Lecture 03: Binary classification and related tasks

Meelis Kull
meelis.kull@ut.ee
Spring 2018
Previous Lecture 02 – Tasks, models, features

✓ Tasks

✓ Models
  ✓ Geometric models
  ✓ Probabilistic models
  ✓ Logical models
  ✓ Compositional models
  ✓ Ensemble models
  ✓ Grading vs grouping models

• Features
Features

• Features determine much of the success of a machine learning application:

  Machine learning model is only as good as its features

• Feature – mapping from the instance space into some domain, usually
  – integers (discrete/categorical feature) or
  – real numbers (continuous feature)
Feature construction

- Feature engineering or construction is an important part of the machine learning pipeline.
- Using and constructing the right features often makes a big difference in the results.
- Feature construction is a way of putting domain knowledge into the machine learning method.
  - E.g. the physicist might know that the target variable must be quadratically related to the first feature.
- Feature construction is crucial if the instances are highly structured.
  - E.g. predicting if a molecule is poisonous based on its molecular structure represented as a graph.
Feature construction and deep learning

- Deep learning methods can learn to construct some useful features automatically
- However, this generally requires huge amounts of training data
- For deep learning it also matters what the input features are
Examples of constructed features

• Discretization a continuous feature
  – E.g. introducing age ranges 0-10, 10-20,… 80+

• Normalization of a continuous feature
  – Rescaling the feature to have mean 0 and std dev 1

• Introduce polynomial features
  – E.g. $x, x^2, x^3$, ...

• Introduce interaction features
  – E.g. $x_1x_2, x_1x_2^2$, ...

• Log-scale and exponentiation
  – E.g. $\log(x), \exp(x)$
Constructed features for time-series

- In time-series there is usually:
  - Noise – a single measurement can be noisy
  - Auto-correlation – consecutive measurements are highly correlated
- Making a prediction at some moment it is important to use information from some window in the past
- Usually beneficial to extract features such as window average, window standard deviation, etc
Previous Lecture 02 – Tasks, models, features

✓ Tasks
✓ Models
  ✓ Geometric models
  ✓ Probabilistic models
  ✓ Logical models
  ✓ Compositional models
  ✓ Ensemble models
  ✓ Grading vs grouping models
✓ Features
Lectures 01-05

✓ Prologue: A machine learning sampler
✓ Chapter 1: The ingredients of machine learning
• Chapter 2: Binary classification and related tasks
• Chapter 3: Beyond binary classification
• Chapter 10: Features
• Chapter 12: Machine learning experiments
• + extra materials not covered in the book
Chapter 2: Binary classification and related tasks
Lecture 03 – Binary classification and related tasks

• Mathematical notation of supervised learning and classification
• Assessing classification performance
• Coverage plots and ROC plots
• Scoring classifiers and rankers
• Turning rankers into classifiers
• Mathematical notation of supervised learning and classification
• Assessing classification performance
• Coverage plots and ROC plots
• Scoring classifiers and rankers
• Turning rankers into classifiers
Supervised learning [Lecture 02]

- **Supervised learning** is a general term about machine learning tasks where:
  - Training data are labelled objects
  - The task is to predict labels on test data

- Tasks in supervised learning:
  - **Classification** (when labels are categorical)
  - **Regression** (when labels are real-valued)
  - **Structured prediction** (when labels are something more complicated)

- Terminology: target variable = label
Mathematical notation of supervised learning

- NB! Notations can vary across authors
- $\mathcal{X}$ - instance space (set of all possible instances)
- $\mathcal{L}$ - label space (set of all possible labels)
- $l : \mathcal{X} \to \mathcal{L}$ - (actual/true) labelling function
- $l(x)$ - the (actual/true) label of instance $x$
- $Tr$ - training set (of all training instances)
- $Te$ - test set (of all test instances) $Tr, Te \subset \mathcal{X}$
Mathematical notation of supervised learning

- $\mathcal{Y}$ - output space (set of all possible model outputs)

- Usually $\mathcal{Y} = \mathcal{L}$ (model outputs labels)
  - Then the task is to learn an approximation $\hat{l} : \mathcal{X} \rightarrow \mathcal{L}$ to the true labelling function $l$

- Other output spaces are possible:
  - E.g., the model could predict the probability of the instance to be positive: $\mathcal{Y} = [0, 1]$
Caveats of this simplified notation

• We stated that $Tr, Te \subset \mathcal{X}$ are sets
  – This implies that no instance occurs twice in $Tr, Te$
  – To accommodate multiple occurrences we could treat $Tr, Te$ as multisets

• We stated that $l : \mathcal{X} \rightarrow \mathcal{L}$ is a function
  – This does not apply to cases where label is not uniquely determined by features, e.g. due to noise in the data
  – To accommodate non-deterministic cases we should use notation with $Tr \subset \mathcal{X} \times \mathcal{L}$

