Applied inductive learning

Bias/variance tradeoff, Model assessment and selection

Pierre Geurts
Department of Electrical Engineering and Computer Science
University of Liège

October 29, 2012

Supervised learning

- Given
  - A learning sample of *N* input-output pairs

    $$LS = \{(x_i, y_i) | i = 1, \ldots, N\} \text{ with } x_i \in \mathcal{X}, y_i \in \mathcal{Y}$$

    independently and identically drawn (i.i.d.) from a (unknown) distribution $p(x, y)$
  - A loss function $L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ measuring the discrepancy between its arguments

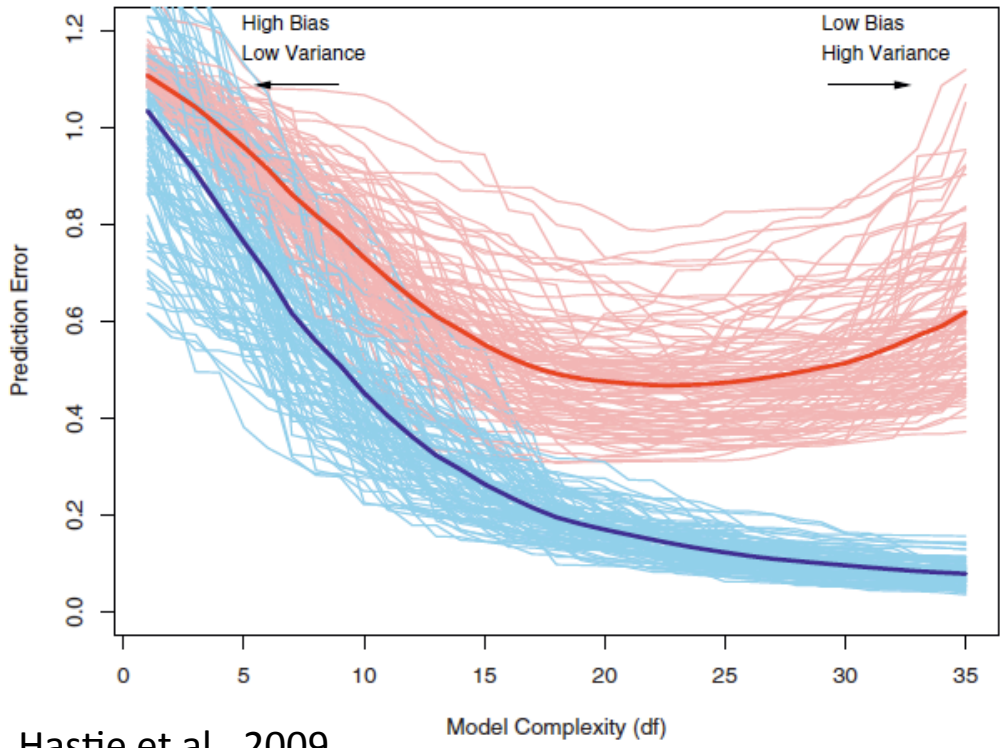
Find a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the following expectation (*generalization error*):

$$E_{x,y}\{L(y, f(x))\}$$

- Classification: $L(y, y') = 1(y = y')$ (error rate)
- Regression: $L(y, y') = (y - y')^2$ (square error)

LS randomness

- Let us denote by $\hat{f}_{LS}$ the function learned from a learning sample LS by a learning algorithm
- The function $\hat{f}_{LS}$ (its prediction at some point) is a random variable



