Performance Evaluation for Learning Algorithms

Nathalie Japkowicz

School of Electrical Engineering
& Computer Science
University of Ottawa

nat@site.uottawa.ca

Motivation: My story

- A student and I designed a new algorithm for data that had been provided to us by the National Institute of Health (NIH).
- According to the standard evaluation practices in machine learning, we found our results to be significantly better than the state-of-the-art.
- The machine learning community agreed as we won a best paper award at ISMIS'2008 for this work.
- NIH disagreed and would not consider our algorithm because it was probably not truly better than the others.

Motivation: My story (cont'd)

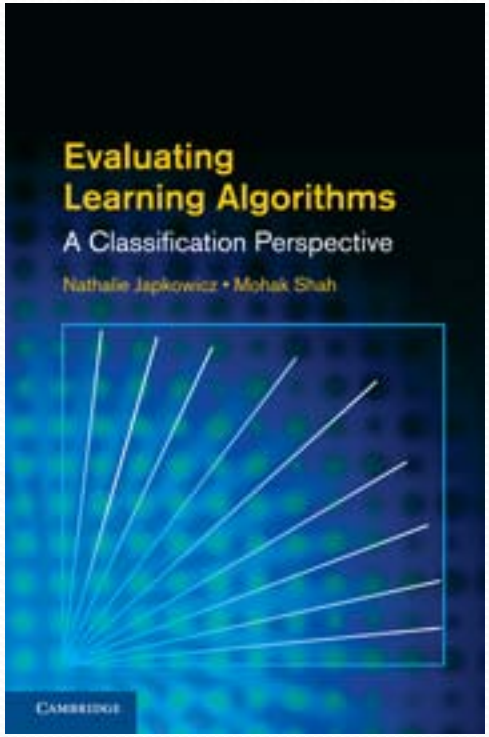
- My reactions were:
  - **Surprise:** Since my student and I properly applied the evaluation methodology that we had been taught and read about everywhere, how could our results be challenged?
  - **Embarrassment:** There is obviously much more to evaluation than what I have been told. How can I call myself a scientist and not know what the scientists of other fields know so well?
  - **Determination:** I needed to find out more about this and share it with my colleagues and students.

Information about this tutorial

- This tutorial is based on the book I have co-written after going through the experience I just described.
- It will give you an overview of the complexity and uncertainty of evaluation.
- It will also give you a brief overview of the issues that come up in different aspects of evaluation and what the possible remedies may be.
- Finally, it will direct you to some resources available that can help you perform more robust evaluations of your systems.

Book Details

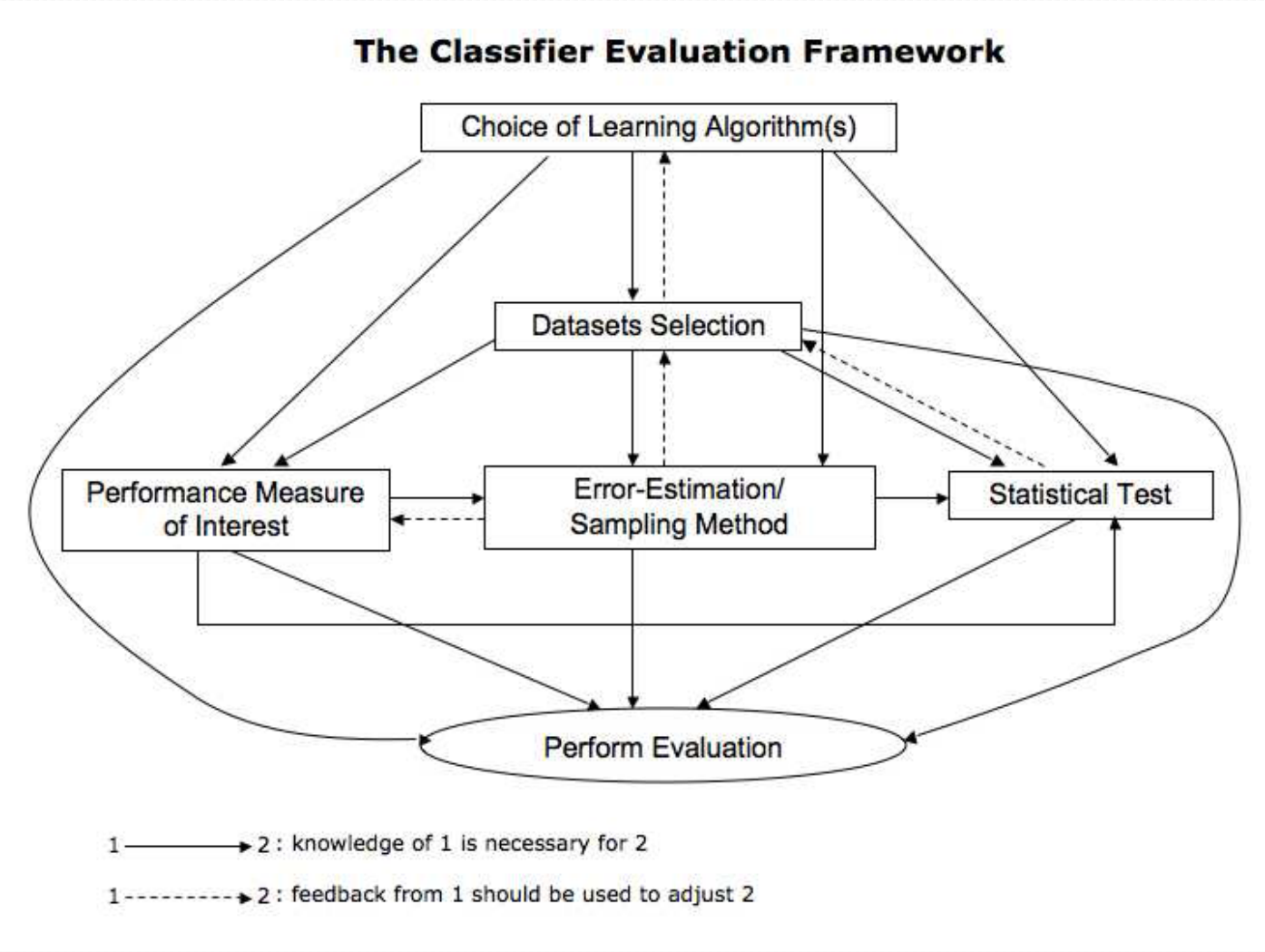
Evaluating Learning Algorithms:
A Classification Perspective
Nathalie Japkowicz & Mohak Shah
Cambridge University Press, 2011



- Review:

  "This treasure-trove of a book covers the important topic of performance evaluation of machine learning algorithms in a very comprehensive and lucid fashion. As Japkowicz and Shah point out, performance evaluation is too often a formulaic affair in machine learning, with scant appreciation of the appropriateness of the evaluation methods used or the interpretation of the results obtained. This book makes significant steps in rectifying this situation by providing a reasoned catalogue of evaluation measures and methods, written specifically for a machine learning audience and accompanied by concrete machine learning examples and implementations in R. This is truly a book to be savoured by machine learning professionals, and required reading for Ph.D students."
  Peter A. Flach, University of Bristol

The main steps of evaluation



What these steps depend on

- These steps depend on the purpose of the evaluation:
  - Comparison of a *new algorithm* to other (may be generic or application-specific) classifiers on a *specific domain* (e.g., when proposing a novel learning algorithm)
  - Comparison of a *new generic algorithm* to other generic ones on a set of *benchmark domains* (e.g. to demonstrate general effectiveness of the new approach against other approaches)
  - Characterization of *generic classifiers* on *benchmarks domains* (e.g. to study the algorithms' behavior on general domains for subsequent use)
  - Comparison of *multiple classifiers* on a *specific domain* (e.g. to find the best algorithm for a given application task)

Outline of the tutorial:

- **Part I** (from now until the coffee break!)
  - Choosing a performance measure.
  - Choosing a statistical test.
- **Part II** (from after the coffee break to the lunch break!)
  - What about sampling?
  - What data sets should we use?
  - Available resources
- **Part III** (if there is time before lunch and we're not too hungry!)
  - Recent research

Part I

Topic 1: Choosing a Performance Measure

Which Classifier is better?

Almost as many answers as there are performance measures! (e.g., UCI Breast Cancer)

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripp	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanF	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

Which Classifier is better?

Ranking the results

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	3	5	1	7	3	1	1	1	2
C4.5	1	1	7	1	1	7	5	7	5
3NN	2	7	6	2	2	6	4	3	3
Ripp	4	3	3	4	4	3	3	6	6
SVM	6	8	4	5	5	4	6	7	1
Bagg	8	4	8	2	8	8	8	3	8
Boost	5	2	2	8	7	2	2	1	4
RanF	7	6	4	5	5	4	7	3	7

What should we make of that?

- Well, for certain pairs of measures, that makes sense, since each measure focuses on a different aspect of learning.
  - For example, the TPR and the FPR are quite different, and often, good results on one yields bad results on the other.
  - Precision and Recall also seem to tradeoff each other.
- How about the global measures (Acc, RMSE, the F-measure, AUC, the Information Score)?
  - They too disagree as they each measure different (though more difficult to pinpoint as they are composite measures) aspects of learning.

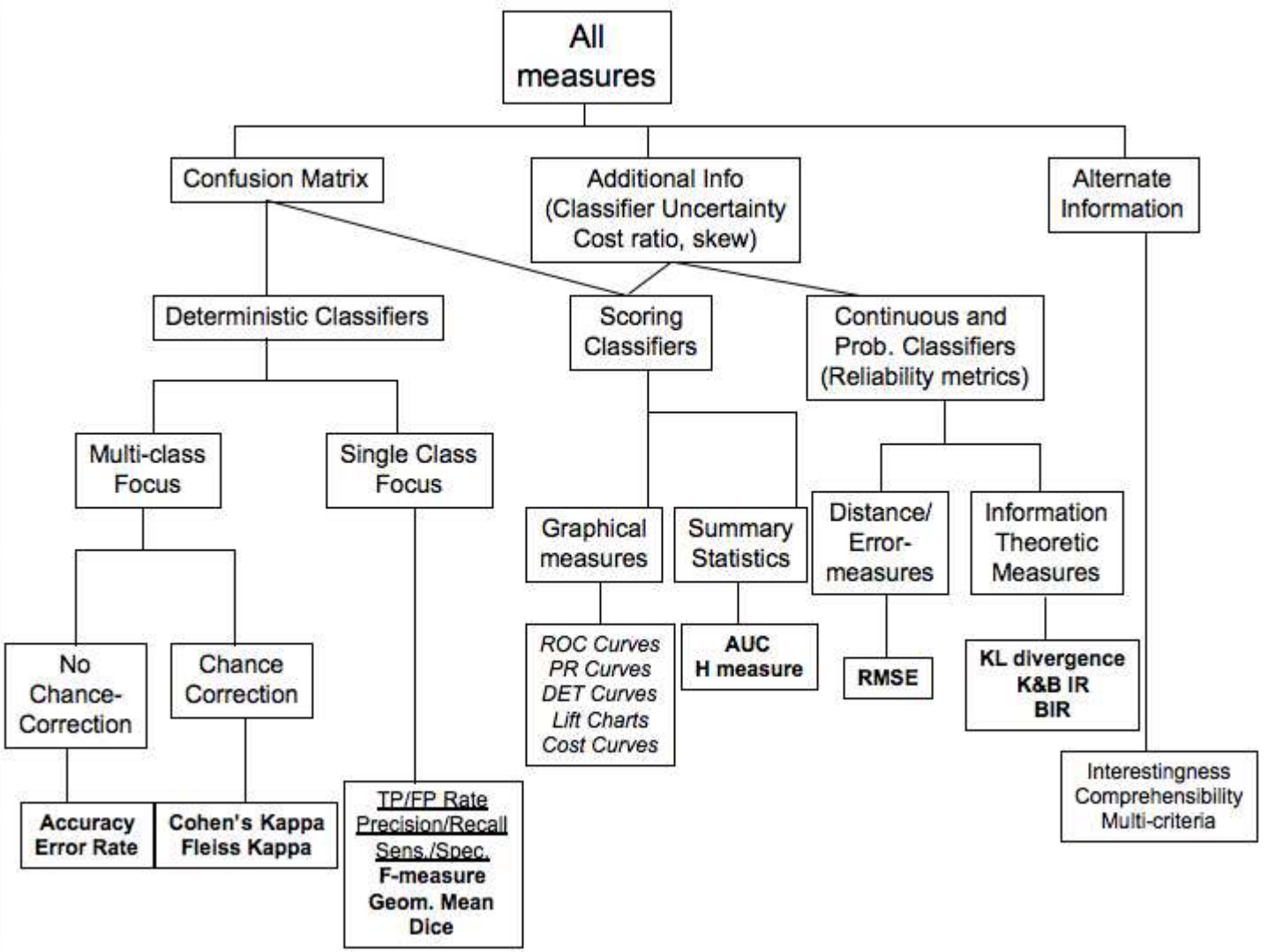
Is this a problem?

