**Assignment4**

Name:Dew

ID:18406040245

***Assignment：Write a brief note on various Evaluation methods you have learned.***

• Issues: training, testing, tuning

• Predicting performance: confidence limits

• Holdout, cross-validation, bootstrap

• Hyperparameter selection

• Comparing machine learning schemes

• Predicting probabilities

• Cost-sensitive evaluation

• The minimum description length principle

[1.Issues: training, testing, tuning 2](#_Toc1461418512)

[Issues in evaluation 2](#_Toc1290460445)

[Training and testing I 2](#_Toc1331348062)

[Training and testing II 2](#_Toc1334759941)

[Note on parameter tuning 3](#_Toc696151825)

[Making the most of the data 3](#_Toc732813919)

[2.Predicting performance: confidence limits 3](#_Toc584821088)

[Predicting performance 3](#_Toc55373697)

[Confidence intervals 3](#_Toc805306328)

[Mean and variance 4](#_Toc1341511302)

[Confidence limits 4](#_Toc349642861)

[Transforming f 4](#_Toc932306635)

[3.Holdout, cross-validation, bootstrap 5](#_Toc1236925933)

[Holdout estimation 5](#_Toc1372452971)

[Repeated holdout method 5](#_Toc695231170)

[Cross-validation 5](#_Toc291750863)

[10-fold cross-validation 6](#_Toc751588340)

[Leave-one-out cross-validation 6](#_Toc446418726)

[Leave-one-out CV Issues 6](#_Toc1799148911)

[The bootstrap 6](#_Toc1725997417)

[4.Hyperparameter selection 7](#_Toc629483843)

[Hyperparameter selection 7](#_Toc1230504179)

[Hyperparameters and cross-validation 7](#_Toc816215843)

[5.Comparing machine learning schemes 8](#_Toc14136265)

[Comparing machine learning schemes 8](#_Toc1365004685)

[Comparing learning schemes II 8](#_Toc65939894)

[6.Predicting probabilities 9](#_Toc150236606)

[Predicting probabilities 9](#_Toc1733351817)

[Quadratic loss function 9](#_Toc1828316764)

[Informational loss function 9](#_Toc176347625)

[Discussion 10](#_Toc347100515)

[7.Cost-sensitive evaluation 10](#_Toc1152770353)

[8.The minimum description length principle 10](#_Toc13859637)

[Minimum Description Length 10](#_Toc1010685183)

# 1.Issues: training, testing, tuning

## Issues in evaluation

•Statistical reliability of estimated differences in performance (→significance tests)

• Choice of performance measure:

• Number of correct classifications

• Accuracy of probability estimates

• Error in numeric predictions

• Costs assigned to different types of errors

• Many practical applications involve costs

## Training and testing I

• Natural performance measure for classification problems: error rate

• Success: instance’s class is predicted correctly

• Error: instance’s class is predicted incorrectly

• Error rate: proportion of errors made over the whole set of instances

• Resubstitution error: error rate obtained by evaluating model on training data

• Resubstitution error is (hopelessly) optimistic!

## Training and testing II

• Test set: independent instances that have played no part in formation of classifier

• Assumption: both training data and test data are representative samples of the underlying problem

• Test and training data may differ in nature

• Example: classifiers built using customer data from two different towns A and B

• To estimate performance of classifier from town A in completely new town, test it on data from B

## Note on parameter tuning

• It is important that the test data is not used in any way to create the classifier

• Some learning schemes operate in two stages:

• Stage 1: build the basic structure

• Stage 2: optimize parameter settings

• The test data cannot be used for parameter tuning!

• Proper procedure uses three sets: training data, validation data, and test data

• Validation data is used to optimize parameters

## 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

• Holdout procedure: method of splitting original data into training and test set

• Dilemma: ideally both training set and test set should be large!

# 2.Predicting performance: confidence limits

## Predicting performance

• Assume the estimated error rate is 25%. How close is this to the true error rate?

