Data Mining

Lesson 6

Engineering data mining tasks

MSc in Computer Science
University of New York Tirana
Assoc. Prof. Dr. Marenglen Biba

Data Mining: Content

- Introduction to data mining and machine learning
- Inductive learning
- Decision trees
- Rule induction
- Instance-based learning
- Bayesian learning
- Neural networks
- Support vector machines
- Other machine learning models
- Engineering data mining tasks

Engineering data mining tasks

- Two parts:
  - Credibility: Evaluating what's been learned - Today
  - Transformations: Engineering the input and output

Credibility: Evaluating what’s been learned

Evaluation

- Evaluation is the key to making real progress in data mining.
- There are lots of ways of inferring structure from data: we have encountered some already.
- But to determine which ones to use on a particular problem we need systematic ways to evaluate how different methods work and to compare one with another.
- Evaluation is not as simple as it might appear at first sight.

Evaluation: What's the problem?

- Performance on the training set is definitely not a good indicator of performance on an independent test set.

- We need ways of predicting performance bounds in practice, based on experiments with whatever data can be obtained.
  - When a vast supply of data is available, this is no problem

- But although data mining sometimes involves "big data" —particularly in marketing, sales, and customer support applications — it is often the case that, quality data is scarce.

Scarce data

- The oil slicks mentioned in Lesson 1 had to be detected and marked manually — a skilled and labor-intensive process — before being used as training data.
- Even in the credit card application, there turned out to be only 1000 training examples of the appropriate type.
- The electricity supply data went back 15 years, 5000 days — but only 15 Christmas Days and Thanksgivings, and just 4 February 29s and presidential elections.
- The electromechanical diagnosis application was able to capitalize on 20 years of recorded experience, but this yielded only 300 usable examples of faults.
- Marketing and sales applications certainly involve big data, but many others do not: training data frequently relies on specialist human expertise — and that is always in short supply.

Error rate

- For classification problems, it is natural to measure a classifier's performance in terms of the *error rate.*

- The classifier predicts the class of each instance: if it is correct, that is counted as a *success;* if not, it is an *error.*

- The error rate is just the proportion of errors made over a whole set of instances, and it measures the overall performance of the classifier.

Resubstitution error

- The error rate on the training data is called the *resubstitution error*, because it is calculated by resubstituting the training instances into a classifier that was constructed from them.
- Although it is not a reliable predictor of the true error rate on new data, it is nevertheless often useful to know.
- To predict the performance of a classifier on new data, we need to assess its error rate on a dataset that played no part in the formation of the classifier.
  - This independent dataset is called the *test set.*
- We assume that both the training data and the test data are representative samples of the underlying problem.

Use of test data

- It is important that the test data was not used *in any way* to create the classifier.

- For example, some learning methods involve two stages, one to come up with a basic structure and the second to optimize parameters involved in that structure, and separate sets of data may be needed in the two stages.

- In such situations people often talk about three datasets: the *training* data, the *validation* data, and the *test* data.

Independent datasets

- The training data is used by one or more learning methods to come up with classifiers.
- The validation data is used to optimize parameters of those classifiers, or to select a particular one.
- Then the test data is used to calculate the error rate of the final, optimized, method.
- Each of the three sets must be chosen independently:
  - the validation set must be different from the training set to obtain good performance in the optimization or selection stage, and
  - the test set must be different from both to obtain a reliable estimate of the true error rate.

Training and test samples

- If lots of data is available, there is no problem: we take a large sample and use it for training; then another, independent large sample of different data and use it for testing.
- Provided that both samples are representative, the error rate on the test set will give a true indication of future performance.
- Generally, the larger the training sample the better the classifier, although the returns begin to diminish once a certain volume of training data is exceeded.
- And the larger the test sample, the more accurate the error estimate.
- The accuracy of the error estimate can be quantified statistically.