- It is not a problem when working on a specific application since the purposes of the application are clear and the developer knows which performance measure is best to optimize in that context.
- It does become a problem, however, when the general usefulness of a new algorithm is assessed. In that case, what performance measure should be chosen?
  - If only one or a few measures are chosen, then it can be argued that the analysis is incomplete, and misleading.
  - If many measures are chosen, the results become too mitigated for any clear statement to be issued.

What to do, then?

- One suggestion which tries to balance pragmatics (getting the paper accepted!) with fairness (acknowledging the weakness of the new algorithm!) is to divide the evaluation procedure into two parts:
  - In the first part, choose a few important metrics on which the new method excels in order to demonstrate this new method's worth.
  - In a second part, overview the results that the new method obtains on a large selection of performance measures.
  - Try to explain these observations; do not be too strict (i.e., if the new method consistently ranks as one of the best three methods tested, then it is not that bad. Similarly, if it ranks very badly on only one performance measure, then it is not that bad either).

Overview of Performance Measures



A Few Confusion Matrix-Based Performance Measures

True class → Hypothesized | class V	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

A Confusion Matrix

- Accuracy = (TP+TN)/(P+N)
- Precision = TP/(TP+FP)
- Recall/TP rate = TP/P
- FP Rate = FP/N
- ROC Analysis moves the threshold between the positive and negative class from a small FP rate to a large one. It plots the value of the Recall against that of the FP Rate at each FP Rate considered.

Issues with Accuracy

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Both classifiers obtain 60% accuracy
- They exhibit very different behaviours:
  - On the left: weak positive recognition rate/strong negative recognition rate
  - On the right: strong positive recognition rate/weak negative recognition rate

Issues with Precision/Recall

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	200	100
No	300	0
	P=500	N=100

- Both classifiers obtain the same precision and recall values of 66.7% and 40% (Note: the data sets are different)
- They exhibit very different behaviours:
  - Same positive recognition rate
  - Extremely different negative recognition rate: strong on the left / nil on the right
- Note: Accuracy has no problem catching this!

Is the AUC the answer?

- Many researchers have now adopted the AUC (the area under the ROC Curve).
- The principal advantage of the AUC is that it is more robust than Accuracy in class imbalanced situations.
- Indeed, given a 95% imbalance (in favour of the negative class, say), the accuracy of the default classifier that issues "negative" all the time will be 95%, whereas a more interesting classifier that actually deals with the issue, is likely to obtain a worse score.
- The AUC takes the class distribution into consideration.

Is the AUC the Answer? (cont')

- While the AUC has been generally adopted as a replacement for accuracy, it met with a couple of criticisms:
  - The ROC curves on which the AUCs of different classifiers are based may cross, thus not giving an accurate picture of what is really happening.
  - The misclassification cost distributions (and hence the skew-ratio distributions) used by the AUC are different for different classifiers. Therefore, we may be comparing apples and oranges as the AUC may give more weight to misclassifying a point by classifier A than it does by classifier B (Hand, 2009) ➔Answer: the H-Measure, but it has been criticized too!

Some other measures that will be discussed in this tutorial:

- Deterministic Classifiers:
  - Chance Correction: Cohen's Kappa
- Scoring Classifiers:
  - Graphical Measures: Cost Curves (Drummond & Holte, 2006)
- Probabilistic Classifiers:
  - Distance measure: RMSE
  - Information-theoretic measure: Kononenko and Bratko's Information Score
- Multi-criteria Measures: The Efficiency Method (Nakhaeizadeh & Schnabl, 1998)

Cohen's Kappa Measure

- Agreement Statistics argue that accuracy does not take into account the fact that correct classification could be a result of coincidental concordance between the classifier's output and the label-generation process.
- Cohen's Kappa statistics corrects for this problem. Its formula is:

$$\kappa = (P_o - P_e^C) / (1 - P_e^C) \qquad \text{where}$$

  - P_o represents the probability of overall agreement over the label assignments between the classifier and the true process, and
  - P_e^C represents the chance agreement over the labels and is defined as the sum of the proportion of examples assigned to a class times the proportion of true labels of that class in the data set.

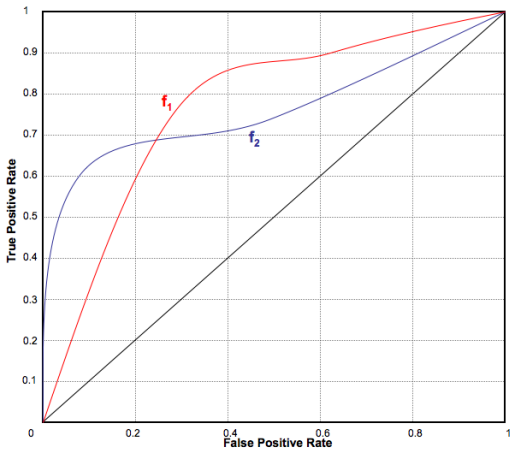
Cohen's Kappa Measure: Example

Predicted -> Actual	A	B	C	Total
A	60	50	10	120
B	10	100	40	150
C	30	10	90	130
Total	100	160	140	

Accuracy = P_0 = (60 + 100 + 90) / 400 = 62.5%
P_e^C = 100/400 * 120/400 + 160/400 * 150/400 + 140/400 * 130/400 = 0.33875
κ = 43.29%

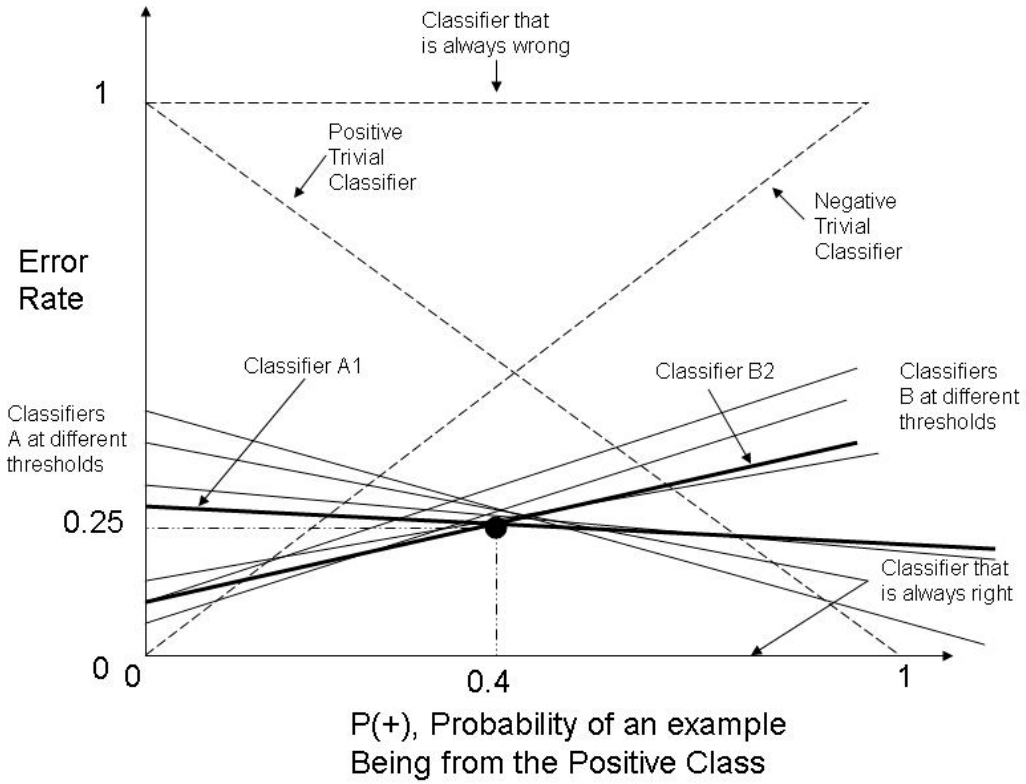
➔ Accuracy is overly optimistic in this example!

Cost Curves



ROC Curves only tell us that *sometimes* one classifier is preferable over the other

Cost-curves are more practical than ROC curves because they tell us *for what class probabilities* one classifier is preferable over the other.



RMSE

- The Root-Mean Squared Error (RMSE) is usually used for regression, but can also be used with probabilistic classifiers. The formula for the RMSE is:

$$RMSE(f) = \text{sqrt}(\ 1/m\ \Sigma_{i=1}^{m}(f(x_i) - y_i)^2))$$

where m is the number of test examples, $f(x_i)$, the classifier's probabilistic output on x_i and y_i the actual label.

ID	$f(x_i)$	y_i	$(f(x_i) - y_i)^2$
1	.95	1	.0025
2	.6	0	.36
3	.8	1	.04
4	.75	0	.5625
5	.9	1	.01

RMSE(f) = sqrt(1/5 * (.0025+.36+.04+.5625+.01))
= sqrt(0.975/5) = 0.4416

Information Score

- Kononenko and Bratko's Information Score assumes a prior P(y) on the labels. This could be estimated by the class distribution of the training data. The output (posterior probability) of a probabilistic classifier is P(y|f), where f is the classifier. I(a) is the indicator fonction.
- $IS(x) = I(P(y|f) \geq P(y)) * (-\log(P(y)) + \log(P(y|f)) + I(P(y|f) < P(y)) * (-\log(1-P(y)) + \log(1-P(y|f)))$
- $IS_{avg} = 1/m \sum_{i=1}^{m}(IS(x_i))$

x	$P(y_i|f)$	y_i	IS(x)
1	.95	1	0.66
2	.6	0	0
3	.8	1	.42
4	.75	0	.32
5	.9	1	.59

$P(y=1) = 3/5 = 0.6$
$P(y=0) = 2/5 = 0.4$
$IS(x_1) = 1 * (-\log(.6) + \log(.95)) + 0 * (-\log(.4) + \log).05))$
$= 0.66$

$IS_{avg} = 1/5 (0.66+0+.42+.32+.59)$
$= 0.40$

Efficiency Method

- The efficiency method is a framework for combining various measures including qualitative metrics, such as interestingness. It considers the positive metrics for which higher values are desirable (e.g, accuracy) and the negative metrics for which lower values are desirable (e.g., computational time).

$$\varepsilon_S(f) = \Sigma_i w_i pm_i^+(f) \; / \; \Sigma_j w_j nm_j^-(f)$$

pm_i^+ are the positive metrics and nm_j^- are the negative metrics. The w_i's and w_j's have to be determined and a solution is proposed that uses linear programming.

Part I

Topic 2: Choosing a Statistical Test

The purpose of Statistical Significance Testing

- The performance metrics just discussed allow us to make observations about different classifiers.
- The question we ask here is: can the observed results be attributed to real characteristics of the classifiers under scrutiny or are they observed by chance?
- The purpose of statistical significance testing is to help us gather evidence of the extent to which the results returned by an evaluation metric are representative of the general behaviour of our classifiers.

Hypothesis Testing

- Hypothesis testing consists of stating a null hypothesis which usually is the opposite of what we wish to test (for example, classifiers A and B perform equivalently)
- We then choose a suitable statistical test and statistic that will be used to reject the null hypothesis.
- We also choose a critical region for the statistic to lie in that is extreme enough for the null hypothesis to be rejected.
- We calculate the observed test statistic from the data and check whether it lies in the critical region. If so, reject the null hypothesis. If not, we fail to reject the null hypothesis, but do not accept it either.
- Rejecting the null hypothesis gives us some confidence in the belief that our observations did not occur merely by chance.

