Data Warehousing and Data Mining

Lecture 10 Classification – Model Evaluation and Selection
Clustering – Measuring Distances between data points

Acknowledgement: The Lecture Slides are adapted from the original slides of Han’s textbook.
Lecture Outline

- Evaluating Classifiers
- Distance Measures
Model Evaluation and Selection

- **Evaluation metrics**: How can we measure accuracy? Other metrics to consider?
- **Use validation test set of class-labeled tuples instead of training set when assessing accuracy**
- **Methods for estimating a classifier’s accuracy**:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap
- **Comparing classifiers**:
  - Confidence intervals
  - Cost-benefit analysis and ROC Curves
Classifier Evaluation: Confusion Matrix

- **Confusion Matrix (binary classes)**

<table>
<thead>
<tr>
<th>Truth \ Predicated</th>
<th>Model Yes</th>
<th>Model No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Yes</td>
<td>True Positives (TP)</td>
<td>False Negatives (FN)</td>
</tr>
<tr>
<td>Actual No</td>
<td>False Positives (FP)</td>
<td>True Negatives (TN)</td>
</tr>
</tbody>
</table>

- **Example of Confusion Matrix**

<table>
<thead>
<tr>
<th>Truth \ Predicated</th>
<th>buy_computer = yes</th>
<th>buy_computer = No</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>buy_computer=yes</td>
<td>6954</td>
<td>46</td>
<td>7000</td>
</tr>
<tr>
<td>buy_computer=no</td>
<td>412</td>
<td>2588</td>
<td>3000</td>
</tr>
<tr>
<td>Total</td>
<td>7366</td>
<td>2634</td>
<td>10000</td>
</tr>
</tbody>
</table>
Confusion Matrix for Multiple Classes

- Given $m$ classes, an entry, $CM_{i,j}$ in a confusion matrix indicates number of tuples in class $i$ that were labeled by the classifier as class $j$
- May have extra rows/columns to provide totals

<table>
<thead>
<tr>
<th></th>
<th>class 1</th>
<th>...</th>
<th>class i</th>
<th>...</th>
<th>class m</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>class 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>class j</td>
<td></td>
<td></td>
<td>$CM_{i,j}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>class m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Accuracy, Error Rate, Sensitivity and Specificity

- **Classifier Accuracy**, or recognition rate:
  percentage of test set tuples that are correctly classified
  
  \[ \text{Accuracy} = \frac{TP + TN}{ALL} \]

- **Error rate**:
  
  \[ \text{Error Rate} = 1 - \text{Accuracy} = \frac{FP + FN}{ALL} \]

- **One class may be rare**, e.g. fraud, or HIV-positive
  
  - Significant *majority of the negative class* and minority of the positive class

- **Sensitivity**: True Positive recognition rate
  
  \[ \text{Sensitivity} = \frac{TP}{P} \]

- **Specificity**: True Negative recognition rate
  
  \[ \text{Specitivity} = \frac{TN}{N} \]
Precision, Recall and F-measures

- Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive
  \[ \text{precision} = \frac{TP}{TP+FP} \]
- Recall: completeness – what % of positive tuples did the classifier label as positive?
  \[ \text{recall} = \frac{TP}{TP+FN} \]
- Perfect score is 1.0
- Inverse relationship between precision & recall
- F measure (\( F_1 \) or F-score): harmonic mean of precision and recall,
  \[ F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]
- \( F_\beta \): weighted measure of precision and recall
  - assigns \( \beta \) times as much weight to recall as to precision
  \[ F_\beta = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}} \]
# Classifier Evaluation Metrics: example

<table>
<thead>
<tr>
<th>Actual Class\Predicted class</th>
<th>cancer = yes</th>
<th>cancer = no</th>
<th>Total</th>
<th>Recognition(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer = yes</td>
<td>90</td>
<td>210</td>
<td>300</td>
<td>30.00 (sensitivity)</td>
</tr>
<tr>
<td>cancer = no</td>
<td>140</td>
<td>9560</td>
<td>9700</td>
<td>98.56 (specificity)</td>
</tr>
<tr>
<td>Total</td>
<td>230</td>
<td>9770</td>
<td>10000</td>
<td>96.40 (accuracy)</td>
</tr>
</tbody>
</table>

