Significance Testing for Classification

Benjamin Evans

Plan

1. Discuss significance testing
2. How to compare 2 methods on a dataset
3. How to compare 2 methods across multiple datasets
4. How to compare multiple methods across multiple datasets
    a. Compare to a control
    b. Compare all methods

What is the goal?

Report the classifier which performs best on some unseen data

Determine best classifier amongst a set of classifiers

Rank classifiers in terms of general performance

Terminology

Type-I Errors:

- Incorrectly determing a difference when no difference exists
- Rejection of a true null hypothesis
- False positive

Type-II Errors

- Saying there is no difference when a difference exists
- Non-rejection of a false null hypothesis
- False Negative

Impossible to completely eliminate either, so we try and minimize one (or both)

Sources of Variation in experiments

- Random variation from selection of test data
- Random variation from selection of training data
- Randomness from the learning algorithm
- Random classification error (mislabelled points)

A good statistical test should not be fooled by these sources of variation

How to compare 2 methods on a dataset

Problematic Tests (Elevated Type-I Errors)

Likely to determine a difference when no difference exists

- Test for difference of two proportions
- Paired-difference T-test based on several random train/test splits
- Paired-difference T-test based on 10-fold CV

Paired-difference T-test based on several random train/test splits

1. Split data randomly into training and test sets
2. Train each method on the training test, record performance on test set
3. Repeat step 1 until we have done 30 runs
4. Perform a students t-test on the results

Paired-difference T-test based on several random train/test splits

1. Split data randomly into training and test sets
2. Train each method on the training test, record performance on test set
3. Repeat step 1 until we have done 30 runs
4. Perform a students t-test on the results

Drawbacks

Differences (p_a-p_b) not normally distributed

Results are not independent, because theres overlapping data in test sets

Also overlap in the training data too

Never use

Paired-difference T-test based on 10-fold CV

1. Split data into 10 non overlapping "folds" (F)
2. For each fold f
    2. Train each method on (F-f), record performance on f
4. Perform a students t-test on the results

Paired-difference T-test based on 10-fold CV

1. Split data into 10 non overlapping "folds" (F)
2. For each fold f
    2. Train each method on (F-f), record performance on f
4. Perform a students t-test on the results

Drawbacks

Differences (p_a-p_b) not normally distributed

Results are not independent, because there's overlap n the training data

Use caution when interpreting results (elevated Type-I error, but nowhere near as elevated as previous

Improved Tests

- 5x2cv Paired t test
- McNemars test (used if we can only run a method once)

5x2cv Paired t-test

1. Repeat 5 times (with different seed)

    2. Split data randomly into 2 non overlapping "folds" (F)
    3. For each fold f
        - Train each method on (F-f), record performance on f

4. Perform a 5x2 t-test on the results, which has corrected numerator in t value to calculations to account for overly large T values in previous tests

http://rasbt.github.io/mlxtend/user_guide/evaluate/paired_ttest_5x2cv/

5x2cv Paired t-test

1. Repeat 5 times (with different seed)
    2. Split data randomly into 2 non overlapping "folds" (F)
    3. For each fold f
        - Train each method on (F-f), record performance on f

4. Perform a 5x2 t-test on the results, which has corrected numerator in t value to calculations to account for overly large T values in previous tests

http://rasbt.github.io/mlxtend/user_guide/evaluate/paired_ttest_5x2cv/
Results are "more" independent than previously seen, because there's no overlap in the training sets (or test sets) since 2 fold

The repetition (5x) helps to stabilise the estimates (higher repetitions, lack of independence can become a problem, lower reptitions the noise in the folds can become a problem)

5x2cv Paired f-test

Note: Alpaydin proposed an improvement to the 5x2 t-test, which should generally be favoured as a more robust extension

http://rasbt.github.io/mlxtend/user_guide/evaluate/combined_ftest_5x2cv/

How to compare 2 methods on a dataset

- Test for difference of two proportions
- Paired-difference T-test based on several random train/test splits
- Paired-difference T-test based on 10-fold CV
- 5x2cv Paired t test (and improved f test)
- McNemars test (used if we can only run a method once)

How to compare 2 methods across multiple datasets

How to compare 2 methods across multiple datasets

When we have several datasets, we should **not** just repeat the steps for comparing on a single dataset!

