# Evaluation Methodology

Ljupčo Todorovski

Department of Knowledge Technologies
Jožef Stefan Institute
http://www-ai.ijs.si/~ljupco/

# Motivation

- evaluating performance of models
  - predictive error (most common)
  - complexity, comprehensibility, ...
- in order to perform tasks such as
  - **model selection**
    choose the best model
  - **model comparison**
    test how significant are differences
  - **model assesment**
    performance on new (future/unseen) data

# Talk Outline

- predictive error/accuracy
  - how to estimate it?
  - bias-variance trade-off
  - comparison of models
- different settings/tasks
  - predicting probabilities
  - misclassification costs
  - regression
- other criteria
  - complexity, comprehensibility

# Basic Notation

- Y – target variable
  - numeric: regression task
  - discrete: classification task
- X – vector of input variables
- D – data set consisting of (x,y) pairs
- unknown function f(X): $Y = f(X) + \varepsilon$
  - $\varepsilon$ – intrinsic target noise
- prediction model f*(X)
- prediction $Y^* = f^*(X)$

1. predictive error (accuracy)

# Loss Function

- loss function measures the error btw.
  - Y – measured/observed target value
  - f*(X) – predicted target value
- classification models
  - 0-1 loss: $L(Y, f^*(X)) = freq(Y \neq f^*(X))$
  - log-likelihood (later)
- regression models
  - squared error: $L(Y, f^*(X)) = (Y - f^*(X))^2$
  - absolute error: $L(Y, f^*(X)) = |Y - f^*(X)|$

# Predictive Error (Accuracy)

- “true” predictive error
  - expected value of the loss function
  - over the whole population

$$Error(f^*) = E[L(Y, f^*(X))]$$

- for 0-1 loss function (classification)
  - the error is between 0 and 1
  - $Accuracy(f^*) = 1 - Error(f^*)$
- How to estimate Error(f*)?

# Sample Error

- sample predictive error
  - average loss over a data sample S
  - consisting of N examples $(x_i, y_i)$

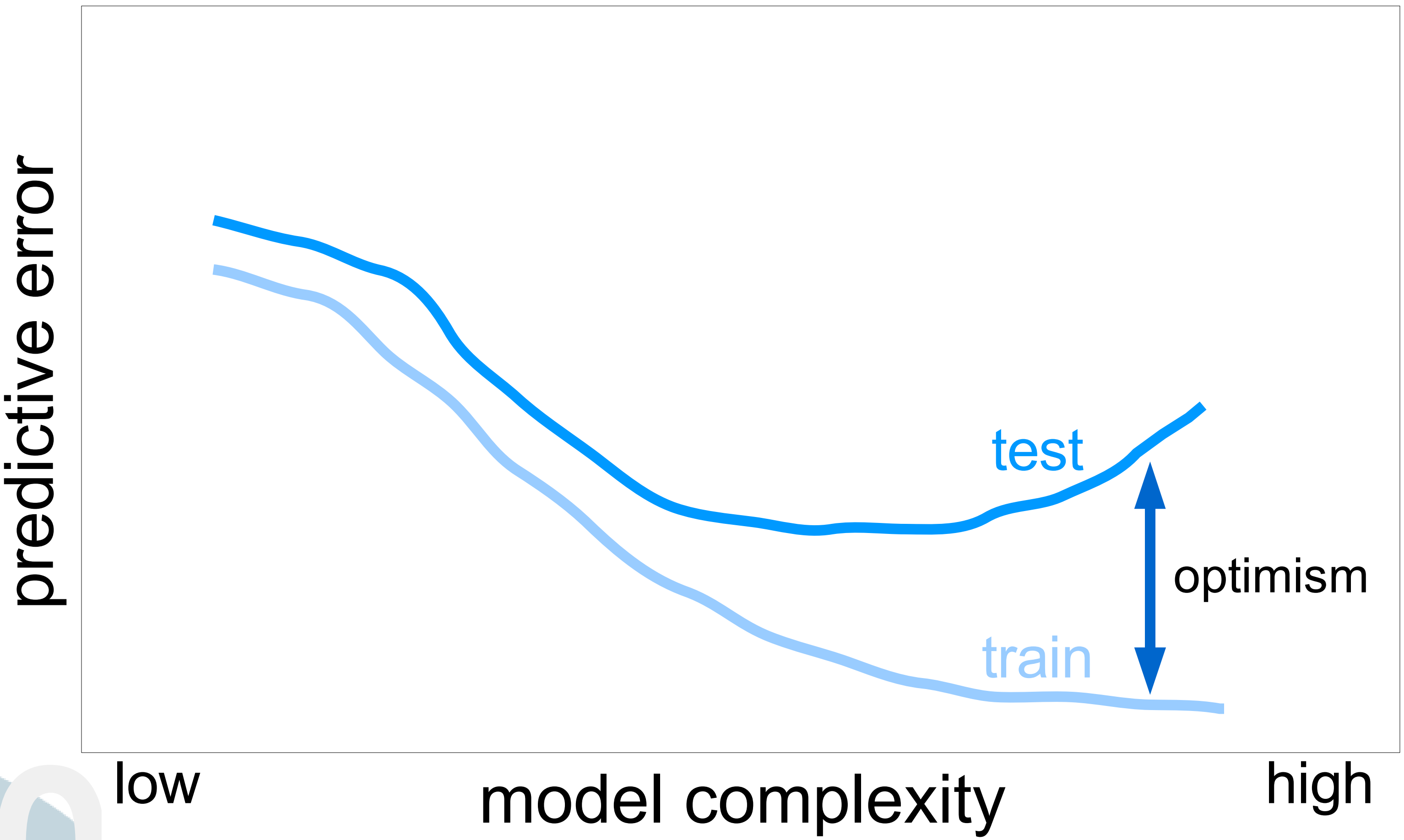
$$\mathrm{Error}_S(f^*) = 1/N \cdot \sum_{(x_i,y_i)\in S} L(y_i, f^*(x_i))$$

- training error
  - error estimated on training data sample
- testing error
  - error estimated on test (unseen) data

# Training vs. Test Error (1)

- common mistake
  - estimate error on train data only
  - resubstitution error
  - too optimistic (lower error)
  - do not reveal the behavior of the model on new (unseen/future) data
- correct approach
  - estimate error on test data
  - unseen in training phase
- WHY IS THIS SO?