- **Precision** = ?
- **Recall** = ?
Holdout & Cross-Validation Methods

• **Holdout method**
  – Given data is randomly partitioned into two independent sets
    • Training set (e.g., 2/3) for model construction
    • Test set (e.g., 1/3) for accuracy estimation

• **Random sampling: a variation of holdout**
  – Repeat holdout k times, accuracy = avg. of the accuracies obtained

• **Cross-validation (k-fold, where k = 10 is most popular)**
  – Randomly partition the data into k mutually exclusive subsets, each approximately equal size
  – At i-th iteration, use Di as test set and others as training set
  – Leave-one-out: k folds where k = # of tuples, for small sized data
  – *Stratified cross-validation*: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data
Bootstrap

• Bootstrap
  – Works well with small data sets
  – Samples the given training tuples uniformly \textit{with replacement}
    • i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set

• Several bootstrap methods, and a common one is .632 bootstrap
  – A data set with \( d \) tuples is sampled \( d \) times, with replacement, resulting in a training set of \( d \) samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2\% of the original data end up in the bootstrap, and the remaining 36.8\% form the test set (since \( (1 - 1/d)^d \approx e^{-1} = 0.368 \))
  – Repeat the sampling procedure \( k \) times, overall accuracy of the model:

\[
\text{Acc}(M) = \frac{1}{k} \sum_{i=1}^{k} (0.632 \times \text{Acc}(M_i)_{\text{test\_set}} + 0.368 \times \text{Acc}(M_i)_{\text{train\_set}})
\]
• ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models
  – Originated from signal detection theory
  – Shows the trade-off between the true positive rate and the false positive rate
  – The area under the ROC curve is a measure of the accuracy of the model

• Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list

• The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model
Summary

- **Effective and advanced classification methods**
  - Selection of Classifier: BPNN vs Support Vector Machine (SVM)
  - comparisons of the different classification methods: No single method has been found to be superior over all others for all data sets
  - Issues such as accuracy, training time, robustness, scalability, and interpretability must be considered and can involve trade-offs.
  - **Other classification methods**: lazy learners (KNN), genetic algorithms, rough set and fuzzy set approaches

- **Evaluation metrics** include: accuracy, sensitivity, specificity, precision, recall, $F$ measure, and $F_β$ measure.

- **Stratified k-fold cross-validation** is recommended for accuracy estimation. **Bagging and boosting** can be used to increase overall accuracy by learning and combining a series of individual models.

- **ROC curves** are often found useful for model selection.
What is Cluster Analysis?

- **Cluster**: Group of objects similar to one another within the same cluster and dissimilar to the objects in other clusters
- **Cluster analysis**: Finding characteristics for similar objects
- **Unsupervised learning**: no predefined classes
- **Typical applications**
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithm
- **Rich Applications**
  - Create thematic maps in GIS
  - market research
  - Document classification
  - DNA analysis
What is good clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity (linkage functions)
  - low inter-class similarity

- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

- The definitions of similarity, measured as a distance functions are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables. Often is highly subjective.
Data Structures

• **Data matrix**
  – (two modes):
    n-observations with p-attributes (measurements).

\[
\begin{bmatrix}
  x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  x_{n1} & \cdots & x_{nf} & \cdots & x_{np}
\end{bmatrix}
\]

• **Dissimilarity matrix**
  – (one mode)
  \(d(i, j)\) is the dissimilarity between objects \(i\) and \(j\)

\[
\begin{bmatrix}
  0 & \cdots & 0 \\
  d(2,1) & 0 & \cdots \\
  d(3,1) & d(3,2) & 0 \\
  \vdots & \vdots & \vdots \\
  d(n,1) & d(n,2) & \cdots & 0
\end{bmatrix}
\]
Type of Data in Clustering Analysis

• Interval-scaled variables (continuous measures)
• Binary variables
• Nominal, ordinal, and ratio variables
• Variables of mixed types
Interval-Valued Variables

• **Standardize data**
  – Calculate the mean absolute deviation:
    \[
    s_f = \frac{1}{n} (|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|)
    \]
  where
  \[
  m = \frac{1}{n} (x_{1f} + x_{2f} + \cdots + x_{nf})
  \]
  – Calculate the standardized measurement (z-score)
    \[
    z_{if} = \frac{x_{if} - m_f}{s_f}
    \]