• We will continue in the simplified notation
More notation

• Let the instance space consist of $d$ features
• $\mathcal{F}_i$ - domain of the i-th feature (set of all possible values of this feature)
• Then $\mathcal{X} = \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_d$ meaning that each instance is a vector of $d$ feature values
Classification notation

• In k-class classification model outputs a class label: \( \mathcal{Y} = \mathcal{L} = \{C_1, C_2, \ldots, C_k\} \)

• For more intuitive notation we denote the set of all classes as \( \mathcal{C} = \{C_1, C_2, \ldots, C_k\} \)

• The task is to find an approximation \( \hat{c} : \mathcal{X} \rightarrow \mathcal{C} \) to the actual class labelling function \( c : \mathcal{X} \rightarrow \mathcal{C} \)

• An instance \( x \) is classified correctly if \( \hat{c}(x) = c(x) \)
Binary classification notation

- In binary classification we refer to the two classes as:
  - Positive, denoted as $\oplus$ or $+1$
  - Negative, denoted as $\ominus$ or $-1$
## Supervised learning scenarios

<table>
<thead>
<tr>
<th>Task</th>
<th>Label space</th>
<th>Output space</th>
<th>Learning problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>$\mathcal{L} = \mathcal{C}$</td>
<td>$\mathcal{Y} = \mathcal{C}$</td>
<td>learn an approximation $\hat{c} : \mathcal{X} \rightarrow \mathcal{C}$ to the true labelling function $c$</td>
</tr>
<tr>
<td>Scoring and ranking</td>
<td>$\mathcal{L} = \mathcal{C}$</td>
<td>$\mathcal{Y} = \mathbb{R}^{\left</td>
<td>\mathcal{C}\right</td>
</tr>
<tr>
<td>Probability estimation</td>
<td>$\mathcal{L} = \mathcal{C}$</td>
<td>$\mathcal{Y} = [0,1]^{\left</td>
<td>\mathcal{C}\right</td>
</tr>
<tr>
<td>Regression</td>
<td>$\mathcal{L} = \mathbb{R}$</td>
<td>$\mathcal{Y} = \mathbb{R}$</td>
<td>learn an approximation $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ to the true labelling function $f$</td>
</tr>
</tbody>
</table>
Lecture 03 – Binary classification and related tasks

✓ Mathematical notation of supervised learning and classification
  • **Assessing classification performance**
  • Coverage plots and ROC plots
  • Scoring classifiers and rankers
  • Turning rankers into classifiers
Accuracy of classifiers

• Accuracy and error rate are the primary evaluation measures of classifiers

• Accuracy is the proportion of correctly classified instances

• Accuracy on the test set is:

\[ acc = \frac{1}{|Te|} \sum_{x \in Te} I[\hat{c}(x) = c(x)] \]

• Here \( I[\cdot] \) denotes the indicator function, which is 1 if its argument evaluates to true and 0 otherwise
Error rate of classifiers

- Error rate is the complement of accuracy
- Error rate is the proportion of incorrectly classified instances
- Error rate on the test set is:

\[ err = \frac{1}{|Te|} \sum_{x \in Te} I[\hat{c}(x) \neq c(x)] = 1 - acc \]

- Here \( I[\cdot] \) denotes the indicator function, which is 1 if its argument evaluates to true and 0 otherwise
Probability of error

• Test set error rate can be seen as an estimate of the probability of an error on a random instance $x \in \mathcal{X}$:

$$P_{\mathcal{X}}(\hat{c}(x) \neq c(x))$$

• Here $P_{\mathcal{X}}$ specifies a probability distribution over instance space $\mathcal{X}$ (all possible instances)
Terminological notes

• We will use the following terms interchangeably:
  – Evaluation measure
  – Evaluation function
  – Performance indicator

• If the measure equals zero for perfect predictions and higher value is worse, then we will also call it:
  – Loss measure
  – Loss function
Confusion matrix

- Often it is useful to see the kind of errors that the classifier makes

<table>
<thead>
<tr>
<th>Predicted ⊗</th>
<th>Predicted ⊘</th>
<th>Actual ⊗</th>
<th>Actual ⊘</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>20</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

- This example of a confusion matrix (also known as contingency table) shows prediction performance on 100 instances
**Terminology in binary classification**

- The counts in the confusion matrix are called as follows:

<table>
<thead>
<tr>
<th>Actual ☑</th>
<th>Predicted ☑</th>
<th>Predicted ☐</th>
<th>Actual positives</th>
<th>Actual negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>True positives</td>
<td>False negatives</td>
<td>Actual positives</td>
<td>Actual negatives</td>
</tr>
<tr>
<td>☐</td>
<td>False positives</td>
<td>True negatives</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted positives</th>
<th>Predicted negatives</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Notation in binary classification

- These counts are denoted as follows:

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>Actual</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑️</td>
<td>TP</td>
<td>☐️</td>
<td>FN</td>
</tr>
<tr>
<td>☐️</td>
<td>FP</td>
<td>☑️</td>
<td>TN</td>
</tr>
</tbody>
</table>

$$\begin{align*}
\text{Pos} & = PPos \\
\text{Neg} & = PNeg \\
|Te| & = |TP| + |TN| + |FP| + |FN|
\end{align*}$$
Confusion matrix

