Transforming Classifier Scores into Accurate Multiclass Probability Estimates

Bianca Zadrozny & Charles Elkan

Presenter: Myle Ott
Motivation
(the same old story)

• Easy to rank examples in order of class-
  membership likelihood
• Hard (or at least not trivial) to turn these
  rankings into probabilities of class-
  membership
• Goal: find $P(c \mid x)$: the probability of
  example $x$ belonging to class $c$
Talking Points

From ranking scores:
• “Obtaining accurate two-class probability estimates”
  – Isotonic regression
• “Obtaining accurate multi-class probability estimates”
“Obtaining accurate two-class probability estimates”

- Problem:

Interpreting re-scaled SVM scores as probabilities. (Note: rescaled based on the maximum and minimum seen distances from the hyperplane)
Platt’s Method

- Fit to a sigmoid:
Naïve Bayes

- Doesn’t work:

Platt’s method applied to Naïve Bayes.
Possible Solutions

• Binning
  – How many bins?
  – Why does it have to be a fixed number?
• Better method: isotonic (non decreasing) regression
  – Binning with variable number of bins
Isotonic Regression

- Pair(Pool)-Adjacent Violators (PAV)

\( \{x_i\}_{i=1}^N \): training examples
\( g(x_i) \): value of the function to be learned via IR
\( g^* \): the isotonic regression

If \( g \) is already isotonic, \( g^* = g \). Otherwise, \( \exists i \) s.t. \( g(x_{i-1}) > g(x_i) \) (i.e. decreasing).

In this case, \( x_{i-1} \) and \( x_i \) are called pair(pool)-adjacent violators.

This is solved by replacing both \( x_{i-1} \) and \( x_i \) by their average.

If this new set of examples is isotonic, \( g^*(x_{i-1}) = g^*(x_i) = \frac{g^*(x_{i-1}) + g^*(x_i)}{2} \), and \( g^*(x_j) = g(x_j) \).

This process is repeated until an isotonic set of values is obtained.

Make the set of training examples
Isotonic Regression

• Making use of the PAV algorithm:
  – Sort examples according to score
  – Let $g(x_i)=0$ if $x_i$ is negative, 1 if $x_i$ is positive
  – Run PAV algorithm on $g$ to get $g^*$
  – $g^*$ is the isotonic regression

• Usually has pretty good results

Typically, this results in 0/1 probabilities if the sorted scores rank examples perfectly, baseline in the random case, and something pretty effective otherwise.
Isotonic Regression
“Obtaining accurate multi-class probability estimates”

• Problem:
  – Calibration methods (Platt’s method, isotonic regression, etc.) are designed for two-class problems

Because “[because] we are mapping between one-dimensional spaces […] it is easy to impose sensible restrictions on the shape of the function being learned” (bottom of page 3, section 4)
“Obtaining accurate multi-class probability estimates”

• Solution:
  – Break the problem into many binary problems, calibrate them separately, and then combine the probabilities

• Two ways:
  – One-against-all: each class one by one
  – All-pairs: try each possible “pair” of classes

One against all: for each class, the problem is predicting “class c” or “not class c (I.e. some other class)”
All pairs: try each possible combination (pair) of classes
How do we “combine” the probabilities?

• One-against-all: since we have $P(c_i \mid x)$ for all $c_i$, just normalize the probabilities to 1.
• What about for all-pairs?
  – Construct a code matrix (a generalization of error-correcting output coding).
$$\begin{array}{ccc}
\text{b}_1 & \text{b}_2 & \text{b}_3 \\
\text{c}_1 & +1 & +1 & 0 \\
\text{c}_2 & -1 & 0 & +1 \\
\text{c}_3 & 0 & -1 & -1
\end{array}$$

- b’s represent various binary problems (all-pairs)
- c’s represent various classes
- +1 indicates that the corresponding c is the positive class in the corresponding binary problem b
- -1 … negative class
- 0 class not used in b
Combining the Probabilities

\[ r_b(x) = P\left(\bigvee_{c \in I} c \bigg| \bigvee_{c \in I \cup J} c, x\right) = \frac{\sum_{c \in I} P(c|x)}{\sum_{c \in I \cup J} P(c|x)} \]

- Where I and J are the sets of classes corresponding to \(M(-, b) = 1\) and \(M(-, b) = -1\), respectively.

