CS456: Machine Learning

Classifier Evaluation and a bit of learning theory

Jakramate Bootkrajang

Department of Computer Science
Chiang Mai University

Objectives

- To understand fundamental background of classifier learning
- To understand several classification performance measures
- To learn about best practice in classifier comparison

Outlines

- Empirical Risk Minimisation
- Performance measures
- Classifier comparison
- Test of significance

Formal view of classification task

- Given a set of features X and a set of labels Y,
- Let $X \times Y$ be a cartesian product of feature set and label set and $\mathcal{D}$ be a distribution over $X \times Y$
- A training data is a set of (features,label) pairs drawn independently and identically from this distribution $(\mathbf{x}, y) \sim \mathcal{D}$
- A classifier $f(\mathbf{x})$ with parameter $\mathbf{w}$ is trained using the training data so as to explain the relationship between $\mathbf{x}$ and y

Empirical risk

- Given a single data pair $(\mathbf{x}_i, y_i) \sim \mathcal{D}$, a classifier's loss can be calculated by $\mathbb{1}(f(\mathbf{x}_i) \neq y_i)$
- Nonetheless, we are more interested in the generalisation performance of $f()$; total loss on all possible $(\mathbf{x}, y) \sim \mathcal{D}$
- The total loss is defined as an expected loss (risk) $\mathbb{E}_{\mathcal{D}}[\mathbb{1}(f(\mathbf{x}_i) \neq y_i)]$ over $\mathcal{D}$
- Since $\mathcal{D}$ is unknown, risk cannot be computed. But we can approximate the risk with empirical risk defined as $\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(f(\mathbf{x}_i) \neq y_i)$

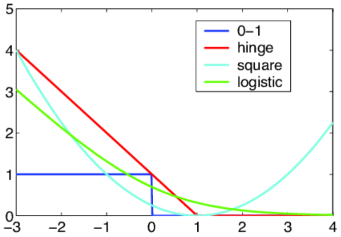
Empirical Risk Minimisation (ERM) philosophy

- Assumption: if training data is representative of data distribution, classifier which does well based on empirical risk should do well on data from this distribution
- This philosophy is fundamental to almost all machine learning algorithm

Minimising Empirical Risk

- $\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(f(\mathbf{x}_i) \neq y_i)$, 0-1 loss in empirical risk is easy to compute but difficult to minimise

- Instead, minimise approximated version of 0-1 loss (surrogate loss)
  - logistic loss (a.k.a binary cross entropy) in LR
  - hinge loss in SVM
  - cross entropy in MLP



Computing empirical risk

- Once learning is completed, we measure actual empirical risk

$$err = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(f(\mathbf{x}_i) \neq y_i) \tag{1}$$

- The risk is often referred to as error rate, $err \in [0, 1]$
- its inverse is classification accuracy: $acc = 1 - err$

Empirical Risk Minimisation caveat

- In practice, empirical risk is computed based on the training data (because it's the only data we have)
- The risk can be biased towards training data and therefore is a poor estimate of the true risk ($\mathbb{E}_{\mathcal{D}}[\mathbb{1}(f(\mathbf{x}_i) \neq y_i)]$)

True risk estimation

- There are several ways to *better* estimate the true risk
- The idea is to calculate empirical risk on a set of unseen data
- Popular examples are
  - Hold-out method
  - Cross validation

Hold out method

- Hold random $p\%$ of training data for empirical risk estimation
- Can be repeated several times

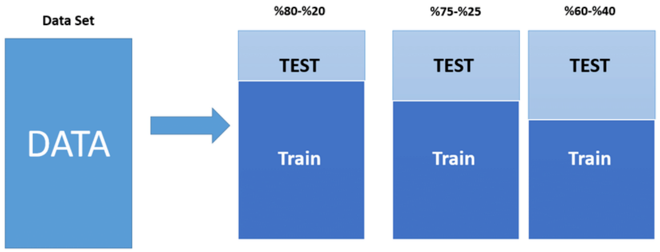


Figure: F.Kayaalp : Open Source Data Mining Programs: A Case Study on R

Cross validation method

- Divide training data into k sets, and repeat training/testing using each of the k sets

- Model performance is the average of k interations

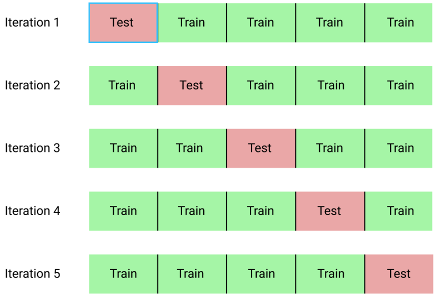


Figure:

`https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85`

Classifier comparison

which one would you choose ?

	training error	validation error
classifier a	0.90	0.79
classifier b	0.85	0.81

Generalisation performance

- We want classifier which gives best generalisation performance
- generalisation performance = good on unseen data
- idea: compare errors on validation set (test set)

Test errors

If we were to use 5-fold cross validation, there will be 5 test errors, which one to compare ?