Hastie et al., 2009

Error on LS (blue) and Error on TS (red) for 100 different learning samples

Two quantities of interest

- Given a model $\hat{f}_{LS}$ built from some learning sample, its *generalization error*:

$$\mathrm{Err}_{LS} = E_{x,y}\{L(y, \hat{f}_{LS}(x))\}$$

$\Rightarrow$ useful for model assessment and selection

- Given a learning algorithm, its *expected generalization error* over random *LS* of size *N*:

$$E_{LS}\{\mathrm{Err}_{LS}\} = E_{LS}\{E_{x,y}\{L(y, \hat{f}_{LS}(x))\}\}$$

$\Rightarrow$ useful to characterize a learning algorithm

Outline

- Bias/variance tradeoff
  - Decomposition of the expected error $E_{LS}\{\mathrm{Err}_{LS}\}$ that helps to understand overfitting
- Performance evaluation
  - Procedures to estimate Err_{LS} (or $E_{LS}\{\mathrm{Err}_{LS}\}$)
- Performance measure
  - Common loss functions L for classification and regression

Outline

- Bias/variance tradeoff
  - Bias and variance definitions
  - Parameters that influence bias and variance
  - Bias and variance reduction techniques

Outline

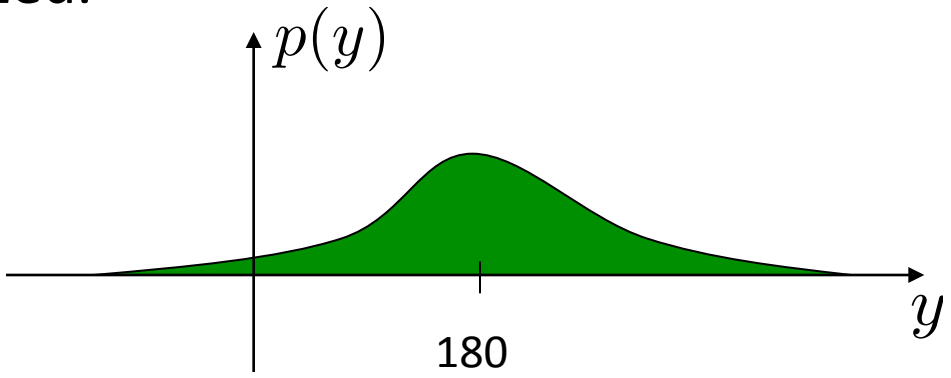
- Bias and variance definitions:
  - A simple regression problem with no input
  - Generalization to full regression problems
  - A short discussion about classification
- Parameters that influence bias and variance
- Bias and variance reduction techniques

Regression problem - no input

- Goal: predict as well as possible the height of a Belgian male adult
- More precisely:
  - Choose an error measure, for example the square error.
  - Find an estimation $\hat{y}$ such that the expectation:

$$E_y\{(y-\hat{y})^2\}$$

over the whole population of Belgian male adult is minimized.



Regression problem - no input

- The estimation that minimizes the error can be computed by taking:

$$\frac{\partial}{\partial y'} E_y\{(y - y')^2\} = 0$$
$$\Leftrightarrow \quad E_y\{-2(y - y')\} = 0$$
$$\Leftrightarrow \quad E_y\{y\} - E_y\{y'\} = 0$$
$$\Leftrightarrow \quad y' = E_y\{y\}$$

- So, the estimation which minimizes the error is $E_y\{y\}$. In AL, it is called the Bayes model.
- **But** in practice, we cannot compute the exact value of $E_y\{y\}$ (this would imply to measure the height of every Belgian male adults).