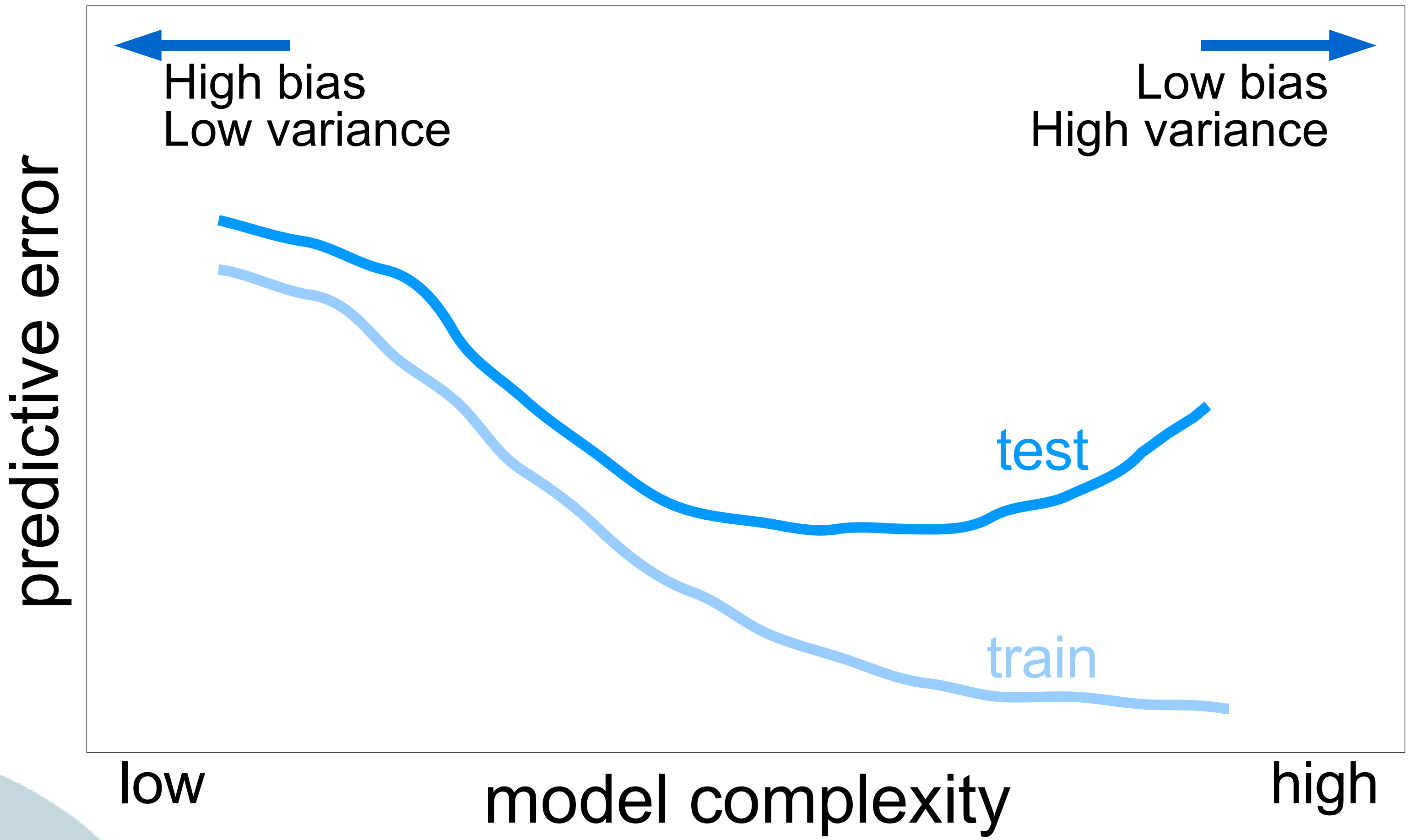
# Training vs. Test Error (2)




Based on Figure 7.1 from the book The Elements of Statistical Learning

# 2. bias-variance trade-off

# Bias-Variance (B-V) Trade-Off




Based on Figure 7.1 from the book The Elements of Statistical Learning

# B-V Decomposition (1)

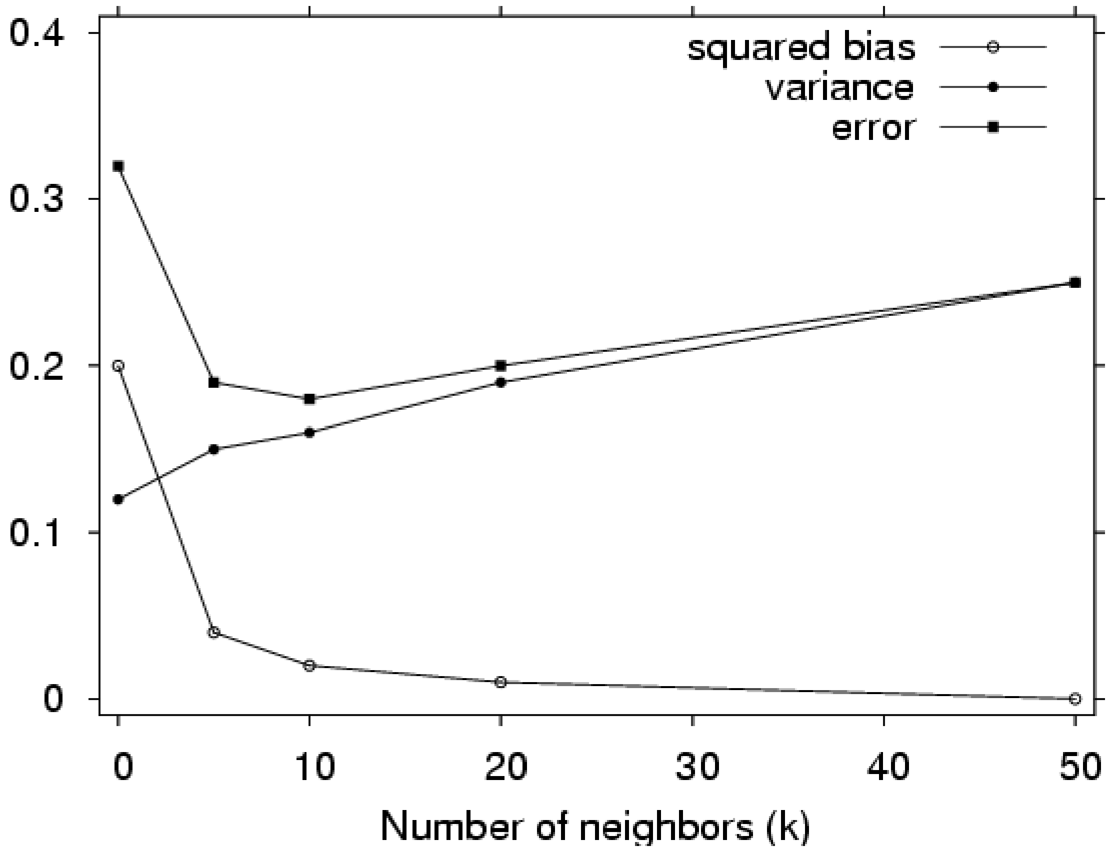
- Error(x)
  $= E[(y - f^*(x))^2]$
  $= E[(y - f(x) + f(x) - f^*(x))^2]$
  $= E[\varepsilon^2] + E[(f(x) - f^*(x))^2]$
  $= E[\varepsilon^2] + E[(f(x) – Ef^*(x) + Ef^*(x) – f^*(X))^2]$
  $=$ noise + bias$^2$ + variance