• Depends on the amount of test data

• Prediction is just like tossing a (biased!) coin

• “Head” is a “success”,”tail” is an “error”

• In statistics, a succession of independent events like this is called a Bernoulli process

• Statistical theory provides us with confidence intervals for the true underlying proportion

## Confidence intervals

• We can say: p lies within a certain specified interval with a certain specified confidence

• Example: S=750 successes in N=1000 trials

• Estimated success rate: 75%

• How close is this to true success rate p?

• Answer: with 80% confidence p is located in [73.2,76.7]

• Another example: S=75 and N=100

• Estimated success rate: 75%

• With 80% confidence p in [69.1,80.1]

## Mean and variance

• Mean and variance for a Bernoulli trial: p, p(1–p)

• Expected success rate f=S/N

• Mean and variance for f : p, p(1–p)/N

• For large enough N, f follows a Normal distribution

• c% confidence interval [–z ≤ X ≤ z] for a random variable X is determined using:

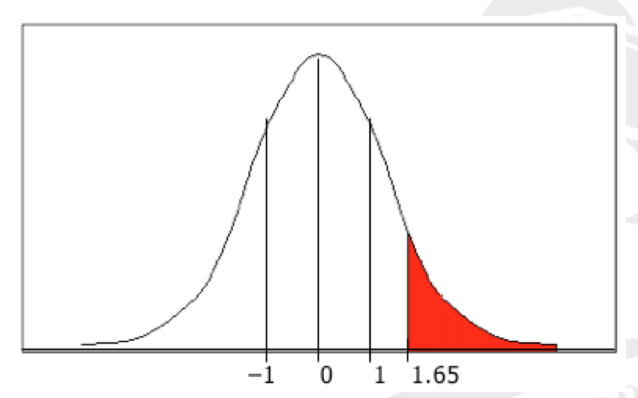
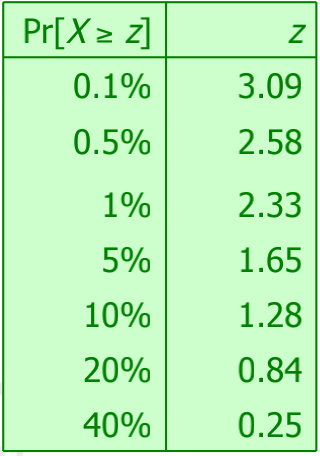
P(−z ≤ X ≤ z) = c

• For a symmetric distribution such as the normal distribution we have:

P(−z ≤ X ≤ z) =1 − 2 × P(x ≥ 2)

## Confidence limits

• Confidence limits for the normal distribution with 0 mean and a variance of 1:



•

Thus:

P(−1.65 ≤ X ≤ 1.65) = 90%

• To use this we have to transform our random variable f to have 0 mean and unit variance

## Transforming f

• Transformed value for f :

• Resulting equation:

• Solving for p yields an expression for the confidence limits:

# 3.Holdout, cross-validation, bootstrap

## Holdout estimation

• What should we do if we only have a single dataset?

• The holdout method reserves a certain amount for testing and uses the remainder for training, after shuffling

• Usually: one third for testing, the rest for training

• Problem: the samples might not be representative

• Example: class might be missing in the test data

• Need to ensure that each class is represented with approximately equal proportions in both subsets

• No over or under representation of class(es) in testing or training sets

• Stratified holdout can be a better method

•Stratification: Random sampling in a way that guarantees each class is properly represented 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 stratificiation)

• The error rates on the different iterations are averaged to yield an overall error rate

• This is called the repeated holdout method

• Still not optimum: the different test sets overlap

• Can we prevent overlapping?

## Cross-validation

• Swap the roles of testing and training in Single holdout estimation. Uses a 50:50 split, which is not ideal

• K-fold cross-validation avoids overlapping test sets

• First step: split data into k subsets of equal size

• Second step: use each subset in turn for testing, the remainder for training

• This means the learning algorithm is applied to k different training sets

• Often the subsets are stratified before the cross-validation is performed to yield stratified k-fold cross-validation

• The error estimates are averaged to yield an overall error estimate; also, standard deviation is often computed

• Alternatively, predictions and actual target values from the k folds are pooled to compute one estimate

• Does not yield an estimate of standard deviation

## 10-fold cross-validation

• Standard method for evaluation: stratified ten-fold cross validation

• Why ten?