Issues with Hypothesis Testing

- Hypothesis testing never constitutes a proof that our observation is valid. It provides added support for our observations. We can never be 100% sure about them.
- Statistical tests come in two forms: parametric and non parametric. Parametric tests make strong assumptions about the distribution of the underlying data. Non-parametric ones make weaker assumptions about the data, but are also typically less powerful (less apt at rejecting the null hypothesis when it is false) than their parametric counterparts. It is often difficult, if not impossible, to verify that all the assumptions hold.
- The results of statistical tests are often misinterpreted:
  - (1-p) does not represent P(H|D)
  - (1-p) does not represent the probability of replication of successful replication of the observations
- It is always possible to show that a difference between two alternatives, no matter how small, is significant, provided that enough data is used.

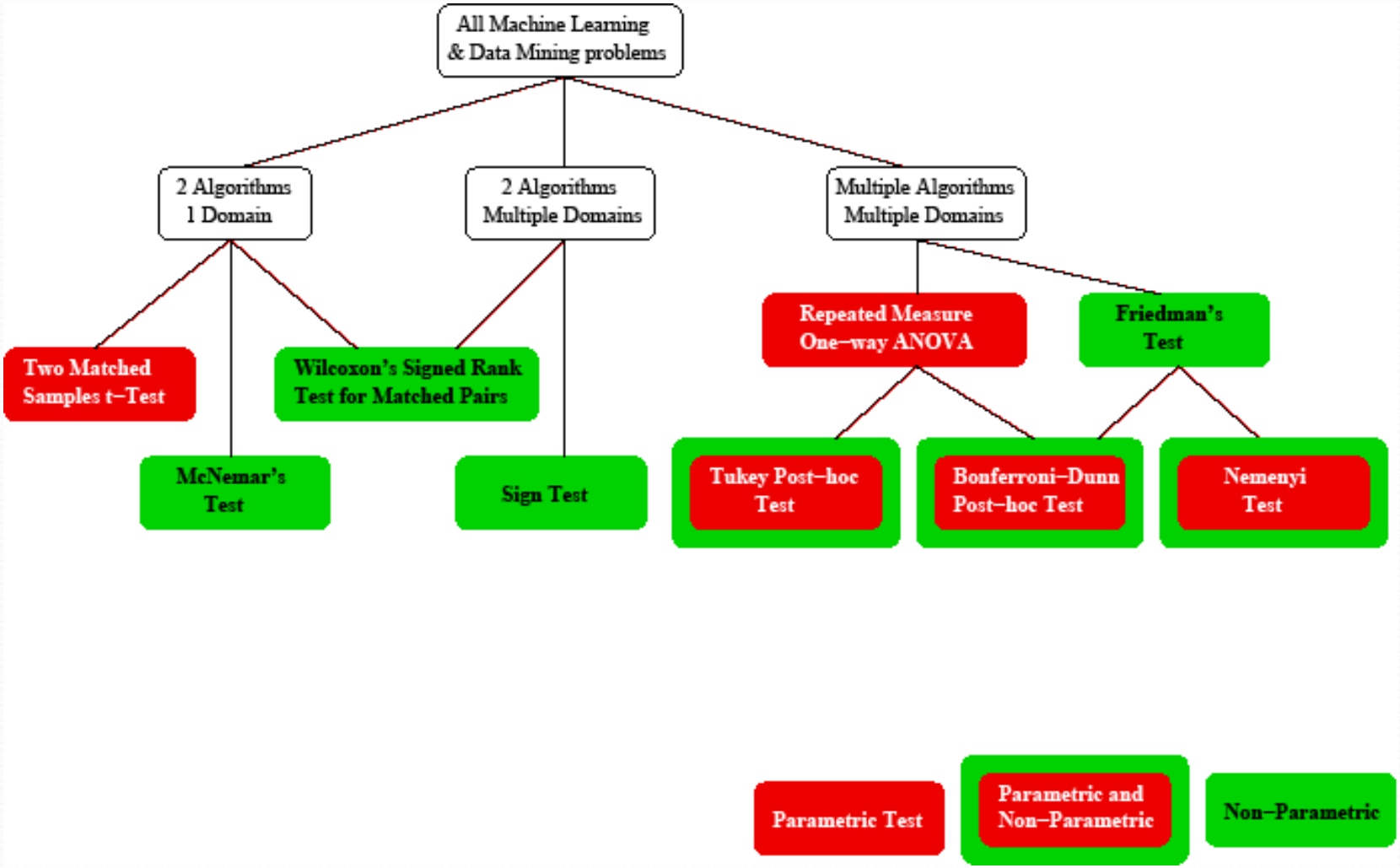
To test or not to test?

- Given the serious issues associated with statistical testing, some researchers (Drummond, 2006, Demsar, 2008) argue that Machine Learning and Data Mining researchers should drop the habit of performing statistical tests. They argue that, this process:
  - Overvalues the results, and
  - Limits the search for novel ideas (because of the excess (not always warranted) confidence in the results).
- The alternative, however, is to train researchers properly in the understanding and application of statistical methods, so that they can decide on their own when a statistical test is warranted, what its limitations are and when the search for new ideas is necessary.

How to choose a statistical test?

- There are several aspects to consider when choosing a statistical test.
  - What kind of problem is being handled?
  - Whether we have enough information about the underlying distributions of the classifiers' results to apply a parametric test?
- Regarding the type of problem, we distinguish between
  - The comparison of 2 algorithms on a single domain
  - The comparison of 2 algorithms on several domains
  - The comparison of multiple algorithms on multiple domains

Statistical tests overview



Statistical Tests we will describe and illustrate in this tutorial

- Two classifiers, one domain:
  - The t-test (parametric)
  - McNemar's test (non-parametric)
  - The Sign Test
- Two classifiers, multiple domains
  - The Sign Test (non-parametric)
  - Wilcoxon's signed-Rank Test (non-parametric)
- Multiple classifiers, multiple domains:
  - Friedman's Test (non-parametric)
  - The Nemenyi Test
- We will also discuss and illustrate the concept of the effect size.

Two classifiers, one domain

- The t-test (parametric)
- McNemar's test (non-parametric)
- The Sign Test (usually used for multiple domains but can also be used on a single domain).

The (2-matched samples) t-test

- Given two matched samples (e.g., the results of two classifiers applied to the same data set with matching randomizations and partitions), we want to test whether the difference in means between these two sets is significant.
- In other words, we want to test whether the two samples come from the same population.
- We look at the difference in observed means and standard deviation.
- We assume that the difference between these means is zero (the null hypothesis) and see if we can reject this hypothesis.

The t-statistic

- $$t = \frac{\bar{d} - 0}{\sigma_{d/\sqrt{n}}}$$

with $\bar{d} = \overline{pm}(f1) - \overline{pm}(f2)$ and $\sigma_d = \sqrt{\frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1}}$

- $\bar{d}$ is the difference of the means of our performance measures obtained when applying classifiers f1 and f2.
- d_i is the difference between the performance measures of classifiers f1 and f2 at trial i. n is the number of trials.
- Labour Example: We compare C4.5 to Naïve Bayes on the Labour data set by testing the two classifiers on the same partitions of 10 runs of 10-fold cross-validation. The result of each 10-fold CV is considered as a single result, so we consider each run of 10-fold CV as a different trial. Consequently, n = 10.
- The results are: $\bar{d} = 0.2175 - 0.0649 = 0.1526$ and $\sigma_d = 0.05969$, and thus $t = 8.0845$

The degree of freedom is n-1 = 9; Using a 2-sided test, the null hypothesis can therefore be rejected at the 0.001 significance level (since the obtained t has to be greater than 4.781for that to be possible).

Assumptions of the t-test

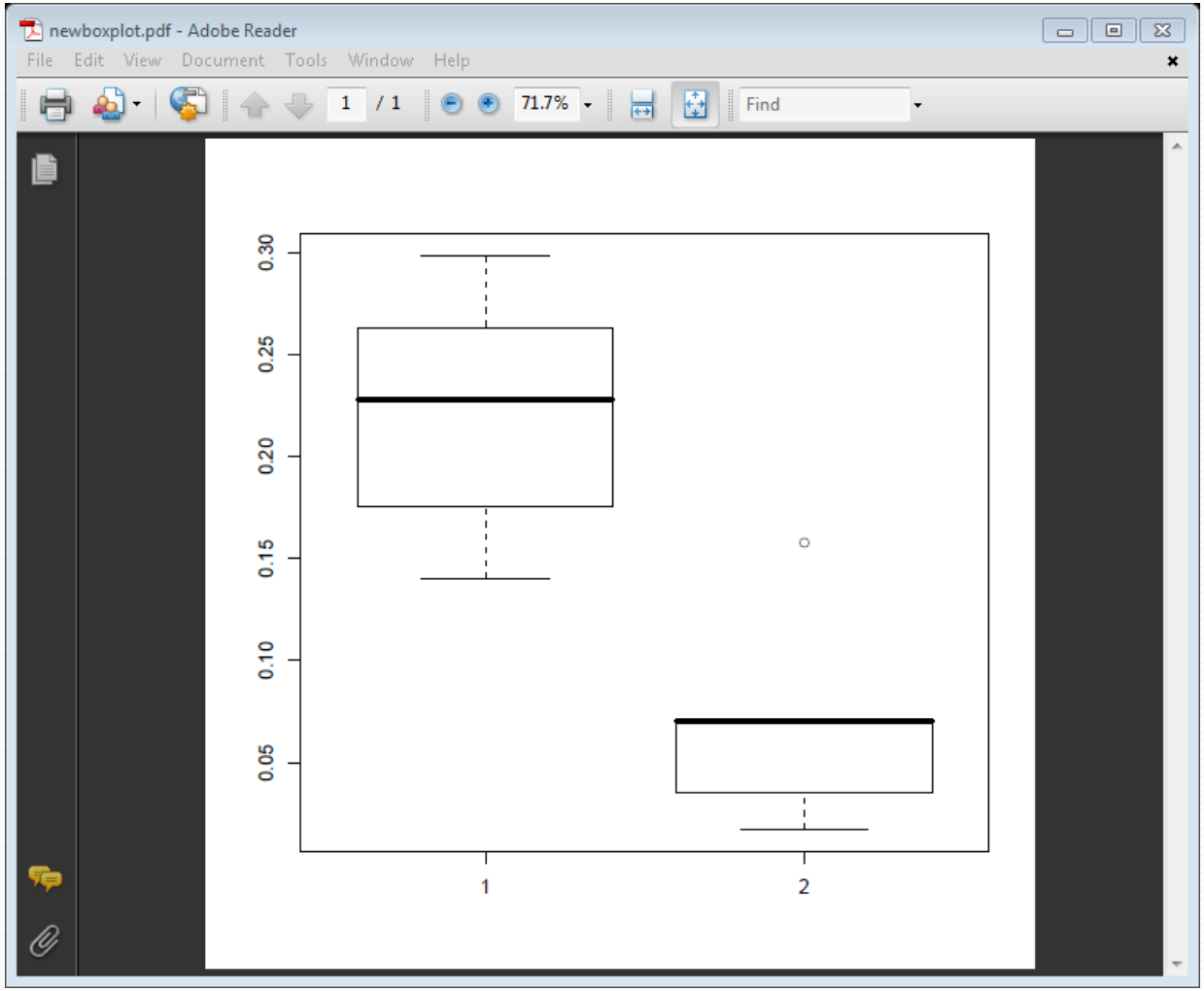
- The Normality or Pseudo-Normality Assumption: The t-test requires that the samples come from normally distributed population. Alternatively, the sample size of the testing set should be greater than 30.
- The Randomness of the Samples: The sample should be representative of the underlying population. Therefore, the instances of the testing set should be randomly chosen from their underlying distribution.
- Equal Variance of the populations: The two sample come from populations with equal variance.
- Labour Example:
  - Normality: The labour data set contains only 57 instances altogether. At each fold of the CV process, only 6 or 7 data points are tested. However, since each trial is a run of 10-fold CV, all 57 examples are tested, so we may be able to assume pseudo-normality.
  - Randomness: The labour data was collected in the 1987-1988 period. All the collective agreements for this period were collected. There is no reason to assume that 1987-1988 was a non-representative year, so we assume that the data was randomly collected.

Assumptions of the t-test (cont'd)

- Equal Variance: The variance of C4.5 (1) and NB (1) cannot be considered equal. (See figure)

➔ We were not warranted to use the t-test to compare C4.5 to NB on the Labour data.

