Classification fundamentals

Elena Baralis, Tania Cerquitelli
Politecnico di Torino

Objectives
- prediction of a class label
- definition of an interpretable model of a given phenomenon

Classification: definition
- Given
  - a collection of class labels
  - a collection of data objects labelled with a class label
- Find a descriptive profile of each class, which will allow the assignment of unlabeled objects to the appropriate class

Definitions
- Training set
  - Collection of labeled data objects used to learn the classification model
- Test set
  - Collection of labeled data objects used to validate the classification model

Classification techniques
- Decision trees
- Classification rules
- Association rules
- Neural Networks
- Naive Bayes and Bayesian Networks
- k-Nearest Neighbours (k-NN)
- Support Vector Machines (SVM)
- ...
**Decision trees**

**Example of decision tree**

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</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>85K</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>

There could be more than one tree that fits the same data!

**Apply Model to Test Data**

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Yes</td>
<td>Married</td>
<td>80K</td>
<td>?</td>
</tr>
</tbody>
</table>

Start from the root of the tree.

**There could be more than one tree that fits the same data!**
**Classification fundamentals**

**Apply Model to Test Data**

Test Data:
- Refund
- MarSt
- TaxInc
- Cheat

Apply Model to Test Data

**Decision tree induction**

- Many algorithms to build a decision tree
  - Hunt’s Algorithm (one of the earliest)
  - CART
  - ID3, C4.5, C5.0
  - SLIQ, SPRINT

**General structure of Hunt’s algorithm**

- If $D_t$ contains records that belong to the same class $y_t$:
  - then $t$ is a leaf node labeled as $y_t$
- If $D_t$ contains records that belong to more than one class:
  - select the “best” attribute $A$ on which to split $D_t$ and label node $t$ as $A$
  - split $D_t$ into smaller subsets and recursively apply the procedure to each subset
- If $D_t$ is an empty set:
  - then $t$ is a leaf node labeled as the default (majority) class $y_d$

**Decision tree induction**

- Adopts a greedy strategy
  - “Best” attribute for the split is selected locally at each step
  - not a global optimum

- Issues
  - Structure of test condition
  - Binary split versus multiway split
  - Selection of the best attribute for the split
  - Stopping condition for the algorithm
Structure of test condition
- Depends on attribute type
  - nominal
  - ordinal
  - continuous
- Depends on number of outgoing edges
  - 2-way split
  - multi-way split

Splitting on nominal attributes
- Multi-way split
  - use as many partitions as distinct values
- Binary split
  - Divides values into two subsets
  - Need to find optimal partitioning

Splitting on ordinal attributes
- Multi-way split
  - use as many partitions as distinct values
- Binary split
  - Divides values into two subsets
  - Need to find optimal partitioning

Splitting on continuous attributes
- Different techniques
  - **Discretization** to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – discretize during tree induction
  - Ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering
  - Binary decision \((A < v)\) or \((A \geq v)\)
    - consider all possible splits and find the best cut
    - more computationally intensive

Selection of the best attribute
Before splitting: 10 records of class 0, 10 records of class 1
Which attribute (test condition) is the best?
Attributes with homogeneous class distribution are preferred. Need a measure of node impurity.

\[
\begin{array}{c|c|c}
\text{Class} & \text{Frequency} \\
\hline
C_0 & 5 & C_0 & 9 \\
C_1 & 5 & C_1 & 1 \\
\end{array}
\]

Non-homogeneous, high degree of impurity

Homogeneous, low degree of impurity

Many different measures available:
- Gini index
- Entropy
- Misclassification error

Different algorithms rely on different measures.

Advantages:
- Inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Accuracy is comparable to other classification techniques for many simple data sets

Disadvantages:
- Accuracy may be affected by missing data

The classification model is defined by means of association rules:

\[(\text{Condition}) \rightarrow \gamma\]

- Rule body is an itemset
- Model generation
  - Rule selection & sorting: based on support, confidence and correlation thresholds
  - Rule pruning
- Database coverage: the training set is covered by selecting topmost rules according to previous sort

Strong points:
- Interpretable model
- Higher accuracy than decision trees
  - Correlation among attributes is considered
- Efficient classification
- Unaffected by missing data
- Good scalability in the training set size

Weak points:
- Rule generation may be slow
  - It depends on support threshold
- Reduced scalability in the number of attributes
- Rule generation may become unfeasible

The strong points of associative classification include:
- Interpretable model
- Higher accuracy than decision trees
- Efficient classification
- Unaffected by missing data
- Good scalability in the training set size

The weak points of associative classification include:
- Rule generation may be slow
  - It depends on support threshold
- Reduced scalability in the number of attributes
- Rule generation may become unfeasible
Neural networks

Inspired to the structure of the human brain
- Neurons as elaboration units
- Synapses as connection network

Structure of a neural network

Structure of a neuron

Construction of the neural network
- For each node, definition of
  - set of weights
  - offset value
  - providing the highest accuracy on the training data
- Iterative approach on training data instances

Neural networks
- Strong points
  - High accuracy
  - Robust to noise and outliers
  - Supports both discrete and continuous output
  - Efficient during classification
- Weak points
  - Long training time
  - weakly scalable in training data size
  - complex configuration
  - Not interpretable model
  - application domain knowledge cannot be exploited in the model