- bias$^2 = E[(f(x) – Ef^*(x))^2]$
- variance $= E[(f^*(x) – Ef^*(x))^2]$

# B-V Decomposition (2)

- intrinsic target noise
- bias term
  - measures how close the average model produced by a particular learning algorithm will be to the target function
- variance term
  - measures how models produced by a learning algorithm vary

# B-V: An Example




Based on Figure 7.3 from the book The Elements of Statistical Learning

# B-V Decomposition: Methods

- empirical B-V decomposition
  - on an arbitrary data set
  - performed by multiple runs of an algorithm
  - on different data samples
- description of methods (further reading):
  - squared loss function [Geman et al. 1992]
  - 0-1 loss function [Kohavi and Wolpert 1996]
  - unified [Domingos 2000]

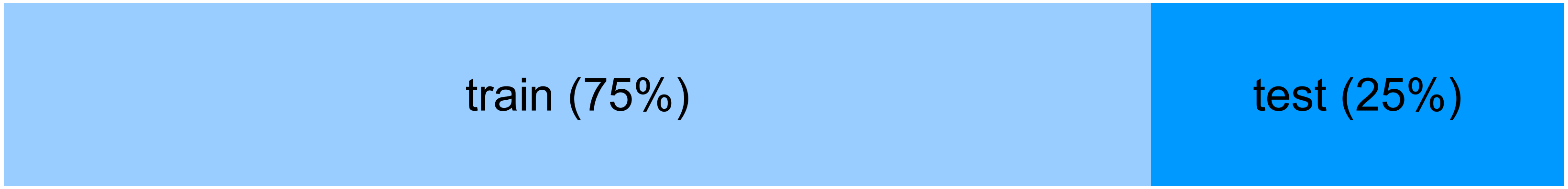
# 3. estimating predictive error

# Data Supply Problems

- all data samples
  - should be large (representative) enough
  - training: obtaining better model
  - test: obtaining better error estimate
- however, in real applications
  - amount of data limited
  - due to practical problems
- usual solution: holdout procedure
  - keep some data out of training sample
  - for testing purposes

# Holdout Procedures (Typical)

- model assessment




- model selection and assessment

# Holdout Estimates: Reliability

- how reliable is the holdout estimate
  - we estimated error rate of 30%
  - (1) on a test sample of 1000 examples
  - (2) on a test sample of 40 examples
  - which is more reliable/confident?
- confidence intervals
- with 95% probability the error lies in
  - (1) interval [30%-3%, 30%+3%] = [27%,33%]
  - (2) interval [30%-14%, 30%+14%] = [16%,44%]

# Confidence Intervals

- different methods for calculating them
  - based on Bernoulli Processes
  - see further reading
- Weka Book
  - Section 5.2
  - Predicting Performance
- ML Book
  - Section 5.2.2
  - Confidence Intervals for Discrete-Valued Hypotheses

# How to Improve Reliability?

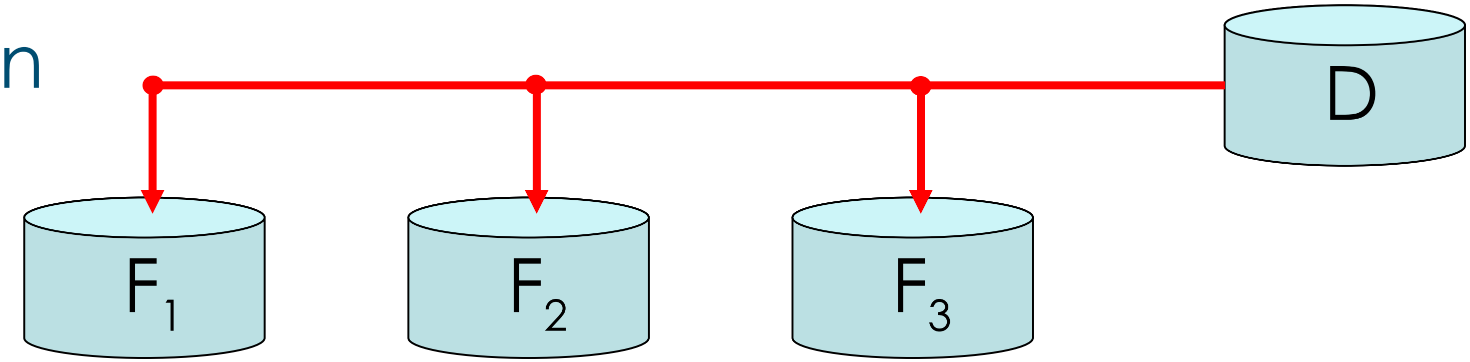
- repetitive holdout estimates
  - instead of running a single holdout
  - repeat it number of times
  - average the estimates obtained

