Classification

Jordan Boyd-Graber
University of Maryland

Slides adapted from Rob Schapire and Fei Xia
Motivation

- Binary and Multi-class: problems and classifiers
- Solving Multi-class problems with binary classifiers
  - One-vs-all
  - All pairs
  - Error correcting codes
Classification Problems

- **Natural binary**
  - Spam classification (spam vs. ham)
  - Segmentation (same or different)
  - Coreference

However, many are multiclass

- Topic classification
- Part of speech tagging
- Scene classification
Classification Problems

- Natural binary
  - Spam classification (spam vs. ham)
  - Segmentation (same or different)
  - Coreference

- However, many are multiclass
  - Topic classification
  - Part of speech tagging
  - Scene classification
Classifiers

- Some are directly multi-class (naïve Bayes, logistic regression, KNN)
- Other classifiers are basically binary
Classifiers

- Some are directly multi-class (naïve Bayes, logistic regression, KNN)
- Other classifiers are basically binary
  - SVM
  - Perceptron
  - Boosting
Reduction

Multiclass Data

\[
\begin{align*}
\langle \text{name=Cindy} , \text{age}=5 , \text{sex=F} \rangle, & \quad \text{yellow} \\
\langle \text{name=Marcia} , \text{age}=15 , \text{sex=F} \rangle, & \quad \text{red} \\
\langle \text{name=Bobby} , \text{age}=6 , \text{sex=M} \rangle, & \quad \text{blue} \\
\langle \text{name=Jan} , \text{age}=12 , \text{sex=F} \rangle, & \quad \text{yellow} \\
\langle \text{name=Peter} , \text{age}=13 , \text{sex=M} \rangle, & \quad \text{green}
\end{align*}
\]
Reduction

Multiclass Data

\[
\langle \text{name=Cindy}, \text{age=5}, \text{sex=F}\rangle, \\
\langle \text{name=Marcia}, \text{age=15}, \text{sex=F}\rangle, \\
\langle \text{name=Bobby}, \text{age=6}, \text{sex=M}\rangle, \\
\langle \text{name=Jan}, \text{age=12}, \text{sex=F}\rangle, \\
\langle \text{name=Peter}, \text{age=13}, \text{sex=M}\rangle,
\]

Binary Classifier

\[
(x_1, +), (x_2, -), (x_3, +), \ldots \rightarrow A \rightarrow h \rightarrow h(x) \in \{+, -\}
\]
Reduction

Multiclass Data

\[
\begin{align*}
&\{\text{name=Cindy, age=5, sex=F}\}, \quad \text{gold} \\
&\{\text{name=Marcia, age=15, sex=F}\}, \quad \text{red} \\
&\{\text{name=Bobby, age=6, sex=M}\}, \quad \text{blue} \\
&\{\text{name=Jan, age=12, sex=F}\}, \quad \text{yellow} \\
&\{\text{name=Peter, age=13, sex=M}\}, \quad \text{green}
\end{align*}
\]

Binary Classifier

\[
(x_1, +), (x_2, -), (x_3, +), \ldots \quad \rightarrow \quad A \quad \rightarrow \quad h \\
\quad \downarrow \\
\quad h(x) \in \{+, -\}
\]
One-Against-All

- Break $k$-class problem into $k$ binary problems and solve separately.
- Combine predictions: evaluate all $h$'s, hope exactly one is + (otherwise, take highest confidence).
One-Against-All

- Break \( k \)-class problem into \( k \) binary problems and solve separately
- Combine predictions: evaluate all \( h \)'s, hope exactly one is + (otherwise, take highest confidence)
- Incorrect prediction if only one is wrong
Does one vs. all work here?
Does one vs. all work here?

Discriminating between class 2 and the rest of the classes, the optimal halfspace would be the all negative classifier.
All-Pairs (Friedman; Hastie & Tibshirani)

- One binary problem for each pair of classes
- Take class with most positives and least negatives
- Faster and more accurate than one-against-all
Time Comparison

Assume training time is $\mathcal{O}(m^\alpha)$ and test time is $\mathcal{O}(c_t)$

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVA</td>
<td>$\mathcal{O}(km^\alpha)$</td>
<td>$\mathcal{O}(kc_t)$</td>
</tr>
<tr>
<td>All-pairs</td>
<td>$\mathcal{O}(k^2\left(\frac{m}{k}\right)^\alpha)$</td>
<td>$\mathcal{O}(k^2ct)$</td>
</tr>
</tbody>
</table>
Time Comparison

Assume training time is $O(m^\alpha)$ and test time is $O(c_t)$

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVA</td>
<td>$O(\frac{km^\alpha}{k})$</td>
<td>$O(\frac{kc_t}{k})$</td>
</tr>
<tr>
<td>All-pairs</td>
<td>$O(k^2(\frac{m}{k})^\alpha)$</td>
<td>$O(k^2c_t)$</td>
</tr>
</tbody>
</table>

OVA better for testing time, all-pairs better for training. (All-pairs usually better for performance.)
Error Correcting Output Codes (Dietterich & Bakiri)

- Reduce to binary using “coding” matrix
Error Correcting Output Codes (Dietterich & Bakiri)

- Reduce to binary using “coding” matrix
- Train classifier for each bit

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>( \downarrow )</td>
<td>(x_1)</td>
<td>( \downarrow )</td>
<td>(x_1)</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>(x_2)</td>
<td>( \downarrow )</td>
<td>(x_2)</td>
<td>( \downarrow )</td>
<td>(x_2)</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>(x_3)</td>
<td>( \downarrow )</td>
<td>(x_3)</td>
<td>( \downarrow )</td>
<td>(x_3)</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>(x_4)</td>
<td>( \downarrow )</td>
<td>(x_4)</td>
<td>( \downarrow )</td>
<td>(x_4)</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>(x_5)</td>
<td>( \downarrow )</td>
<td>(x_5)</td>
<td>( \downarrow )</td>
<td>(x_5)</td>
<td>( \downarrow )</td>
</tr>
</tbody>
</table>

\(h_1\), \(h_2\), \(h_3\), \(h_4\), \(h_5\)
Error Correcting Output Codes (Dietterich & Bakiri)

- Reduce to binary using “coding” matrix
- Train classifier for each bit

Choose closest row of coding matrix to predict
ECOC

- If rows of $M$ are far apart, will be robust to error
- Much faster if $k$ is large
- Disadvantage: binary problems may be unnatural
How to construct codes

- Exhaustive (if $k$ small): length $2^{k-1} - 1$
  - Row 1 has only ones
  - Row 2: $2^{k-2}$ zeros followed by $2^{k-2} - 1$ ones
  - Row 3: $2^{k-3}$ zeros, $2^{k-3}$ ones, $2^{k-3}$ zeros, $2^{k-3} - 1$ ones
  - ...
How to construct codes

- Exhaustive (if $k$ small): length $2^{k-1} - 1$
  - Row 1 has only ones
  - Row 2: $2^{k-2}$ zeros followed by $2^{k-2} - 1$ ones
  - Row 3: $2^{k-3}$ zeros, $2^{k-3}$ ones, $2^{k-3}$ zeros, $2^{k-3} - 1$ ones
  - ...

- Random codes: James and Hastie ’98 showed that this reduces variance through model averaging
That’s it for classification!

- You can implement multiple forms of classification
- Derive theoretical bounds for many classification tasks
- Today is bridge to the future: classification foundation of other ML tasks