Knowledge Discovery and Data Mining

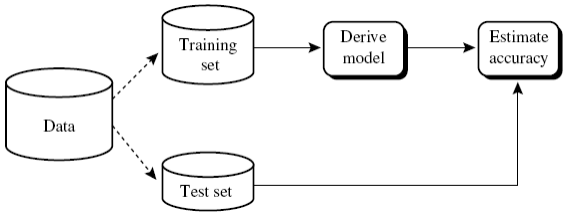
Unit # 10

Supervised Learning Process

- Data Collection/Preparation
  - Data Cleaning
  - Discretization
    - Supervised/Unspervised
  - Identification of right number of discrete states for nominal attributes
- Feature Selection
  - Supervised/Unsupervised
- Classification Techniques
  - Decision Tree, Neural Networks, Naïve Bayes, Nearest Neighbor
- Model Testing and Evaluation
  - Accuracy, Recall/Precision/F-Measure, Bagging, Boosting, Cross-Validation, ROC Curve, Lift Charts, Hypothesis Testing
- The whole activity is not a sequential process. More importantly, you would always be required to make good use of common sense.

Evaluating the Accuracy of a Classifier

- Holdout, random subsampling, crossvalidation, and the bootstrap are common techniques for assessing accuracy based on randomly sampled partitions of the given data.



Holdout Method

- The holdout method is what we have alluded to so far in our discussions about accuracy.
- In this method, the given data are randomly partitioned into two independent sets, a training set and a test set.
- Typically, two-thirds of the data are allocated to the training set, and the remaining one-third is allocated to the test set.
- The training set is used to derive the model, whose accuracy is estimated with the test set.
- The estimate is pessimistic because only a portion of the initial data is used to derive the model.

Random Subsampling

- Random subsampling is a variation of the holdout method in which the holdout method is repeated k times.
- The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration.

K-fold Cross-Validation

- In k-fold cross-validation, the initial data are randomly partitioned into k mutually exclusive subsets or "folds," D1, D2, ….., Dk, each of approximately equal size.
- Training and testing is performed k times.
- In iteration i, partition Di is reserved as the test set, and the remaining partitions are collectively used to train the model. That is, in the first iteration, subsets D2, …, Dk collectively serve as the training set in order to obtain a first model, which is tested on D1; the second iteration is trained on subsets D1, D3, …., Dk and tested on D2; and so on.
- Unlike the holdout and random subsampling methods above, here, each sample is used the same number of times for training and once for testing.
- The accuracy estimate is the overall number of correct classifications from the k iterations, divided by the total number of tuples in the initial data.

Leave-One-Out

- Leave-one-out is a special case of *k-fold cross-validation where k is set to the number* of initial tuples.
- That is, only one sample is "left out" at a time for the test set.
- In stratified cross-validation, the folds are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data.

Bootstrap

- The bootstrap method samples the given training tuples uniformly with replacement.
- That is, each time a tuple is selected, it is equally likely to be selected again and re-added to the training set.
- There are several bootstrap methods. A commonly used one is the .632 bootstrap, which works as follows.
- Suppose we are given a data set of d tuples.
- The data set is sampled d times, with replacement, resulting in a bootstrap sample or training set of d samples.

Bootstrap (Cont'd)

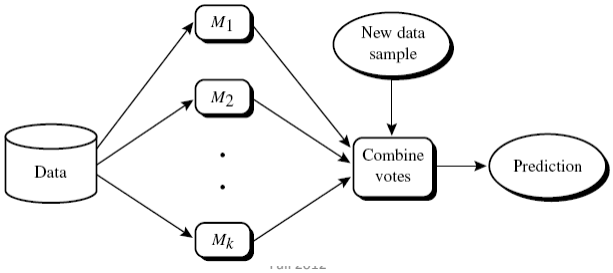
- It is very likely that some of the original data tuples will occur more than once in this sample.
- The data tuples that did not make it into the training set end up forming the test set.
- Suppose we were to try this out several times.
- As it turns out, on average, 63.2% of the original data tuples will end up in the bootstrap, and the remaining 36.8% will form the test set

