>0.00%</td>
<td>5.88%</td>
<td>1.22%</td>
<td>3.61%</td>
<td>6.02%</td>
<td>4.17%</td>
<td>4.55%</td>
<td>2.94%</td>
<td>0.00%</td>
<td>2.38%</td>
</tr>
<tr>
<td>Jazz</td>
<td>0.00%</td>
<td>2.94%</td>
<td>0.00%</td>
<td>3.66%</td>
<td>6.02%</td>
<td>4.17%</td>
<td>4.55%</td>
<td>8.82%</td>
<td>0.00%</td>
<td>5.95%</td>
</tr>
<tr>
<td>New Age</td>
<td>29.41%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>4.88%</td>
<td>4.82%</td>
<td>4.33%</td>
<td>4.55%</td>
<td>32.35%</td>
<td>0.00%</td>
<td>4.76%</td>
</tr>
<tr>
<td>Punk</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>4.17%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>52.38%</td>
</tr>
<tr>
<td>Rock</td>
<td>5.88%</td>
<td>14.71%</td>
<td>0.00%</td>
<td>21.95%</td>
<td>7.23%</td>
<td>0.00%</td>
<td>4.55%</td>
<td>2.94%</td>
<td>0.00%</td>
<td>4.76%</td>
</tr>
</tbody>
</table>
Midterm report

- 4 pages, citing 10 relevant sources
- Be sure to consider feedback!
- Data collection should be completed
- You should specify a validation strategy to be performed at the end
- Present initial experimental results
http://mybinder.org/repo/dbamman/dds