Holdout

- Now consider what to do when the amount of data for training and testing is limited.
- The *holdout method* reserves a certain amount for testing and uses the remainder for training (and sets part of that aside for validation, if required).
- In practical terms, it is common to hold out one-third of the data for testing and use the remaining two-thirds for training.
- Of course, you may be unlucky: the sample used for training (or testing) might not be representative.

Stratification

- In general, you cannot tell whether a sample is representative or not.
- But there is one simple check that might be worthwhile: each class in the full dataset should be represented in about the right proportion in the training and testing sets.
- If, by bad luck, all examples with a certain class were missing from the training set, you could hardly expect a classifier to perform well on the examples of that class.
- The situation would be exacerbated by the fact that the class would necessarily be overrepresented in the test set because none of its instances made it into the training set!

Stratification

- Instead, we should ensure that the random sampling is done in such a way as to guarantee that each class is properly represented in both training and test sets.

- This procedure is called *stratification,* and we might speak of *stratified holdout.*

- Although it is generally well worth doing, stratification provides only a primitive safeguard against uneven representation in training and test sets.

Cross-validation

- In cross-validation, you decide on a fixed number of *folds,* or partitions of the data.
- Suppose we use three. Then the data is split into three approximately equal partitions and each in turn is used for testing and the remainder is used for training.
- That is, use two-thirds for training and one-third for testing and repeat the procedure three times so that, in the end, every instance has been used exactly once for testing.
- This is called *threefold cross-validation,* and if stratification is adopted as well — which it often is — it is *stratified threefold cross-validation.*

10-fold cross-validation

- The standard way of predicting the error rate of a learning technique given a single, fixed sample of data is to use stratified 10-fold cross-validation.
- Why 10?
- Extensive tests on numerous datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up.
- Although these arguments are by no means conclusive, and debate continues to rage in machine learning and data mining circles about what is the best scheme for evaluation, 10-fold cross-validation has become the standard method in practical terms.

10 times 10-fold cross-validation

- A single 10-fold cross-validation might not be enough to get a reliable error estimate.
- Different 10-fold cross-validation experiments with the same learning method and dataset often produce different results, because of the effect of random variation in choosing the folds themselves.
- Stratification reduces the variation, but it certainly does not eliminate it entirely.
- When seeking an accurate error estimate, it is standard procedure to repeat the cross-validation process 10 times — that is, 10 times 10-fold cross-validation — and average the results.
- This involves invoking the learning algorithm 100 times on datasets that are all nine-tenths the size of the original.
- Obtaining a good measure of performance is a computation-intensive undertaking.

Leave-one-out cross-validation

- Leave-one-out cross-validation is simply *n*-fold cross-validation, where *n* is the number of instances in the dataset.
- Each instance in turn is left out, and the learning method is trained on all the remaining instances.
- It is judged by its correctness on the remaining instance — one or zero for success or failure, respectively.
- The results of all *n* judgments, one for each member of the dataset, are averaged, and that average represents the final error estimate.

Leave-one-out cross-validation: advantages

- This procedure is an attractive one for two reasons.
  - First, the greatest possible amount of data is used for training in each case, which presumably increases the chance that the classifier is an accurate one.
  - Second, the procedure is deterministic: no random sampling is involved. There is no point in repeating it 10 times, or repeating it at all: the same result will be obtained each time.
- Set against this is the high computational cost, because the entire learning procedure must be executed *n* times and this is usually quite infeasible for large datasets.
- Nevertheless, leave-one-out seems to offer a chance of squeezing the maximum out of a small dataset and obtaining as accurate an estimate as possible.

Leave-one-out cross-validation: disadvantages

- But there is a disadvantage to leave-one-out cross-validation, apart from the computational expense.

- By its very nature, it cannot be stratified — worse than that, it *guarantees* a nonstratified sample.

- Stratification involves getting the correct proportion of examples in each class into the test set, and this is impossible when the test set contains only a single example.

Comparing data mining methods

Comparing data mining methods

- We often need to compare two different learning methods on the same problem to see which is the better one to use.

