UNION COLLEGE

CSC321 Data Mining & Machine Learning

Prof. Nick Webb
webbn@union.edu

Credibility: Evaluating what's been learned

- Issues: training, testing, tuning
- Predicting performance: confidence limits
- Holdout, cross-validation
- Comparing schemes: the t-test
- Predicting probabilities: loss functions
- Cost-sensitive measures
- Evaluating numeric prediction
- The Minimum Description Length principle

Evaluation: the key to success

- How predictive is the model we learned?

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

- Simple solution, 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 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 from 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 can't 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!

Holdout estimation

- What to do if the amount of data is limited?
- The *holdout* method reserves a certain amount for testing and uses the remainder for training
  - 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
- Advanced version uses *stratification*
  - Ensures 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 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

- *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
- 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

More on 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 still further)

UNION
COLLEGE

Judging Performance

- Using cross-validation can help us address
  - Variance in data
  - Overfitting

- But what else do we have to think about?
  - How much does the amount of data impact our understanding of the resulting scores?

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

UNION COLLEGE

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 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]

UNION COLLEGE

Mean, variance, standard deviation

- Mean
  - Simple average of all the values
- Variance
  - The average of the squared differences of the mean
- Standard deviation
  - The squared root of the variance

Confidence intervals

- Solving for *p* :

$$p=\left(f+\frac{z^2}{2N}\mp z\sqrt{\frac{f}{N}-\frac{f^2}{N}+\frac{z^2}{4N^2}}\right)/\left(1+\frac{z^2}{N}\right)$$

- Where:
  - F: frequency of successful event
  - N: number of trials
  - C: Confidence level
  - Z: found from corresponding table

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

Examples

- f = 75%, N = 1000, c = 80% (so that z = 1.28):

$$p \in [0.732, 0.767]$$

- f = 75%, N = 100, c = 80% (so that z = 1.28):

$$p \in [0.691, 0.801]$$

- Note that normal distribution assumption is only valid for large N (i.e. N > 100)
- f = 75%, N = 10, c = 80% (so that z = 1.28):

$$p \in [0.549, 0.881]$$

(should be taken with a grain of salt)

UNION COLLEGE

Comparing schemes

- Frequent question: which of two learning schemes performs better?
- Note: this is domain dependent!
- Obvious way: compare 10-fold CV estimates
- Generally sufficient in applications (we don't 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

Comparing schemes II

- Want to show that scheme A is better than scheme B in a particular domain
  - For a given amount of training data
  - On average, across all possible training sets
- Let's assume we have an infinite amount of data from the domain:
  - Sample infinitely many dataset of specified size
  - Obtain cross-validation estimate on each dataset for each scheme
  - Check if mean accuracy for scheme A is better than mean accuracy for scheme B