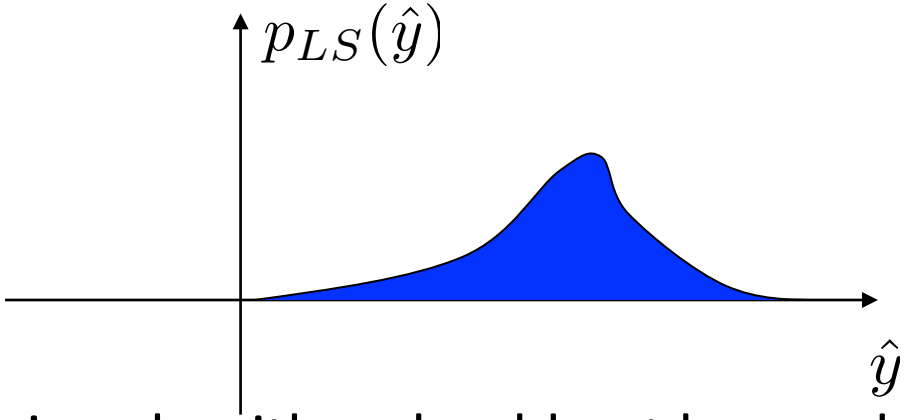
Learning algorithm

- As $p(y)$ is unknown, find an estimation $\hat{y}$ from a sample of individuals, $LS = \{y_1, y_2, \ldots, y_N\}$, (independently) drawn from the Belgian male adult population.
- Example of learning algorithms:
  - $\hat{y}_1 = \frac{1}{N} \sum_{i=1}^{N} y_i$
  - $\hat{y}_2 = \frac{\lambda 180 + \sum_{i=1}^{N} y_i}{\lambda + N}, \lambda \in [0, +\infty[$

    (if we know that the height is close to 180)

Good learning algorithm

- As *LS* are randomly drawn, the prediction $\hat{y}$ will also be a random variable



- A good learning algorithm should not be good only on one learning sample but in average over all learning samples (of size *N*) $\Rightarrow$ we want to minimize:

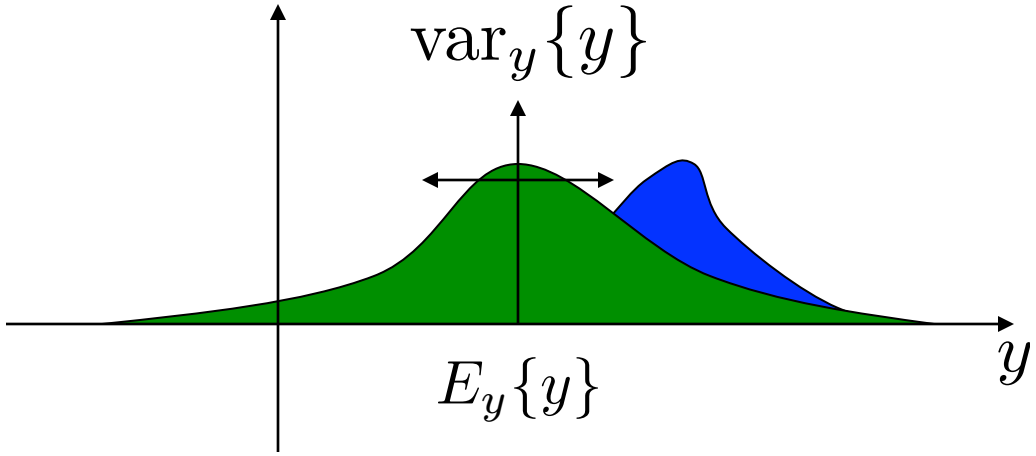
$$E = E_{LS}\{E_y\{(y - \hat{y})^2\}\}$$

- Let us analyse this error in more detail

Bias/variance decomposition (1)

$$
\begin{aligned}
& E_{LS}\{E_y\{(y-\hat{y})^2\} \\
= \quad & E_{LS}\{E_y\{(y-E_y\{y\}+E_y\{y\}-\hat{y})^2\}\} \\
= \quad & E_{LS}\{E_y\{(y-E_y\{y\})^2\}\} + E_{LS}\{E_y\{(E_y\{y\}-\hat{y})^2\}\} \\
+ \quad & E_{LS}\{E_y\{2(y-E_y\{y\})(E_y\{y\}-\hat{y})\}\} \\
= \quad & E_y\{(y-E_y\{y\})^2\} + E_{LS}\{(E_y\{y\}-\hat{y})^2\} \\
+ \quad & E_{LS}\{2(E_y\{y\}-E_y\{y\})(E_y\{y\}-\hat{y})\}\} \\
= \quad & E_y\{(y-E_y\{y\})^2\} + E_{LS}\{(E_y\{y\}-\hat{y})^2\}
\end{aligned}
$$