Essentially, \(r_b(x)\) is equal to the probability of the positive class divided by the combined probabilities of the positive and negative classes (which should always be 1, right?) I only include this because it is included in the paper.
Combining the Probabilities

- There are two methods for solving this problem:
  - Least-squares method with non-negativity constraints
  - Coupling, an iterative algorithm for minimizing log-loss instead of squared error

These methods are not explained in the paper, but references are given.
Major things to note: PAV (i.e. isotonic regression) works in a way comparable to Platt’s method on SVMs and better for NB.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE Training</th>
<th>MSE Test</th>
<th>Error Rate Training</th>
<th>Error Rate Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>0.25112</td>
<td>0.25198</td>
<td>0.17100</td>
<td>0.17321</td>
</tr>
<tr>
<td>Sigmoid NB</td>
<td>0.21530</td>
<td>0.21515</td>
<td>0.15270</td>
<td>0.15190</td>
</tr>
<tr>
<td>PAV NB</td>
<td>0.20312</td>
<td>0.20452</td>
<td>0.14665</td>
<td>0.14831</td>
</tr>
<tr>
<td>SVM</td>
<td>0.28719</td>
<td>0.28684</td>
<td>0.15190</td>
<td>0.14968</td>
</tr>
<tr>
<td>Sigmoid SVM</td>
<td>0.20980</td>
<td>0.20962</td>
<td>0.15156</td>
<td>0.14993</td>
</tr>
<tr>
<td>PAV SVM</td>
<td>0.20815</td>
<td>0.20924</td>
<td>0.15115</td>
<td>0.15113</td>
</tr>
</tbody>
</table>

Table 3: MSE and error rate on the Adult dataset.
Results (multi-class)

Major things to note: Normalization is very close in performance to least-squares and coupling. PAV (i.e. isotonic regression) does help boost performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB Normalization</td>
<td>0.0326</td>
<td>0.1672</td>
</tr>
<tr>
<td>NB Least-Squares</td>
<td>0.0319</td>
<td>0.1672</td>
</tr>
<tr>
<td>NB Coupling</td>
<td>0.0304</td>
<td>0.1715</td>
</tr>
<tr>
<td>PAV NB Normalization</td>
<td>0.0241</td>
<td>0.1498</td>
</tr>
<tr>
<td>PAV NB Least-Squares</td>
<td>0.0260</td>
<td>0.1498</td>
</tr>
<tr>
<td>PAV NB Coupling</td>
<td>0.0260</td>
<td>0.1512</td>
</tr>
<tr>
<td>BNB Normalization</td>
<td>0.0163</td>
<td>0.0963</td>
</tr>
<tr>
<td>BNB Least-Squares</td>
<td>0.0164</td>
<td>0.0958</td>
</tr>
<tr>
<td>BNB Coupling</td>
<td>0.0160</td>
<td>0.1023</td>
</tr>
<tr>
<td>PAV BNB Normalization</td>
<td>0.0150</td>
<td>0.0946</td>
</tr>
<tr>
<td>PAV BNB Least-Squares</td>
<td>0.0150</td>
<td>0.0946</td>
</tr>
<tr>
<td>PAV BNB Coupling</td>
<td>0.0149</td>
<td>0.0935</td>
</tr>
</tbody>
</table>

Table 4: MSE and error rate on Pendigits (test set)
Conclusion

• Isotonic regression works for various models (i.e. SVMs and NB) in two-class problems

• One-against-all with normalized probabilities works well for multi-class problems, although using some of the more sophisticated methods might perform slightly better