- how to split into train/test samples?
  - cross validation (CV)
  - leave-one-out (special case of CV)
  - bootstrap sampling

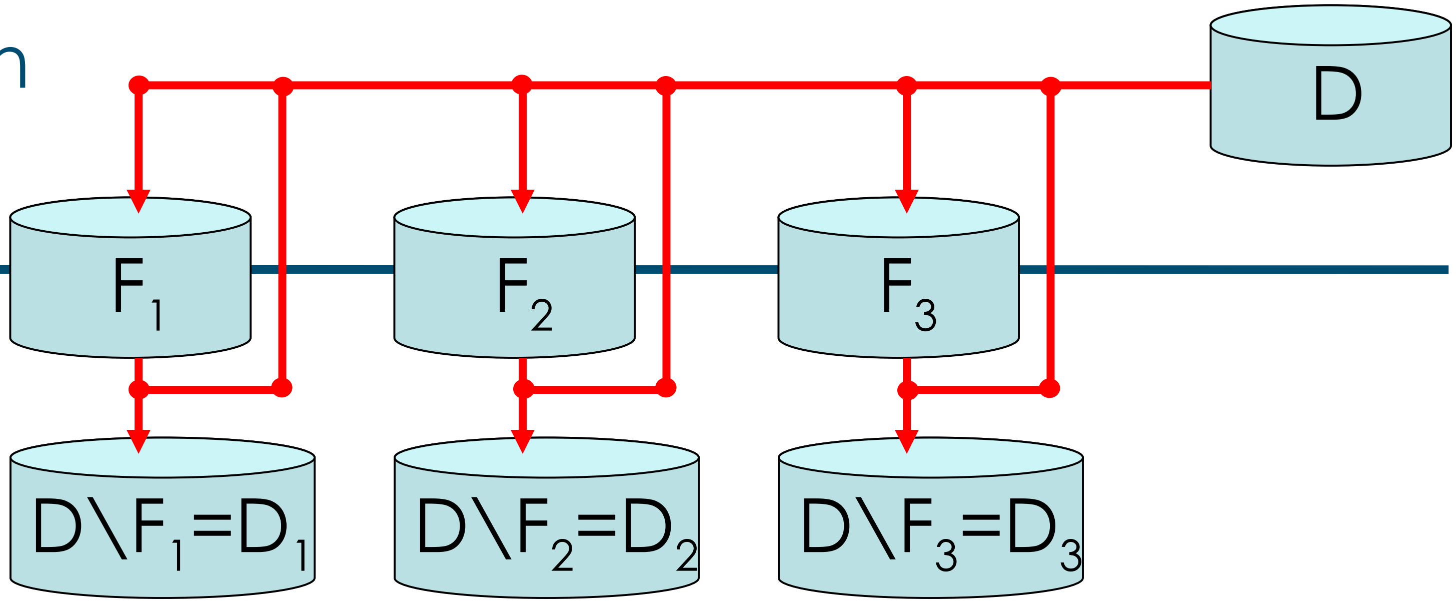
# Cross Validation (CV)

- three steps: partition, train, and test
- **partition**
  - randomly into k folds ($F_1$, $F_2$, ... $F_k$)
- repeat k times (once for each $F_i$)
  - **train** on $D \setminus F_i$
  - **test** (estimate sample error) on $F_i$
- average error estimates