- It seems simple: estimate the error using cross-validation (or any other suitable estimation procedure), perhaps repeated several times, and choose the scheme whose estimate is smaller.

- However, it may be that the difference is simply caused by the particular dataset or randomness and in some circumstances it is important to determine whether one scheme is really better than another on a particular problem.

Comparing data mining methods

- This is a standard challenge for machine learning researchers.
- If a new learning algorithm is proposed, its proponents must show that it improves on the state of the art for the problem at hand and demonstrate that the observed improvement is not just a chance effect in the estimation process.
- This is a job for a statistical test that gives confidence bounds.

Significance of the difference

- If the difference turns out to be significant we must ensure that this is not just because of the particular dataset we happened to base the experiment on.

- What we want to determine is whether one scheme is better or worse than another on average, across all possible training and test datasets that can be drawn from the domain.

- Because the amount of training data naturally affects performance, all datasets should be the same size: indeed, the experiment might be repeated with different sizes to obtain a learning curve.

Comparing means

- Assume that the supply of data is unlimited.
- For definiteness, suppose that cross-validation is being used to obtain the error estimates (other estimators, such as repeated cross-validation, are equally viable).
- For each learning method we can draw several datasets of the same size, obtain an accuracy estimate for each dataset using cross-validation, and compute the mean of the estimates.
- Each cross-validation experiment yields a different, independent error estimate.
- What we are interested in is the mean accuracy across all possible datasets of the same size, and whether this mean is greater for one scheme or the other.

Paired t-test

- From this point of view, we are trying to determine whether the mean of a set of samples — cross-validation estimates for the various datasets that we sampled from the domain — is significantly greater than, or significantly less than, the mean of another.

- This is a job for a statistical device known as the *t-test*, or *Student's t-test.*

- Because the same cross-validation experiment can be used for both learning methods to obtain a matched pair of results for each dataset, a more sensitive version of the *t*-test known as a *paired t-test* can be used.

Paired t-test

- There is a set of samples $x_1, x_2, \ldots, x_k$ obtained by successive 10-fold cross-validations using one learning scheme, and a second set of samples $y_1, y_2, \ldots, y_k$ obtained by successive 10-fold cross-validations using the other.
- Each cross-validation estimate is generated using a different dataset (but all datasets are of the same size and from the same domain).
- We will get the best results if exactly the same cross-validation partitions are used for both schemes so that x_1 and y_1 are obtained using the same cross-validation split, as are x_2 and y_2, and so on.
- Denote the mean of the first set of samples by $\bar{x}$ and the mean of the second set by $\bar{y}$.
- We are trying to determine whether $\bar{x}$ is significantly different from $\bar{y}$.

Cost-sensitive machine learning

Cost examples

- Diagnosis: the cost of misidentifying problems with a machine that turns out to be free of faults is less than the cost of overlooking problems with one that is about to fail.

- Promotional mailing: the cost of sending junk mail to a household that doesn't respond is far less than the lost-business cost of not sending it to a household that would have responded.

Possible outcomes for two class prediction

Table 5.3 Different outcomes of a two-class prediction.

		Predicted class: yes	Predicted class: no
Actual class	yes	true positive	false negative
	no	false positive	true negative

Cost-sensitive classification

- If the costs are known, they can be incorporated into a financial analysis of the decision-making process.

- Taking the cost matrix into account replaces the success rate by the average cost (or, thinking more positively, profit) per decision.

Cost-sensitive learning

- A classifier can be built without taking costs into consideration, but can be used to make predictions that are cost sensitive.
  - In this case, costs are ignored at training time but used at prediction time.

- An alternative is to do just the opposite: take the cost matrix into account during the training process and ignore costs at prediction time.

- In principle, better performance might be obtained if the classifier were tailored by the learning algorithm to the cost matrix.