➔A better test to use in this case is the non-parametric alternative to the t-test: McNemar's Test (See Japkowicz & Shah, 2011 for a description)



McNemar's test

- McNemar's test is the non-parametric counterpart of the t-test. It relies on 4 values, observed on the testing set:
  - The number of instances misclassified by both classifiers (c_{00})
  - The number of instances misclassified by f1 but correctly classified by f2 (c_{01})
  - The number of instances misclassified by f2 but correctly classified by f1 (c_{10})
  - The number of instances correctly classified by both classifiers (c_{11})
  - The McNemar χ^2 statistics is given by $\chi^2_{MC} = \frac{(|c_{01} - c_{10}| - 1)^2}{c_{01} + c_{10}}$
  - If $c_{01} + c_{10} \geq 20$, then χ^2_{MC} is compared by to the χ^2 statistics. If χ^2_{MC} exceeds the $\chi^2_{1,1-\alpha}$ statistic, then we can reject the null hypothesis that assumes that f1 and f2 perform equally with 1-α confidence.
  - If $c_{01} + c_{10} < 20$, this test cannot be used, and the sign test should be used instead (that is the case in our Labour example)

The Sign Test

- Since the Sign test is usually used on multiple domains (though can be used on a single one with several trial (e.g., 10 folds of cross-validation)), we will discuss it in the next section which looks at multiple domains.

Two classifiers multiple domains

- There are no clear parametric way to deal with the problem of comparing the performance of two classifiers on multiple domains:
  - The t-test is not a very good alternative because it is not clear that we have commensurability of the performance measures in such a setting.
  - The normality assumption is difficult to establish as the number of domains on which the test is run must exceed 30.
  - The t-test is susceptible to outliers, which is more likely when many different domains are considered.
- Therefore we will describe two non-parametric alternatives
  - ➔ The Sign Test
  - ➔ Wilcoxon's signed-Rank Test

The Sign test

- The sign test can be used either to compare two classifiers on a single domain (using the results at each fold as a trial) or more commonly, to compare two classifiers on multiple domains.
- We count the number of times that f1 outperforms f2, n_{f1} and the number of times that f2 outperforms f1, n_{f2}.
- The null hypothesis (stating that the two classifiers perform equally well) holds if the number of wins follows a binomial distribution.
- Practically speaking, a classifier should perform better on at least w_α datasets to be considered statistically significantly better at the α significance level, where w_α is the critical value for the sign test at the α significance level .

Illustration of the sign test

Dataset	NB	SVM	Adaboost	Rand Forest
Anneal	96.43	99.44	83.63	99.55
Audiology	73.42	81.34	46.46	79.15
Balance Scale	72.30	91.51	72.31	80.97
Breast Cancer	71.70	66.16	70.28	69.99
Contact Lenses	71.67	71.67	71.67	71.67
Pima Diabetes	74.36	77.08	74.35	74.88
Glass	70.63	62.21	44.91	79.87
Hepatitis	83.21	80.63	82.54	84.58
Hypothyroid	98.22	93.58	93.21	99.39
Tic-Tac-Toe	69.62	99.90	72.54	93.94

- NB vs SVM: n_{f1}= 4.5, n_{f2}= 5.5 and $w_{0.05}$ = 8 ➔ we cannot reject the null hypothesis stating that NB and SVM perform similarly on these data sets for level α=0.05 (1-tailed)
- Ada vs RF: n_{f1}=1 and n_{f2}=8.5 ➔ we can reject the null hypothesis at level α=0.05 (1-tailed) and conclude that RF is significantly better than Ada on these data sets at that significance level.

Wilcoxon's signed-Rank Test

- Wilcoxon's signed-Rank Test, like the sign test, deals with two classifiers on multiple domains. It is also non-parametric, however, it is more powerful than the sign test. Here is its description:
  - For each domain, we calculate the difference in performance of the two classifiers.
  - We rank the absolute values of these differences and graft the signs in front of the ranks.
  - We calculate the sum of positive and negative ranks, respectively (W_{S1} and W_{S2})
  - $T_{Wilcox} = \min(W_{S1}, W_{S2})$
  - Compare to critical value V_α. If $V_\alpha \geq T_{Wilcox}$ we reject the null hypothesis that the performance of the two classifiers is the same, at the α confidence level.

Wilcoxon's signed-Rank Test: Illustration

Data	NB	SVM	NB-SVM	|NB-SVM|	Ranks	± Ranks
1	.9643	.9944	-0.0301	0.0301	3	-3
2	.7342	.8134	-0.0792	0.0792	6	-6
3	.7230	.9151	-0.1921	0.1921	8	-8
4	.7170	.6616	+0.0554	0.0554	5	+5
5	.7167	.7167	0	0	Remove	Remove
6	.7436	.7708	-0.0272	0.0272	2	-2
7	.7063	.6221	+0.0842	0.0842	7	+7
8	.8321	.8063	+0.0258	0.0258	1	+1
9	.9822	.9358	+0.0464	0.0464	4	+4
10	.6962	.9990	-0.3028	0.3028	9	-9

W_{S1} = 17 and W_{S2} = 28 ➔ T_{Wilcox} = min(17, 28) = 17

For n= 10-1 degrees of freedom and α = 0.005, V = 8 for the 1-sided test. V must be larger than T_{Wilcox} in order to reject the hypothesis. Since 17 > 8, we cannot reject the hypothesis that NB's performance is equal to that of SVM at the 0.005 level.

Multiple classifiers, Multiple Domains

- For the case of multiple classifiers and multiple domains, two alternatives are possible. The parametric alternative is (one-way repeated measure) ANOVA and the non-parametric alternative is Friedman's Test.
- These two tests are multiple-hypothesis tests, also called Omnibus tests. Their null hypotheses is that all the classifiers perform equally, and rejection of that null hypothesis means that: there exists at least one pair of classifiers with significantly different performances.
- In case of rejection of this null hypothesis, the omnibus test is followed by a Post-hoc test whose job is to identify the significantly different pairs of classifiers.
- In this tutorial, we will discuss Friedman's Test (omnibus) and the Nemenyi test (post-hoc test).

Friedman's Test

- All the algorithms are ranked on each domain separately. Ties lead to the rank between the number of classifiers involved in the rank.
- For each classifier, the sum of their ranks obtained on all the domains is computed and named $R.j$, where j is the classifier considered.
- The Friedman Statistic can then be calculated as:
- $\chi_F^2 = [\frac{12}{n \times k \times (k+1)} \times \sum_{j=1}^{k} (R_{.j})^2] - 3 \times n \times (k+1)$

With n representing the number of domains and k, the number of classifiers.

Illustration of the Friedman test

Domain	Classifier fA	Classifier fB	Classifier fC
1	85.83	75.86	84.19
2	85.91	73.18	85.90
3	86.12	69.08	83.83
4	85.82	74.05	85.11
5	86.28	74.71	86.38
6	86.42	65.90	81.20
7	85.91	76.25	86.38
8	86.10	75.10	86.75
9	85.95	70.50	88.03
19	86.12	73.95	87.18

Domain	Classifier fA	Classifier fB	Classifier fC
1	1	3	2
2	1.5	3	1.5
3	1	3	2
4	1	3	2
5	2	3	1
6	1	3	2
7	2	3	1
8	2	3	1
9	2	3	1
10	2	3	1
$R_{.j}$	15.5	30	14.5

$\chi_F^2 = [\frac{12}{10 \times 3 \times (3+1)} \times \sum_{j=1}^{3} (R_{.j})^2] - 3 \times 10 \times (3+1) = 15.05$ For a 2-tailed test at the 0.05 level of significance, the critical value is 7.8. $\chi_F^2 > 7.8$, i.e., Rejection of the NH.

The Nemenyi test

- If Friedman's test shows that there is a significant difference among the algorithms being tested, the Nemenyi test can be used to pinpoint where that difference lies.
- If R_{ij} is the rank of classifier f_j on data set S_i, we compute the mean rank of classifier f_j on all data sets as:

$$\overline{R}_{.j} = \frac{1}{n} \sum_{i=1}^{n} R_{ij}$$

- The q_{yz} statistic between classifier f_y and f_z is

$$q_{yz} = \frac{\overline{R}_y - \overline{R}_z}{\sqrt{\frac{k(k+1)}{6n}}}$$

(n is the number of domains and k, the number of classifiers)

Illustration of the Nemenyi Test

- From the Friedman test, we had:

$$\overline{R}_{.A} = 15.5,\ \overline{R}_{.B} = 30 \text{ an } \overline{R}_{.C} = 14.5$$

- Replacing $R_{.y}$ and $R_{.Z}$ by the above values in

$$q_{yz} = \frac{\overline{R}_{.y} - \overline{R}_{.z}}{\sqrt{\frac{k(k+1)}{6n}}}$$

- We get $q_{AB} = -32.22$, $q_{AC} = 2.22$, and $q_{BC} = 34.44$
- $q_\alpha = 2.55$ for $\alpha = 0.05$ (q_α must be larger than q_{yz} for the hypothesis that y and z perform equally to be rejected)
- There fore, we reject the null hypothesis in the case of classifiers A and B and B and C, but not in the case of A and C.

Effect Size

- Statistical tests can determine whether a difference between classifiers is significant, but not whether it is of practical importance.
- Statistical significant is known as the *effect*, and practical relevance is obtained by measuring the *size* of this effect.
- Cohen's d statistic is the most appropriate measurement of effect size for classification. It is calculated as:
  - $d_{cohen} = \frac{(\overline{pm}(f_1) - \overline{pm}(f_2)}{\sigma_p}$

where σ_p is the pooled standard deviation and is defined as:

- $\sigma_p = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}$

where σ_1^2 and σ_2^2 are the variances of pm(f1) and pm(f2), respectively.

Part II

Topic 1: Sampling

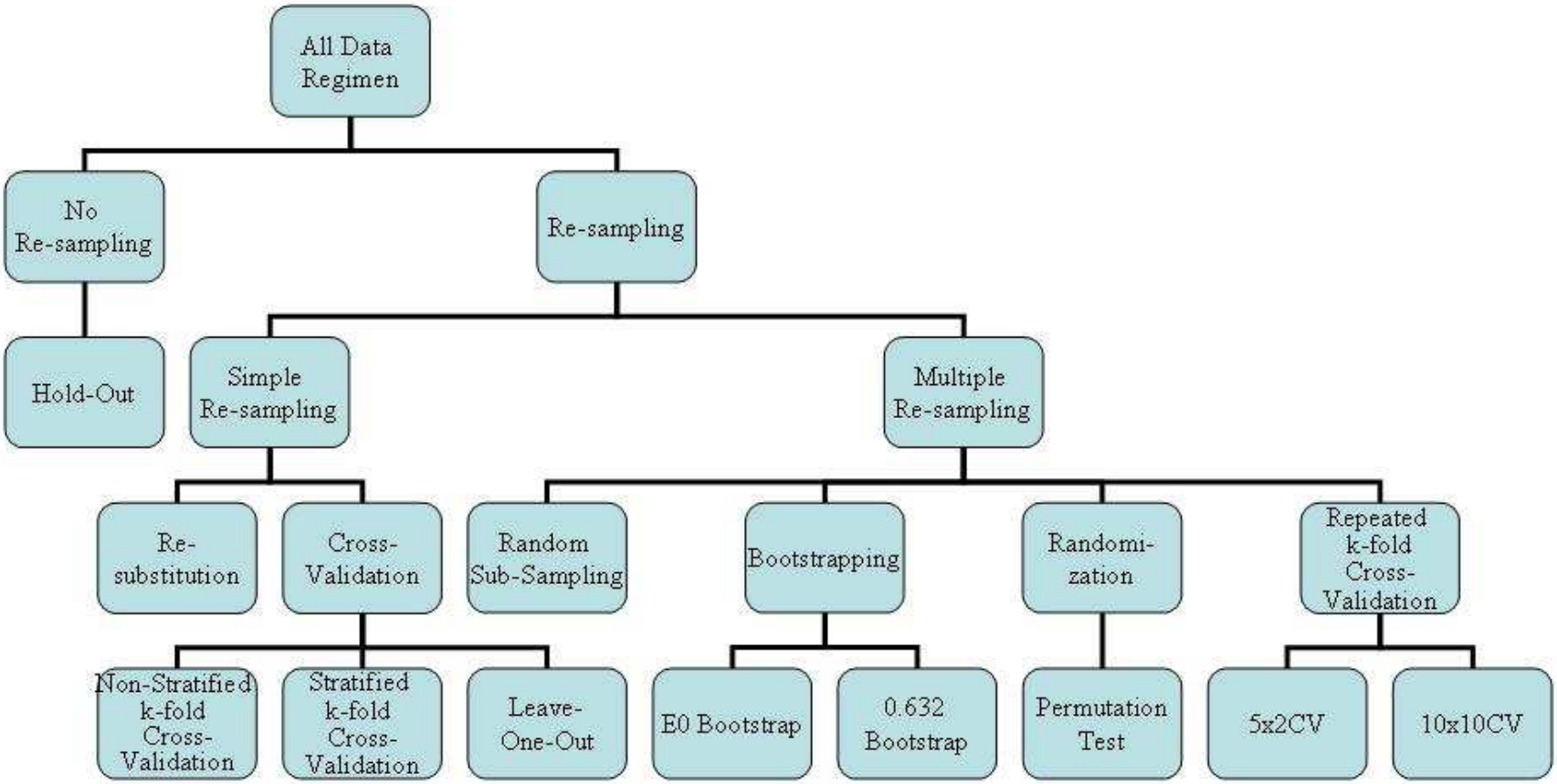
What is the Purpose of Resampling?