• Extensive experiments have shown that this is the best choice to get an accurate estimate

• There is also some theoretical evidence for this

• 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:

A particular form of k-fold 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

• Deterministic: Involves no random subsampling

• Very computationally expensive (exception: using lazy classifiers such as the nearest-neighbor classifier)

## Leave-one-out CV Issues

• Disadvantage of Leave-one-out CV: 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 gives 100% error!

## The bootstrap

• CV uses sampling without replacement

• The same instance, once selected, can not be selected again for a particular training/test set

• The bootstrap uses sampling with replacement to form the training set

• Sample a dataset of n instances n times with replacement to form a new dataset of n instances

• Use this data as the training set

• Use the instances from the original dataset that do not occur in the new training set for testing

# 4.Hyperparameter selection

## Hyperparameter selection

• Hyperparameter: parameter that can be tuned to optimize the performance of a learning algorithm

– Different from basic parameter that is part of a model, such as a coefficient in a linear regression model

– Example hyperparameter: k in the k-nearest neighbour classifier

• We are not allowed to peek at the final test data to choose the value of this parameter

– Adjusting the hyperparameter to the test data will lead to optimistic performance estimates on this test data!

– Parameter tuning needs to be viewed as part of the learning algorithm and must be done using the training data only

• But how to get a useful estimate of performance for different parameter values so that we can choose a value?

– Answer: split the data into a smaller “training” set and a validation set” (normally, the data is shuffled first)

– Build models using different values of k on the new, smaller training set and evaluate them on the validation set

– Pick the best value of k and rebuild the model on the full original training set

## Hyperparameters and cross-validation

• Note that k-fold cross-validation runs k different train-test evaluations

– The above parameter tuning process using validation sets must be applied separately to each of the k training sets!

• This means that, when hyperparameter tuning is applied, k different hyperparameter values may be selected

– This is OK: hyperparameter tuning is part of the learning process

– Cross-validation evaluates the quality of the learning process, not the quality of a particular model

• What to do when the training sets are very small, so that performance estimates on a validation set are unreliable?

• We can use nested cross-validation (expensive!)

– For each training set of the “outer” k-fold cross-validation, run “inner” p-fold cross-validations to choose the best hyperparameter value

– Outer cross-validation is used to estimate quality of learning process

– Inner cross-validations are used to choose hyperparameter values

– Inner cross-validations are part of the learning process!

# 5.Comparing machine learning schemes

## Comparing machine learning schemes

• Frequent question: which of two learning schemes performs better?

• Note: this is domain dependent!

• Obvious way: compare 10-fold cross-validation estimates

• Generally sufficient in applications (we do not loose if the chosen method is not truly better)

• However, what about machine learning research?

• Need to show convincingly that a particular method works better in a particular domain from which data is taken

## Comparing learning schemes II

• Want to show that scheme A is better than scheme B in a particular domain

– For a given amount of training data (i.e., data size)

– On average, across all possible training sets from that domain

• Let's assume we have an infinite amount of data from the domain

• Then, we can simply

– sample infinitely many dataset of a specified size

– obtain a cross-validation estimate on each dataset for each scheme

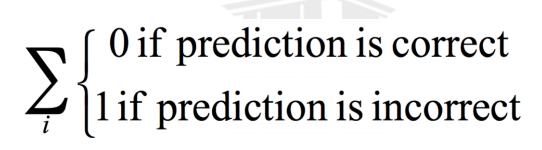
– check if the mean accuracy for scheme A is better than the mean accuracy for scheme B

# 6.Predicting probabilities

## Predicting probabilities

• Performance measure so far: success rate

• Also called 0-1 loss function:



• Most classifiers produces class probabilities

• Depending on the application, we might want to check the accuracy of the probability estimates

