Lecture outline

• Introduction to classification
• Evaluating classifiers
• $k$-NN

Some of the material presented here is from the supplementary material of the book: Introduction to Data Mining by Tan, Steinbach and Kumar
What is classification?

<table>
<thead>
<tr>
<th>Tid</th>
<th>Home Owner</th>
<th>Marital Status</th>
<th>Annual Income</th>
<th>Defaulted Borrower</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 4.6. Training set for predicting borrowers who will default on loan payments.

Figure from Introduction to Data Mining (Tan, Steinbach and Kumar)
What is classification?

- **Classification** is the task of **learning a target function** $f$ that maps attribute set $x$ to one of the predefined class labels $y$.
What is classification?

Figure 4.2. Classification as the task of mapping an input attribute set \( x \) into its class label \( y \).
Why classification?

• The target function \( f \) is known as a classification model

• **Descriptive modeling:** Explanatory tool to distinguish between objects of different classes (e.g., description of who can pay back his loan)

• **Predictive modeling:** Predict a class of a previously unseen record
Typical applications

- credit approval
- target marketing
- medical diagnosis
- treatment effectiveness analysis
General approach to classification

- **Training set** consists of records with known class labels

- Training set is used to **build a classification model**

- The classification model is applied to the **test set** that consists of records with **unknown labels**
General approach to classification

Figure 4.3. General approach for building a classification model.

Figure from Introduction to Data Mining (Tan, Steinbach and Kumar)
Evaluating your classifier

• Metrics for Performance Evaluation
  – How to evaluate the performance of a classifier?

• Methods for Performance Evaluation
  – How to obtain reliable estimates?

• Methods for Classifier Comparison
  – How to compare the relative performance of different classifiers?
Evaluation of classification models

- Counts of test records that are correctly (or incorrectly) predicted by the classification model
- Confusion matrix

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class = 1</td>
<td>$f_{11}$</td>
</tr>
<tr>
<td>Class = 0</td>
<td>$f_{01}$</td>
</tr>
</tbody>
</table>

Accuracy = \( \frac{\text{# correct predictions}}{\text{total # of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}} \)

Error rate = \( \frac{\text{# wrong predictions}}{\text{total # of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}} \)
Supervised vs. Unsupervised Learning

• **Supervised learning (classification)**
  – Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  – New data is classified based on the training set

• **Unsupervised learning (clustering)**
  – The class labels of training data is unknown
  – Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data
Metrics for Performance Evaluation

• Focus on the predictive capability of a model
  – Rather than how fast it takes to classify or build models, scalability, etc.

• Confusion Matrix:

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
<td>Class=No</td>
<td></td>
</tr>
<tr>
<td>Class=Yes</td>
<td>a: TP</td>
<td>b: FN</td>
<td></td>
</tr>
<tr>
<td>Class=No</td>
<td>c: FP</td>
<td>d: TN</td>
<td></td>
</tr>
</tbody>
</table>

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)
## Metrics for Performance Evaluation

- Most widely-used metric:

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
<td>Class=No</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>a (TP)</td>
<td>b (FN)</td>
</tr>
<tr>
<td>Class=No</td>
<td>c (FP)</td>
<td>d (TN)</td>
</tr>
</tbody>
</table>

\[
\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}
\]
Limitation of Accuracy

• Consider a 2-class problem
  – Number of Class 0 examples = 9990
  – Number of Class 1 examples = 10

• If model predicts everything to be class 0, accuracy is $9990/10000 = 99.9\%$
  – Accuracy is misleading because model does not detect any class 1 example
### Cost Matrix

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C(i</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>Class=Yes</td>
</tr>
<tr>
<td></td>
<td>C(Yes</td>
</tr>
<tr>
<td>Class=No</td>
<td>Class=No</td>
</tr>
<tr>
<td></td>
<td>C(Yes</td>
</tr>
</tbody>
</table>

**C(i|j):** Cost of misclassifying class \( j \) example as class \( i \)
Computing Cost of Classification

<table>
<thead>
<tr>
<th>Cost Matrix</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTUAL CLASS</td>
<td>C(i</td>
</tr>
<tr>
<td>+</td>
<td>-1</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model M₁</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTUAL CLASS</td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>150</td>
</tr>
<tr>
<td>-</td>
<td>60</td>
</tr>
</tbody>
</table>

Accuracy = 80%  
Cost = 3910

<table>
<thead>
<tr>
<th>Model M₂</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTUAL CLASS</td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>250</td>
</tr>
<tr>
<td>-</td>
<td>5</td>
</tr>
</tbody>
</table>

Accuracy = 90%  
Cost = 4255
Cost vs Accuracy

<table>
<thead>
<tr>
<th>Count</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
</tr>
<tr>
<td>ACTUAL CLASS</td>
<td></td>
</tr>
<tr>
<td>Class=Yes</td>
<td>a</td>
</tr>
<tr>
<td>Class=No</td>
<td>c</td>
</tr>
</tbody>
</table>

\[ N = a + b + c + d \]

Accuracy = \( \frac{a + d}{N} \)

Cost = \( p \ (a + d) + q \ (b + c) \)

\[ = p \ (a + d) + q \ (N - a - d) \]

\[ = q \ N - (q - p)(a + d) \]

\[ = N \ [q - (q-p) \times \text{Accuracy}] \]

Accuracy is proportional to cost if

1. \( C(Yes|No)=C(No|Yes) = q \)
2. \( C(Yes|Yes)=C(No|No) = p \)
Cost–Sensitive Measures

Precision (p) = \frac{a}{a + c} = \frac{TP}{TP + FP}

Recall (r) = \frac{a}{a + b} = \frac{TP}{TP + FN}

F - measure (F) = \frac{2rp}{r + p} = \frac{2a}{2a + b + c} = \frac{2TP}{2TP + FP + FN}

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F–measure is biased towards all except C(No|No)

Weighted Accuracy = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}
Model Evaluation

• Metrics for Performance Evaluation
  – How to evaluate the performance of a model?

• Methods for Performance Evaluation
  – How to obtain reliable estimates?

• Methods for Model Comparison
  – How to compare the relative performance of different models?
Methods for Performance Evaluation

• How to obtain a reliable estimate of performance?

• Performance of a model may depend on other factors besides the learning algorithm:
  – Class distribution
  – Cost of misclassification
  – Size of training and test sets
Learning Curve

- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve

Effect of small sample size:
- Bias in the estimate
- Variance of estimate
Methods of Estimation

- **Holdout**
  - Reserve $2/3$ for training and $1/3$ for testing

- **Random subsampling**
  - Repeated holdout

- **Cross validation**
  - Partition data into $k$ disjoint subsets
  - $k$-fold: train on $k-1$ partitions, test on the remaining one
  - **Leave-one-out**: $k=n$

- **Bootstrap**
  - Sampling with replacement
Definition

• Given: a set $X$ of $n$ points in $\mathbb{R}^d$
• Nearest neighbor: for any query point $q \in \mathbb{R}^d$ return the point $x \in X$ minimizing $D(x,q)$

• **Intuition:** Find the point in $X$ that is the closest to $q$
Motivation

• **Learning:** Nearest neighbor rule
• **Databases:** Retrieval
• **Data mining:** Clustering
• Donald Knuth in vol. 3 of *The Art of Computer Programming* called it the post-office problem, referring to the application of assigning a resident to the nearest-post office
Nearest-neighbor rule
MNIST dataset “2”
Methods for computing NN

• **Linear scan**: $O(nd)$ time

• This is pretty much all what is known for exact algorithms with theoretical guarantees

• In practice:
  – **kd-trees** work “well” in “low-medium” dimensions