- Ideally, we would have access to the entire population or a lot of representative data from it.
- This is usually not the case, and the limited data available has to be re-used in clever ways in order to be able to estimate the error of our classifiers as reliably as possible, i.e., to be re-used in clever ways in order to obtain sufficiently large numbers of samples.
- Resampling is divided into two categories: *Simple re-sampling* (where each data point is used for testing only once) and *Multiple re-sampling* (which allows the use of the same data point more than once for testing)

What are the dangers of Resampling?

- Re-sampling is usually followed by Statistical testing. Yet statistical testing relies on the fundamental assumption that the data used to obtain a sample statistics must be independent.
- However, if data is re-used, then this important independence assumption is broken and the result of the statistical test risks being invalid.
- In addition to discussing a few re-sampling approaches, we will underline the issues that may arise when applying them.

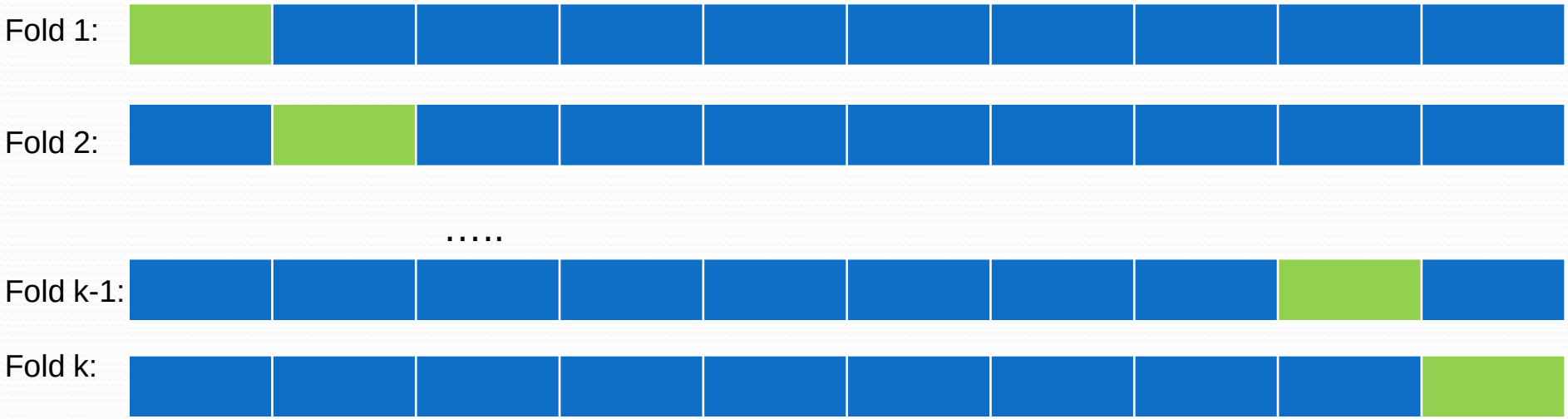
Overview of Re-Sampling Methods



Resampling Approaches Discussed in this Tutorial

- Simple Resampling:
  - Cross-Validation and its variants
- Multiple Resampling:
  - The 0.632 Bootstrap
  - The Permutation Test
  - Repeated k-fold Cross-Validation

k-fold Cross-Validation



: training data subset
: testing data subset

In Cross-Validation, the data set is divided into k folds and at each iteration, a different fold is reserved for testing and all the others, used for training the classifiers.

Some variations of Cross-Validation

- Stratified k-fold Cross-Validation:
  - This variation is useful when the class-distribution of the data is skewed. It ensures that the distribution is respected in the training and testing sets created at every fold. This would not necessarily be the case if a pure random process were use.
- Leave-One-Out
  - In k-fold Cross-Validation, each fold contains m/k data points where m is the overall size of the data set. In Leave-one-out, k =m and therefore, each fold contains a single data point.

Considerations about k-fold Cross-Validation and its variants

- k-fold Cross-Validation is the best known and most commonly used resampling technique.
- K-fold Cross-Validation is less computer intensive than Leave-One-Out.
- The testing sets are independent of one another, as required, by many statistical testing methods, but the training sets are highly overlapping. This can affect the bias of the error estimates.
- Leave-One-Out produces error estimates with high variance given that the testing set at each fold contains only one example. The classifier is practically unbiased since each fold trains on almost all the data.

Boootsrapping

- Bootstrapping assumes that the available sample is representative and creates a large number of new samples by drawing from replacement from the available sample.
- Bootstrapping is useful in practice when the sample is too small for Cross-Validation or Leave-One-Out approaches to yield a good estimate
- There are two Bootstrap estimates that are useful in the context of classification: the Є0 and the e632 Bootstrap.
- The Є0 Bootstrap tends to be pessimistic because it is only trained on 63.2% of the data in each run. The e632 attempts to correct for this.

The Є0 and e632 Bootstraps

- Given a data set D of size m, we create k bootstrap samples B_i of size m, by sampling from D with replacement (k is typically ≥ 200).
- At each run, each of the k bootstraps represent the training set while the testing set is made up of a single copy of the examples from D that did not make it to B_i.
- At each run, a classifier is trained and tested and $Є0_i$ represents the performance of the classifier at that run.
- Є0 represents the average of all the $Є0_i$'s.

$$e632 = 0.632 \times Є0 + 0.368 \times err(f)$$

where err(f) is the optimistically biased re-substitution error (error rate

Considerations about Bootstrapping

- Bootstrapping yields better estimates than Cross-validation and its variants when the data set is very small.
- The result of bootstrapping in such cases was shown to have low variance.
- The Ɛo is a good estimator when the true error rate is very high.
- E632 is a good estimator on small data sets especially if the true error rate is small.
- Bootstrapping is a poor estimator for for certain types of classifiers like Nearest Neighbours or FOIL which do not benefit from the presence of duplicate instances.

The Permutation Test

- Given the error estimate found on a data set, the question is:
  - Is this error estimate significant, or
  - Could it also have been obtained on 'bogus' data?
- The 'bogus' data is created by taking the genuine samples and randomly choosing to either leave their label intact or switch them.
- Once this 'bogus' data set is created, the classifier is ran on it and its error estimated.
- This process is repeated a very large number of times in an attempt to establish whether the error estimate obtained on the true data is truly different from those obtained on large numbers of 'bogus' data sets.

Repeated k-Fold Cross-Validation(1)

- In order to obtain more stable estimates of an algorithm's performance, it is useful to perform multiple runs of simple re-sampling schemes. This can also enhance replicability of the results.
- Two specific schemes have been suggested in the context of Cross-Validation: 5x2 CV and 10x10 CV
- K-fold CV does not estimate the mean of the difference between 2 learning algorithms properly. The mean at a single fold behaves better. This lead Dietterich (1998) to propose the 5x2 CV fold, in which 2-fold CV is repeated 5 times.

Repeated k-Fold Cross-Validation(2)

- Dietterich found that the paired t-test based on the the 5x2 CV scheme had lower probability of issuing a type-I error but had less power than the k-fold CV paired t-test.
- Alpaydyn (1999) proposed to substitute the t-test at the end of the 5x2CV scheme by an F-test. That test had an even lower chance of issuing a type-I error and had increased power (though that new test was not compared to k-fold CV in terms of type I error or power)
- Bouckaert (2003) proposed several variations of a 10x10 CV scheme. Generally speaking these schemes show a higher probability of Type-I error than 10-fold CV, but higher power.

Part II

Topic 2: Choosing Appropriate Data Sets for Testing Classifiers

Considerations to keep in mind while choosing an appropriate test bed

- Wolpert's "No Free Lunch" Theorems: if one algorithm tends to perform better than another on a given class of problems, then the reverse will be true on a different class of problems.
- LaLoudouana and Tarate (2003) showed that even mediocre learning approaches can be shown to be competitive by selecting the test domains carefully.

➔The purpose of data set selection should not be to demonstrate an algorithm's superiority to another in all cases, but rather to identify the areas of strengths of various algorithms with respect to domain characteristics or on specific domains of interest.

Where can we get our data from?

- Repository Data: Data Repositories such as the UCI repository and the UCI KDD Archive have been extremely popular in Machine Learning Research.
- Artificial Data: The creation of artificial data sets representing the particular characteristic an algorithm was designed to deal with is also a common practice in Machine Learning
- Web-Based Exchanges: Could we imagine a multi-disciplinary effort conducted on the Web where researchers in need of data analysis would “lend” their data to macine learning researchers in exchange for an analysis?

Pros and Cons of Repository Data

- Pros:
  - Very easy to use: the data is available, already processed and the user does not need any knowledge of the underlying field.
  - The data is not artificial since it was provided by labs and so on. So in some sense, the research is conducted in real-world setting (albeit a limited one)
  - Replication and comparisons are facilitated, since many researchers use the same data set to validate their ideas.
- Cons:
  - The use of data repositories does not guarantee that the results will generalize to other domains.
  - The data sets in the repository are not representative of the data mining process which involves many steps other than classification.
  - Community experiment/Multiplicity effect: since so many experiments are run on the same data set, by chance, some will yield interesting (though meaningless) results

Pros and Cons of Artificial Data

- Pros:
  - Data sets can be created to mimic the traits that are expected to be present in real data (which are unfortunately unavailable)
  - The researcher has the freedom to explore various related situations that may not have been readily testable from accessible data.
  - Since new data can be generated at will, the multiplicity effect will not be a problem in this setting.
- Cons:
  - Real data is unpredictable and does not systematically behave according to a well defined generation model.
  - Classifiers that happen to model the same distribution as the one used in the data generation process have an unfair advantage.

Pros and Cons of Web-Based Exchanges

- Pros:
  - Would be useful for three communities:
    - Domain experts, who would get interesting analyses of their problem
    - Machine Learning researchers who would be getting their hand on interesting data, and thus encounter new problems
    - Statisticians, who could be studying their techniques in an applied context.
- Cons:
  - It has not been organized. Is it truly feasible?
  - How would the quality of the studies be controlled?

Part II

Topic 3: Available Resources

What help is available for conducting proper evaluation?

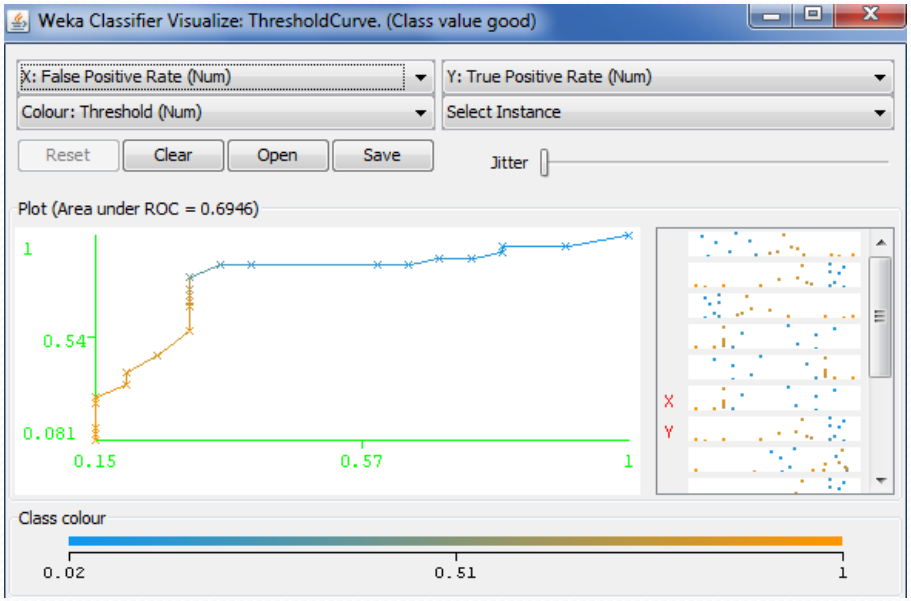
- There is no need for researchers to program all the code necessary to conduct proper evaluation nor to do the computations by hand.
- Resources are readily available for every steps of the process, including:
  - Evaluation metrics
  - Statistical tests
  - Re-sampling
  - And of course, Data repositories
- Pointers to these resources and explanations on how to use them are discussed in our book: <*"Evaluating Learning Algorithms: A Classification Perspective"* by Japkowicz and Shah, Cambridge University Press, 2011>.

