Classification
Lecture 1: Basics and Methods

Guo-Jun Qi
University of Central Florida
Outline

• Basics
  • Problem, goal and evaluation

• Methods
  • Nearest Neighbor
  • Decision Tree
  • Naïve Bayes
  • Rule-based Classification
  • Logistic Regression
  • Support Vector Machines

• Advanced Topics
  • Semi-supervised learning
  • Multi-view learning
  • Transfer learning
Readings

• Aggarwal, Chapter 10: Data classification
• Han and Kamber. Data Mining: Concepts and Technologies, Chapter 6.
• Extended readings
  • Bishop, Pattern recognition and Machine Learning. Section 4.1.
Classification Definition

• Given a collection of records (training set)
  Each record contains a set of attributes, one of the attributes is the class.
• Find a model for class attribute as a function of the values of other attributes.
• Goal: previously unseen records should be assigned a class as accurately as possible (generalization ability).
  A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.
Illustrating Classification Task

<table>
<thead>
<tr>
<th>Training Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>

Learning algorithm

Induction

Learn Model

Apply Model

Deduction
Examples of classification task

• Predicting tumor cells as benign or malignant
• Classifying credit card transactions as legitimate or fraudulent
• Classifying emails as spams or normal emails
• Categorizing news stories as finance, weather, entertainment, sports, etc.
Metrics for Performance Evaluation

- Focus on the predicative capability of a model on future test set
  - Rather than how accurately it classifies training examples

- Confusion matrix:

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

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

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

• Most widely used metric:

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

• Consider a binary classification problem
  • Number of Class 0 examples = 9990
  • Number of Class 1 examples = 10

• If model predicted everything as Class 0, the accuracy would be 9990/10000 = 99.9%!
  • No sense
  • Accuracy is misleading on the imbalanced dataset (e.g., predicting rare events like terrorism attack)
  • Detecting rare events is more important than trivially predicting everyt is normal.
Cost-Sensitive Measures

• Precision: the chance of correctly predicting the attacks
  • Among all the predicted attacks, how large portion is true?

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

• Recall: the chance of predicting all the attacks
  • Among all the true attacks, how large portion is predicted?

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

• F-measure: harmonic combination of two

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

• Holdout
  • Reserve 2/3 for training and 1/3 for testing

• Random subsampling
  • Repeated holdout by randomly selecting training and test sets
  • Both holdout and random subsampling are pessimistic because only a portion of samples are used for training the model

• Solution: Cross validation
  • Partition data into k disjoint subsets
  • k-fold: training on k-1 partition and testing on the remaining one
  • Repeat over all k times
  • Extreme case: Leave-One-Out (LOO) when k=number of all samples
k-Fold Cross-Validation

k-fold cross-validation
- Fix the parameter to be tuned
- Divide dataset into k subsets of equal size
- Every time, use k-1 subsets to train a model
- Test the trained model on the remaining subset of validation examples
- Repeat k times
- Average the accuracy/error over k times experiments

Validation example

Training example
Special case: Leave one out (LOO) cross-validation

Set $k=n$
- Every time, only one example is left for validation, others are used as training examples

- Validation example
- Training example
Methods of Estimation: Bootstrap

• Bootstrap: Sampling with replacement
  • Each time, one sample is chosen into training set with equal probability from a given dataset
  • With replacement: a sample might be chosen multiple times with repetition
  • Test set: the remaining samples not chosen in the training set are used as test examples.
  • Suppose there are \( n \) examples in a dataset, we repeat bootstrap sampling with \( n \) times, ending up with \( n \) training examples (with replacement)
    • It can be proven that when the dataset is sufficiently large, on average 63.2% of examples are chosen into the training set.
Classification Methods

• Nearest Neighbor
• Decision Tree
• Naïve Bayes
• Rule-based Classification
• Logistic Regression
• Support Vector Machines
• Ensemble methods
• Neural Networks
• Deep Learning
Nearest Neighbor Classification

- Store the training records
- Use training records to predict the class label of unseen cases
Nearest-Neighbor Classification

• Requiring
  • The set of stored labeled examples
  • Distance Metric to compute the distance between examples in a *feature space*
  • The value of k, the number of the nearest neighbors to retrieve

• To classify an unknown example
  • Computing its distances to the training examples
  • Identifying k nearest neighbors
  • Use the labels of k nearest neighbors to determine the class label of unknown example, usually by majority voting
\( k \) Nearest Neighbors

- \( k \)-nearest neighbors of a test example (denoted by \( x \)) are data points in a metric feature space with \( k \) smallest distance to \( x \)

(a) 1-nearest neighbor  
(b) 2-nearest neighbor  
(c) 3-nearest neighbor
1 nearest neighbor

- Voronoi Diagram: dividing the whole space into regions, each of which contains the data points belong to the 1 nearest neighbor of a certain training examples
Distance

• Computing distance between two points
  • Euclidean distance

\[ d(p, q) = \sqrt{\sum_i (p_i - q_i)^2} \]

• Determining the class from nearest neighbors
  • Take the majority vote on the class labels
  • Weighted majority vote by the distance
  • Weighting factor, \( w = 1/d^2 \)
Nearest Neighbor Classification

• Choosing the value of k:
  • When k is too small, sensitive to noise points
  • When k is too large, neighbors might involve irrelevant points from the other classes
  • Making a trade-off: selecting k by cross-validation by varying k and choosing the one generating highest cross-validation accuracy
Nearest neighbor classification

• Scaling issues
  • A given dataset might have dramatically different scales for different dimensions in the feature space
  • E.g., a feature vector representing a person with height (4~7ft) and weight (100~200 lbs)
  • Have to scale the dimensions to prevent one from dominating the computation of distance

• Whitening scale
  • Normalizing each dimension to have zero mean and unit variance.
  • Subtracting each data point by mean, followed by dividing with standard variance.
Nearest Neighbor Classification

• K-NN classifiers are lazy learners
  • No need to build models explicitly
  • In contrast to eager learners like decision trees (next lecture)
  • However, it is usually not scalable to very large datasets
    • Computing distances to all examples in a very large dataset is prohibitive
    • Solution: approximate neighbor neighbors (by hashing) – nearest neighbors are mapped into the same hashing bucket.