• **Using mean absolute deviation is more robust than using standard deviation**
Distance Measures

- Distances are normally used to measure the **similarity** or **dissimilarity** between two data objects
- Some popular ones include: Minkowski distance:
  - \( d(i, j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q} \)
  - where \( i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \) and \( j = (x_{j1}, x_{j2}, \ldots, x_{jp}) \) are two \( p \)-dimensional data objects, and \( q \) is a positive integer.
  - If \( q = 1 \), \( d \) is Manhattan Distance
    - \( d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \)
  - If \( q = 2 \), \( d \) is Euclidean Distance
    - \( d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2} \)
- Properties:
  - \( d(i, j) \geq 0 \); \( d(i, j) = 0 \) if \( i = j \) and \( d(i, j) = d(j, i) \)
  - \( d(i, j) \leq d(i, k) + d(j, k) \)
- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures
Binary Values

- A contingency table for binary data
- Distance Measure for
  - Symmetric binary variables:
    \[ d(i, j) = \frac{b+c}{a+b+c+d} = \frac{b+c}{p} \]
  - Asymmetric binary variables:
    \[ d(i, j) = \frac{b+c}{a+b+c} \]
- Jaccard Coefficient (similarity measure for asymmetric binary variables):
  \[ \text{sim}_{\text{Jaccard}}(i,j) = \frac{a}{a+b+c} \]
Dissimilarity between Binary Variables

- **Given the following example**

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
<th>Fever</th>
<th>Cough</th>
<th>Test-1</th>
<th>Test-2</th>
<th>Test-3</th>
<th>Test-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>M</td>
<td>Y</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Mary</td>
<td>F</td>
<td>Y</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td>Jim</td>
<td>M</td>
<td>Y</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

- Gender is a symmetric attribute
- The remaining attributes are asymmetric binary
- Let the values of Y and P to be 1, and value N to be 0.

- We have
  - \( d(jack, mary) = \frac{0+1}{2+0+1} = 0.33 \)
  - \( d(jack, jim) = \frac{1+1}{1+1+1} = 0.67 \)
  - \( d(jack, mary) = \frac{1+2}{1+1+2} = 0.75 \)
Nominal Values

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - $m$: # of matches, $p$: total # of variables
    - $d(i, j) = \frac{p-m}{p}$
- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the $M$ nominal states
• An ordinal variable can be discrete or continuous
• Order is important, e.g., rank
• Can be treated like interval-scaled
  – replace $x_{if}$ by their rank $r_{if} \in \{1, \ldots, M_f\}$
  – map the range of each variable onto $[0, 1]$ by replacing $i$-th object in the $f$-th variable by
    • $z_{if} = (r_{if} - 1)/(M_{if} - 1)$
    – compute the dissimilarity using methods for interval-scaled variables
Ratio-Scaled Variables

- **Ratio-scaled variable**: a positive measurement on a nonlinear scale, approximately at exponential scale, such as $Ae^{Bt}$ or $Ae^{-Bt}$

- **Methods**:
  - treat them like interval-scaled variables—*not a good choice!* (why?—the scale can be distorted)
  - apply logarithmic transformation
    \[ y_{if} = \log(x_{if}) \]
  - treat them as continuous ordinal data treat their rank as interval-scaled
Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio
- One may use a weighted formula to combine their effects
  \[ d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^f d_{ij}^f}{\sum_{f=1}^{p} \delta_{ij}^f} \]
  - \( f \) is binary or nominal:
    \[ d_{ij}^f = 0, \quad \text{if } x_{if} = x_{jf} \quad \text{or} \]
    \[ d_{ij}^f = 1, \quad \text{otherwise} \]
  - \( f \) is interval-based: use the normalized distance
  - \( f \) is ordinal or ratio-scaled
    - compute ranks \( r_{if} \) and
    - and treat \( z_{if} \) as interval-scaled
Vector Objects

- Vector objects: keywords in documents, gene features in micro-arrays, etc.
- Broad applications: information retrieval, biologic taxonomy, etc.
- Cosine measure
  \[ s(x, y) = \frac{x^T \cdot y}{\|x\| \|y\|} \]
- A variant: Tanimoto coefficient - used in information retrieval and biology taxonomy
  \[ s(x, y) = \frac{x^T \cdot y}{x^T \cdot x + y^T \cdot y - x^T \cdot y} \]