Bias/variance decomposition (2)



$$E = \underbrace{E_y\{(y - E_y\{y\})^2\}} + E_{LS}\{(E_y\{y\} - \hat{y})^2\}$$

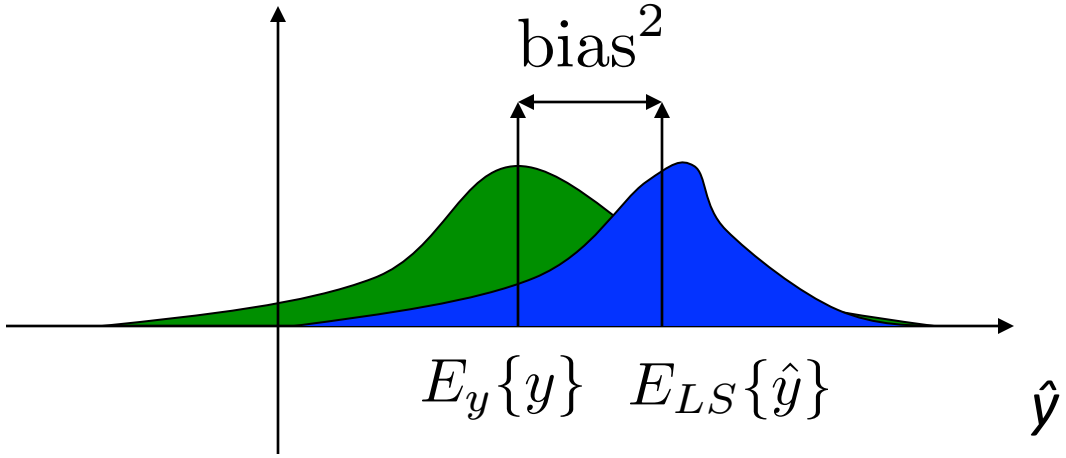
= residual error = minimal attainable error

= $\mathrm{var}_y\{y\}$

Bias/variance decomposition (3)

$$
\begin{aligned}
& E_{LS}\{(E_y\{y\} - \hat{y})^2\} \\
= \quad & E_{LS}\{(E_y\{y\} - E_{LS}\{\hat{y}\} + E_{LS}\{\hat{y}\} - \hat{y})^2\} \\
= \quad & E_{LS}\{(E_y\{y\} - E_{LS}\{\hat{y}\})^2\} + E_{LS}\{(E_{LS}\{\hat{y}\} - \hat{y})^2\} \\
+ \quad & E_{LS}\{2(E_y\{y\} - E_{LS}\{\hat{y}\})(E_{LS}\{\hat{y}\} - \hat{y})\} \\
= \quad & (E_y\{y\} - E_{LS}\{\hat{y}\})^2 + E_{LS}\{(\hat{y} - E_{LS}\{\hat{y}\})^2\} \\
+ \quad & 2(E_y\{y\} - E_{LS}\{\hat{y}\})(E_{LS}\{\hat{y}\} - E_{LS}\{\hat{y}\}) \\
= \quad & (E_y\{y\} - E_{LS}\{\hat{y}\})^2 + E_{LS}\{(\hat{y} - E_{LS}\{\hat{y}\})^2\}
\end{aligned}
$$

Bias/variance decomposition (4)