- Often it is useful to see the kind of errors that the classifier makes

<table>
<thead>
<tr>
<th></th>
<th>Predicted ☑</th>
<th>Predicted ☐</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual ☑</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Actual ☐</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

- This example of a confusion matrix (also known as contingency table) shows prediction performance on 100 instances
Confusion matrix and evaluation measures

- Accuracy and error rate can be calculated from the confusion matrix:

\[
\text{acc} = \frac{\text{TP} + \text{TN}}{|\text{Te}|} \quad \text{err} = \frac{\text{FP} + \text{FN}}{|\text{Te}|}
\]

- There are many other evaluation measures – Accuracy and error rate do not describe the full confusion matrix
Based on contingency table, the accuracy of classifier is

A. 30
B. 40
C. 0.3
D. 0.7
E. 100
F. None of these
G. I don’t know
True positive rate

- Often it is usual to assess the classifier separately on positives and negatives

- **True positive rate** is the proportion of positives correctly classified

\[
 tpr = \frac{TP}{Pos} = \frac{\sum_{x \in Te} I[\hat{c}(x) = c(x) = \oplus]}{\sum_{x \in Te} I[c(x) = \oplus]}
\]

- TPR is the estimator of:

\[
 P_{\mathcal{X}}(\hat{c}(x) = \oplus | c(x) = \oplus)
\]
**True negative rate**

- **True negative rate** is the proportion of negatives correctly classified (accuracy on the negatives)

\[
\text{tnr} = \frac{\text{TN}}{\text{Neg}} = \frac{\sum_{x \in \text{Te}} I[\hat{c}(x) = c(x) = \emptyset]}{\sum_{x \in \text{Te}} I[c(x) = \emptyset]}
\]

- TNR is the estimator of:

\[
P_{\mathcal{X}}(\hat{c}(x) = \emptyset \mid c(x) = \emptyset)
\]
Per-class error rates: FNR, FPR

- False negative rate (error rate on positives):
  \[
  fnr = \frac{\text{FN}}{\text{Pos}} = \frac{\sum_{x \in \text{Te}} I[\hat{c}(x) = \Theta, c(x) = \oplus]}{\sum_{x \in \text{Te}} I[c(x) = \oplus]}
  \]

- False positive rate (error rate on negatives):
  \[
  fpr = \frac{\text{FP}}{\text{Neg}} = \frac{\sum_{x \in \text{Te}} I[\hat{c}(x) = \oplus, c(x) = \ominus]}{\sum_{x \in \text{Te}} I[c(x) = \ominus]}
  \]
Synonyms for the per-class measures

- **TPR** – sensitivity, recall
  \[ tpr = \frac{TP}{Pos} \]
- **TNR** – specificity
  \[ tnr = \frac{TN}{Neg} \]
- **FNR** – miss rate
  \[ fnr = \frac{FN}{Pos} \]
- **FPR** – false alarm rate
  \[ fpr = \frac{FP}{Neg} \]
Based on contingency table the TPR of classifier is

A. 30
B. 20
C. 0.3
D. 0.6
E. None of these
F. I don’t know
Precision

- **Precision** (also known as positive predictive value) is the proportion of actual positives among predicted positives

\[ prec = \frac{TP}{PPos} = \frac{\sum_{x \in Te} I[\hat{c}(x) = c(x) = \oplus]}{\sum_{x \in Te} I[\hat{c}(x) = \oplus]} \]

- Precision is the estimator of:

\[ P_{\mathcal{X}}(c(x) = \oplus | \hat{c}(x) = \oplus) \]
Based on contingency table, the precision of classifier is

A. 0.3
B. 0.6
C. 0.75 (Correct)
D. 0.9
E. 1.0
F. None of these
G. I don’t know

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>Actual</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
# Summary (1)

