Evaluation of Learning Models

Powered by:

LEARNING OUTCOMES

- Motivation
- Metrics for Classifier’s Evaluation
- Methods for Classifier’s Evaluation
- Comparing the Performance of two Classifiers

Motivation

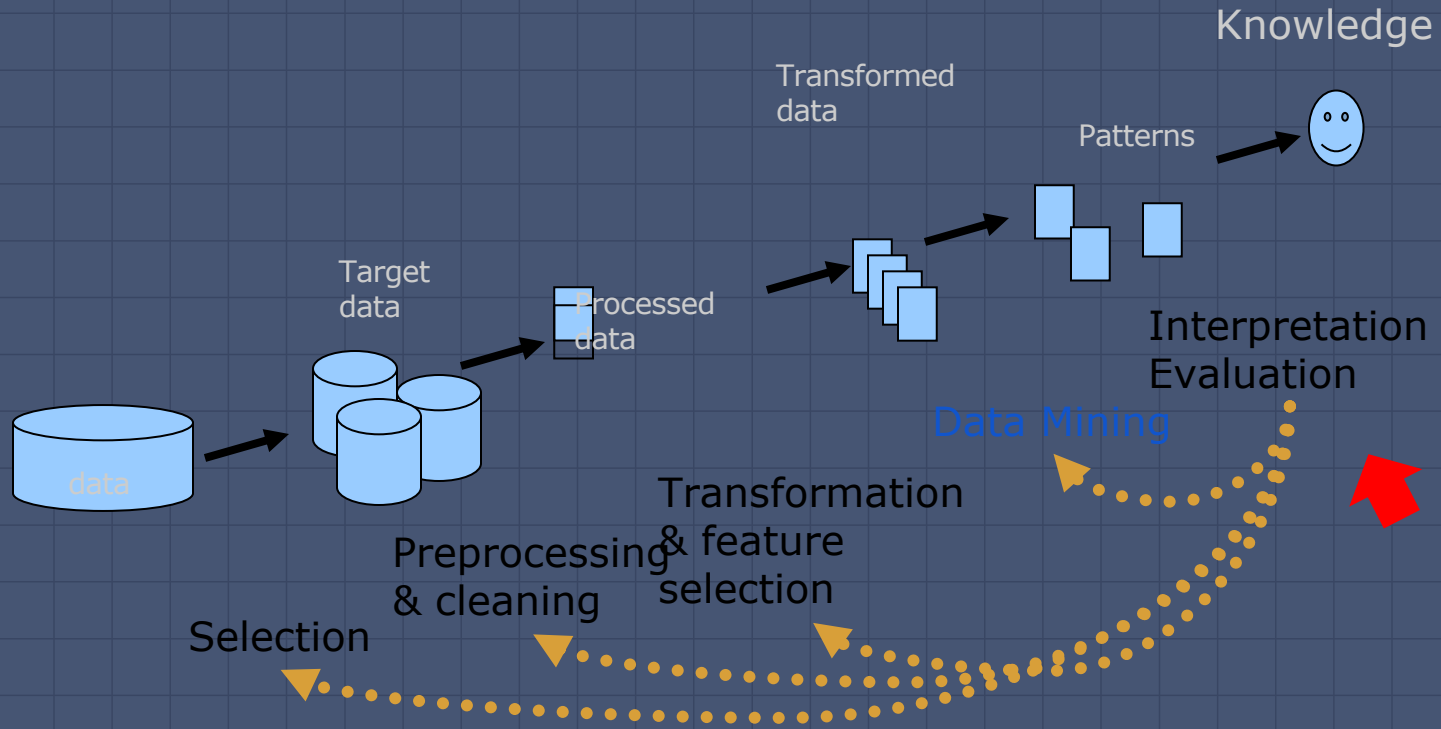
It is important to evaluate classifier's generalization performance in order to:

- Determine whether to employ the classifier;

  (*For example: when learning the effectiveness of medical treatments from a limited-size data, it is important to estimate the accuracy of the classifiers.*)
- Optimize the classifier.

  (*For example: when post-pruning decision trees we must evaluate the accuracy of the decision trees on each pruning step.*)

Model's Evaluation in the KDD Process



How to evaluate the Classifier's Generalization Performance?

- Assume that we test a classifier on some test set and we derive at the end the following *confusion matrix*:

		Predicted class		
		Pos	Neg	
Actual class	Pos	*TP*	*FN*	*P*
	Neg	*FP*	*TN*	*N*

How to Estimate the Metrics?