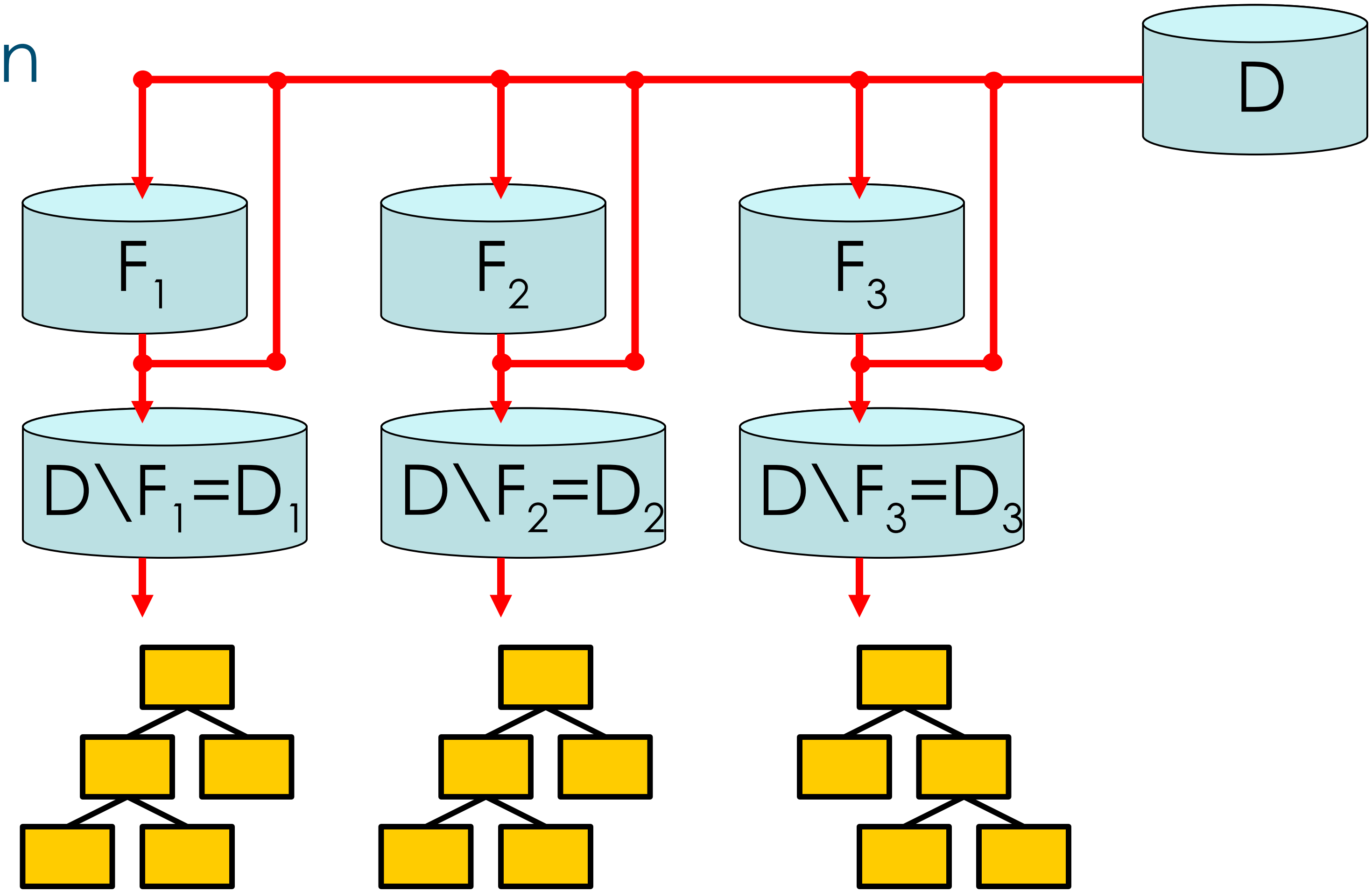
- Partition



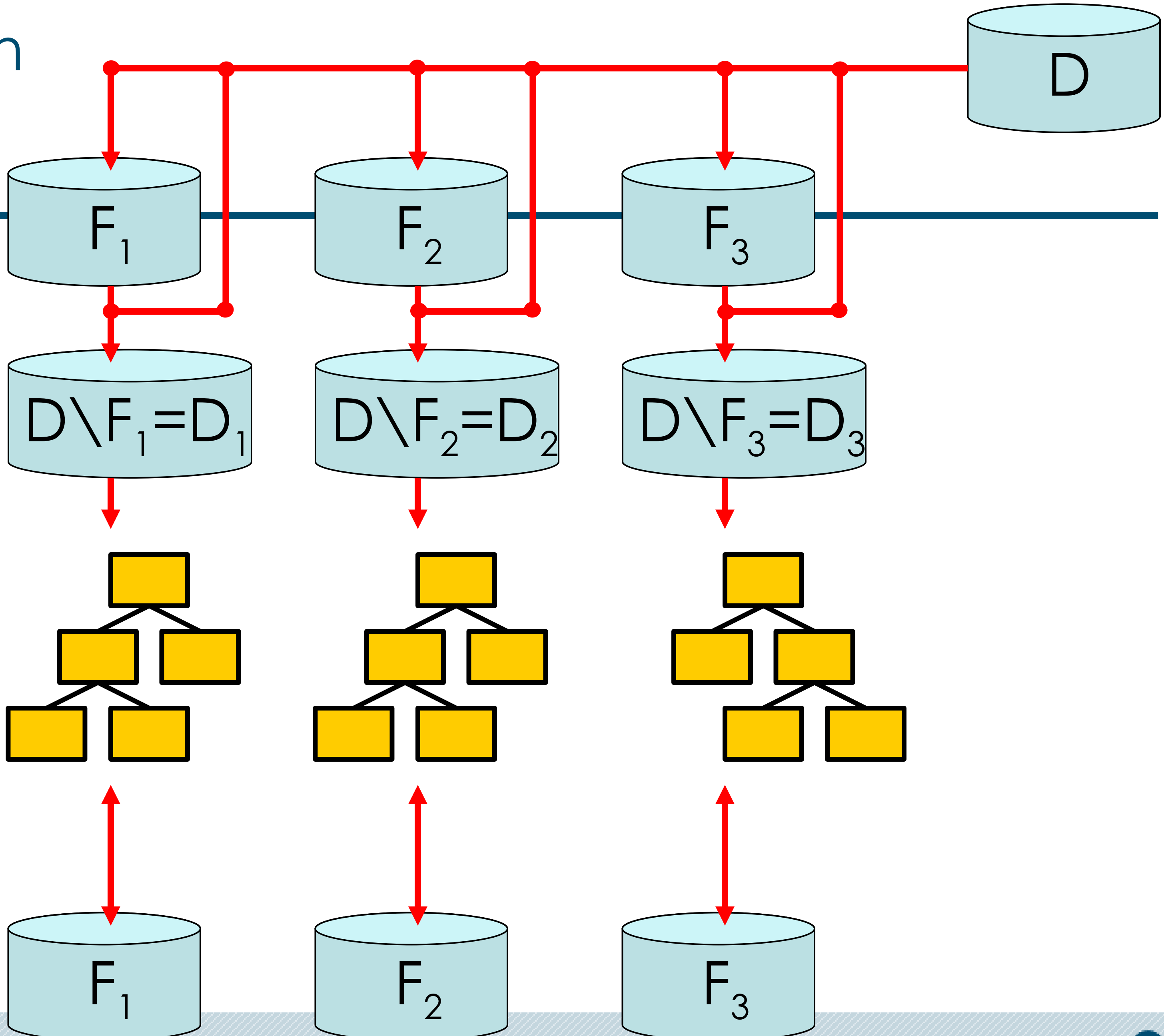
- Partition
- Train



- Partition
- Train



- Partition
- Train
- Test



# CV: Number of Folds

- large number of folds:
  - training sets very similar to each other
  - high variance of the estimate
  - maximal number of folds N: **leave-one-out**
  - illustrate high variance on an example
- small number of folds:
  - lower variance, but
  - training set might be too small
- recommended compromise: 5 or 10!