<table>
<thead>
<tr>
<th>Measure</th>
<th>Definition</th>
<th>Equal to</th>
<th>Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of positives</td>
<td>$\text{Pos} = \sum_{x \in \mathcal{T}_e} I[c(x) = \oplus]$</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>number of negatives</td>
<td>$\text{Neg} = \sum_{x \in \mathcal{T}_e} I[c(x) = \ominus]$</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>number of true positives</td>
<td>$\text{TP} = \sum_{x \in \mathcal{T}_e} I[\hat{c}(x) = c(x) = \oplus]$</td>
<td></td>
<td>$\text{Neg} - \text{TN}$</td>
</tr>
<tr>
<td>number of true negatives</td>
<td>$\text{TN} = \sum_{x \in \mathcal{T}_e} I[\hat{c}(x) = c(x) = \ominus]$</td>
<td></td>
<td>$\text{Pos} - \text{TP}$</td>
</tr>
<tr>
<td>number of false positives</td>
<td>$\text{FP} = \sum_{x \in \mathcal{T}_e} I[\hat{c}(x) = \oplus, c(x) = \ominus]$</td>
<td></td>
<td>$\text{Neg} - \text{TN}$</td>
</tr>
<tr>
<td>number of false negatives</td>
<td>$\text{FN} = \sum_{x \in \mathcal{T}_e} I[\hat{c}(x) = \ominus, c(x) = \oplus]$</td>
<td></td>
<td>$\text{Pos} - \text{TP}$</td>
</tr>
<tr>
<td>proportion of positives</td>
<td>$\text{pos} = \frac{1}{</td>
<td>\mathcal{T}_e</td>
<td>} \sum_{x \in \mathcal{T}_e} I[c(x) = \oplus]$</td>
</tr>
<tr>
<td>proportion of negatives</td>
<td>$\text{neg} = \frac{1}{</td>
<td>\mathcal{T}_e</td>
<td>} \sum_{x \in \mathcal{T}_e} I[c(x) = \ominus]$</td>
</tr>
<tr>
<td>class ratio</td>
<td>$\text{clr} = \text{pos} / \text{neg}$</td>
<td></td>
<td>$\text{Pos} / \text{Neg}$</td>
</tr>
<tr>
<td>accuracy</td>
<td>$\text{acc} = \frac{1}{</td>
<td>\mathcal{T}_e</td>
<td>} \sum_{x \in \mathcal{T}_e} I[\hat{c}(x) = c(x)]$</td>
</tr>
<tr>
<td>error rate</td>
<td>$\text{err} = \frac{1}{</td>
<td>\mathcal{T}_e</td>
<td>} \sum_{x \in \mathcal{T}_e} I[\hat{c}(x) \neq c(x)]$</td>
</tr>
</tbody>
</table>
## Summary (2)

<table>
<thead>
<tr>
<th>Measure</th>
<th>Definition</th>
<th>Equal to</th>
<th>Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>true positive rate, sensitivity, recall</td>
<td>( tpr = \frac{\sum_{x \in Te} I[\hat{c}(x) = c(x) = \oplus]}{\sum_{x \in Te} I[c(x) = \oplus]} )</td>
<td>( TP/Pos )</td>
<td>( P(\hat{c}(x) = \oplus</td>
</tr>
<tr>
<td>true negative rate, specificity</td>
<td>( tnr = \frac{\sum_{x \in Te} I[\hat{c}(x) = c(x) = \ominus]}{\sum_{x \in Te} I[c(x) = \ominus]} )</td>
<td>( TN/Neg )</td>
<td>( P(\hat{c}(x) = \ominus</td>
</tr>
<tr>
<td>false positive rate, false alarm rate</td>
<td>( fpr = \frac{\sum_{x \in Te} I[\hat{c}(x) = \oplus, c(x) = \ominus]}{\sum_{x \in Te} I[c(x) = \ominus]} )</td>
<td>( FP/Neg = 1 - tnr )</td>
<td>( P(\hat{c}(x) = \oplus</td>
</tr>
<tr>
<td>false negative rate</td>
<td>( fnr = \frac{\sum_{x \in Te} I[\hat{c}(x) = \ominus, c(x) = \oplus]}{\sum_{x \in Te} I[c(x) = \oplus]} )</td>
<td>( FN/Pos = 1 - tpr )</td>
<td>( P(\hat{c}(x) = \ominus</td>
</tr>
<tr>
<td>precision, confidence</td>
<td>( prec = \frac{\sum_{x \in Te} I[\hat{c}(x) = c(x) = \oplus]}{\sum_{x \in Te} I[\hat{c}(x) = \oplus]} )</td>
<td>( TP/(TP + FP) )</td>
<td>( P(c(x) = \oplus</td>
</tr>
</tbody>
</table>
Accuracy from per-class accuracies

• Accuracy can be calculated from the per-class accuracies (TPR, TNR):

\[ \text{acc} = \text{pos} \cdot \text{tpr} + \text{neg} \cdot \text{tnr} \]

• Example:

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted ⊕</th>
<th>Predicted ⊖</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊕</td>
<td>60</td>
<td>15</td>
<td>75</td>
</tr>
<tr>
<td>⊖</td>
<td>10</td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>30</td>
<td>100</td>
</tr>
</tbody>
</table>

\[ \text{acc} = (60 + 15)/100 = 0.75 \]

\[ \text{pos} = 75/100 = 0.75 \]

\[ \text{tpr} = 60/75 = 0.80 \]

\[ \text{neg} = 25/100 = 0.25 \]

\[ \text{tnr} = 15/25 = 0.60 \]
Example: ‘Reluctant’ search engine

- Consider a query to internet search engine
- Each web page is either relevant or not
- Suppose 1 relevant page in 1000 \((pos = \frac{1}{1000})\)
- Consider ‘reluctant’ search engine that doesn’t return any pages (all predicted)

\[
acc = pos \cdot tpr + neg \cdot tnr
\]

\[
0.999 = 0.001 \cdot 0.0 + 0.999 \cdot 1.0
\]