Lift charts

- If you knew the different costs involved, you could work them out for each sample size and choose the most profitable.
- But a graphical depiction of the various possibilities will often be far more revealing than presenting a single “optimal” decision.
- Repeating the preceding operation for different-sized samples allows you to plot a lift chart like that of Figure 5.1. => next slide
- The horizontal axis shows the sample size as a proportion of the total possible mailout.
- The vertical axis shows the number of responses obtained.
- The lower left and upper right points correspond to no mailout at all, with a response of 0, and a full mailout, with a response of 1000.

Lift charts

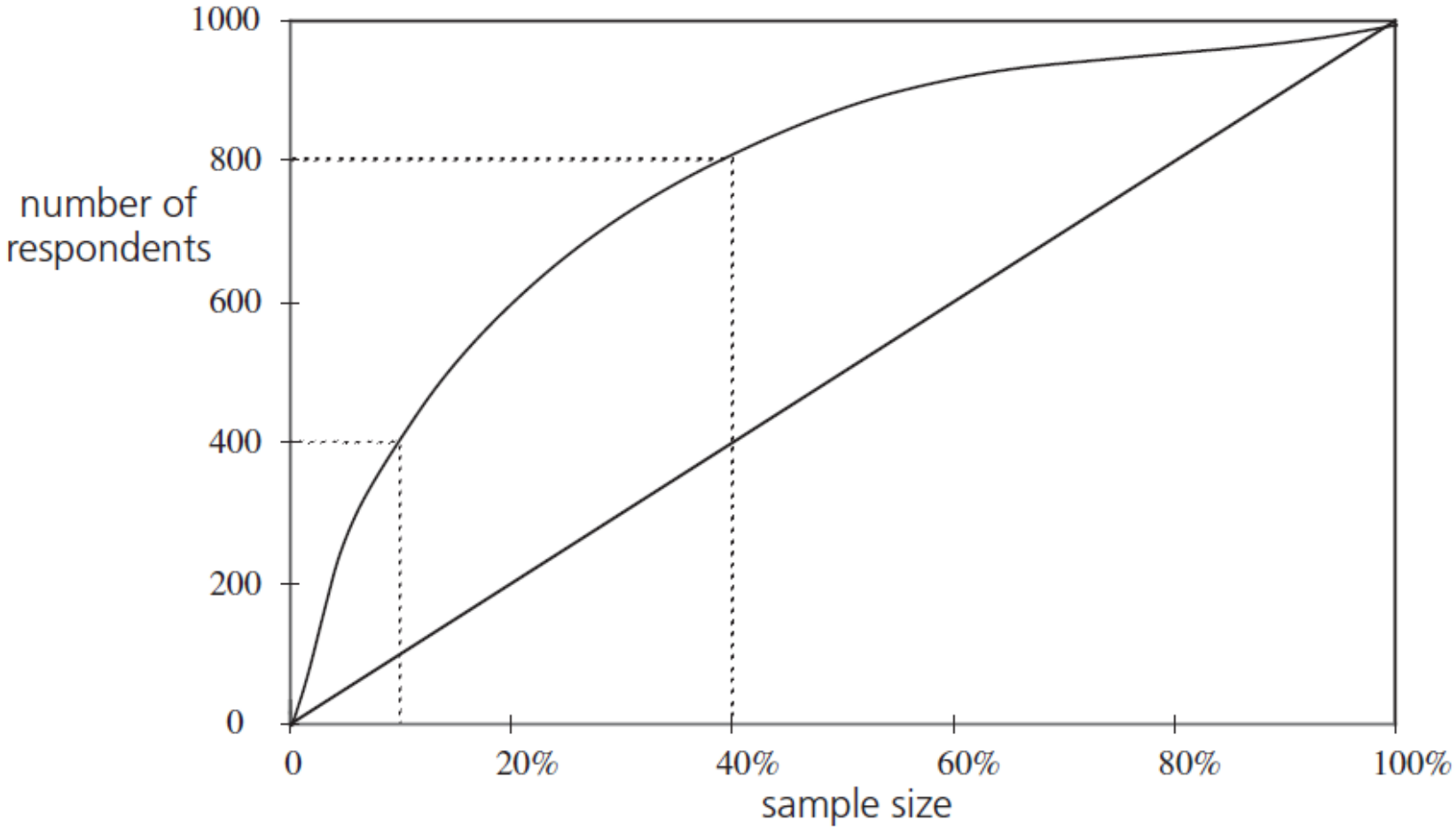

Figure 5.1 A hypothetical lift chart.

