Real World ML

Lyle Ungar

Evaluation metrics
The final project
Real-world ML projects
Future classes: visualization and causality
Evaluating ML

Lyle Ungar

Probability vs loss
Confusion matrix: TP/TN/FP/FN
Precision, Recall, Sensitivity, Specificity
ROC curves
What is Netflix trying to do?
Loss functions come from decision making

- We often optimize a loss function which is a surrogate for our true loss function.
- Don’t confuse *probability* or *score* with loss.
  - One can optimize a model for probability and then use the probability in a decision rule.
  - Or just directly optimize the loss resulting from a decision rule.
Regression loss function

- For a linear regression predicting dollar amounts (e.g. income, housing prices)
  - What is the loss function being optimized for
  - What is the residual plot likely to look like?

- Does this meet the assumptions of the linear regression model?
  - If not, how could you fix it?
Precision, Recall, Sensitivity, Specificity and ROC curves

Have you seen ROC curves?

A) Yes
B) No
## Ways to be right or wrong

<table>
<thead>
<tr>
<th>Claim\Is</th>
<th>True Yes</th>
<th>True No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classify Yes</td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>Classify No</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>

Accuracy = \frac{(TP + TN)}{(TP+FP+FN+TN)}
Measuring Performance

- **Accuracy (symmetric)**
  - % correctly classified

- **Asymmetric measures**
  - **Precision**
    - \( P(\text{yes} \mid \text{predicted as yes}) \)
  - **Recall (or Sensitivity)**
    - \( P(\text{predicted as yes} \mid \text{yes}) \)
  - **Specificity**
    - \( P(\text{predicted as no} \mid \text{no}) \)
### Precision/Recall Sensitivity/Specificity

<table>
<thead>
<tr>
<th>Claim\s</th>
<th>True Yes</th>
<th>True No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classify Yes</td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>Classify No</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>

- **Precision**
  - $P(\text{yes} \mid \text{predicted as yes}) = \frac{TP}{TP+FP}$
- **Recall (or Sensitivity)**
  - $P(\text{predicted as yes} \mid \text{yes}) = \frac{TP}{TP+FN}$
- **Specificity**
  - $P(\text{predicted as no} \mid \text{no}) = \frac{TN}{TN+FP}$
Precision/Recall Example

<table>
<thead>
<tr>
<th>Claim</th>
<th>True Good</th>
<th>True Not Good</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classify “Good”</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>Classify “Not good”</td>
<td>30</td>
<td>350</td>
</tr>
</tbody>
</table>

- **Precision**
  - \( P(\text{good} | \text{predicted as good}) = \frac{70}{70+50} \)

- **Recall (or Sensitivity) = True Positive Rate (TPR)**
  - \( P(\text{predicted as good} | \text{good}) = \frac{70}{70+30} \)

- **Specificity = 1 – (False Positive Rate)**
  - \( P(\text{predicted as bad} | \text{bad}) = \frac{350}{350+50} \)
F1 combines Precision and Recall

<table>
<thead>
<tr>
<th>Claim\s</th>
<th>True Yes</th>
<th>True No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classify Yes</td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>Classify No</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>

- Precision
  - $\frac{TP}{TP+FP}$

- Recall
  - $\frac{TP}{TP+FN}$

- F1
  - $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
ROC  (Receiver Operating Characteristic) Curve

- Sort all examples from highest probability (or score) of being ‘yes’, \( p(y=\text{‘yes’}|x) \), to lowest
- Sweep the threshold for predicting an example to be labeled ‘yes’ from 1 down to 0
  - This varies specificity from 1 to 0.
- At each threshold compute the sensitivity
  - i.e., the fraction of the true positives you found
- Plot the curve

https://en.wikipedia.org/wiki/Receiver_operating_characteristic
ROC Chart Varies Threshold

AUC = Area Under Curve

$p(y|x) > \text{threshold}$ to be in class
ROC charts support comparison

AUC = 0.5 is random guessing
AUC = 1.0 is perfection

AUC = Area Under the Curve
Where does google care about?

\[ p(y|x) > \text{threshold} \text{ to be in class} \]
Which method is most likely to be better for generating an ROC curve?
A) Logistic regression
B) SVM
### The Truth

<table>
<thead>
<tr>
<th>Has the disease</th>
<th>Does not have the disease</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True Positives</strong> (TP)</td>
<td><strong>False Positives</strong> (FP)</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td><strong>False Negatives</strong> (FN)</td>
<td><strong>True Negatives</strong> (TN)</td>
</tr>
<tr>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

PPV = \( \frac{TP}{TP + FP} \)

NPV = \( \frac{TN}{TN + FN} \)

### Sensitivity

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \quad \text{or} \quad \frac{a}{a + c}
\]

### Specificity

\[
\text{Specificity} = \frac{TN}{TN + FP} \quad \text{or} \quad \frac{d}{d + b}
\]
A confusion matrix shows the counts of the actual versus predicted class values.

Example (overall accuracy rate of 73.9%)

<table>
<thead>
<tr>
<th></th>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class A</td>
<td>20</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Class B</td>
<td>6</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>Class C</td>
<td>4</td>
<td>2</td>
<td>25</td>
</tr>
</tbody>
</table>
For the confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>purchase  no purchase</td>
</tr>
<tr>
<td>Predicted</td>
<td></td>
</tr>
<tr>
<td>purchase</td>
<td>10</td>
</tr>
<tr>
<td>no purchase</td>
<td>20</td>
</tr>
</tbody>
</table>

◆ What is its precision?
◆ What is its recall?
◆ How do you
  a) increase precision (but decrease recall)
  b) increase both precision and recall

<table>
<thead>
<tr>
<th></th>
<th>a) 10/20</th>
<th>b) 10/(10+20)</th>
<th>c) 10/60</th>
<th>d) 10/(10+60)</th>
<th>e) other</th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Optimizing for true utility

- Could one directly learn a model to optimize
  - An asymmetric loss function?
  - AUC?
You should know

- **Probability vs. loss**
  - Often use model to estimate score; then threshold for decision

- **Loss function vs. utility function**

- **Confusion matrix:**
  - TP/TN/FP/FN or TPR/TNR/FPR/FNR

- **Precision, Recall, Sensitivity, Specificity, F1**

- **ROC curves**
  - AUC
Final project

- Pick a group of 2-4 students – and a team name
- Pick a problem and data set
- Look up related problems
- Formulate as an ML problem
- Run 3-5 methods, plus a baseline
  - Optimize hyperparameters
  - Table showing results
- What can you do that is clever?
Final project deliverables

- **11/16 Project proposal**
  - Give us enough information to give you feedback
- **11/27 Project checkpoint**
  - Show that you are making progress
- **12/9 Project report, code and notebook**
Real World ML

- Loss functions
- Model form
  - Feature engineering
- Visualization
- Causality: “what if?”
What you should know

- Think about the true loss function
  - Distinguish modeling from decision making

- Think about the features
  - What do you have? What can you get?
  - How should they be regularized (blocks)

- Think about what ML methods fit best
  - Compare several