- Accuracy is high - a random page on the web classified correctly with probability 99.9%
Example: ‘Reluctant’ search engine

- We want a better TPR (usually results in lower TNR and hence lower accuracy)
- Accuracy not meaningful in this context and should not be optimized for
- This is true more generally for cases where we care mostly about the minority class and not so much about the majority class
- Precision, recall and F-measure are then used instead of TNR, TPR and accuracy

\[ F = \frac{\frac{2}{1/\text{prec}+1/rec}} \] (harmonic mean of precision and recall)
Lecture 03 – Binary classification and related tasks

✓ Mathematical notation of supervised learning and classification
✓ Assessing classification performance
  • Coverage plots and ROC plots
  • Scoring classifiers and rankers
  • Turning rankers into classifiers
How to visualize?

• How to visualize the contents of the contingency table?

<table>
<thead>
<tr>
<th>Predicted ⊕</th>
<th>Predicted ⊖</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual ⊕</td>
<td>30</td>
</tr>
<tr>
<td>Actual ⊖</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
</tbody>
</table>

• Fully determined by 4 values:

<table>
<thead>
<tr>
<th>Predicted ⊕</th>
<th>Predicted ⊖</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual ⊕</td>
<td>30</td>
</tr>
<tr>
<td>Actual ⊖</td>
<td>10</td>
</tr>
</tbody>
</table>
Coverage plot

NB!
Related to but not the same as ROC plot (axes ranges are different)
Coverage plot

NB!
Related to but not the same as ROC plot (axes ranges are different)
Coverage plot

• Coverage plot:
  – X-axis: **FP** (false positives), range $[0, Neg]$
  – Y-axis: **TP** (true positives), range $[0, Pos]$
Coverage plot

```
<table>
<thead>
<tr>
<th></th>
<th>Actual</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>C2</td>
<td>20</td>
<td>50</td>
</tr>
</tbody>
</table>
```

Points:
- **C1**: (10, 50)
- **C2**: (20, 50)
Coverage plot

Meelis Kull - Spring 2018 - MTAT.03.227 - Machine Learning - Lecture 03
Coverage plot with imbalanced classes

<table>
<thead>
<tr>
<th>Positives</th>
<th>Predicted ⊕</th>
<th>Predicted ⊗</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual ⊕</td>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>Actual ⊗</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Classifier dominance

- Definition: classifier C1 **dominates** over classifier C2 if it has better (or same) per-class accuracy on each class.

- In binary classification this is equivalent to higher TPR and higher TNR:

  \[ tpr_1 > tpr_2 \iff \frac{TP_1}{Pos} > \frac{TP_2}{Pos} \iff TP_1 > TP_2 \]

  \[ tnr_1 > tnr_2 \iff fpr_1 < fpr_2 \iff \frac{FP_1}{Neg} < \frac{FP_2}{Neg} \iff FP_1 < FP_2 \]

- Therefore, dominance can be seen visually from coverage plots.
Coverage plot

\[ TP_1 > TP_2 \]

\[ FP_1 < FP_2 \]
Domination example

C1 dominates C2
C3 dominates C2
C1 does not dominate C3
C3 does not dominate C1
Domination example

C1 dominates C2
C3 dominates C2
C1 does not dominate C3
C3 does not dominate C1

All classifiers on an ascending line with slope 1 (45 degrees) have the same accuracy
Domination example

All classifiers on an ascending line with slope 1 (45 degrees) have the same accuracy.

Imagine travelling up that line: gaining true positives, losing true negatives.

C1 dominates C2
C3 dominates C2
C1 does not dominate C3
C3 does not dominate C1
From coverage plot to ROC plot

- Coverage plot has X-axis \([0, \text{Neg}]\) and Y-axis \([0, \text{Pos}]\)
- If we renormalize both to \([0, 1]\):
  - \(\text{FP}/\text{Neg} = fpr\)
  - \(\text{TP}/\text{Pos} = tpr\)
- Then we get the ROC plot
  (Receiver Operating Characteristic):
  - X-axis: FPR
  - Y-axis: TPR
ROC plot
ROC plot

ROC heaven = perfect classifier

Always positive

Always negative

ROC hell = always wrong

False positive rate

True positive rate
Lecture 03 – Binary classification and related tasks

✓ Mathematical notation of supervised learning and classification
✓ Assessing classification performance
✓ Coverage plots and ROC plots
  • Scoring classifiers and rankers
  • Turning rankers into classifiers
Scoring classifier

• In binary classification, a scoring classifier is a mapping \( \hat{s} : \mathcal{X} \rightarrow \mathbb{R} \)

• The score \( \hat{s}(x) \) indicates how likely it is that the instance \( x \) is a positive (higher value of \( \hat{s}(x) \) means more likely)

• Most classifier learning algorithms can return a scoring classifier, if needed
  – KNN – number of neighbours predicting positive
  – SVM – signed distance from decision boundary
  – NB – probability to be positive
Scoring and ranking

- A scoring classifier can be applied to rank all test instances according to how likely they are to be positive.

