Evaluation Metrics
CS229

Yining Chen
(Adapted from slides by Anand Avati)
May 1, 2020
Topics

- Why are metrics important?
- Binary classifiers
  - Rank view, Thresholding
- Metrics
  - Confusion Matrix
  - Point metrics: Accuracy, Precision, Recall / Sensitivity, Specificity, F-score
- Choosing Metrics
- Class Imbalance
  - Failure scenarios for each metric
- Multi-class
Why are metrics important?

- Training objective (cost function) is only a proxy for real world objectives.
- Metrics help capture a business goal into a quantitative target (not all errors are equal).
- Helps organize ML team effort towards that target.
  - Generally in the form of improving that metric on the dev set.
- Useful to quantify the “gap” between:
  - Desired performance and baseline (estimate effort initially).
  - Desired performance and current performance.
  - Measure progress over time.
- Useful for lower level tasks and debugging (e.g. diagnosing bias vs variance).
- Ideally training objective should be the metric, but not always possible. Still, metrics are useful and important for evaluation.
Binary Classification

- x is input
- y is binary output (0/1)
- Model is $\hat{y} = h(x)$
- Two types of models
  - Models that output a categorical class directly (K-nearest neighbor, Decision tree)
  - Models that output a real valued score (SVM, Logistic Regression)
    - Score could be margin (SVM), probability (LR, NN)
    - Need to pick a threshold
    - We focus on this type (the other type can be interpreted as an instance)
Score based models

Example of Score: Output of logistic regression. For most metrics: Only ranking matters. If too many examples: Plot class-wise histogram.

\[
\text{Prevalence} = \frac{\# \text{ positive examples}}{\# \text{ positive examples} + \# \text{ negative examples}}
\]
Threshold -> Classifier -> Point Metrics