ROC curves

- *ROC curves* are a graphical technique for evaluating data mining schemes, which are used in situations where the learner is trying to select samples of test instances that have a high proportion of positives.

- The acronym stands for *receiver operating characteristic*, a term used in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel.

- ROC curves depict the performance of a classifier without regard to class distribution or error costs.

ROC curves

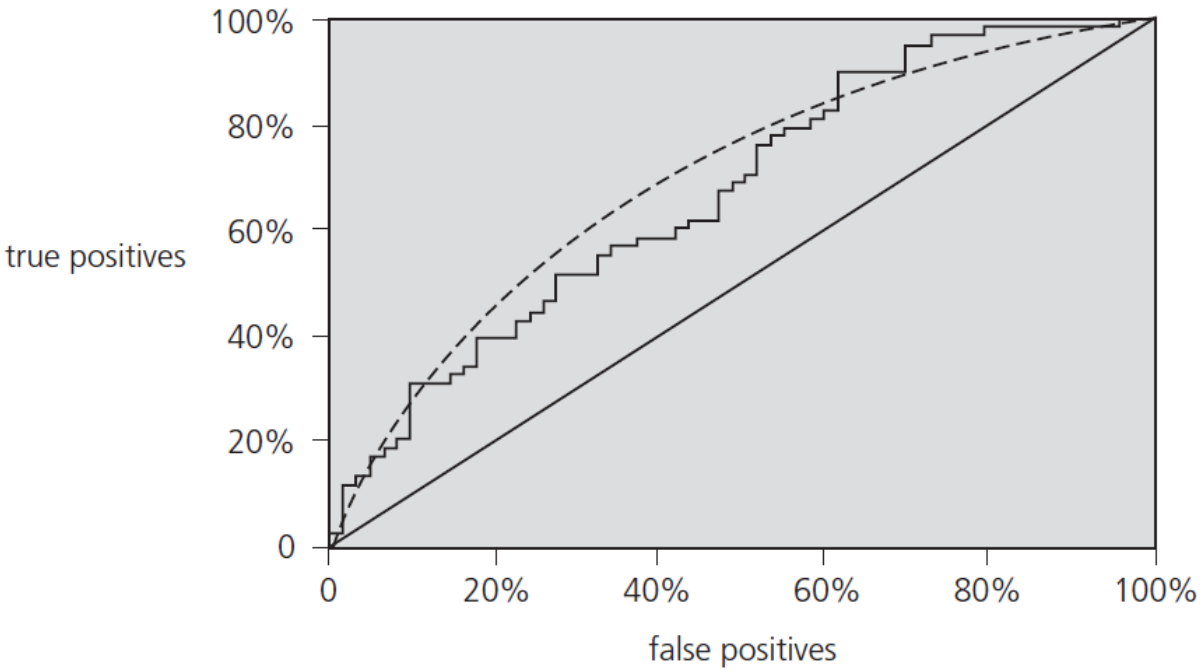

Figure 5.2 A sample ROC curve.