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
Thresholding

- A scoring classifier can be thresholded to obtain a binary classifier (positive if the score above the threshold)

\[ w \]

Threshold A: \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)

Threshold B: \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)

Threshold C: \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)
How good is a ranking?

• In a good ranking positives come before negatives
• Ranking error – a negative before a positive
• Number of ranking errors:

$$\sum_{x \in Te^+, x' \in Te^\ominus} I[\hat{s}(x) < \hat{s}(x')]$$

where

$$Te^\oplus = \{x \in Te \mid c(x) = \oplus\}$$

$$Te^\ominus = \{x \in Te \mid c(x) = \ominus\}$$
Ranking error rate and accuracy

- Additionally, if $\hat{s}(x) = \hat{s}(x')$ (a scoring tie) then to be counted as half of ranking error.

- Ranking error rate = number of ranking errors in proportion to maximal possible

$$rank\text{-}err = \frac{\sum_{x \in T^+, x' \in T^\Theta} I[\hat{s}(x) < \hat{s}(x')] + \frac{1}{2} I[\hat{s}(x) = \hat{s}(x')]}{Pos \cdot Neg}$$

- Ranking accuracy: \(rank\text{-}acc = 1 - rank\text{-}err\)
  - Estimate of the probability that an arbitrary positive-negative pair is ranked correctly
Example of ranking accuracy

• Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5

• 4 ranking errors:
  – n1 before p4
  – n1 before p5
  – n2 before p5
  – n3 before p5

Ranking error rate = Proportion of red area = 4/25 = 0.16
Ranking accuracy = Proportion of green area = 21/25 = 0.84
Count the number of ranking errors in the following ranking: $+1, +1, -1, +1, -1, +1, -1$

A. 0  
B. 1  
C. 2  
D. 3  
E. 4  
F. 5  
G. None of these  
H. I don’t know
Example of ranking accuracy

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
- 4 ranking errors:
  - n1 before p4
  - n1 before p5
  - n2 before p5
  - n3 before p5

Ranking error rate = \frac{\text{Propotion of red area}}{\text{Total area}} = \frac{4}{25} = 0.16
Ranking accuracy = \frac{\text{Propotion of green area}}{\text{Total area}} = \frac{21}{25} = 0.84
Coverage plot shows ranking performance

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
Coverage plot shows ranking performance

- Ranking: $p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5$
Coverage plot shows ranking performance

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
Coverage plot shows ranking performance

• Ranking: \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)
Coverage plot shows ranking performance

- **Ranking:** \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)
Coverage plot shows ranking performance

- Ranking: $p_1\text{-}p_2\text{-}p_3\text{-}n_1\text{-}p_4\text{-}n_2\text{-}n_3\text{-}p_5\text{-}n_4\text{-}n_5$
Coverage plot shows ranking performance

• Ranking: $p1-p2-p3-n1-p4-n2-n3-p5-n4-n5$
Coverage plot shows ranking performance

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
Coverage plot shows ranking performance

• Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
Coverage plot shows ranking performance

- Ranking: \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)
Coverage plot shows ranking performance

- **Ranking:** \( p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5 \)
Coverage plot shows ranking performance

- Ranking: \( p_1 - p_2 - p_3 - n_1 - p_4 - n_2 - n_3 - p_5 - n_4 - n_5 \)
Coverage plot shows ranking performance

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5
Coverage plot shows ranking performance

- Ranking: \( p_1 - p_2 - p_3 - n_1 - p_4 - n_2 - n_3 - p_5 - n_4 - n_5 \)
Coverage plot shows ranking performance

- **Ranking:** $p_1-p_2-p_3-n_1-p_4-n_2-n_3-p_5-n_4-n_5$
Coverage plot shows ranking performance

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5

Proportion of area under coverage plot = ranking accuracy = 21/25 = 0.84
ROC plot shows ranking performance

- Ranking: p1-p2-p3-n1-p4-n2-n3-p5-n4-n5

\[
\text{AUC} = \text{Area under ROC curve} = \frac{21}{25} = 0.84
\]
Example: decision tree

- Instances within a leaf share the same score
- We denote such a tie between 20 positives and 5 negatives as $[20+,5-]$

\[ \hat{s}(x) = +2 \]
\[ \hat{s}(x) = -1 \]
\[ \hat{s}(x) = +1 \]

- Ranking: $[20+,5-] [10+,5-] [20+,40-]$
Example: decision tree

- Ranking: \([20+, 5-] [10+, 5-] [20+, 40-]\)
- Full errors (negatives before positives):
  \[5 \cdot 10 + 5 \cdot 20 + 5 \cdot 20 = 250\]
- Half-errors (negatives tied with positives):
  \[20 \cdot 5 + 10 \cdot 5 + 20 \cdot 40 = 950\]
- Maximum possible ranking errors:
  \[50 \cdot 50 = 2500\]
- Ranking error rate:
  \[
  \frac{(250 + 950/2)}{2500} = \frac{725}{2500} = 0.29
  \]
- Ranking accuracy:
  \[1 - 0.29 = 0.71\]
Example: decision tree