We can use:

- Training data;
- Independent test data;
- Hold-out method;
- k-fold cross-validation method;
- Leave-one-out method;
- Bootstrap method;
- And many more…

Estimation with Training Data

The accuracy/error estimates on the training data are *not* good indicators of performance on future data.

- **Q: Why?**
- **A:** Because [illegible] probably [illegible] same as the training data!

The accuracy/error estimates on the training data measure the degree of classifier's overfitting.

Estimation with Independent Test Data

Estimation with independent test data is used when we have plenty of data and there is a natural way to forming training and test data.

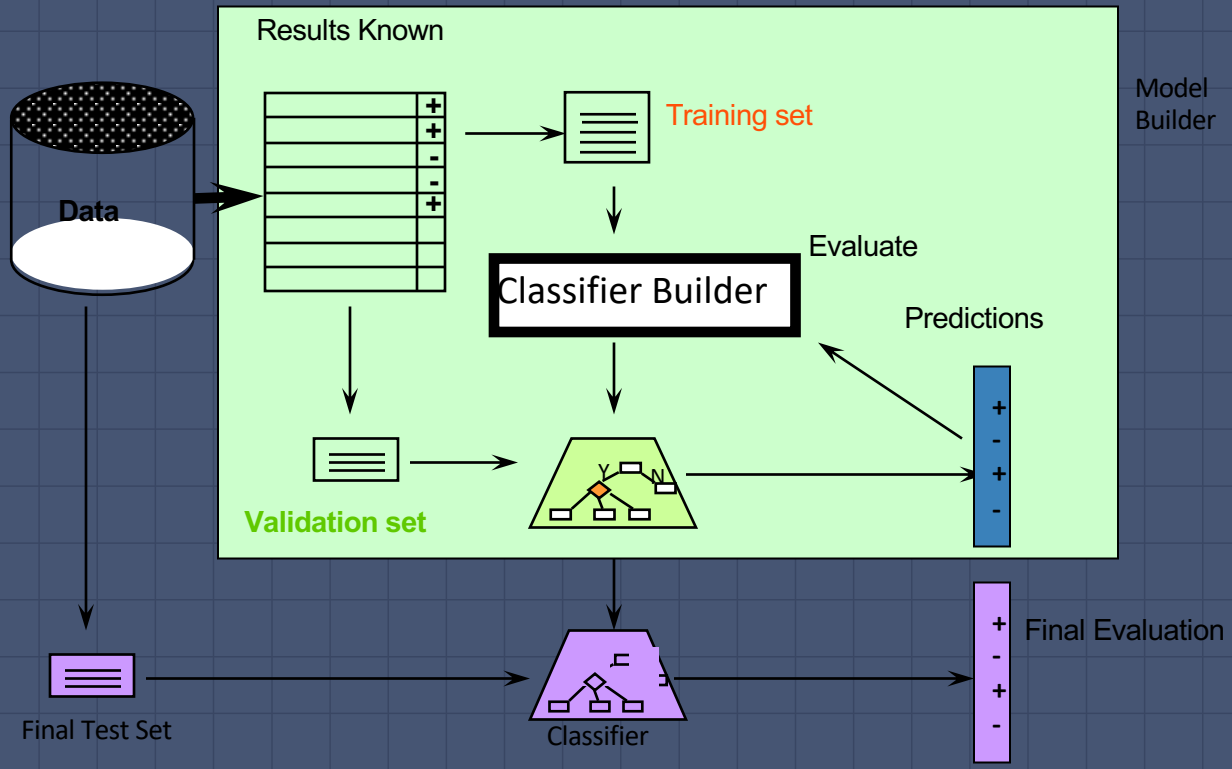
For example: Quinlan in 1987 reported experiments in a medical domain for which the classifiers were trained on data from 1985 and tested on data from 1986.

Hold-out Method

The hold-out method splits the data into training data and test data (usually 2/3 for train, 1/3 for test). Then we build a classifier using the train data and test it using the test data.

The hold-out method is usually used when we have thousands of instances, including several hundred instances from each class.

Classification: Train, Validation, Test Split



The test data can't be used for parameter tuning!

Making the Most of the Data

Once evaluation is complete, *all the data* can be used to build the final classifier.

Generally, the larger the training data the better the classifier (but returns diminish).

The larger the test data the more accurate the error estimate.