Paired t-test

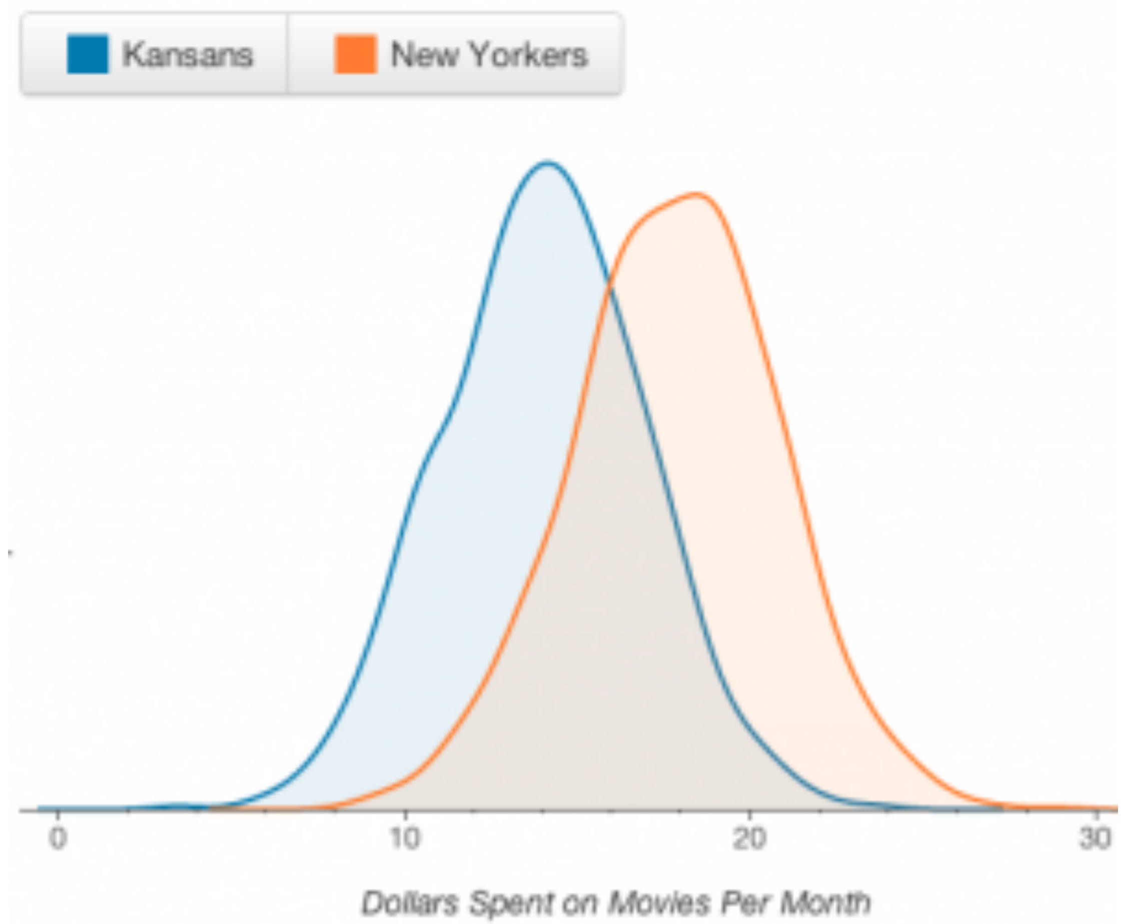
- In practice we have limited data and a limited number of estimates for computing the mean
- *Student's t-test* tells whether the means of two samples are significantly different
- In our case the samples are cross-validation estimates for different datasets from the domain
- Use a *paired* t-test because the individual samples are paired
  - The same CV is applied twice

William Gosset

Born: 1876 in Canterbury; Died: 1937 in Beaconsfield, England

Obtained a post as a chemist in the Guinness brewery in Dublin in 1899. Invented the t-test to handle small samples for quality control in brewing. Wrote under the name "Student".

Is a difference in means REAL?



Calculate a value t

- Approximately

$$t = \frac{mean(1) - mean(2)}{s / \sqrt{n}}$$

- So what does this mean?

Calculating t

- mean(1) – mean(2)
  - Gives you the size of the difference you're trying to measure
  - The strength of a signal
  - Larger difference, stronger signal

Calculating t

- s / sqrt(n)
  - s is the standard deviation
    - How spread out the data is
  - sqrt(n) is the size of your sample size
  - Together they give you a sense of the surrounding noise
  - Louder noise, the stronger a signal you need to hear it

Calculating t

- t is the ratio of signal to noise
- If the signal is weak relative to the noise, you'll get a smaller t
- If you have a small t, then there are three possible causes:
  - The difference between the means isn't large enough
  - The variation in the data is too large
  - The sample is too small

Performing the test

- Fix a significance level
  - If a difference is significant at the α% level, there is a (100-α)% chance that the true means differ
- Divide the significance level by two because the test is two-tailed
  - I.e. the true difference can be +ve or – ve
- Look up the value for z that corresponds to $\alpha/2$
- If $t \leq -z$ or $t \geq z$ then the difference is significant
  - I.e. the *null hypothesis* (that the difference is zero) can be rejected