- **Ranking**: [20+, 5-] [10+, 5-] [20+, 40-]
  - Threshold A: [20+, 5-] [10+, 5-] [20+, 40-]
  - Threshold B: [20+, 5-] [10+, 5-] [20+, 40-]
  - Threshold C: [20+, 5-] [10+, 5-] [20+, 40-]
  - Threshold D: [20+, 5-] [10+, 5-] [20+, 40-]
Linear interpolation in ROC plots

• Why can we interpolate linearly in the ROC plots?

• Define a new classifier

• With probability $p$ use classifier C:
  - $[20+,5-] [10+,5-] [20+,40-]$

• With probability $1-p$ use classifier D:
  - $[20+,5-] [10+,5-] [20+,40-]$

\[ \text{spam: 20 ham: 5} \]
\[ \text{spam: 10 ham: 5} \]
\[ \text{spam: 20 ham: 40} \]
\[ \text{spam: 10 ham: 5} \]
\[ \text{$\delta(x) = -1$} \]
\[ \text{$\delta(x) = +1$} \]
\[ \text{$\delta(x) = +2$} \]
Linear interpolation in ROC plots

- \( p \) : \([20+,5-] [10+,5-] [20+,40-]\)
- \( 1 - p \) : \([20+,5-] [10+,5-] [20+,40-]\)
- \( fpr = (5 + 5 + 40(1 - p)) / 50 = 1 - 0.8p \)
- \( tpr = (20 + 10 + 20(1 - p)) / 50 = 1 - 0.4p \)

\[
p = 1 \Rightarrow (fpr, tpr) = (0.2, 0.6)
\]

\[
p = 0.5 \Rightarrow (fpr, tpr) = (0.6, 0.8)
\]

\[
p = 0 \Rightarrow (fpr, tpr) = (1.0, 1.0)
\]
Lecture 03 – Binary classification and related tasks

✓ Mathematical notation of supervised learning and classification
✓ Assessing classification performance
✓ Coverage plots and ROC plots
✓ Scoring classifiers and rankers

• Turning rankers into classifiers
Turning rankers into classifiers

• Where to threshold a ranker?
• Depends on the evaluation measure
• Simple general algorithm:
  – Try out all thresholds
  – Find threshold which has best performance according to the chosen evaluation measure
• When maximizing accuracy:
  – Draw a line with slope \( \frac{1}{clr} = \frac{\text{Neg}}{\text{Pos}} \) through ROC heaven and slide it downwards
  – Best threshold is where it hits the ROC curve
Example of optimizing accuracy

- Scores:
  - 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), 0.13 (ham)

- Slope of accuracy isometric:
  \[ \frac{\text{Neg}}{\text{Pos}} = \frac{4}{6} \]
Example of optimizing accuracy

- Scores:
  - 0.89 (spam), 0.80 (spam),
  - 0.74 (ham), 0.71 (spam),
  - 0.63 (spam), 0.49 (ham),
  - 0.42 (spam), 0.32 (spam),
  - 0.24 (ham), 0.13 (ham)

- Slope of accuracy isometric:
  \[ \text{Neg}/\text{Pos} = 4/6 \]
Example of optimizing accuracy

- Scores:
  - 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), 0.13 (ham)

- Slope of accuracy isometric:

  \[ \text{Neg} / \text{Pos} = 4/6 \]
Example of optimizing accuracy

• Scores:
  – 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), 0.13 (ham)

• Slope of accuracy isometric:
  \( \frac{\text{Neg}}{\text{Pos}} = \frac{4}{6} \)
Example of optimizing accuracy

• Scores:
  – 0.89 (spam), 0.80 (spam),
    0.74 (ham), 0.71 (spam),
    0.63 (spam), 0.49 (ham),
    0.42 (spam), 0.32 (spam),
    0.24 (ham), 0.13 (ham)

• Slope of accuracy isometric:
  \[ \frac{\text{Neg}}{\text{Pos}} = 4/6 \]
Example of optimizing accuracy

• Scores:
  – 0.89 (spam), 0.80 (spam),
    0.74 (ham), 0.71 (spam),
    0.63 (spam), 0.49 (ham),
    0.42 (spam), 0.32 (spam),
    0.24 (ham), 0.13 (ham)

• Slope of accuracy isometric:
  \[ \text{Neg} / \text{Pos} = 4/6 \]
Example of optimizing accuracy

• Scores:
  - 0.89 (spam), 0.80 (spam),
    0.74 (ham), 0.71 (spam),
    0.63 (spam), 0.49 (ham),
    0.42 (spam), 0.32 (spam),
    0.24 (ham), 0.13 (ham)