Metrics for Classifier's Evaluation

- Accuracy = $(TP+TN)/(P+N)$
- Error = $(FP+FN)/(P+N)$
- Precision = $TP/(TP+FP)$
- Recall/TP rate = TP/P
- FP Rate = FP/N

		Predicted class		
		Pos	Neg	
Actual class	Pos	*TP*	*FN*	*P*
	Neg	*FP*	*TN*	*N*

Confusion Matrix

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

201 1's correctly classified as "1"

85 1's incorrectly classified as "0"

25 0's incorrectly classified as "1"

2689 0's correctly classified as "0"

TABLE 5.1 CLASSIFICATION MATRIX: MEANING OF EACH CELL

	Predicted Class	
Actual Class	C_0	C_1
C_0	$n_{0,0}$ = number of C_0 cases classified correctly	$n_{0,1}$ = number of C_0 cases classified incorrectly as C_1
C_1	$n_{1,0}$ = number of C_1 cases classified incorrectly as C_0	$n_{1,1}$ = number of C_1 cases classified correctly

Error Rate

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

Overall error rate = (25+85)/3000 = 3.67%

Accuracy = 1 – err = (201+2689) = 96.33%

If multiple classes, error rate is:

(sum of misclassified records)/(total records)

Cutoff for classification

Most DM algorithms classify via a 2-step process:

For each record,

1. Compute **probability of belonging to class “1”**
2. Compare to cutoff value, and classify accordingly

Default cutoff value is 0.50

If >= 0.50, classify as “1”

If < 0.50, classify as “0”

Can use different cutoff values

Typically, error rate is lowest for cutoff = 0.50

Cutoff Table

Actual Class	Prob. of "1"	Actual Class	Prob. of "1"
1	0.996	1	0.506
1	0.988	0	0.471
1	0.984	0	0.337
1	0.980	1	0.218
1	0.948	0	0.199
1	0.889	0	0.149
1	0.848	0	0.048
0	0.762	0	0.038
1	0.707	0	0.025
1	0.681	0	0.022
1	0.656	0	0.016
0	0.622	0	0.004

If cutoff is 0.50: 13 records are classified as “1”

If cutoff is 0.80: seven records are classified as “1”

Confusion Matrix for Different Cutoffs

Cut off Prob.Val. for Success (Updatable)	0.25

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	11	1
non-owner	4	8

Cut off Prob.Val. for Success (Updatable)	0.75

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	7	5
non-owner	1	11

Stratification

The *holdout* method reserves a certain amount for testing and uses the remainder for training.

- *Usually: one third for testing, the rest for training.*

For "unbalanced" datasets, samples might not be representative.

- *Few or none instances of some classes.*

Stratified sample: advanced version of balancing the data.

- *Make sure that each class is represented with approximately equal proportions in both subsets.*

Repeated Holdout Method

Holdout estimate can be made more reliable by repeating the process with different subsamples.

- In each iteration, a certain proportion is randomly selected for training (possibly with stratification).
- The error rates on the different iterations are averaged to yield an overall error rate.

This is called the *repeated holdout* method.

Repeated Holdout Method, 2

Still not optimum: the different test sets overlap, but we would like all our instances from the data to be tested at least ones.

Can we prevent overlapping?

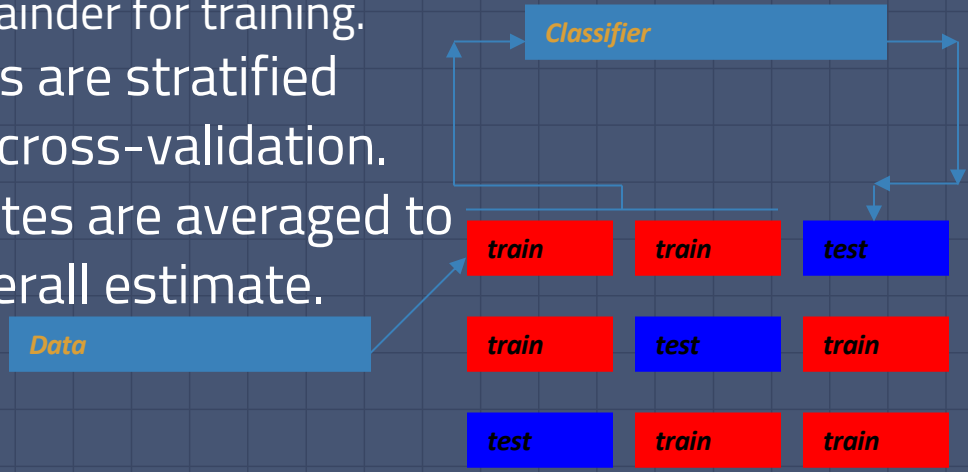
k-Fold Cross-Validation

k-fold cross-validation avoids overlapping test sets:

- *First step*: data is split into *k* subsets of equal size;
- *Second step*: each subset in turn is used for testing and the remainder for training.