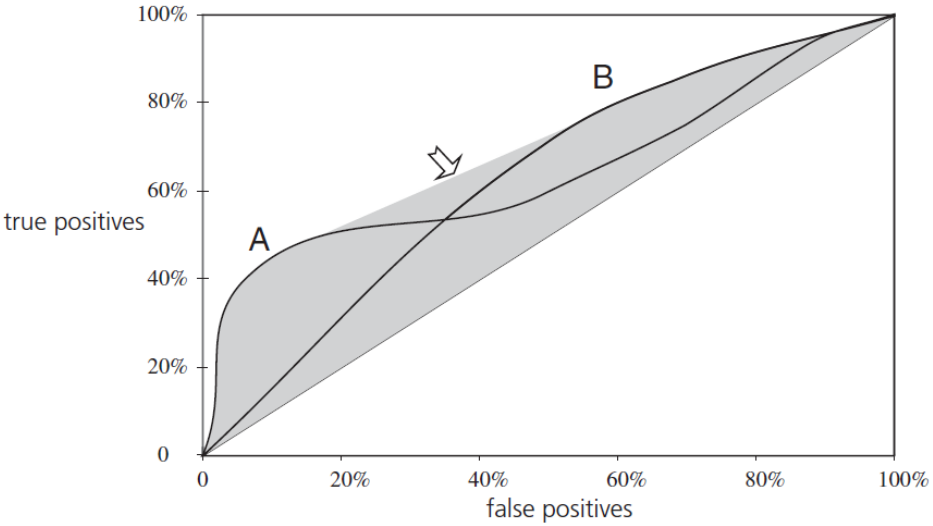

Figure 5.3 ROC curves for two learning methods.

- Method A excels if a small, focused sample is sought; that is, if you are working toward the left-hand side of the graph.
- Clearly, if you aim to cover just 40% of the true positives you should choose method A, which gives a false positive rate of around 5%, rather than method B, which gives more than 20% false positives.
- But method B excels if you are planning a large sample: if you are covering 80% of the true positives, method B will give a false positive rate of 60% as compared with method A's 80%.
- The shaded area is called the convex hull of the two curves, and you should always operate at a point that lies on the upper boundary of the convex hull.

Recall–precision curves

- Given a query, a Web search engine produces a list of hits that represent documents supposedly relevant to the query.
- Compare one system that locates 100 documents, 40 of which are relevant, with another that locates 400 documents, 80 of which are relevant.
- Which is better? The answer should now be obvious: it depends on the relative cost of false positives, documents that are returned that aren't relevant, and false negatives, documents that are relevant that aren't returned.
- Information retrieval researchers define parameters called *recall* and *precision*.

Recall–precision curves

$$\text{recall} = \frac{\text{number of documents retrieved that are relevant}}{\text{total number of documents that are relevant}}$$

$$\text{precision} = \frac{\text{number of documents retrieved that are relevant}}{\text{total number of documents that are retrieved}}.$$

- Information retrieval experts use *recall–precision curves* that plot one against the other, for different numbers of retrieved documents.

Different measures

Table 5.7 **Different measures used to evaluate the false positive versus the false negative tradeoff.**

	Domain	Plot	Axes	Explanation of axes
lift chart	marketing	TP vs. subset size	TP subset size	number of true positives $\frac{TP+FP}{TP+FP+TN+FN} \times 100\%$
ROC curve	communications	TP rate vs. FP rate	TP rate FP rate	$tp = \frac{TP}{TP+FN} \times 100\%$ $fp = \frac{FP}{FP+TN} \times 100\%$
recall–precision curve	information retrieval	recall vs. precision	recall precision	same as TP rate *tp* $\frac{TP}{TP+FP} \times 100\%$

Evaluating numeric prediction

- Several alternative measures can be used to evaluate the success of numeric prediction.

- The predicted values on the test instances are p_1, p_2, . . ., p_n; the actual values are a_1, a_2, . . ., a_n.

Table 5.8 Performance measures for numeric prediction*.

Performance measure	Formula
mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
mean absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
relative squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$, where $\bar{a}=\frac{1}{n}\sum_i a_i$
root relative squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
relative absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
correlation coefficient	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_p=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, and $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$

* p are predicted values and a are actual values.