• Slope of accuracy
  isometric:

  $$\frac{Neg}{Pos} = \frac{4}{6}$$
Example of optimizing accuracy

• Scores:
  - 0.89 (spam), 0.80 (spam),
    0.74 (ham), 0.71 (spam),
    0.63 (spam), 0.49 (ham),
    0.42 (spam), 0.32 (spam),
    0.24 (ham), 0.13 (ham)

• Slope of accuracy isometric:
  \( \frac{\text{Neg}}{\text{Pos}} = 4/6 \)
Example of optimizing accuracy

• Scores:
  – 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), 0.13 (ham)

• Slope of accuracy isometric:

\[ \text{Neg/Pos} = 4/6 \]
Example of optimizing accuracy

- **Scores:**
  - 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), 0.13 (ham)

- **Slope of accuracy isometric:**
  \[ \text{Neg/Pos} = 4/6 \]

- **Threshold** \( 0.32 > t > 0.24 \)
  - For example at \( t = 0.28 \)
Cost sensitive learning

• In many applications the cost \( c_{FP} \) of one false positive and cost \( c_{FN} \) of one false negative can be different

• Then we must find a threshold which minimizes the total cost \( FP \cdot c_{FP} + FN \cdot c_{FN} \)

• If \( c_{FP} = c_{FN} \) then we can optimize accuracy

• Otherwise, we must slide the line with slope \( \frac{c_{FP}}{c_{FN}} \cdot \frac{Neg}{Pos} \) down from ROC heaven
Example of cost-sensitive learning

- **Scores:**
  - 0.89 (spam), 0.80 (spam),
  - 0.74 (ham), 0.71 (spam),
  - 0.63 (spam), 0.49 (ham),
  - 0.42 (spam), 0.32 (spam),
  - 0.24 (ham), 0.13 (ham)

- **Costs:** $c_{FP} = 2$, $c_{FN} = 1$

- **Slope:** $\frac{c_{FP}}{c_{FN}} \cdot \frac{\text{Neg}}{\text{Pos}} = \frac{2.4}{1.6} = \frac{4}{3}$
Example of cost-sensitive learning

- Scores:
  - 0.89 (spam), 0.80 (spam),
  - 0.74 (ham), 0.71 (spam),
  - 0.63 (spam), 0.49 (ham),
  - 0.42 (spam), 0.32 (spam),
  - 0.24 (ham), 0.13 (ham)

- Costs: $c_{FP} = 2, c_{FN} = 1$

- Slope: $\frac{c_{FP}}{c_{FN}} \cdot \frac{\text{Neg}}{\text{Pos}} = \frac{2.4}{1.6} = \frac{4}{3}$
Example of cost-sensitive learning

- Scores:
  - 0.89 (spam), 0.80 (spam),
    0.74 (ham), 0.71 (spam),
    0.63 (spam), 0.49 (ham),
    0.42 (spam), 0.32 (spam),
    0.24 (ham), 0.13 (ham)

- Costs: \[ c_{FP} = 2, c_{FN} = 1 \]

- Slope: \[ \frac{c_{FP}}{c_{FN}} \cdot \frac{\text{Neg}}{\text{Pos}} = \frac{2.4}{1.6} = \frac{4}{3} \]
Example of cost-sensitive learning

- **Scores:**
  - 0.89 (spam), 0.80 (spam),
    0.74 (ham), 0.71 (spam),
    0.63 (spam), 0.49 (ham),
    0.42 (spam), 0.32 (spam),
    0.24 (ham), 0.13 (ham)

- **Costs:** \( c_{FP} = 2, c_{FN} = 1 \)

- **Slope:** \( \frac{c_{FP}}{c_{FN}} \cdot \frac{Neg}{Pos} = 2.4 \cdot \frac{1}{1.6} = \frac{4}{3} \)
Example of cost-sensitive learning

• Scores:
  - 0.89 (spam), 0.80 (spam),
  0.74 (ham), 0.71 (spam),
  0.63 (spam), 0.49 (ham),
  0.42 (spam), 0.32 (spam),
  0.24 (ham), 0.13 (ham)

• Costs: \( c_{FP} = 2, c_{FN} = 1 \)

• Slope: \( \frac{c_{FP}}{c_{FN}} \cdot \frac{\text{Neg}}{\text{Pos}} = \frac{2.4}{1.6} = \frac{4}{3} \)
Example of cost-sensitive learning

• Scores:
  – 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), 0.13 (ham)

• Costs: $c_{FP} = 2, c_{FN} = 1$

• Slope: $\frac{c_{FP}}{c_{FN}} \cdot \frac{\text{Neg}}{\text{Pos}} = \frac{2.4}{1.6} = \frac{4}{3}$

• Threshold: $0.80 > t > 0.74$ or $0.63 > t > 0.49$ or $0.32 > t > 0.24$
Mathematical notation of supervised learning and classification

Assessing classification performance

Coverage plots and ROC plots

Scoring classifiers and rankers

Turning rankers into classifiers