Label positive  Label negative

Th=0.5

Predict Positive

Th=0.5

Predict Negative

Th

0.5
Point metrics: Confusion Matrix

Properties:
- Total sum is fixed (population).
- Column sums are fixed (class-wise population).
- Quality of model & threshold decide how columns are split into rows.
- We want diagonals to be “heavy”, off diagonals to be “light”.
Point metrics: True Positives

```
<table>
<thead>
<tr>
<th>Label positive</th>
<th>Label negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predict Positive</td>
<td>Predict Negative</td>
</tr>
<tr>
<td>Th=0.5</td>
<td>Th=0.5</td>
</tr>
<tr>
<td><img src="image.png" alt="Diagram" /></td>
<td></td>
</tr>
<tr>
<td><img src="table.png" alt="Table" /></td>
<td></td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
</tr>
</tbody>
</table>
Point metrics: True Negatives

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>
Point metrics: False Positives

<table>
<thead>
<tr>
<th></th>
<th>Label positive</th>
<th>Label negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Predict Positive</td>
<td>Predict Negative</td>
</tr>
<tr>
<td>2</td>
<td>Predict Positive</td>
<td>Predict Negative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>
Point metrics: False Negatives

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
FP and FN also called Type-1 and Type-2 errors
Point metrics: Accuracy

Equivalent to 0-1 Loss!
Point metrics: Precision

Predict Negative                      Predict Positive

<table>
<thead>
<tr>
<th></th>
<th>Label positive</th>
<th>Label negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>TN</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Acc</td>
<td>.85</td>
<td></td>
</tr>
<tr>
<td>Pr</td>
<td>.81</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Acc</th>
<th>Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>.85</td>
<td>.81</td>
</tr>
</tbody>
</table>
Point metrics: Positive Recall (Sensitivity)

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Acc</th>
<th>Pr</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>.85</td>
<td>.81</td>
<td>.9</td>
</tr>
</tbody>
</table>

Trivial 100% recall = pull everybody above the threshold.
Trivial 100% precision = push everybody below the threshold except 1 green on top.
(Hopefully no gray above it!)

Striving for good precision with 100% recall = pulling up the lowest green as high as possible in the ranking.
Striving for good recall with 100% precision = pushing down the top gray as low as possible in the ranking.
Point metrics: Negative Recall (Specificity)

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Acc</th>
<th>Pr</th>
<th>Recall</th>
<th>Spec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>.85</td>
<td>.81</td>
<td>.9</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Point metrics: F1-score

\[
F_1 = \left( \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.
\]

<table>
<thead>
<tr>
<th>Th</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Acc</th>
<th>Pr</th>
<th>Recall</th>
<th>Spec</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>.85</td>
<td>.81</td>
<td>.9</td>
<td>.8</td>
<td>.857</td>
</tr>
</tbody>
</table>
Point metrics: Changing threshold

# effective thresholds = # examples + 1
<table>
<thead>
<tr>
<th>Threshold</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>Specificity</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0.50</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.95</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>9</td>
<td>0.55</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td>0.182</td>
</tr>
<tr>
<td>0.90</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>8</td>
<td>0.60</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
<td>0.333</td>
</tr>
<tr>
<td>0.85</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td>8</td>
<td>0.55</td>
<td>0.667</td>
<td>0.2</td>
<td>0.9</td>
<td>0.308</td>
</tr>
<tr>
<td>0.80</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>0.60</td>
<td>0.750</td>
<td>0.3</td>
<td>0.9</td>
<td>0.429</td>
</tr>
<tr>
<td>0.75</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>6</td>
<td>0.65</td>
<td>0.800</td>
<td>0.4</td>
<td>0.9</td>
<td>0.533</td>
</tr>
<tr>
<td>0.70</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>0.70</td>
<td>0.833</td>
<td>0.5</td>
<td>0.9</td>
<td>0.625</td>
</tr>
<tr>
<td>0.65</td>
<td>5</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>0.65</td>
<td>0.714</td>
<td>0.5</td>
<td>0.8</td>
<td>0.588</td>
</tr>
<tr>
<td>0.60</td>
<td>6</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>0.70</td>
<td>0.750</td>
<td>0.6</td>
<td>0.8</td>
<td>0.667</td>
</tr>
<tr>
<td>0.55</td>
<td>7</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>0.75</td>
<td>0.778</td>
<td>0.7</td>
<td>0.8</td>
<td>0.737</td>
</tr>
<tr>
<td>0.50</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>0.80</td>
<td>0.800</td>
<td>0.8</td>
<td>0.8</td>
<td>0.800</td>
</tr>
<tr>
<td>0.45</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>0.85</td>