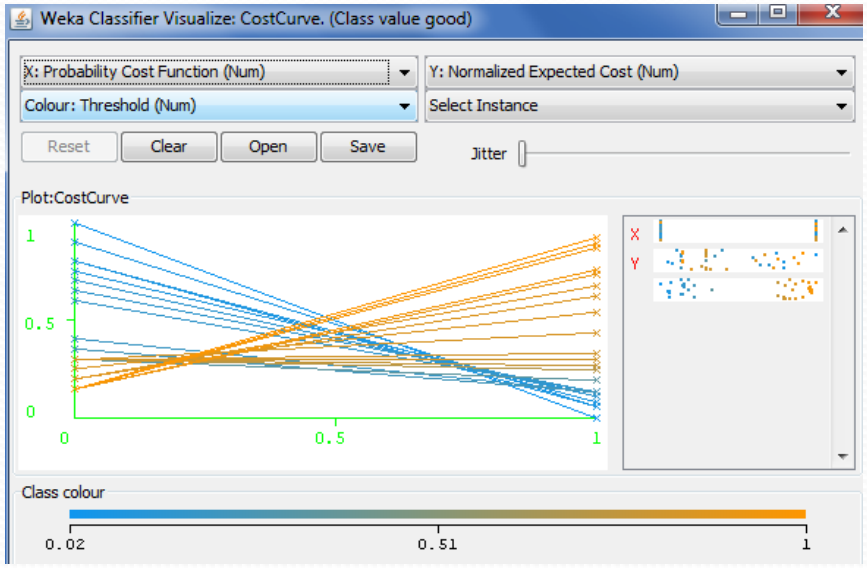
Where to look for evaluation metrics?

Actually, as most people know, Weka is a great source for, not only, classifiers, but also computation of the results according to a variety of evaluation metrics

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          42               73.6842 %
Incorrectly Classified Instances        15               26.3158 %
Kappa statistic                          0.4415
K&B Relative Info Score               1769.6451 %
K&B Information Score                   16.5588 bits      0.2905 bits/instance
Class complexity | order 0              53.3249 bits      0.9355 bits/instance
Class complexity | scheme             3267.2456 bits     57.3201 bits/instance
Complexity improvement     (Sf)      -3213.9207 bits    -56.3846 bits/instance
Mean absolute error                      0.3192
Root mean squared error                  0.4669
Relative absolute error                 69.7715 %
Root relative squared error             97.7888 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.7       0.243      0.609     0.7       0.651      0.695    bad
                 0.757     0.3        0.824     0.757     0.789      0.695    good
Weighted Avg.    0.737     0.28       0.748     0.737     0.74       0.695

=== Confusion Matrix ===

  a  b   <-- classified as
 14  6 |  a = bad
  9 28 |  b = good
```

WEKA even performs ROC Analysis and draws Cost-Curves





Although better graphical analysis packages are available in R, namely the ROCR package, which permits the visualization of many versions of ROC and Cost Curves, and also Lift curves, P-R Curves, and so on

Where to look for Statistical Tests?

```
> c45 = c(.2456, .1754, .1754, .2632, .1579, .2456, .2105, .1404, .2632,.2982)
> nb = c(.0702, .0702, .0175, .0702, .0702, .0526, .1579, .0351, .0351,.0702)
> t.test(c45, nb, paired= TRUE)

        Paired t-test

data:  c45 and nb
t = 7.8645, df = 9, p-value = 2.536e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1087203 0.1965197
sample estimates:
mean of the differences
                0.15262
```

```
> | > classA= c(85.83, 85.91, 86.12, 85.82, 86.28, 86.42, 85.91, 86.10, 85.95, 86.12)
  > classB= c(75.86, 73.18, 69.08, 74.05, 74.71, 65.90, 76.25, 75.10,  70.50, 73.95)
  > classC= c(84.19, 85.90, 83.83, 85.11, 86.38, 81.20, 86.38, 86.75, 88.03, 87.18)
  > t=matrix(c(classA, classB, classC), nrow=10, byrow=FALSE)
  > t
        [,1]  [,2]  [,3]
 [1,] 85.83 75.86 84.19
 [2,] 85.91 73.18 85.90
 [3,] 86.12 69.08 83.83
 [4,] 85.82 74.05 85.11
 [5,] 86.28 74.71 86.38
 [6,] 86.42 65.90 81.20
 [7,] 85.91 76.25 86.38
 [8,] 86.10 75.10 86.75
 [9,] 85.95 70.50 88.03
[10,] 86.12 73.95 87.18
  > friedman.test(t)

        Friedman rank sum test

data:  t
Friedman chi-squared = 15, df = 2, p-value = 0.0005531

>
```

```
> c4510folds= c(3, 0, 2, 0, 2, 2, 2, 1, 1, 1)
> nb10folds= c(1, 0, 0, 0, 0, 1, 0, 2, 0, 0)
> wilcox.test(nb10folds, c4510folds, paired= TRUE)

        Wilcoxon signed rank test with continuity correction

data:  nb10folds and c4510folds
V = 2.5, p-value = 0.03125
alternative hypothesis: true location shift is not equal to 0
```

The R Software Package contains implementations of all the statistical tests discussed here and many more. They are very simple to run.

Where to look for Re-sampling methods?

- In our book, we have implemented all the re-sampling schemes described in this tutorial. The actual code is also available upon request. (e-mail me at <nat@site.uottawa.ca>)