How to compare 2 methods across multiple datasets

When we have several datasets, we should **not** just repeat the steps for comparing on a single dataset!

Each test is approximate. We specify an alpha (saying 0.05), which is represents our "acceptable" probability of a Type I error for a given test.

If we perform N tests, probability one of those gives us a significant result by chance (i.e. a type 1 error) isn't 5%, its 1−(1−0.05)^N

How to compare 2 methods across multiple datasets

When we have several datasets, we should **not** just repeat the steps for comparing on a single dataset!

Each test is approximate. We specify an alpha (saying 0.05), which is represents our "acceptable" probability of a Type I error for a given test.

If we perform N tests, probability one of those gives us a significant result bu chance (i.e. a type 1 error) isnt 5%, its 1−(1−0.05)^N

N=10, 40% chance
N=20, 65% chance
N=30, 78% chance

How to compare 2 methods across multiple datasets

Instead we should either:

- Adjust our alpha (Bonferroni correction) based on the number of tests (can be overly conservative)

- Use a Wilcoxon signed rank test across datasets

Bonferroni Correction

Simply divide the alpha rate (i.e. 0.05) by the number of tests being performed. For example, 10 datasets, alpha = 0.05/10.

Now to be significant, p_value < 0.005

Note: Can be overly conservative. Consider if we had a method doing better on all datasets, but it was right above the cut off alpha. It's unlikely this just does better by chance on every dataset.

Wilcoxon Signed Rank Test

Non-parametric alternative to t-test - Fewer assumptions!

It assumes commensurability of differences, but only qualitatively: greater differences still count more, which is probably desired, but the absolute magnitudes are ignored.

Doesn't assume normal distribution

Outliers have less effect compared to t-test

Wilcoxon Test Across Datasets

Rather than performing across folds on a dataset, should perform across datasets.

	Method 1	Method 2	Significant
Dataset 1	0.99	0.98	No
Dataset 2	0.77	0.65	Yes (+)
Dataset 3	0.85	0.87	Yes (-)

	Method 1	Method 2
Dataset 1	0.99	0.98
Dataset 2	0.77	0.65
Dataset 3	0.85	0.87
Significant	No	

Wilcoxon Test Across Datasets

Rather than performing across folds on a dataset, should perform across datasets.

	Method 1	Method 2	Significant
Dataset 1	0.99	0.98	No
Dataset 2	0.77	0.65	Yes (+)
Dataset 3	0.85	0.87	Yes (-)

	Method 1	Method 2
Dataset 1	0.99	0.98
Dataset 2	0.77	0.65
Dataset 3	0.85	0.87
Significant	No	

Note: our only assumption for the averages presented is they are "reliable". Should be generated by some form of cross validation with matched samples between methods

How do we compare multiple classifiers across multiple datasets?

Problematic Approaches

- Ranking based on average across all datasets
- Repeating the pairwise Wilcoxon test we did before
- Counting number of wins/losses
- ANOVA

Problematic Approaches

- Ranking based on average across all datasets (Not commensurable between datasets and influenced by outliers)

- Repeating the pairwise Wilcoxon test we did before (Inflated errors)

- Counting number of wins/losses (weaker than the Wilcoxon signed-ranks test, so less likely to find a difference when one exists)

- Counting number of significant wins/losses (Even less reliable than above, since introduce an arbitrary cut off at alpha)

- ANOVA (Assumes normal distributions* and equal variance)

Friedman Testing

Non parametric alternative to ANOVA

Overcomes all the previously mentioned issue

Ranks each algorithm for each dataset, then computes the average rank.

Tells us if the average ranks are equal (null hypothesis expects them to be equal)

Friedman Testing

Non parametric alternative to ANOVA

Overcomes all the previously mentioned issue

Ranks each algorithm for each dataset, then computes the average rank.

Tells us if the average ranks are equal (null hypothesis expects them to be equal)

Friedman Testing + Post Hoc Analysis

If the null-hypothesis is rejected (ranks are not equal), we can proceed with a post-hoc test.

