Performance Measures for Machine Learning

Performance Measures

• Accuracy
• Weighted (Cost-Sensitive) Accuracy
• Lift
• ROC
  – ROC Area
• Precision/Recall
  – F
  – Break Even Point
• Similarity of Various Performance Metrics via MDS (Multi-Dimensional Scaling)

Accuracy

• Target: 0/1, -1/+1, True/False, …
• Prediction = f(inputs) = f(x): 0/1 or Real
• Threshold: f(x) > thresh => 1, else => 0
• If threshold(f(x)) and targets both 0/1:
  \[
  \text{accuracy} = \frac{\#\text{right}}{\#\text{total}} \text{ or } \frac{\sum_{i=1}^{N} (\text{target}_i \& \text{threshold}_i(f(x)))}{N}
  \]
• #right / #total
• p("correct"): p(threshold(f(x)) = target)

Confusion Matrix

\[
\begin{array}{cc}
\text{Predicted 1} & \text{Predicted 0} \\
\text{True 1} & a & b \\
\text{True 0} & c & d
\end{array}
\]

\[
\text{accuracy} = \frac{(a+d)}{(a+b+c+d)}
\]
Prediction Threshold

<table>
<thead>
<tr>
<th>Predicted 1</th>
<th>Predicted 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>True 1</td>
<td>TP</td>
</tr>
<tr>
<td>Predicted 0</td>
<td>FN</td>
</tr>
<tr>
<td>True 0</td>
<td>FP</td>
</tr>
<tr>
<td>Predicted 0</td>
<td>TN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted 1</th>
<th>Predicted 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>True 1</td>
<td>P(pr1</td>
</tr>
<tr>
<td>Predicted 0</td>
<td>P(pr0</td>
</tr>
<tr>
<td>True 0</td>
<td>P(pr1</td>
</tr>
<tr>
<td>Predicted 0</td>
<td>P(pr0</td>
</tr>
</tbody>
</table>

- threshold > MAX(f(x))
- all cases predicted 0
- (b+d) = total
- accuracy = %False = %0’s

- threshold < MIN(f(x))
- all cases predicted 1
- (a+c) = total
- accuracy = %True = %1’s

Problems with Accuracy

- Assumes equal cost for both kinds of errors
  - cost(b-type-error) = cost (c-type-error)
- is 99% accuracy good?
  - can be excellent, good, mediocre, poor, terrible
  - depends on problem
- is 10% accuracy bad?
  - information retrieval
- BaseRate = accuracy of predicting predominant class
  (on most problems obtaining BaseRate accuracy is easy)
Percent Reduction in Error

• 80% accuracy = 20% error
• suppose learning increases accuracy from 80% to 90%
• error reduced from 20% to 10%
• 50% reduction in error

• 99.90% to 99.99% = 90% reduction in error
• 50% to 75% = 50% reduction in error
• can be applied to many other measures

Costs (Error Weights)

Predicted 1 | Predicted 0
---|---
True 1 | \( w_a \) | \( w_b \)
True 0 | \( w_c \) | \( w_d \)

• Often \( w_a = w_d = \text{zero} \) and \( w_b \neq w_c \neq \text{zero} \)
Lift

- not interested in accuracy on entire dataset
- want accurate predictions for 5%, 10%, or 20% of dataset
- don’t care about remaining 95%, 90%, 80%, resp.
- typical application: marketing

\[
lift(\text{threshold}) = \frac{\% \text{ positives } > \text{ threshold}}{\% \text{ dataset } > \text{ threshold}}
\]

- how much better than random prediction on the fraction of the dataset predicted true (f(x) > threshold)

Lift

\[
lift = \frac{af(a+b)}{(a+c)(a+b+c+d)}
\]

Lift and Accuracy do not always correlate well

Problem 1

Problem 2

(thresholds arbitrarily set at 0.5 for both lift and accuracy)
**ROC Plot and ROC Area**

- Receiver Operator Characteristic
- Developed in WWII to statistically model false positive and false negative detections of radar operators
- Better statistical foundations than most other measures
- Standard measure in medicine and biology
- Becoming more popular in ML

**ROC Plot**

- Sweep threshold and plot
  - TPR vs. FPR
  - Sensitivity vs. 1-Specificity
  - $P(\text{true|true})$ vs. $P(\text{true|false})$
- Sensitivity = $a/(a+b)$ = LIFT numerator = Recall (see later)
- $1 - \text{Specificity} = 1 - d/(c+d)$

**Properties of ROC**

- ROC Area:
  - 1.0: perfect prediction
  - 0.9: excellent prediction
  - 0.8: good prediction
  - 0.7: mediocre prediction
  - 0.6: poor prediction
  - 0.5: random prediction
  - <0.5: something wrong!
Wilcoxon-Mann-Whitney

\[ ROCA = 1 \cdot \frac{\# \text{pairwise inversions}}{\# \text{POS} - \# \text{NEG}} \]

where

\[ \# \text{pairwise inversions} = \sum_{i,j} I\left[ (P(x_i) > P(x_j)) \land (T(x_i) < T(x_j)) \right] \]
Properties of ROC

- Slope is non-increasing
- Each point on ROC represents different tradeoff (cost ratio) between false positives and false negatives
- Slope of line tangent to curve defines the cost ratio
- ROC Area represents performance averaged over all possible cost ratios
- If two ROC curves do not intersect, one method dominates the other
- If two ROC curves intersect, one method is better for some cost ratios, and other method is better for other cost ratios

Precision and Recall

- typically used in document retrieval
- Precision:
  - how many of the returned documents are correct
  - precision(threshold)
- Recall:
  - how many of the positives does the model return
  - recall(threshold)
- Precision/Recall Curve: sweep thresholds

Precision/Recall

\[
\text{PRECISION} = \frac{a}{a + c} \quad \text{(for Predicted 1)}
\]

\[
\text{RECALL} = \frac{a}{a + b}
\]
Summary Stats: F & BreakEvenPt

- **PRECISION** = \( \frac{a}{a + c} \)
- **RECALL** = \( \frac{a}{a + b} \)
- \( F = \frac{2 \times (\text{PRECISION} \times \text{RECALL})}{\text{PRECISION} + \text{RECALL}} \)

BreakEvenPoint = PRECISION = RECALL

F and BreakEvenPoint do not always correlate well

Problem 1

Problem 2
Many Other Metrics

- Mitre F-Score
- Kappa score
- Balanced Accuracy
- RMSE (squared error)
- Log-loss (cross entropy)
- Calibration
  - reliability diagrams and summary scores
- ...

Summary

- the measure you optimize to makes a difference
- the measure you report makes a difference
- use measure appropriate for problem/community
- accuracy often is not sufficient/appropriate
- ROC is gaining popularity in the ML community
- only a few of these (e.g. accuracy) generalize easily to >2 classes
Different Models Best on Different Metrics

Really does matter what you optimize!
2-D Multi-Dimensional Scaling