Average of Test errors

We can compare the average of test errors

	average training error	average validation error
classifier a	0.90	0.79
classifier b	0.85	0.81

- which classifier is better ?

Deviations of error

- To decide which classifier is better we need to see the deviation of errors in each fold
- The deviation can be summarised using stadard deviation or standard error (s.d/number of folds)

Deviations of error example

which classifier is better ?

	average training error $\pm$ s.d.	average validation error $\pm$ s.d.
classifier a	0.90 ± 0.010	0.79 ± 0.010
classifier b	0.85 ± 0.005	0.81 ± 0.005

Deviations of error, another example

which classifier is better ?

	average training error $\pm$ s.d.	average validation error $\pm$ s.d.
classifier a	0.90 ± 0.010	0.79 ± 0.010
classifier b	0.85 ± 0.15	0.81 ± 0.1

Statistical tests

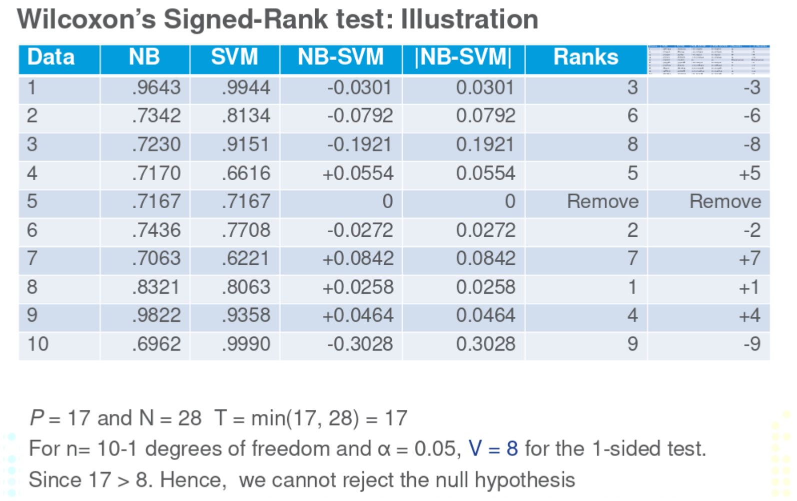
- To better compare the classifiers, one may employ statistical test
- Commonly used method = Wilcoxon's ranksum test
- Null hypothesis = two classifiers have similar performance

Wilcoxon's ranksum test

Steps

- Calculate performance difference between 2 classifier on each fold
- Rank absolute differences and note the sign in front of the ranks
- Compute sum of positive ranks P and sum of negative ranks N and $T = \min(P, N)$
- Reject null hypothesis if $T < V_\alpha$ where α is critical value

Example



Wilcoxon's Signed-Rank test: Illustration

Data	NB	SVM	NB-SVM	|NB-SVM|	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

P = 17 and N = 28 T = min(17, 28) = 17

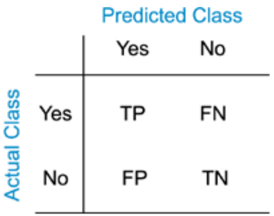
For n= 10-1 degrees of freedom and α = 0.05, V = 8 for the 1-sided test.

Since 17 > 8. Hence, we cannot reject the null hypothesis

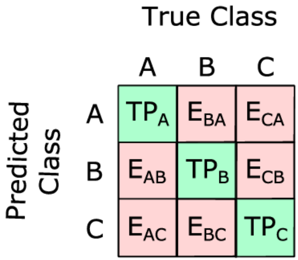
Figure: Mohak Shah and Nathalie Japkowicz, Performance Evaluation of Machine Learning Algorithms

Other performance measures

Confusion matrix summarises model's performance in details



		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN



		True Class		
		A	B	C
Predicted Class	A	TP_A	E_{BA}	E_{CA}
	B	E_{AB}	TP_B	E_{CB}
	C	E_{AC}	E_{BC}	TP_C

Cell naming convention: [Is prediction correct ?][Type of prediction]

True Positive: Prediction is “positive” and it was correct

Performance measures from confusion matrix

- For general classification task (every class is equally important)
  - Use error defined as $acc = \frac{FP+FN}{TP+TN+FP+FN}$
  - or accuracy defined as $acc = \frac{TP+TN}{TP+TN+FP+FN}$
- For detection (classification with focus on one class)
  - Use precision defined as $prec = \frac{TP}{TP+FP}$
  - Use recall defined as $recall = \frac{TP}{TP+FN}$

Precision and Recall at the airport

- Precision: (ratio of) passengers over people that were let in
- Recall: (ratio of) passengers over all passengers in the airport

Precision affects Recall (and vice-versa)

- Increasing precision often ...... recall
- Increasing recall often ...... precision
- Depending on the task, we might need to find perfect balance between precision and recall
- F_1-score summarises precision and recall in single number
  $F_1 = \frac{2}{recall^{-1} + prec^{-1}}$

Objectives: revisited

- To understand fundamental background of classifier learning
- To understand several classification performance measures
- To learn about best practice in classifier comparison