Bootstrap (Cont'd)

- We can repeat the sampling procedure k times, where in each iteration, we use the current test set to obtain an accuracy estimate of the model obtained from the current bootstrap sample.
- The overall accuracy of the model is then estimated as $Acc(M) = (i=1 \text{ to } K) \Sigma (0.632 \times Acc(Mi)_{train\_set} + 0.368 \times Acc(Mi)_{test\_set})$
- where Acc(Mi)test set is the accuracy of the model obtained with bootstrap sample I when it is applied to test set i. Acc(Mi)train set is the accuracy of the model obtained with bootstrap sample i when it is applied to the original set of data tuples.
- The bootstrap method works well with small data sets.

Ensemble Models

- Bagging and boosting are examples of ensemble methods, or methods that use a combination of models.
- Each combines a series of k learned models (classifiers or predictors), M1, M2, …, Mk, with the aim of creating an improved composite model, M*.
- Both bagging and boosting can be used for classification as well as prediction



Algorithm: Bagging. The bagging algorithm—create an ensemble of models (classifiers or predictors) for a learning scheme where each model gives an equally-weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a learning scheme (e.g., decision tree algorithm, backpropagation, etc.)

Output: A composite model, $M*$.

Method:

(1) for $i = 1$ to k do // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement;
(3) use D_i to derive a model, M_i;
(4) endfor

To use the composite model on a tuple, X:

(1) if classification then
(2) let each of the k models classify X and return the majority vote;
(3) if prediction then
(4) let each of the k models predict a value for X and return the average predicted value;

Bagging (Cont'd)

- The bagged classifier often has significantly greater accuracy than a single classifier derived from D, the original training data.
- It will not be considerably worse and is more robust to the effects of noisy data.
- The increased accuracy occurs because the composite model reduces the variance of the individual classifiers.
- For prediction, it was theoretically proven that a bagged predictor will always have improved accuracy over a single predictor derived from D.

Boosting

- In boosting, weights are assigned to each training tuple.
- A series of k classifiers is iteratively learned.
- After a classifier Mi is learned, the weights are updated to allow the subsequent classifier,Mi+1, to "pay more attention" to the training tuples that were misclassified by Mi.
- The final boosted classifier, M, combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy.

Algorithm: Adaboost. A boosting algorithm—create an ensemble of classifiers. Each one gives a weighted vote.

Input:

- D, a set of d class-labeled training tuples;
- k, the number of rounds (one classifier is generated per round);
- a classification learning scheme.

Output: A composite model.

Method:

(1) initialize the weight of each tuple in D to $1/d$;
(2) for $i = 1$ to k **do** // for each round:
(3) sample D with replacement according to the tuple weights to obtain D_i;
(4) use training set D_i to derive a model, M_i;
(5) compute $error(M_i)$, the error rate of M_i (Equation 6.66)
(6) if $error(M_i) > 0.5$ then
(7) reinitialize the weights to $1/d$
(8) go back to step 3 and try again;
(9) endif
(10) for each tuple in D_i that was correctly classified **do**
(11) multiply the weight of the tuple by $error(M_i)/(1 - error(M_i))$; // update weights
(12) normalize the weight of each tuple;
(13) endfor

5

Boosting Algorithm (Cont'd)

To use the composite model to classify tuple, X:

(1) initialize weight of each class to 0;
(2) for $i = 1$ to k **do** // for each classifier:
(3) $w_i = log\frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
(4) $c = M_i(X)$; // get class prediction for X from M_i
(5) add w_i to weight for class c
(6) endfor
(7) return the class with the largest weight;

Bagging vs. Boosting

- *Because of the way boosting focuses on* the misclassified tuples, it risks overfitting the resulting composite model to such data.
- Therefore, sometimes the resulting "boosted" model may be less accurate than a single model derived from the same data.
- Bagging is less susceptible to model overfitting.
- While both can significantly improve accuracy in comparison to a single model, boosting tends to achieve greater accuracy.