The subsets are stratified before the cross-validation.

The estimates are averaged to yield an overall estimate.



More on Cross-Validation

Standard method for evaluation: stratified 10-fold cross-validation.

Why 10? Extensive experiments have shown that this is the best choice to get an accurate estimate.

Stratification reduces the estimate's variance.

Even better: repeated stratified cross-validation:

- E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance).

Leave-One-Out Cross-Validation

Leave-One-Out is a particular form of cross-validation:

- Set number of folds to number of training instances;
- I.e., for *n* training instances, build classifier *n* times.

Makes best use of the data.

Involves no random sub-sampling.

Very computationally expensive.

Leave-One-Out Cross-Validation and Stratification

A disadvantage of Leave-One-Out-CV is that stratification is not possible:

- It *guarantees* a non-stratified sample because there is only one instance in the test set!

Extreme example - random dataset split equally into two classes:

- Best inducer predicts majority class;
- 50% accuracy on fresh data;
- Leave-One-Out-CV estimate is 100% error!

When One Class is More Important

In many cases it is more important to identify members of one class

- Tax fraud
- Credit default
- Response to promotional offer
- Detecting electronic network intrusion
- Predicting delayed flights

In such cases, we are willing to tolerate greater overall error, in return for better identifying the important class for further attention

Alternate Accuracy Measures

If "C_1" is the important class,

Sensitivity = % of "C_1" class correctly classified

Sensitivity = $n_{1,1} / (n_{1,0} + n_{1,1})$

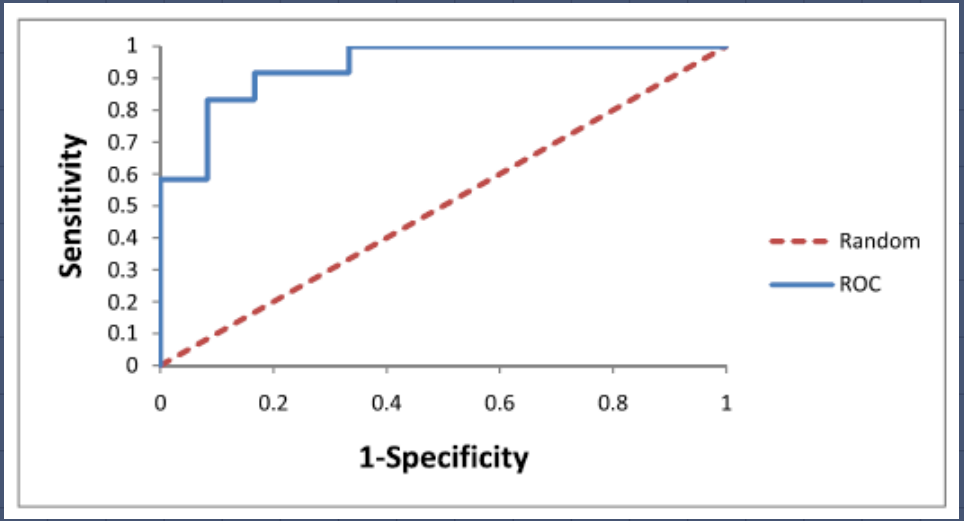
Specificity = % of "C_0" class correctly classified

Specificity = $n_{0,0} / (n_{0,0} + n_{0,1})$

False positive rate = % of predicted "C_1's" that were not "C_1's"

False negative rate = % of predicted "C_0's" that were not "C_0's"

ROC Curve


Sensitivity
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1-Specificity
Random
ROC

Metric Evaluation Summary:

Use test sets and the hold-out method for “large” data;

Use the cross-validation method for “middle-sized” data;

Use the leave-one-out and bootstrap methods for small data;

Don’t use test data for parameter tuning - use separate validation data.

Note

To generate ROC curves or Lift charts we need to use some evaluation methods considered in this lecture.

ROC curves and Lift charts can be used for internal optimization of classifiers.

Reading

http://www.cs.bham.ac.uk/~axk/EvaluatingClassifiers.pdf

Walkthrough

https://www.ritchieng.com/machine-learning-evaluate-classification-model/