Post Hoc Analysis

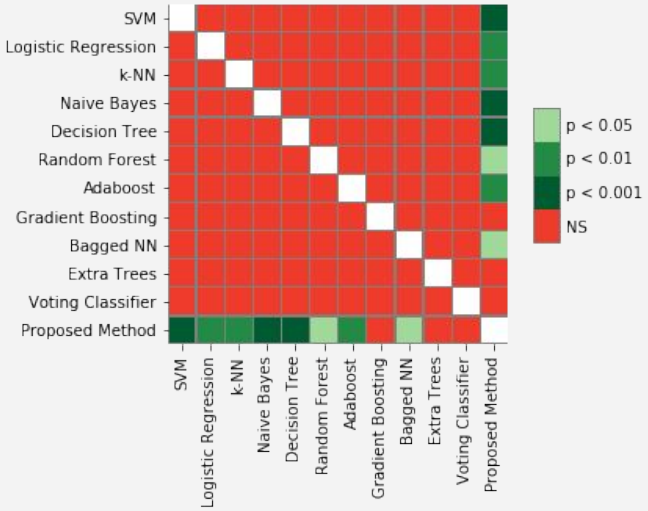
Nemenyi and Holm Test

Nemenyi: When comparing all algorithms to **each other** (i.e. in a survey paper)

General FWER corrections: Bonferonni-Dunn, Holm. When comparing to a **control** classifier (i.e. we propose a single method, compare to others)