```
eoBoot = function(iter, dataSet, setSize, dimension, classifier1, classifier2){
 classifier1eoBoot <- numeric(iter)
classifier2eoBoot <- numeric(iter)
 for(i in 1:iter) {
 Subsamp <-  sample(setSize, setSize, replace=TRUE)
 Basesamp <- 1:setSize
  oneTrain <- dataSet[Subsamp ,1:dimension ]
 oneTest <- dataSet[setdiff(Basesamp,Subsamp), 1:dimension]
 classifier1model <- classifier1(class~., data=oneTrain)
 classifier2model <- classifier2(class~., data=oneTrain)
     classifier1eval <- evaluate_Weka_classifier(classifier1model, newdata=oneTest)
  classifier1acc <- as.numeric(substr(classifier1eval$string, 70,80))
  classifier2eval <- evaluate_Weka_classifier(classifier2model,
                        newdata=oneTest)
      classifier2acc <- as.numeric(substr(classifier2eval$string, 70,80))
 classifier1eoBoot[i]= classifier1acc
 classifier2eoBoot[i]= classifier2acc
 }
return(rbind(classifier1eoBoot, classifier2eoBoot))}
```

Part III

Recent Research

Topic 1: A Visualization-based framework for classifier evaluation

(Nathalie Japkowicz, Pritika Sanghi and Peter Tischer)

[ECML'2008, ISAIM'2008]

A New Framework for Classifier Evaluation

- Classifier evaluation can be viewed as a problem of analyzing high-dimensional data.
- The performance measures currently used are but one class of projections that can be applied to these data.
- Why not apply other (standard or not) projections to the data with various kinds (standard or not) distance measures?

Some Advantages of this new Framework

- Projection approaches are typically intended for visualization. This yields two advantages:
  - A quick and easy way for human-beings to assess classifier performance results.
  - The possibility to offer simultaneous multiple views of classifier performance evaluation.
- The framework offers a solution to the problem of aggregating the results obtained by a classifier on several domains.
- The framework offers a way to deal with multi-class domains.

The Framework and its Implementation

- The framework is implemented as a function of the following steps:
  1. All the classifiers in the study are run on all the domains of the study.
  2. The performance matrices (e.g., confusion matrix) of a single classifiers on every domain are aggregated into a single vector. This is repeated for each classifier
  3. A projection and a distance measure for that projection are chosen and applied to the vectors of Step 2.

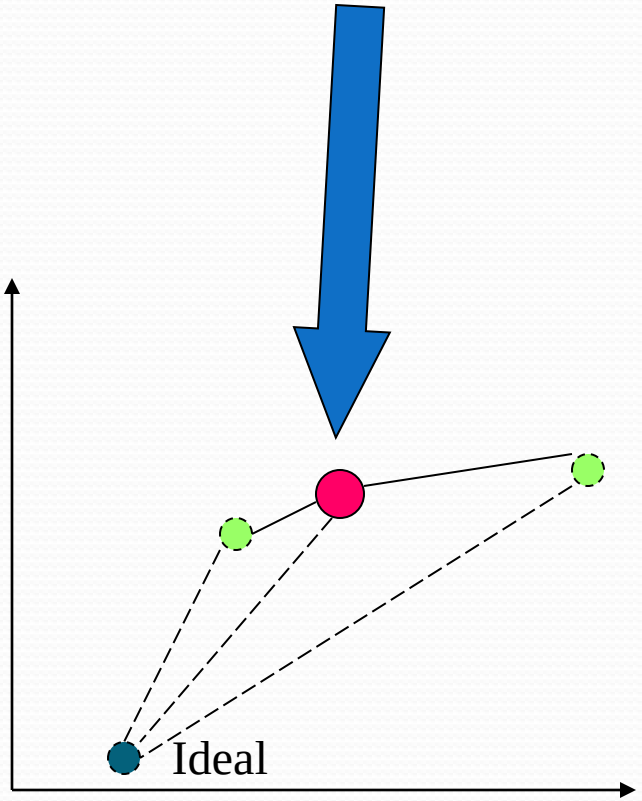
Illustration of the Framework

True class →	Pos	Neg
Yes	82	17
No	12	114

True class →	Pos	Neg
Yes	15	5
No	25	231

True class →	Pos	Neg
Yes	99	6
No	1	94

82	17	12	114	15	5	25	231	99	6	1	94



Confusion matrices for a single classifier on three domains

A Few Remarks about our Framework

- It decomposes the evaluation problem neatly, separating the issue of projection from that of distance measure.
- By going from a projection (or two) into a one-dimensional space to one into a two-dimensional space, we allow for two rather than one relationships to be established:
  - The ranking of classifiers with respect to the ideal classifier.
  - The comparison of each classifier to the others.

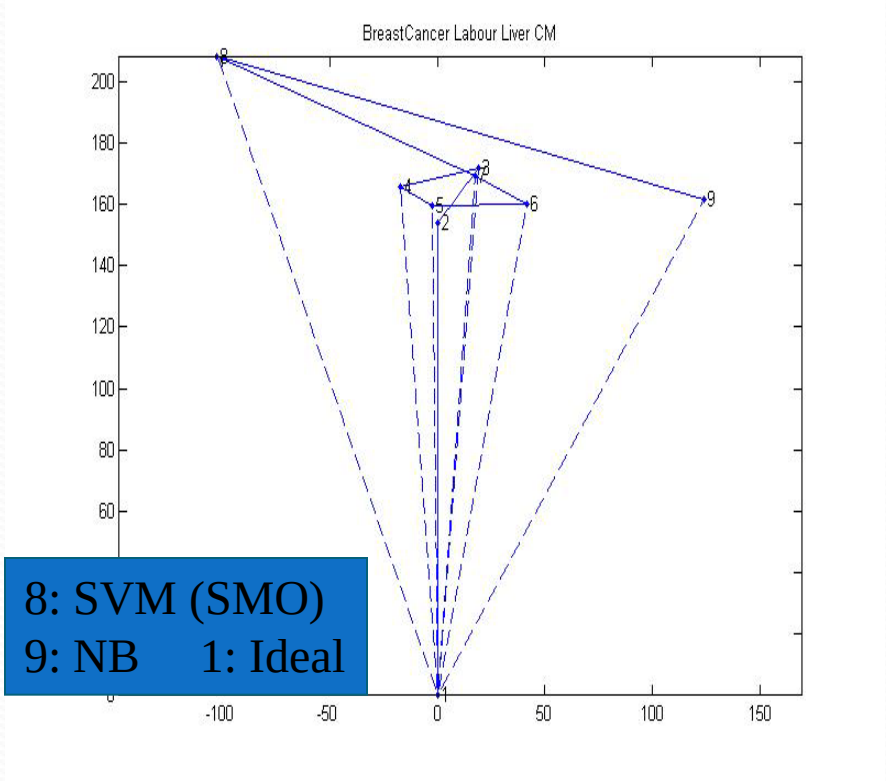
Specific Implementation Details

- Our framework can be used with any projection technique and any distance function associated to the projection.
- In this work, we experimented with:
  - A Minimal Cost Spanning Tree (MCST) distance-preserving projection [Yang, 2004].
  - Two distance functions: the Euclidean Distance (L2-norm) and the Manhattan Distance (L1-Norm). In our experiments, the L2 norm is the one normally used, unless otherwise specified.
- We focus on the results obtained
  - When combining the confusion matrices of various classifiers on several domains,
  - When dealing with a multi-class domain.

Experimental Set-Up

- We tested our approach on four UCI domains:
  - 3 Binary ones: Breast Cancer, Labour and Liver.
  - 1 Multi-Class One: Anneal.
- We compared the performance of eight WEKA classifiers on these domains: NB, J48, Ibk, JRip, SMO, Bagging, AdaBoost, RandFor.
- The focus of our study is not to discover which classifier wins or loses on our data sets. Rather, we are using our experiments to illustrate the advantages and disadvantages of our framework over other evaluation methods. We, thus, used Weka's default parameters in all cases.
- Also, though we test our approach with the MCST projection, others could have been used. This is also true of our distance functions.

Illustration on Multiple domains: Breast Cancer, Labour and Liver

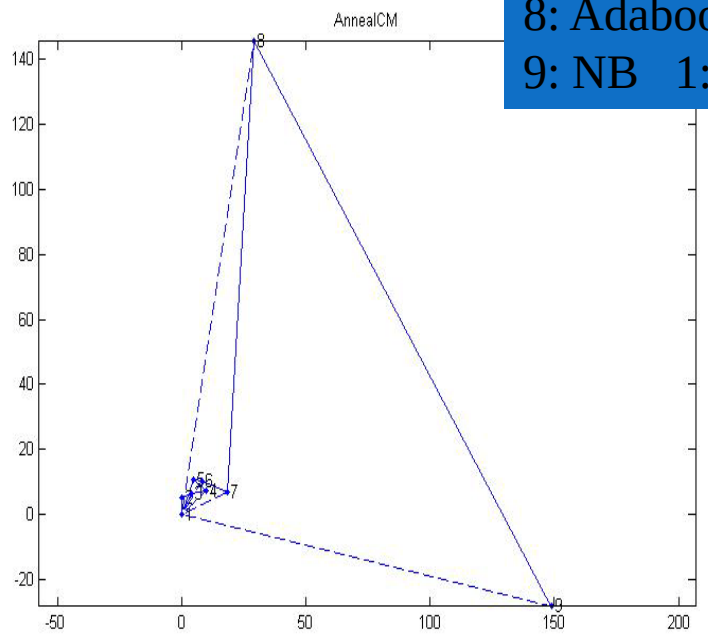


		Acc.	F-Meas.	AUC
NB	BC:	71.7	.48	.7
	La:	89.5	.92	.97
	Li:	55.4	.6	.64
	Avg:	72.2	.67	.77
SMO	BC:	69.6	.39	.59
	La:	89.5	.92	.87
	Li:	58.3	.014	.5
	Avg:	72.46	.44	.65
Boost.	BC:	70.3	.46	.7
	La:	87.7	.91	.87
	Li:	66.1	.534	.68
	Avg:	74.7	.64	.75

Abnormality detection with our new approach is a lot easier and accurate than it is, when relying on Accuracy, F-Measure, or AUC listings on each domain or their average on all domains.

Also, our new approach Allows us to mix binary and multi-class domains. Averaging does not!

Illustration on a Multiclass domain: Anneal (L2-Norm)



Adaboost:

a	b	c	d	e	f	← classified as
0	0	8	0	0	0	| a
0	0	99	0	0	0	| b
0	0	684	0	0	0	| c
0	0	0	0	0	0	| d
0	0	0	0	67	0	| e
0	0	40	0	0	0	| f

NB:

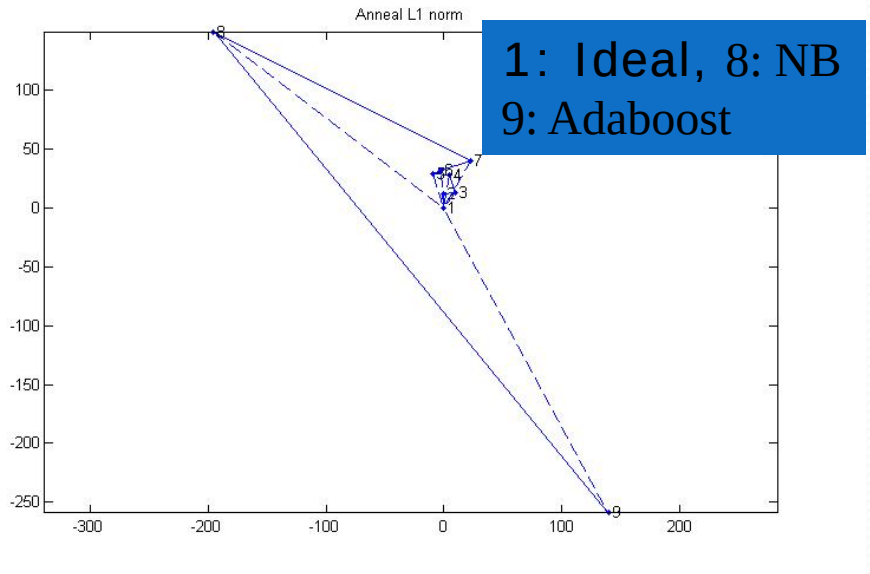
a	b	c	d	e	f	←classified as
7	0	1	0	0	0	| a
0	99	0	0	0	0	| b
3	38	564	0	0	79	| c
0	0	0	0	0	0	| d
0	0	0	0	67	0	| e
0	0	2	0	0	38	| f

NB	Other Classifiers	Ada boost
86.3	97.4 to 99.3	83.6

Accuracy

Accuracy does not tell us whether NB and Adaboost make the same kind of errors!

Illustration on Anneal using the L1-Norm



- When using the L2-norm, NB and Adaboost were at approximately the same distance to Ideal.
- When using the L1-norm, NB is significantly closer.
- NB makes fewer errors than Adaboost, but the majority of its errors are concentrated on one or several large classes.

Because our new evaluation framework is visual in nature, we can process quickly the results obtained using various distance Measures (evaluation measure), and, thus, interpret our results In a more informed manner. It is easier done this way than by staring at large tables of numbers!

Summary

- We present a new framework for classifier evaluation that recognizes that classifier evaluation consists of projecting high-dimensional data into low-dimensional ones.
- By using a projection into 2-dimensional space rather than one, we propose a visualization approach to the problem. This allows for quick assessments of the classifiers' behaviour based on the results obtained using multiple performance measures.
- Each entry of the evaluation vectors we project is compared in pair-wise fashion to its equivalent in other vectors. Thus, our aggregation technique is more precise than that used with traditional performance measures. This is an advantage when considering results over various domains, or in the case of multi-class domains.

Future Work

- As presented, our approach seems limited to the comparison of single classifier's performance.
  - How about threshold-insensitive classifiers?
  - How about the computation of statistical guarantees on our results?

➢ This can be solved by plotting either the results obtained at various thresholds, or the results obtained at various folds of a cross-validation regimen, thus plotting clouds of classifiers that could then be analyzed.

❑ We also plan to experiment with other distance measures and projection methods.

Part III

Recent Research

Topic 2: Assessing the Impact of Changing environments on Classifier Performance

(Rocio Alaiz-Rodriguez and Nathalie Japkowicz)

[Canadian AI ' 2008]

Purpose of the Work

- **Direct purpose:** To test the hypothesis by David Hand (2006) that simple classifiers are more robust to changing environments than complex ones.
- **Indirectly:** To demonstrate the feasibility and value of generating artificial, but realistic domains.
- **More generally:** To propose an alternative to the use of the UCI domains.

Specific hypotheses under review

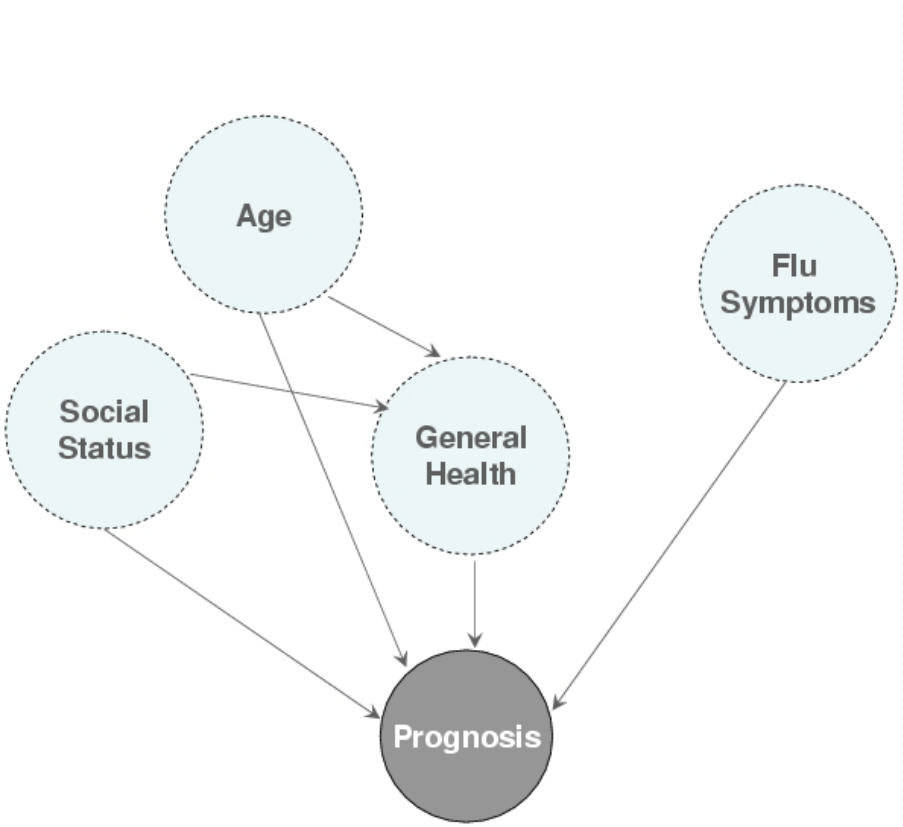
- **Preliminaries:** Different kinds of changing environments:
  - **Population Drift** — p(d|x) remains unchanged, but p(x) differs from training to testing set. Also known as: covariate shift or sample selection bias.
  - **Class Definition Change** — p(x) does not change, but p(d|x) varies from training to testing set. Also known as: concept drift or functional relation change.
- **Hypotheses under review:**
  - **Hypothesis 1:** When either or both a population drift and a class definition change occurs, can we generally observe a drop in performance by all kinds of classifiers?
  - **Hypothesis 2:** Do simpler classifiers maintain their performance more reliably than more complex ones in such cases?

Our Experimental Framework I

- Our domain is a simulated medical domain that states the prognostic of patients infected with the flu and described as follows:
  - Patient's age [Infant, Teenager, YoungAdult, Adult, OldAdult, Elderly]