# CV: Stratification

- folds sampling not completely random
  - "due to bad luck" we can end-up with non-representative data sample
  - distribution of target variable values vary
- stratified sampling
  - each fold has similar distribution of target variable values
- different stratification methods for
  - classification (similar distributions)
  - regression (similar average values)

# Bootstrap Sampling

- three steps: sample, train and test
  - **sample** N examples from D with **replacement** (an example can be used more than once)
  - **train** on the (multi)set of sampled examples S
  - **test** (estimate sample error) on D\S

- number of distinct training examples
  - 0.632·N (see ESL or Weka Book)
  - comparable to 2-fold CV: pessimistic estimate
  - combine estimated test error ($Error_{D\setminus S}$) with the training error ($Error_S$)

$$Error_{0.632} = 0.632 \cdot Error_{D\setminus S} + 0.368 \cdot Error_S$$

# Alternatives to Sampling

- in-sample estimates
  - $Error_{TEST} = Error_{TRAIN} + Optimism$
  - problem reduced to estimating “optimism”
- several in-sample estimates
  - Akaike information criterion (AIC)
  - Bayesian information criterion (BIC)
  - Minimum description length (MDL)
  - further details in the ESL book

# MDL Principle

- the best model is the one that minimizes
  - the model size
  - the amount of information necessary to encode model errors
  - i.e., information necessary to reconstruct training data
- model estimate thus is a sum of
  - model size: L(M)
  - training data D w.r.t. M: L(D|M)
- coding method important

# 4. comparing predictive errors

# Paired t-test

- perform CV for both models ($M_1$, $M_2$)
  - on same $k$ data folds $F_1$, $F_2$, ... $F_k$
  - obtain estimates $Error_{Fi}(M_1)$ and $Error_{Fi}(M_2)$
  - calculate $Diff_i = Error_{Fi}(M_1) - Error_{Fi}(M_2)$
  - t-statistic $t = mean(Diff) / sqrt(var(Diff)/k)$
- calculated t-statistic
  - follows Student's distribution
  - with $k-1$ degrees of freedom
  - see ML or Weka Book for details

# Non-Paired t-test

- allows for comparison with models
  - estimated using different CV folds
  - or even different number of CV folds
- Different estimate of var(Diff) needed
  - see Weka book for details

# Comparison: Open Issue

- comparing models on limited data
  - is still an open issue
- ongoing research work focus on
  - criticism of existing methods [Bengio and Grandvalet 2004]
  - comparing existing and proposing new alternatives [Diettrich 1998; Bouckaert 2004]

# 5. different settings/tasks

# Predicting Probabilities (1)

- predicting distribution of Y values
  - instead of predicting Y value itself
  - example: weather forecast (sunny/rainy)
  - prediction: sunny – 75%, rainy – 25%
- 0-1 loss function not good
  - wrong prediction with 55% probability
  - is better than
  - wrong prediction with 75% probability
  - different loss function needed

# Predicting Probabilities (2)

- Notation:
  - $p_j$ – predicted probability of j-th value of Y
  - $p_k$ – predicted probability of actual Y value
  - $a_j$ – actual probability of j-th value of Y
    - Note that only $a_k = 1$, rest are 0
- alternative loss-functions
  - quadratic

$$L(Y,p^*(X)) = \sum_j (a_j - p_j)^2 = 1 - 2\,p_k + \sum_j p_j^2$$

  - log-likelihood

$$L(Y,p^*(X)) = -2 \sum_j a_j \cdot \log(p_j) = -2 \log(p_k)$$

# Errors of Regression Models

- mean squared error (MSE) correspond to
  - squared error loss function
  - $L(Y, f^*(X)) = (Y - f^*(X))^2$
- commonly used RMSE = sqrt(MSE)
- mean absolute error correspond to
  - absolute error loss function
  - $L(Y, f^*(X)) = |Y - f^*(X)|$
- these error measures are scale dependent

# Relative and Scale Independent Errors

- relative squared error (RSE)
  - RSE = MSE / var(Y)
  - error relative to the error of the simplest predictor (predicting mean(Y))
  - RSE value greater than 1 (one) means that the predictor performs worse than simplest
  - comparable across domains
- correlation coefficient ($r^2$)
  - scale independent
  - see Weka book

# Misclassification Costs

- binary classification problem
- two kind of errors
  - false positive
    negative example predicted as positive
  - false negative
    positive example predicted as negative
- different costs assigned to each
  - examples: loan decisions, diagnosis