<td>0.818</td>
<td>0.9</td>
<td>0.8</td>
<td>0.857</td>
</tr>
<tr>
<td>0.40</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>0.80</td>
<td>0.750</td>
<td>0.9</td>
<td>0.7</td>
<td>0.818</td>
</tr>
<tr>
<td>0.35</td>
<td>9</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>0.75</td>
<td>0.692</td>
<td>0.9</td>
<td>0.6</td>
<td>0.783</td>
</tr>
<tr>
<td>0.30</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>0.70</td>
<td>0.643</td>
<td>0.9</td>
<td>0.5</td>
<td>0.750</td>
</tr>
<tr>
<td>0.25</td>
<td>9</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>0.65</td>
<td>0.600</td>
<td>0.9</td>
<td>0.4</td>
<td>0.720</td>
</tr>
<tr>
<td>0.20</td>
<td>9</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>0.60</td>
<td>0.562</td>
<td>0.9</td>
<td>0.3</td>
<td>0.692</td>
</tr>
<tr>
<td>0.15</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>0.55</td>
<td>0.529</td>
<td>0.9</td>
<td>0.2</td>
<td>0.667</td>
</tr>
<tr>
<td>0.10</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>0.50</td>
<td>0.500</td>
<td>0.9</td>
<td>0.1</td>
<td>0.643</td>
</tr>
<tr>
<td>0.05</td>
<td>10</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0.55</td>
<td>0.526</td>
<td>1</td>
<td>0.1</td>
<td>0.690</td>
</tr>
<tr>
<td>0.00</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0.50</td>
<td>0.500</td>
<td>1</td>
<td>0</td>
<td>0.667</td>
</tr>
</tbody>
</table>

**Score = 1**

**Score = 0**

**Threshold Scanning**

Threshold = 1.00

Threshold = 0.00
Summary metrics: Rotated ROC (Sen vs. Spec)

Score = 1
Score = 0

Specificity = True Neg / Neg

Random Guessing

Pos examples

AUROC = Area Under ROC
= Prob[Random Pos ranked higher than random Neg]
Agnostic to prevalence!

Neg examples

Receiver operating characteristic

Sensitivity = True Pos / Pos

Sensitivity

Specificity

0.0 0.2 0.4 0.6 0.8 1.0

Sensitivity

ROC curve (area = 0.80)
Summary metrics: PRC (Recall vs. Precision)

Precision = True Pos / Predicted Pos

Score = 1

Score = 0

Recall = Sensitivity = True Pos / Pos

Precision

AUPRC = Area Under PRC

= Expected precision for Random threshold

When threshold = 0:

Precision = prevalence

Pos examples

Neg examples

Average Precision / AUPRC = 0.81

Recall = Sensitivity = True Pos / Pos
Summary metrics:

Two models scoring the same data set. Is one of them better than the other?
Summary metrics: Log-Loss vs Brier Score

- Same ranking, and therefore the same AUROC, AUPRC, accuracy!

\[
\text{Log Loss} = \frac{1}{N} \sum_{i=1}^{N} -y_i \log \hat{y}_i - (1 - y_i) \log (1 - \hat{y}_i).
\]

- Rewards confident correct answers, heavily penalizes confident wrong answers.

- One perfectly confident wrong prediction is fatal.

-> Well-calibrated model

- **Proper** scoring rule: Minimized at \( \hat{y} = y \)

\[
\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2
\]
Calibration vs Discriminative Power

Logistic (th=0.5):
- Precision: 0.872
- Recall: 0.851
- F1: 0.862
- Brier: 0.099

SVC (th=0.5):
- Precision: 0.872
- Recall: 0.852
- F1: 0.862
- Brier: 0.163

Fraction of Positives

Output

Histogram
Unsupervised Learning

- Log $P(x)$ is a measure of fit in Probabilistic models (GMM, Factor Analysis)
  - High log $P(x)$ on training set, but low log $P(x)$ on test set is a measure of overfitting
  - Raw value of log $P(x)$ hard to interpret in isolation

- K-means is trickier (because of fixed covariance assumption)
Class Imbalance

Symptom: Prevalence < 5% (no strict definition)

Metrics: May not be meaningful.

Learning: May not focus on minority class examples at all

(majority class can overwhelm logistic regression, to a lesser extent SVM)
What happen to the metrics under class imbalance?

Accuracy: Blindly predicts majority class -> prevalence is the baseline.

Log-Loss: Majority class can dominate the loss.

AUROC: Easy to keep AUC high by scoring most negatives very low.

AUPRC: Somewhat more robust than AUROC. But other challenges.

In general: Accuracy < AUROC < AUPRC
Score = 1

Score = 0

1%

1%

98%

Score = 0

Score = 1

“Fraudulent”

Specificity = True Neg / Neg

Rotated ROC

AUC = 98/99

Sensitivity = True Pos / Pos
Multi-class

- Confusion matrix will be N * N (still want heavy diagonals, light off-diagonals)
- Most metrics (except accuracy) generally analyzed as multiple 1-vs-many
- Multiclass variants of AUROC and AUPRC (micro vs macro averaging)
- Class imbalance is common (both in absolute and relative sense)
- Cost sensitive learning techniques (also helps in binary Imbalance)
  - Assign weights for each block in the confusion matrix.
  - Incorporate weights into the loss function.
Choosing Metrics

Some common patterns:

- High precision is hard constraint, do best recall (search engine results, grammar correction): Intolerant to FP
  - Metric: Recall at Precision = XX %
- High recall is hard constraint, do best precision (medical diagnosis): Intolerant to FN
  - Metric: Precision at Recall = 100 %
- Capacity constrained (by K)
  - Metric: Precision in top-K.
- ......
Thank You!