UNION COLLEGE

Example t test

- Perform cross-validation experiments
- Collect means of results
- Calculate value of t
- Compare to value of z (confidence level)
  - Typically 0.05 (or 95% confidence level)
- Report results

UNION COLLEGE

Reporting significance

- If results are significant
  - "our results showed statistical significance ($p<0.05$)"

- If results ARE NOT significant
  - "Our study did not show statistically significant results ($p<0.05$)"

What we DON'T say

- Let's say we run the test, and get a score of 0.059
- With a chosen confidence level of 0.05
- This result is NOT statistically significant

- We do NOT say:
  - Almost significant
  - Equally we never say VERY significant

Challenges with the test

- The difference between the means isn't large enough
  - Not a lot you can do to improve this
- The variation in the data is too large
  - Can see if there are genuine outliers that SHOULD be removed
  - But be careful about p-hacking
- The sample is too small
  - Ah. Bit of an issue
  - With enough data, ANYTHING can become significant

Unpaired observations

- If the CV estimates are from different datasets, they are no longer paired (or maybe we have *k* estimates for one scheme, and *j* estimates for the other one)
- Then we have to use an *un*paired t-test
- The estimate of the variance of the difference of the means is slightly different

Evaluating numeric prediction

- Same strategies: independent test set, cross-validation, significance tests, etc.
- Difference: error measures
- Actual target values: $a_1\ a_2 \ldots a_n$
- Predicted target values: $p_1\ p_2 \ldots\ p_n$
- Most popular measure: *mean-squared error*

$$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$$

  - Easy to manipulate mathematically

Other measures

- The *root mean-squared error* :

$$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$$

- The *mean absolute error* is less sensitive to outliers than the mean-squared error:

$$\frac{|p_1-a_1|+\ldots+|p_n-a_n|}{n}$$

Improvement on the mean

- How much does the scheme improve on simply predicting the average?

- The *relative squared error* is:

$$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(\bar{a}-a_1)^2+\ldots+(\bar{a}-a_n)^2}$$

- The *relative absolute error* is:

$$\frac{|p_1-a_1|+\ldots+|p_n-a_n|}{|\bar{a}-a_1|+\ldots+|\bar{a}-a_n|}$$

UNION COLLEGE

Correlation coefficient

- Measures the *statistical correlation* between the predicted values and the actual values

- Scale independent, between –1 and +1

- Good performance leads to large values!

- But be careful: Correlation does NOT mean causation

Which measure?

- Best to look at all of them
- Often DOESN'T MATTER

	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 rel 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

The MDL principle

- MDL stands for *minimum description length*
- The description length is defined as:

space required to describe a theory

+

space required to describe the theory's mistakes

- In our case the theory is the classifier and the mistakes are the errors on the training data
- Aim: we seek a classifier with minimal DL
- MDL principle is a *model selection criterion*

Model selection criteria

- Model selection criteria attempt to find a good compromise between:
  - The complexity of a model
  - Its prediction accuracy on the training data
- Reasoning: a good model is a simple model that achieves high accuracy on the given data
- Also known as *Occam's Razor* : the best theory is the smallest one that describes all the facts

William of Ockham, born in the village of Ockham in Surrey (England) about 1285, was the most influential philosopher of the 14th century and a controversial theologian.

Elegance vs. errors

- Theory 1: very simple, elegant theory that explains the data almost perfectly
- Theory 2: significantly more complex theory that reproduces the data without mistakes
- Theory 1 is probably preferable
- Classical example: Kepler's three laws on planetary motion
  - Less accurate than Copernicus's latest refinement of the Ptolemaic theory of epicycles

MDL and compression

- MDL principle relates to data compression:
  - The best theory is the one that compresses the data the most
  - I.e. to compress a dataset we generate a model and then store the model and its mistakes
- We need to compute
  (a) size of the model, and
  (b) space needed to encode the errors
- (b) easy: use the informational loss function
- (a) need a method to encode the model