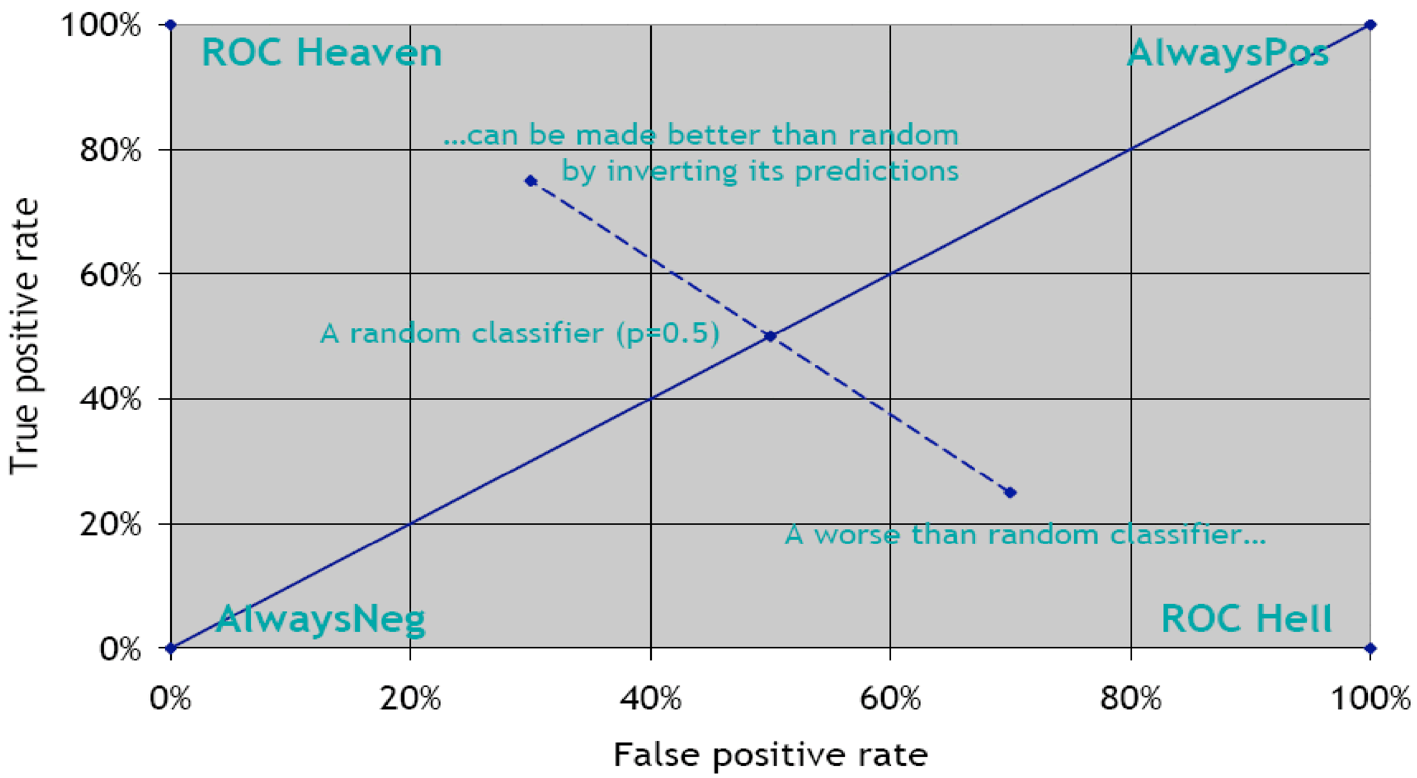
# Confusion Matrix

| | predicted class | |
|---|---|---|
| actual class | yes | no |
| yes | true positives | false negatives |
| no | false positives | true negatives |

- Error = (FP + FN) / N
- Accuracy = (TP + TN) / N

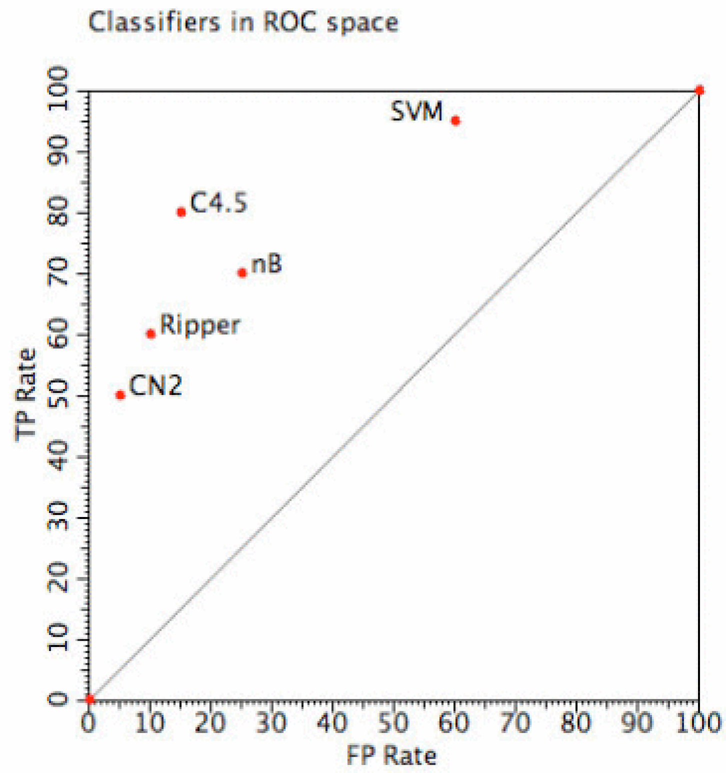
- TPrate = Recall = TP / (TP + FN)
- FPrate = FP / (FP + TN)

# ROC Space



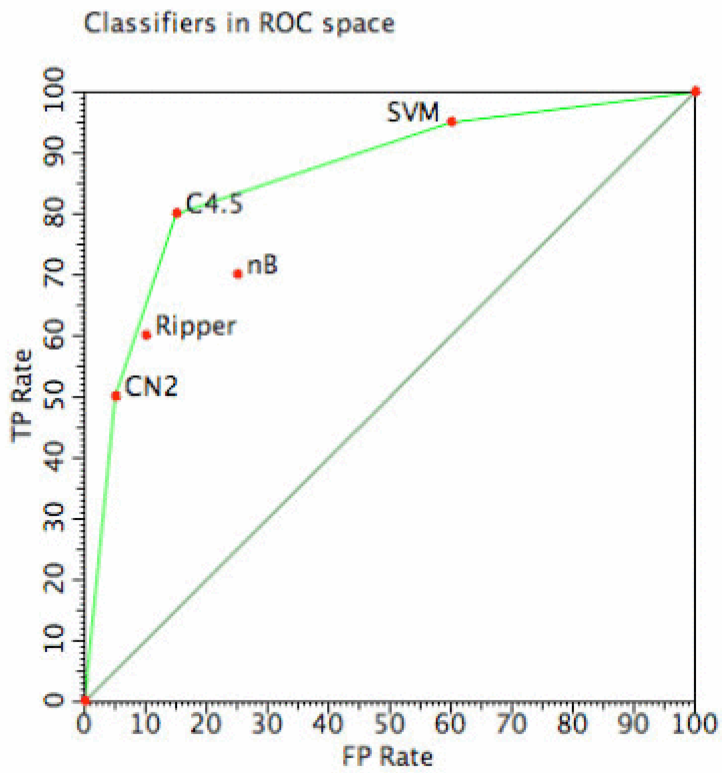
100%
80%
60%
40%
20%
0%
True positive rate
ROC Heaven
AlwaysPos
...can be made better than random
by inverting its predictions
A random classifier (p=0.5)
A worse than random classifier...
AlwaysNeg
ROC Hell
0%
20%
40%
60%
80%
100%
False positive rate