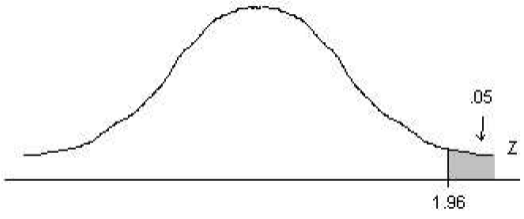
Exercise

- Suppose that we would like to select between two prediction models, M1 and M2.
- We have performed 10 rounds of 10-fold cross-validation on each model, where the same data partitioning in round i is used for both M1 and M2.
- The error rates obtained for M1 are 30.5, 32.2, 20.7, 20.6, 31.0, 41.0, 27.7, 26.0, 21.5, 26.0.
- The error rates for M2 are 22.4, 14.5, 22.4, 19.6, 20.7, 20.4, 22.1, 19.4, 16.2, 35.0.
- Comment on whether one model is significantly better than the other considering a significance level of 1%.

Hypothesis Testing: Example 1 (Source: Univ. of North Carolina)

- An insurance company is reviewing its current policy rates.
- When originally setting the rates they believed that the average claim amount was $1,800.
- They are concerned that the true mean is actually higher than this, because they could potentially lose a lot of money.
- They randomly select 40 claims, and calculate a sample mean of $1,950. Assuming that the standard deviation of claims is $500, and set alpha = 0.05,test to see if the insurance company should be concerned.

Example 1 (Cont'd)

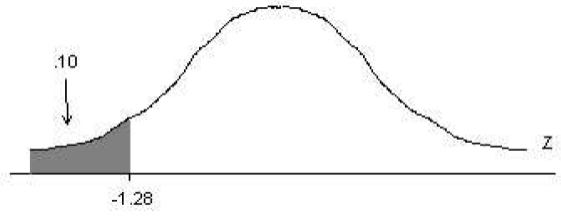


- We can see that 1.897 < 1.96, thus our test statistic is not in the rejection region. Therefore we fail to reject the null hypothesis.
- We cannot conclude anything statistically significant from this test, and cannot tell the insurance company whether or not they should be concerned about their current policies.

Hypothesis Testing: Example 2 (Source: Univ. of North Carolina)

- Trying to encourage people to stop driving to campus, the university claims that on average it takes people 30 minutes to find a parking space on campus.
- I don't think it takes so long to find a spot.
- In fact I have a sample of the last five times I drove to campus, and I calculated x = 20.
- Assuming that the time it takes to find a parking spot is normal, and that sigma = 6 minutes, then perform a hypothesis test with level alpha = 0.10 to see if my claim is correct.

Example 2 (Cont'd)

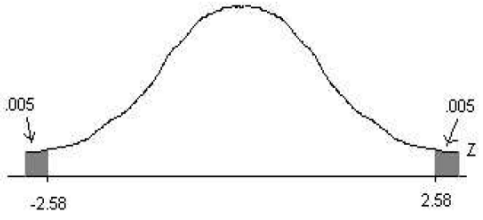


- We can see that *-3:727 < -1:28, thus our test statistic is in the rejection region.*
- *Therefore we reject* the null hypothesis in favor of the alternative.
- We can conclude that the mean is significantly less than 30, thus I have proven that the mean time to find a parking space is less than 30.

Hypothesis Testing: Example 3 (Source: Univ. of North Carolina)

- A sample of 40 sales recepts from a grcery store has x = $137 and sigma = $30.2.
- Use these values to test whether or not the mean is sales at the grocery store are different from $150.

Example 3 (Cont'd)



- We can see that |-2.722| = 2.722 > 2:58, thus our test statistic is in the rejection region.
- Therefore we reject the null hypothesis in favor of the alternative. We can conclude that the mean is significantly different from $150, thus I have proven that the mean sales at the grocery store is not $150.

Exercise Continued

- t-statistics for mean comparison

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\left(\frac{SD_1^2}{n_1} + \frac{SD_2^2}{n_2}\right)}}$$

- Or Use Excel's "TDIST" function
  - TDIST (2.43, 20, 2)
  - Where 2.43 is the t-statistics
  - 20 is the number of observation
  - 2 is for 2-tail (1 is for one-tail)