From: Han, Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann 2006
Bayesian Classification

Let C and X be random variables

\[
P(C,X) = P(C|X) P(X)
\]

\[
P(C,X) = P(X|C) P(C)
\]

Hence

\[
P(C|X) P(X) = P(X|C) P(C)
\]

and also

\[
P(C|X) = \frac{P(X|C) P(C)}{P(X)}
\]

Bayesian classification

Let the class attribute and all data attributes be random variables

- C = any class label
- X = \(<x_1, ..., x_k>\) record to be classified

Bayesian classification

- compute \(P(C|X)\) for all classes
- probability that record \(X\) belongs to \(C\)
- assign \(X\) to the class with maximal \(P(C|X)\)

Applying Bayes theorem

\[
P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}
\]

- \(P(X)\) constant for all \(C\), disregarded for maximum computation
- \(P(C)\) a priori probability of \(C\)

\[
P(C) = \frac{N_c}{N}
\]

How to estimate \(P(X|C)\), i.e. \(P(x_1, ..., x_k|C)\)?

Naive hypothesis

\[
P(x_1, ..., x_k|C) = P(x_1|C) P(x_2|C) ... P(x_k|C)
\]

- statistical independence of attributes \(x_1, ..., x_k\)
- not always true
- model quality may be affected

Computing \(P(x_k|C)\)

- for discrete attributes
  \[
P(x_k|C) = \frac{|x_k|_C}{N_c}
\]
  - where \(|x_k|_C\) is number of instances having value \(x_k\) for attribute \(k\) and belonging to class \(C\)
- for continuous attributes, use probability distribution

Bayesian networks

- allow specifying a subset of dependencies among attributes

### Bayesian classification: Example

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>N</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>N</td>
</tr>
<tr>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
</tbody>
</table>

From: Han, Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann 2006
Bayesian classification: Example

- Data to be labeled
  \(X = \langle \text{rain}, \text{hot}, \text{high}, \text{false} \rangle\)
- For class \(p\)
  \[P(X|p) \cdot P(p) = P(\text{rain}|p) \cdot P(\text{hot}|p) \cdot P(\text{high}|p) \cdot P(\text{false}|p) \cdot P(p) = \frac{3}{9} \cdot \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{6}{9} \cdot \frac{9}{14} = 0.010582\]
- For class \(n\)
  \[P(X|n) \cdot P(n) = P(\text{rain}|n) \cdot P(\text{hot}|n) \cdot P(\text{high}|n) \cdot P(\text{false}|n) \cdot P(n) = \frac{2}{5} \cdot \frac{2}{5} \cdot \frac{4}{5} \cdot \frac{2}{5} \cdot \frac{5}{14} = 0.018286\]

From: Han, Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann 2006

Model evaluation

- Methods for performance evaluation
  - Partitioning techniques for training and test sets
  - Metrics for performance evaluation
    - Accuracy, other measures
  - Techniques for model comparison
    - ROC curve

Methods of estimation

- Partitioning labeled data in
  - training set for model building
  - test set for model evaluation
- Several partitioning techniques
  - holdout
  - cross validation
- Stratified sampling to generate partitions
  - without replacement
- Bootstrap
  - Sampling with replacement

Holdout

- Fixed partitioning
  - reserve 2/3 for training and 1/3 for testing
  - Appropriate for large datasets
  - may be repeated several times
    - repeated holdout

Cross validation

- Cross validation
  - partition data into \(k\) disjoint subsets (i.e., folds)
  - \(k\)-fold: train on \(k-1\) partitions, test on the remaining one
    - repeat for all folds
  - reliable accuracy estimation, not appropriate for very large datasets
- Leave-one-out
  - cross validation for \(k=n\)
  - only appropriate for very small datasets
Metrics for model evaluation
- Evaluate the predictive accuracy of a model
- Confusion matrix
  - binary classifier

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

- a: TP (true positive)
- b: FN (false negative)
- c: FP (false positive)
- d: TN (true negative)

Accuracy
- Most widely-used metric for model evaluation
  \[ \text{Accuracy} = \frac{\text{Number of correctly classified objects}}{\text{Number of classified objects}} \]
- Not always a reliable metric

Accuracy for a binary classifier

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

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

Limitations of accuracy
- Classes may have different importance
  - Misclassification of objects of a given class is more important
  - e.g., ill patients erroneously assigned to the healthy patients class
- Accuracy is not appropriate for
  - unbalanced class label distribution
  - different class relevance

Class specific measures
- Evaluate separately for each class
  - Recall (r) = \frac{\text{Number of objects correctly assigned to C}}{\text{Number of objects belonging to C}}
  - Precision (p) = \frac{\text{Number of objects correctly assigned to C}}{\text{Number of objects assigned to C}}
- Maximize
  \[ F\text{-measure} (F) = \frac{2rp}{r + p} \]
Class specific measures

For a binary classification problem

- on the confusion matrix, for the positive class

\[
\text{Precision}(p) = \frac{a}{a + c}
\]

\[
\text{Recall}(r) = \frac{a}{a + b}
\]

\[
\text{F}-\text{measure} (F) = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}
\]