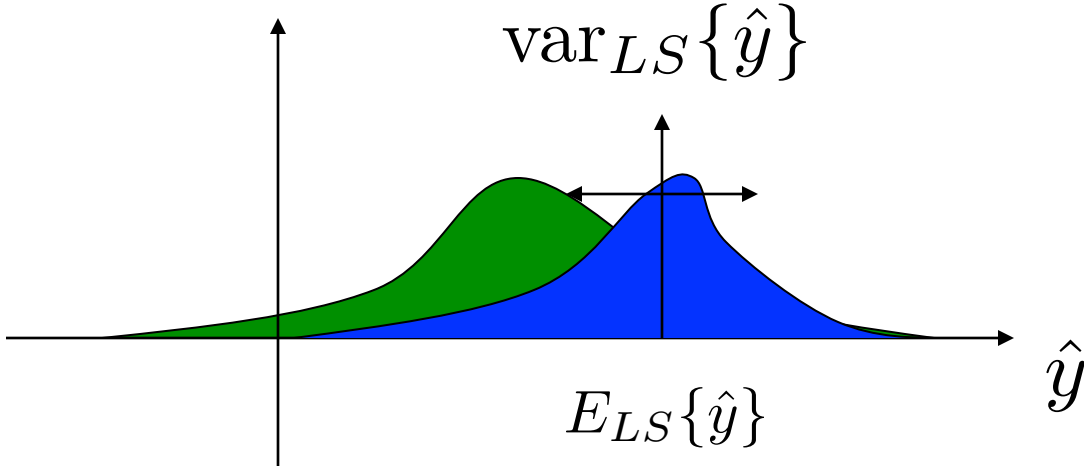


$$E = \text{var}_y\{y\} + (E_y\{y\} - E_{LS}\{\hat{y}\})^2 + \dots$$

$E_{LS}\{\hat{y}\}$ = average model (over all LS)

bias^2 = error between Bayes and average model

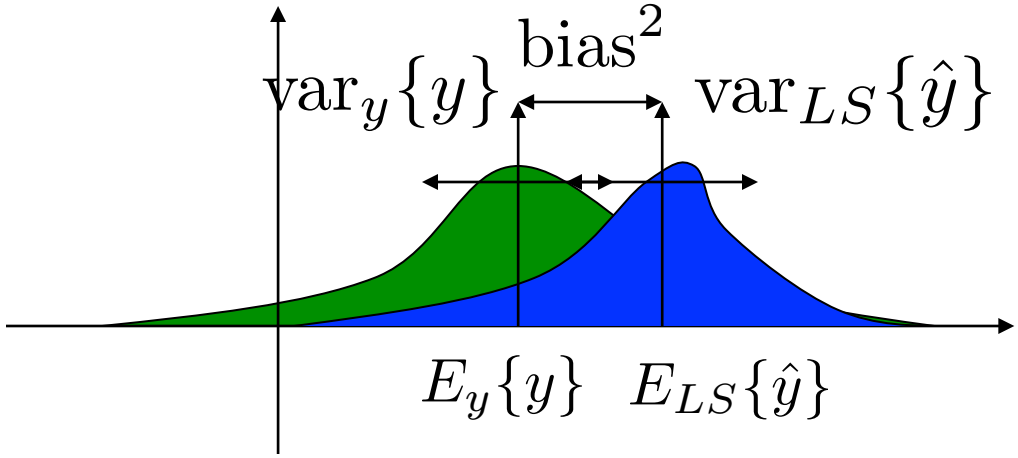
Bias/variance decomposition (5)



$$E = \mathrm{var}_y\{y\} + \mathrm{bias}^2 + E_{LS}\{(\hat{y} - E_{LS}\{\hat{y}\})^2\}$$

$\mathrm{var}_{LS}\{\hat{y}\}$ = estimation variance = consequence of over-fitting

Bias/variance decomposition (6)



$$E = \mathrm{var}_y\{y\} + \mathrm{bias}^2 + \mathrm{var}_{LS}\{\hat{y}\}$$

Our simple example

- $\hat{y}_1 = \frac{1}{N}\sum_{i=1}^{N} y_i$

$$\text{bias}^2 = (E_y\{y\} - E_{LS}\{\hat{y}_1\})^2 = 0$$

$$\text{var}_{LS}\{\hat{y}_1\} = \frac{1}{N}\text{var}_y\{y\}$$

From statistics, $\hat{y}_1$ is the best estimate with zero bias

- $\hat{y}_2 = \frac{\lambda 180 + \sum y_i}{\lambda + N}$

$$\text{bias}^2 = \left(\frac{\lambda}{\lambda + N}\right)^2 (E_y\{y\} - 180)^2$$

$$\text{var}_{LS}\{\hat{y}_2\} = \frac{N}{(\lambda + N)^2}\text{var}_y\{y\}$$

So, the first one may not be the best estimator because of variance (There is a bias/variance tradeoff w.r.t. λ)