Post Hoc Analysis: Nemenyi

The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference



Result is all pairwise comparisons

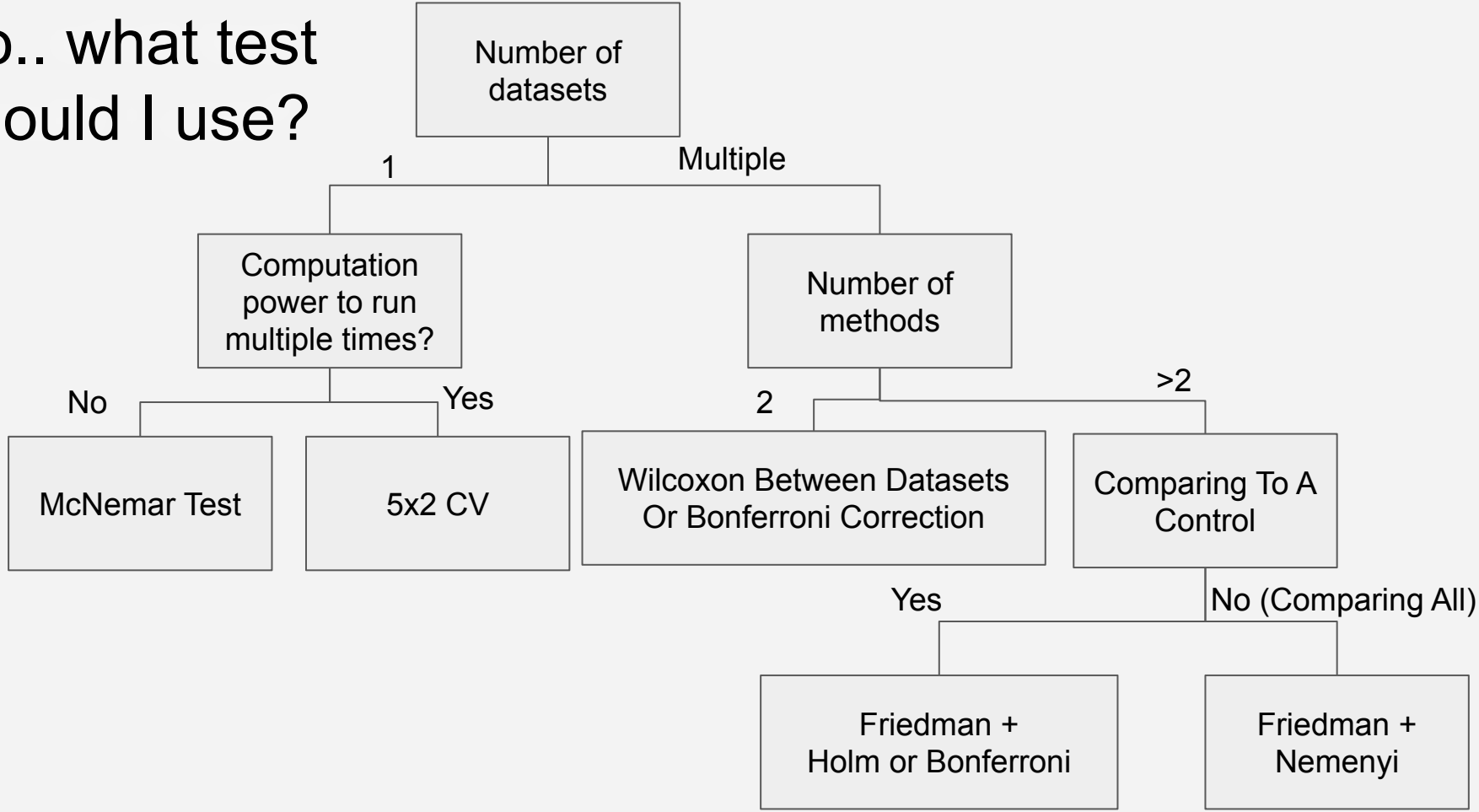
Post Hoc Analysis: Holm/Bonferroni

The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference

	p	sig
Proposed Method vs Decision Tree	0.000002	True
Proposed Method vs SVM	0.000003	True
Proposed Method vs Naive Bayes	0.000083	True
Proposed Method vs k-NN	0.000150	True
Proposed Method vs Adaboost	0.001084	True
Proposed Method vs Logistic Regression	0.001084	True
Proposed Method vs Random Forest	0.001609	True
Proposed Method vs Bagged NN	0.001609	True
Proposed Method vs Extra Trees	0.027584	True
Proposed Method vs Gradient Boosting	0.047193	True
Proposed Method vs Voting Classifier	0.047194	True

Only compares one method (control) to the others. Not all pairwise comparisons!

So.. what test should I use?



So.. what test should I use? (Alternative perspective)

	Method 2 Correct	Method 2 Incorrect
Method 1 Correct	10	2
Method 1 Incorrect	9	8

McNemar Test

Method 1	Method 2
0.98	0.97

5x2 CV Test

	Method 1	Method 2
Dataset 1	0.98	0.97
Dataset 2	0.9	0.86
Dataset 3	0.5	0.54
….		

Wilcoxon Signed Rank (Cols not rows) or Bonferonni Correction

	Method 1	Method 2	Method 3	
Dataset 1	0.98	0.97	0.95	.....
Dataset 2	0.9	0.86	0.81	
Dataset 3	0.5	0.54	0.6	
….				

Friedman + Nemenyi

Friedman + Holm or Bonferroni

Summary

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, *10*(7), pp.1895-1923.

Alpaydm, E., 1999. Combined 5× 2 cv F test for comparing supervised classification learning algorithms. *Neural computation*, *11*(8), pp.1885-1892.

Demšar, J., 2006. **Statistical comparisons of classifiers over multiple data sets.** ***Journal of Machine learning research*****,** *7*(Jan), pp.1-30.

But what about the 30 runs "requirement"?

This is for normality, not needed with the mentioned non parametric tests! But to get more "robust" answers, we can still run our EC techniques several times. And if they're fast to run, it's good to do so (but its not *required* for computationally intensive experiments).

For the single datasets: The variability is already captured with the repeated runs (either in 10-fold CV or 5x2CV). For each of these 10 values, we could run our algorithm several times to improve the robustness though (particularly for EC)

For multiple datasets: The tests make no assumptions about the variance. The only assumption for the averages presented is they are "reliable". To improve reliability, we could do say 10x10 CV (with higher repetition if not too costly)

Python Resources

Mlxtend: http://rasbt.github.io/mlxtend/

Stac: https://github.com/citiususc/stac

Scikit-posthocs: https://github.com/maximtrp/scikit-posthocs

Self promotion: https://github.com/ben-ix/MethodComparisonsInPython
(Just summarises the links above with examples on Friedman tests, also has the downloadable slides).