# ROC Plot



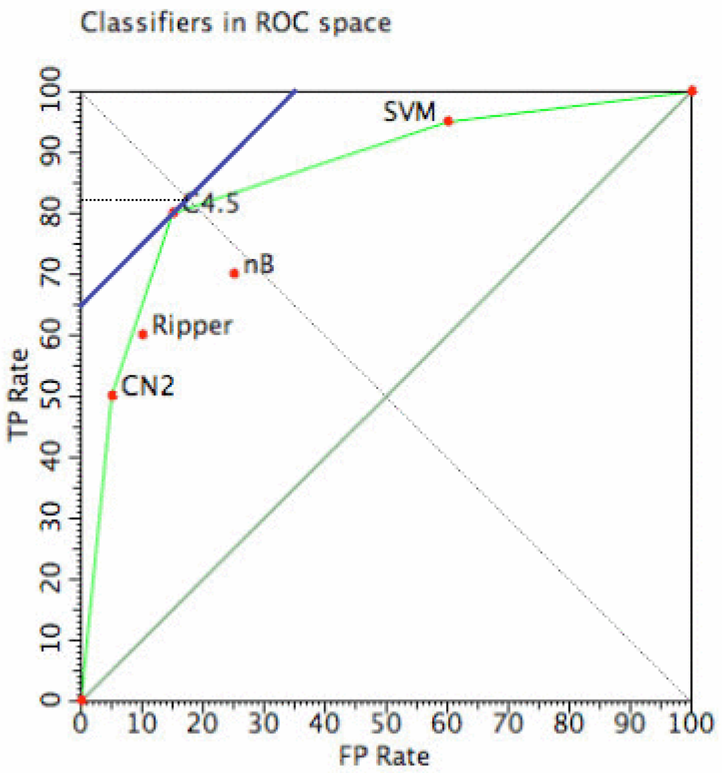
Classifiers in ROC space
SVM
C4.5
nB
Ripper
CN2
TP Rate
FP Rate
0
10
20
30
40
50
60
70
80
90
100

# ROC Convex Hull



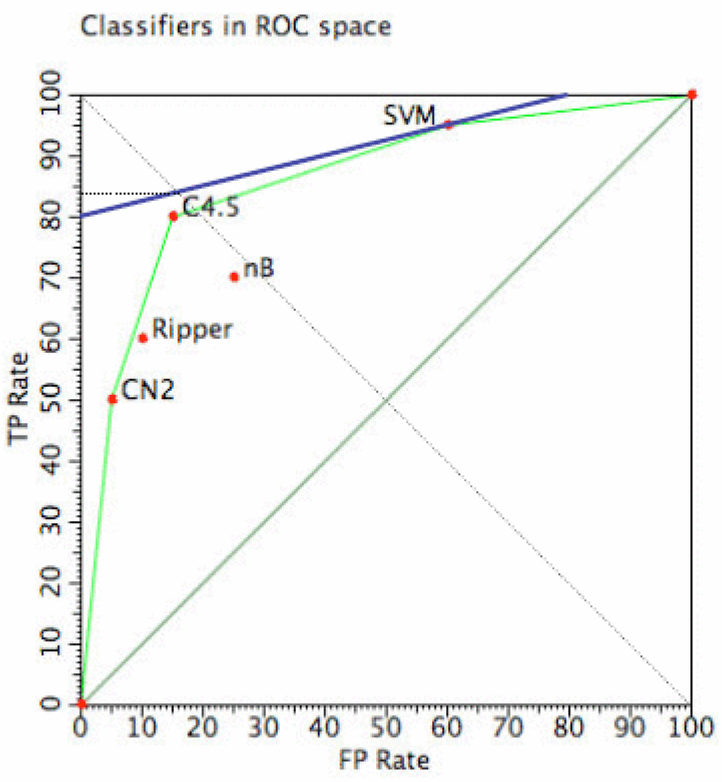

- classifiers on the CH achieve best accuracy for some class distributions
- classifiers not on the CH are always suboptimal

# Optimal Classifier (1)




- C4.5 optimal for uniform class distribution (slope of the blue line)
- Accuracy: 82%

# Optimal Classifier (2)




- SVM optimal for class distribution where we have 4 times as many positives as negatives (slope of the blue line)
- Accuracy: 84%

# Incorporating Costs

- for skewed class distribution
  - slope equals neg/pos
- for misclassification costs
  - slope equals (neg*C(+/-))/(pos*C(-/+))
- further details
  - [Provost and Fawcett 2001]
  - [Flach 2003]

# 6. other performance measures

# Model Complexity

- many different measures
  - model dependent
- decision trees
  - number of nodes, parameters in leaf nodes
- decision rules
  - number of rules, literals, coverage
- in general
  - number of parameters
  - encoding length (MDL like)

# Model Comprehensibility

- difficult to assess
  - most methods involve manual work
  - can not be fully automated
- tests
  - can human expert understand the model?
  - can he/she use it for manual prediction?
  - how well?
- roughly related
  - rule interestigness [Fuernkranz and Flach 05]

# 7. further reading

# Further Reading: Books

- Weka Book
  I.H.Witten and E.Frank (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann. [Chapter 5].

- ML Book
  T.M.Mitchell (1997) *Machine Learning*. McGraw-Hill. [Chapter 5].

- ESL Book
  T.Hastie, R. Tibshirani, and J. Friedman (2001) *The Elements of Statistical Learning*. Springer-Verlag. [Chapter 7].

# Further Reading: Articles (1)


- Y.Bengio and Y.Grandvalet (2004) No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research* 5: 1089-1105.

- R.R.Bouckaert (2004) Estimating Replicability of Classifier Learning Experiments. In *Proceedings of Twenty-First International Conference on Machine Learning*.

- T.Dietterich (1998) Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* 10(7): 1895-1924.

# Further Reading: Articles (2)


- P.Domingos (2000) A unified bias-variance decomposition and its applications. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, pages 231-238.

- S.Geman, G.Beinenstock, and R.Doursat (1992) Neural networks and the bias/variance dilemma. *Neural Computation* 4: 1-58.

- R. Kohavi and D.H.Wolpert (1996) Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on Machine Learning (IMCL-1996)*, pages 275-283.

# Further Reading: Articles (3)


- P.A.Flach (2003) The geometry of ROC space. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 194-201.

- J.Fuernkranz and P.A.Flach (2005) ROC'n'Rule learning – towards a better understanding of covering algorithms. *Machine Learning* 58(1): 39-77.

- F.J.Provost and T.Fawcett (2001) Robust classification analysis for performance evaluation. *Machine Learning* 42(3): 203-231.