• 0-1 loss is not the right thing to use in those cases

## Quadratic loss function

• p1…pk are probability estimates for an instance

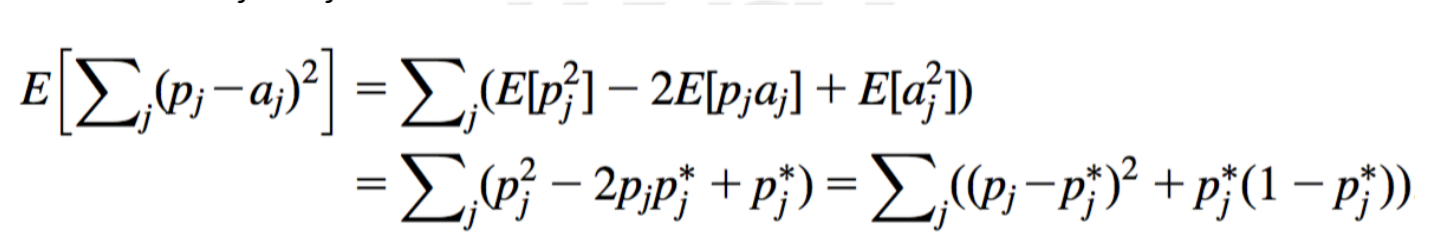
• c is the index of the instance’s actual class

• a1… ak = 0, except for ac which is 1

• Quadratic loss is:

• Want to minimize

• Can show that the expected value of this is minimized when pj = pj\*, where the latter are the true probabilities:



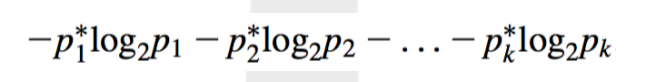
## Informational loss function

• The informational loss function is –log(pc), where c is the index of the instance’s actual class

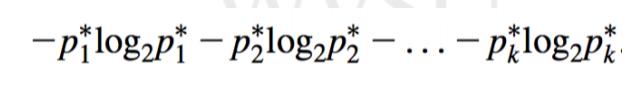
• Number of bits required to communicate the actual class

• Let p1\*…pk\* be the true class probabilities

• Then the expected value for the loss function is:



• Justification for informational loss is that this is minimized when pj = pj\*:



• Difficulty with informational loss : zero-frequency problem

## Discussion

• Which loss function to choose?

• Both encourage honesty

• Quadratic loss function takes into account all class probability estimates for an instance

• Informational loss focuses only on the probability estimate for the actual class

• Quadratic loss is bounded by

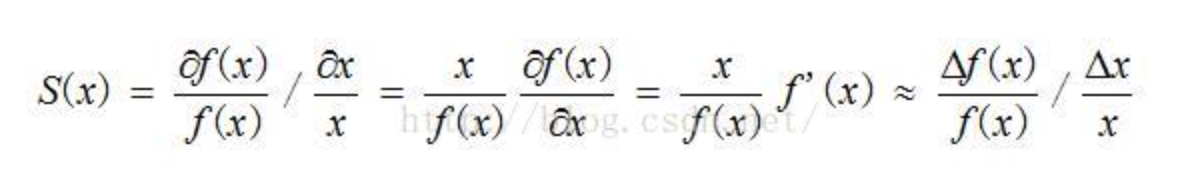
it can never exceed 2

• Informational loss can be infinite

• Informational loss is related to the MDL principle

# 7.Cost-sensitive evaluation

Sensitivity analysis is to study the uncertainty in the output of a model, and further determine the source of uncertainty, that is, to study which input parameter changes the degree of output changes. So sensitivity analysis is an indispensable routine step in the process of mathematical modeling.



# 8.The minimum description length principle

## Minimum Description Length

• objective

• The purpose of minimum description length (MDL) is to explain map according to the basic concepts of information theory.

• theoretical basis

• A. Maximum a posteriori hypothesis (map)

• B. Optimal coding

• Information theory explanation of minimum description length criterion