  - Severity of flu symptoms [Light, Medium, Strong]
  - Patient's general health [Good, Medium, Poor]
  - Patient's social position [Rich, MiddleClass, Poor]
  - Class: NormalRemission, Complications

Our Experimental Framework II

- In order to make the problem interesting for classifiers, we assumed that the features are not independent of one another.
- We also assumed that certain feature values were irrelevant.



Attribute Dependency Graph

Our Experimental Framework III

We used different distributions to model the various features and the class:

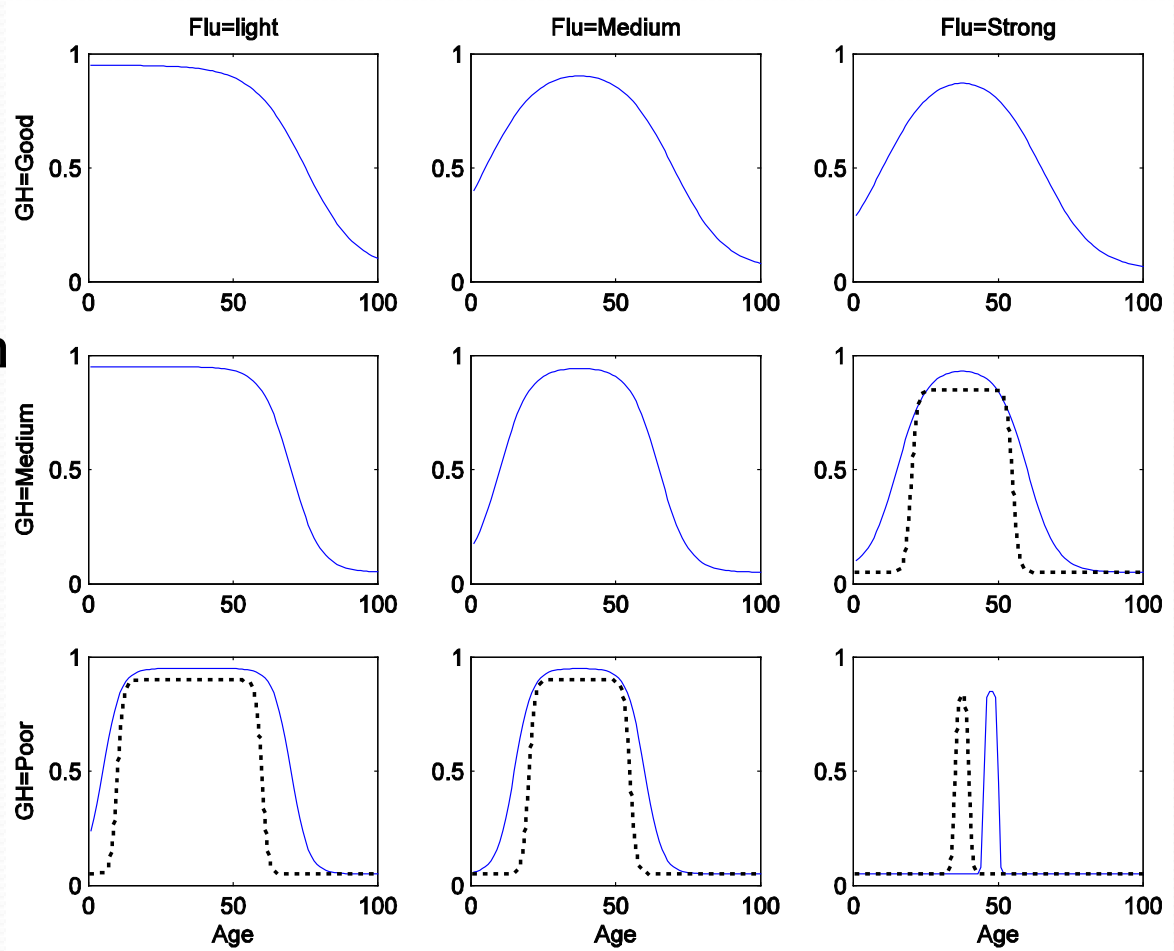
- **Age:** we assumed a region with negative growth which, according to the Population Reference Bureau contains many people in the Adult and Young Adult categories, and few people in the other categories. We used uniform distributions to model the first five categories, and an exponential distribution for the elderly category (which is not bounded upwards).
- **Social Status:** Normal Distribution distributed around the Middle class (= 2), with variance .75. [Poor=3 and Rich= 1]
- **Severity of Flu Symptoms:** (= Severity of the virus strain) Same distribution as Social Status.

Our Experimental Framework IV

- **General Health:** We used an unobserved binary variable called "delicate person" whose probability increases with age, and generated rules based on the assumptions that (1) delicate people have worse general health than other members of the population; and (2) poorer people have worse geberal health than the richer members of society.
- **Class:** The class labels were assigned automatically and abided by the following general principles:
  - In case of stronger flu symptoms, the chances of complications are greater.
  - Infants and Elderly have greater chances of complications
  - People with poorer general health are more susceptible to complications

Our Experimental Framework V

- Class Distribution, or probability of normal remission as a function of age, severity of the flu symptoms and general health. When two lines appear, the discontinuous one applies to instances with poor social status



Experimental Set Up

- We compared two pairs of simple/complex classifiers: 1R/Decision Tree and Simple Perceptron/Multi-Layer Perceptron
- We looked at three situations:
  - Population drifts with full representation
  - Population drifts with some cases not represented
  - Concept drifts
- We studied the performance deterioration that occurs when going from no drift to one of the three types of drifts above.

Changes to the Testing Set I

- Population Drifts with full representation:
  - Developing Population (DP): high birth and death rates.
  - Zero Growth Population (ZGP): Similar birth and death rates.
  - Season changes (NGP/W): Winter: Flu symptoms get stronger
  - Season changes (NGP/SW): Soft Winter: Flu Symptoms get milder
  - Season changes (NGP/DW): Drastic Winter: Flu Symptoms get stronger and general health declines
  - Population is much poorer (NGP/P)
  - Population is much poorer and the winter is trastic (NGP/P+DW)

Changes to the Testing Set II

- Population Drifts with Non-Represented cases
  - We considered several situations where one or two population groups are not represented.
- Class Definition Changes:
  - More Complications (MC): the probability of normal remission decreases for certain ages, social statuses and flu symptoms.
  - Fewer Complications (FC): the age group for which the probability of normal remission is high is widened.

Evaluation Measure

- In order to measure the changes in performance caused by environmental changes, we introduce a new metric, called ***performance Deterioration (pD)***, defined as follows:

$$pD = \begin{cases} \dfrac{E_{test} - E_{ideal}}{E_o - E_{ideal}}, & \text{if } E_{test} <= E_o \\ 1, & \text{Otherwise} \end{cases}$$

With E_o representing the error rate of the trivial classifier, E_{test} is the classifier's error rate on the test set, and E_{ideal} is the classifier's error rate when trained and tested on data abiding by the same distribution.

Results I: Population Drifts with Full Representation

averaged value:

		Test Sets								Averaged pD
		NGP	DP	ZGP	NGP W	NGP SW	NGP P	NGP DW	NGP P+DW	
Trivial Classifier	Error rate	36.0	32.9	40.8	37.2	34.4	39.8	41.4	46.2	
1R	Error rate(Training conditions: NGP)	23.7	34.2	27.5	28.5	22.6	24.1	30.7	33.7	
	Error rate(Training = Test conditions)	23.7	32.8	27.3	27.2	19.0	23.6	28.3	27.4	
	pD		1	0.01	0.13	0.24	0.03	0.19	0.33	0.28
Decision Tree	Error rate(Training conditions: NGP)	18.9	24.7	21.1	20.8	15.7	18.2	19.1	18.8	
	Error rate(Training = Test conditions)	18.9	23.4	20.7	20.2	15.1	16.8	18.7	16.8	
	pD		0.14	0.02	0.04	0.03	0.06	0.02	0.07	0.05
Simple NN	Error rate(Training conditions: NGP)	21.2	33.4	26.1	24.8	16.9	21.1	24.9	25.6	
	Error rate(Training = Test conditions)	21.2	25.9	25.6	23.7	16.4	20.4	23	22.9	
	pD		1	0.03	0.08	0.03	0.04	0.10	0.12	0.20
Complex NN	Error rate(Training conditions: NGP)	18.6	24.4	21.1	20.6	15.4	17.3	19.7	18.5	
	Error rate(Training = Test conditions)	18.6	23.8	20.8	20.2	14.9	16.6	18.6	16.7	
	pD		0.07	0.02	0.02	0.03	0.03	0.05	0.06	0.04

Verification of our hypotheses:

(a) A drop in performance is observed by all classifiers

(b) Simpler classifiers suffer much more.

Results II: Population Drifts with Non-Represented Cases

as the averaged value.

		Test Set: NGP							
		1R		Decision Tree		Simple NN		Complex NN	
		Error rate	pD	Error rate	pD	Error rate	pD	Error rate	pD
	Trivial classifier	36.0		36.0		36.0		36.0	
Training =	Test conditions	23.7		18.9		21.2		18.6	
Training set:	No Infant	23.7	0.00	19.0	0.01	22.2	0.07	19.2	0.04
	No Teenager	24.0	0.02	19.4	0.03	21.3	0.00	18.6	0.00
	No Young	25.0	0.11	18.9	0.00	23.3	0.14	19.8	0.07
	No Adult	26.5	0.23	22.2	0.19	23.5	0.16	20.2	0.09
	No Old Adult	24.6	0.07	19.6	0.04	23.8	0.17	19.8	0.07
	No Elderly	24.1	0.03	19.5	0.03	22.9	0.11	19.3	0.04
	No Old Adult + No Elderly	33.3	0.78	24.5	0.33	32.7	0.77	27.4	0.50
	No Infant + No Teenager	24.4	0.06	20.7	0.11	21.4	0.01	20.0	0.08
	No Elderly + No Infant	24.2	0.04	20.0	0.06	21.4	0.01	19.2	0.04
	No Old Adult+ No Teenager	25.0	0.11	20.2	0.08	21.8	0.04	19.2	0.04
	Averaged pD		0.29		0.17		0.23		0.18

Verification of our hypotheses:

(a) A drop in performance is observed by all classifiers

(b) Simpler classifiers suffer slightly more.

Results III: Class Definition Changes

			Test set: NGP		Test set: MC		Test set: FC		Averaged pD
			Error rate	pD	Error rate	pD	Error rate	pD	
Trivial classifier	Training conditions		36.0		28.4		34.4		
1R	Training set	NGP	23.7	-	27.1	0.33	22.8	0.10	
		MC	26.1	0.20	26.5	-	26.1	0.36	
		FC	25.8	0.17	29.5	1	21.5	-	
									0.36
Decision Tree	Training set	NGP	18.6	-	21.6	0.08	19.0	0.08	
		MC	20.6	0.11	21.0	-	24.1	0.38	
		FC	20.4	0.11	25.9	0.67	17.7	-	
									0.24
Simple NN	Training set	NGP	21.2	-	26.3	0.17	19.4	0.08	
		MC	23.1	0.13	25.9	-	23.0	0.30	
		FC	22.0	0.05	25.9	0.73	18.1	-	
									0.24
Complex NN	Training set	NGP	18.6	-	21.6	0.12	19.0	0.10	
		MC	20.8	0.13	20.7	-	23.4	0.36	
		FC	20.5	0.11	25.0	0.56	17.3	-	
									0.23

Verification of our hypotheses:

(a) A drop in performance is observed by all classifiers

(b) Simpler classifiers don't necessarily suffer more than complex ones. (SimpleNN does not. 1R does)

Summary

- Our results show that the trend hypothesized by David Hand does happen in some cases, but does not happen in others.
- In all cases, however, complex classifiers that generally obtain lower error rates in the original scenario (with no changing conditions) remain the best choice since their performance remains higher than that of the simple classifiers even though their performance deterioration are sometimes equivalent.
- Given the dearth of data sets representing changing environments, none of the results we present here could have been obtained had we not generated artificial though realistic data sets simulating various conditions.

Future Work

- Develop a systematic way to generate realistic artificial data sets that could replace, or, at least, supplement the UCI domains.
- Find a way to verify the realistic nature of these data sets.
- Rather than generate data sets from intuition as we've done it here, start from actual real-world data sets and expand them artificially.

General Conclusions

- A first step to improve the evaluation process in machine learning is to be more aware of the tradeoffs involved in choosing one evaluation method over another. This tutorial as well as the book I co-wrote will help researchers understand these issues more deeply.
- In particular, the tutorial overviewed:
  - Evaluation Metrics
  - Statistical Tests
  - Re-sampling methods
  - Data set selection criteria
  - Available resources
  - Some recent research from our group

If you need help, advice, etc…

- Please, contact me at:

nat@site.uottawa.ca

- Thank you for attending the workshop!!!

References

- Too many to put down here, but pp. 393-402 of the book.
- If you want specific ones, come and see me at the end of the tutorial or send me an e-mail.