Mean-squared and mean-absolute error

- *Mean-squared error* is the principal and most commonly used measure; sometimes the square root is taken to give it the same dimensions as the predicted value itself.
- Many mathematical techniques, such as linear regression, use the mean-squared error because it tends to be the easiest measure to manipulate mathematically.
- The question is, is it an appropriate measure for the task at hand?
- *Mean absolute error* is an alternative: just average the magnitude of the individual errors without taking account of their sign.
- Mean-squared error tends to exaggerate the effect of outliers — instances whose prediction error is larger than the others — but absolute error does not have this effect: all sizes of error are treated evenly according to their magnitude.

Relative squared error

- The error is made relative to what it would have been if a simple predictor had been used.
- The simple predictor in question is just the average of the actual values from the training data.
- Thus relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the default predictor.
- The next error measure goes by the name of *relative absolute error* and is just the total absolute error, with the same kind of normalization.
- In these three relative error measures, the errors are normalized by the error of the simple predictor that predicts average values.

Correlation coefficient

- The *correlation coefficient*, which measures the statistical correlation between the *a*'s and the *p*'s.

- The correlation coefficient ranges from 1 for perfectly correlated results, through 0 when there is no correlation, to -1 when the results are perfectly correlated negatively.

Different performance measures

- The squared error measures and root squared error measures weigh large discrepancies much more heavily than small ones, whereas the absolute error measures do not.

Different performance measures

Table 5.9 Performance measures for four numeric prediction models.

	A	B	C	D
root mean-squared error	67.8	91.7	63.3	57.4
mean absolute error	41.3	38.5	33.4	29.2
root relative squared error	42.2%	57.2%	39.4%	35.8%
relative absolute error	43.1%	40.1%	34.8%	30.4%
correlation coefficient	0.88	0.88	0.89	0.91

- Which of these measures is appropriate in any given situation is a matter that can only be determined by studying the application itself.
- What are we trying to minimize?
- What is the cost of different kinds of error? Often it is not easy to decide.

Which measure to use?

- Fortunately, it turns out that in most practical situations the best numeric prediction method is still the best no matter which error measure is used.
- For example, Table 5.9 shows the result of four different numeric prediction techniques on a given dataset, measured using cross-validation.
- Method D is the best according to all five metrics: it has the smallest value for each error measure and the largest correlation coefficient.
- Method C is the second best by all five metrics.
- The performance of methods A and B is open to dispute: they have the same correlation coefficient, method A is better than method B according to both mean-squared and relative squared errors, and the reverse is true for both absolute and relative absolute error.
- It is likely that the extra emphasis that the squaring operation gives to outliers accounts for the differences in this case.

Comparing learning schemes

- When comparing two different learning schemes that involve numeric prediction, the methodology developed previously still applies.

- The only difference is that success rate is replaced by the appropriate performance measure (e.g., root mean-squared error) when performing the significance test.

Readings

- Data Mining. Witten and Frank
  - Chapter 7, except section 7.5

Lab Session

1. Comparing data mining methods in Weka
2. Comparing models in Oracle

Comparison of methods in Weka

- Experimenter in Weka
- Discrete-valued functions
- Percentage correct comparison
  - Iris dataset
    - Naïve Bayes, Decision Trees, SVMs (no sign. diff)
      - 10-fold and 10 repetitions
  - Credit-g
    - Naïve Bayes, Decision Trees, SVMs (DT are sign. worst)
      - 10-fold and 10 repetitions
  - Vehicle
    - Naïve Bayes, Decision Trees, SVMs (DT and SVM sign better)
      - 10-fold and 10 repetitions

Comparison of methods in Weka

- Experimenter in Weka
- Continuous-valued functions
  - Autoprice
    - Linear Reg, SVMs, Perceptron, 1BK
      - Root. Rel. Sq. Error. (no sign.diff. even though values are different)
      - Rel.Abs.Error. SVMs sign. Worse
  - CPU
    - Linear Reg, SVMs, Perceptron, 1BK
      - Correlation Coeff. (Perceptron better)
      - Mean.Abs. Error. (Lin.Reg better)
      - Root,M.S.Error (Perceptron is worse)

Comparing Models in Oracle

- Comparing classification models

End of Class