Evaluation

Based on slides by Ian Witten and Frank Eibe

Evaluation

- How predictive is the model we learned?

- Error on the training data is *not* a good indicator of performance on future data

- Simple solution that can be used if lots of (labeled) data is available:
  - Split data into training and test set

- However: (labeled) data is usually limited
  - More sophisticated techniques need to be used

Training and testing

- 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 from training data
  - Resubstitution error is (hopelessly) optimistic!

- *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

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

- 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!

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 would like to state propositions such as:

  The true success rate lies within a certain interval

  with a certain confidence

How to derive such numbers? See following slides.

- Example: 750 successes in N=1000 trials
  - Estimated success rate: 75%
  - How close is this to the true success rate p?
    - Answer: with 80% confidence $p \in [73.2, 76.7]$
- Another example: 75 successes in N=100 trials
  - Estimated success rate: 75%
  - With 80% confidence $p \in [69.1, 80.1]$
    - i.e. the probability that $p \in [69.1, 80.1]$ is 0.8.

- Bigger the N more confident we are, i.e. the surrounding interval is smaller.
  - Above: for N=100 we were less confident than for N=1000.

Mean and Variance of S/N

- Let S_N be the random variable for the success rate in N trials.
- Let the true probability of success be p.
- Then the true probability of error is $q=1-p$.

- What's the mean of S_1 i.e. $N=1$?

  $1*p + 0*q = p$
- What's the variance?

  $(1-p)^2*p + (0-p)^2*q$

  $= q^2*p+p^2*q$

  $= pq(p+q)$

  $= pq$
- In general for S_N:
  - the mean continues to be p, however
  - the variance is: pq/N.

Estimating p

- **We approximate p with the success rate in N trials, i.e. S/N.**
  - We denote the approximation by p'

- By the Central Limit Theorem, when N is big, the probability distribution of S_N is approximated by a normal distribution with
  - mean p and
  - variance pq/N.

Estimating *p*

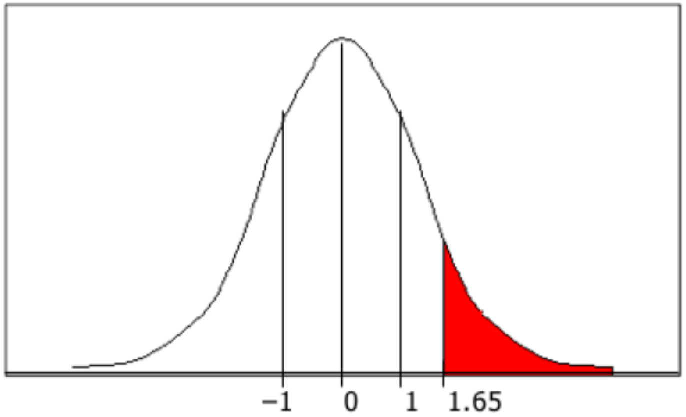
- c% confidence interval [$-z \le X \le z$] for random variable with 0 mean is given by:

$$\Pr[-z \le X \le z] = c$$

- With a symmetric distribution:

$$\Pr[-z \le X \le z] = 1 - 2 \times \Pr[x \ge z]$$

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



$\Pr[X \ge z]$	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

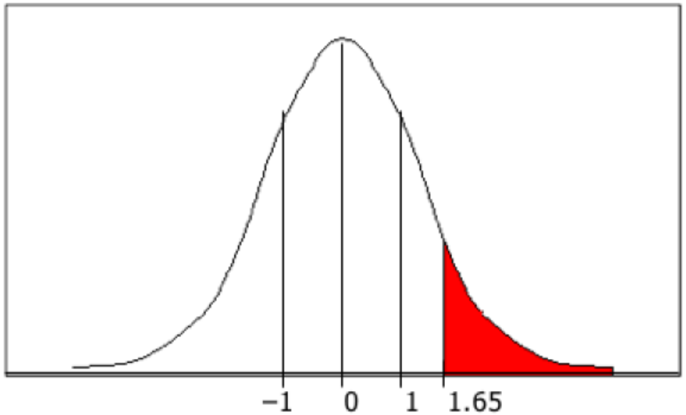
Thus: $\Pr[-1.65 \le X \le 1.65] = 90\%$

Estimating p

To use this we have to transform our random variable S_N to have mean=0 and variance=1:

$$\Pr\left[-1.65 \leq \frac{S_N - p'}{\sqrt{\frac{p'q'}{N}}} \leq 1.65\right] = 90\%$$

Estimating p



$\Pr[X \geq z]$	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

Let N=100, and 70 successes

p' =.7 q'=.3

sqrt(.7*.3/100) = .046

Two equations to solve:

$(S_N - 0.7) / .046 = -1.65$

$S_N = .7 - 1.65*.046 = .624$

$(S_N - 0.7) / .046 = 1.65$

$S_N = .7 + 1.65*.046 = .776$

Thus, we say:

With a 90% confidence we have

$$\mathbf{0.624 \leq S_N \leq 0.776}$$

***p*, the true success rate of the classifier, being the mean of S_N, will also be**

$$\mathbf{0.624 \leq p \leq 0.776}$$

Summary

- Suppose I want to be C% confident in my estimation.
  - Looking at a table we find *z* such that: $\Pr[-z \leq X \leq z] \approx C\%$
  - E.g. for C=90, we got z=1.65

- Solving equations we get: $\dfrac{S}{N} = p' \pm z \cdot \sqrt{\dfrac{p'q'}{N}}$

- We say: With confidence C%, we have

$$p \in \left[p' - z \cdot \sqrt{\frac{p'q'}{N}},\ p' + z \cdot \sqrt{\frac{p'q'}{N}} \right]$$

Cross-validation

- **First step**: split data into *k* subsets of equal size
- **Second step**: use each subset in turn for testing, the remainder for training
- Called *k-fold cross-validation*
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

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

Leave-One-Out cross-validation

- Leave-One-Out:
  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 subsampling
- But, computationally expensive

Leave-One-Out-CV and stratification

- 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: completely random dataset split equally into two classes
  - Best classifier predicts majority class
  - 50% accuracy on fresh data
  - Leave-One-Out-CV estimate is 100% error!