Evaluating classifiers

CS440

The 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive prediction</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>negative prediction</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>
**Precision and recall**

- **Precision**: % of predicted items that are correct  \( P = \frac{tp}{(tp + fp)} \)
- **Recall**: % of correct items that are predicted  \( R = \frac{tp}{(tp + fn)} \)

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive prediction</td>
<td>( tp )</td>
<td>( fp )</td>
</tr>
<tr>
<td>negative prediction</td>
<td>( fn )</td>
<td>( tn )</td>
</tr>
</tbody>
</table>

**Combining precision and recall with the F measure**

- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):
  \[
  F = \frac{1}{\frac{1}{P} \alpha + \frac{1}{R} (1 - \alpha)} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}
  \]
- The harmonic mean is a very conservative average (tends towards the minimum)
- People usually use the F1 measure
  - i.e., with \( \beta = 1 \) (that is, \( \alpha = \frac{1}{2} \)):
  - \( F_1 = \frac{2PR}{(P+R)} \)
Prediction problems with more than two classes

• Dealing with muti-class problems
  – A document can belong one of several classes.

• Given test doc d,
  – Evaluate it for membership in each class
  – d belongs to the class which has the maximum score

Prediction problems with more than two classes

• Dealing with muti-label problems
  – A document can belong to 0, 1, or >1 classes.

• Given test doc d,
  – Evaluate it for membership in each class
  – d belongs to any class for which the score is higher than some threshold
Evaluation: the Reuters-21578 Data Set

- Most (over)used data set, 21,578 docs (each 90 types, 200 tokens)
- 9603 training, 3299 test articles (ModApte/Lewis split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Each document has 1.24 classes on average
- Only about 10 out of 118 categories are large

Common categories
(#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369, 119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

Reuters-21578 example document

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="#798">
<DATE>2-MAR-1987 16:51:43.42</DATE>
<TOPICS><D>livestock</D><D>hog</D></TOPICS>
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>
<DATELINE>CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter
</BODY></TEXT></REUTERS>
Confusion matrix

• For each pair of classes \(c_1, c_2\) how many documents from \(c_1\) were assigned to \(c_2\)?

<table>
<thead>
<tr>
<th>Docs in test set</th>
<th>Assigned UK</th>
<th>Assigned poultry</th>
<th>Assigned wheat</th>
<th>Assigned coffee</th>
<th>Assigned interest</th>
<th>Assigned trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>True UK</td>
<td>95</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>True poultry</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>True wheat</td>
<td>10</td>
<td>90</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>True coffee</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>True interest</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>13</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>True trade</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

Per class evaluation measures

**Recall:**

\[
\frac{c_{ii}}{\sum_j c_{ij}}
\]

Fraction of documents in class \(i\) classified correctly

**Precision:**

\[
\frac{c_{ii}}{\sum_j c_{ji}}
\]

Fraction of documents assigned class \(i\) that are actually about class \(i\)

**Accuracy:** (1 - error rate)

\[
\frac{\sum_i c_{ii}}{\sum_i \sum_j c_{ij}}
\]

Fraction of docs classified correctly
Micro- vs. Macro-Averaging

If we have two or more classes, how do we combine multiple performance measures into one quantity?

- **Macroaveraging**: Compute performance for each class, then average.
- **Microaveraging**: Collect decisions for all classes, compute contingency table, evaluate.

**Micro- vs. Macro-Averaging: Example**

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
<th>Micro Ave. Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class: yes</td>
<td>Truth: yes</td>
<td>10</td>
</tr>
<tr>
<td>Class: no</td>
<td>10</td>
<td>970</td>
</tr>
<tr>
<td>Class: yes</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>Class: no</td>
<td>10</td>
<td>890</td>
</tr>
<tr>
<td>Class: yes</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Class: no</td>
<td>20</td>
<td>1860</td>
</tr>
</tbody>
</table>

- Macroaveraged precision: \((0.5 + 0.9)/2 = 0.7\)
- Microaveraged precision: \(100/120 = .83\)
- Microaveraged score is dominated by score on common classes
Validation sets and cross-validation

- **Metric:** P/R/F1 or Accuracy
- **Unseen test set**
  - avoid overfitting (‘tuning to the test set’)
  - more realistic estimate of performance

Cross-validation over multiple splits
- Pool results over each split
- Compute pooled validation set performance

The Real World

- Gee, I’m building a text classifier for real, now!
- What should I do?
No training data? Manually written rules

If (wheat or grain) and not (whole or bread) then
Categorize as grain

• Need careful crafting
  – Human tuning on development data
  – Time-consuming

Very little data?

• Use Naïve Bayes
• Get more labeled data
  – Find clever ways to get humans to label data for you
• Try semi-supervised training methods
A reasonable amount of data?

• Perfect for all the clever classifiers
  – SVM
  – Regularized Logistic Regression
  – Random forests and boosting classifiers
  – Deep neural networks
• Look into decision trees
  – Considered a more interpretable classifier

A huge amount of data?

• Can achieve high accuracy!

• At a cost:
  – SVMs/deep neural networks (train time) or kNN (test time) can be too slow
  – Regularized logistic regression can be somewhat better
• So Naïve Bayes can come back into its own again!
Accuracy as a function of dataset size

• Classification accuracy improves with amount of data

Real-world systems generally combine:

• Automatic classification
• Manual review of uncertain/difficult cases
How to tweak performance

- Domain-specific features and weights: *very* important in real performance
- Sometimes need to collapse terms:
  - Part numbers, chemical formulas, …
  - But stemming generally doesn’t help
- Upweighting: Counting a word as if it occurred twice:
  - title words *(Cohen & Singer 1996)*
  - first sentence of each paragraph *(Murata, 1999)*
  - In sentences that contain title words *(Ko *et al*, 2002)*