Bayesian approach (1)

- Hypotheses :
  - The average height is close to 180cm:

$$P(\bar{y}) = A \exp(-\frac{(\bar{y} - 180)^2}{2\sigma_{\bar{y}}})$$

  - The height of one individual is Gaussian around the mean:

$$P(y_i|\bar{y}) = B \exp(-\frac{(y_i - \bar{y})^2}{2\sigma_y})$$

- What is the most probable value of $\bar{y}$ after having seen the learning sample ?

$$\hat{y} = \arg\max_{\bar{y}} P(\bar{y}|LS)$$

Bayesian approach (2)

$$
\begin{aligned}
\hat{y} &= \arg\max_{\bar{y}} P(\bar{y}|LS) \\
&= \arg\max_{\bar{y}} P(LS|\bar{y})P(\bar{y}) \\
&= \arg\max_{\bar{y}} P(y_1, \ldots, y_N|\bar{y})P(\bar{y}) \\
&= \arg\max_{\bar{y}} \prod_{i=1}^{N} P(y_i|\bar{y})P(\bar{y}) \\
&= \arg\min_{\bar{y}} -\sum_{i=1}^{N} \log(P(y_i|\bar{y})) - \log(P(\bar{y})) \\
&= \arg\min_{\bar{y}} \sum_{i=1}^{N} \frac{(y_i - \bar{y})^2}{2\sigma_y^2} + \frac{(\bar{y} - 180)^2}{2\sigma_{\bar{y}}^2} \\
&= \ldots \\
&= \frac{\lambda 180 + \sum_i y_i}{\lambda + N} \text{ with } \lambda = \frac{\sigma_y^2}{\sigma_{\bar{y}}^2}
\end{aligned}
$$

Bayes theorem and *P*(*LS*) is constant

Independence of the learning cases

Regression problem – full (1)

- Actually, we want to find a function $\hat{y}(\underline{x})$ of several inputs => average over the whole input space:

- The error becomes:

$$E_{\underline{x},y}\{(y - \hat{y}(\underline{x}))^2\}$$

- Over all learning sets:

$$\begin{aligned} E &= E_{LS}\{E_{\underline{x},y}\{(y - \hat{y}(\underline{x}))^2\}\} \\ &= E_{\underline{x}}\{E_{LS}\{E_{y|\underline{x}}\{(y - \hat{y}((\underline{x}))^2\}\}\} \\ &= E_{\underline{x}}\{\mathrm{var}_{y|\underline{x}}\{y\}\} + E_{\underline{x}}\{\mathrm{bias}^2(\underline{x})\} + E_{\underline{x}}\{\mathrm{var}_{LS}\{\hat{y}(\underline{x})\}\} \end{aligned}$$

Regression problem – full (2)

$$E_{LS}\{E_{y|\underline{x}}\{(y-\hat{y}(\underline{x}))^2\}\} = \text{noise}(\underline{x}) + \text{bias}^2(\underline{x}) + \text{variance}(\underline{x})$$

- $\text{noise}(\underline{x}) = E_{y|\underline{x}}\{(y - h_B(\underline{x}))^2\}$

  Quantifies how much y varies from $h_B(\underline{x}) = E_{y|\underline{x}}\{y\}$, the Bayes model.
- $\text{bias}^2(\underline{x}) = (h_B(\underline{x}) - E_{LS}\{\hat{y}(\underline{x})\})^2$

  Measures the error between the Bayes model and the average model.
- $\text{variance}(\underline{x}) = E_{LS}\{(\hat{y} - E_{LS}\{\hat{y}(\underline{x})\})^2\}$

  Quantify how much $\hat{y}(\underline{x})$ varies from one learning sample to another.

Illustration (1)

- Problem definition:
  - One input *x*, uniform random variable in [0,1]
  - $y = h(x) + \epsilon$ where $\epsilon \sim N(0,1)$

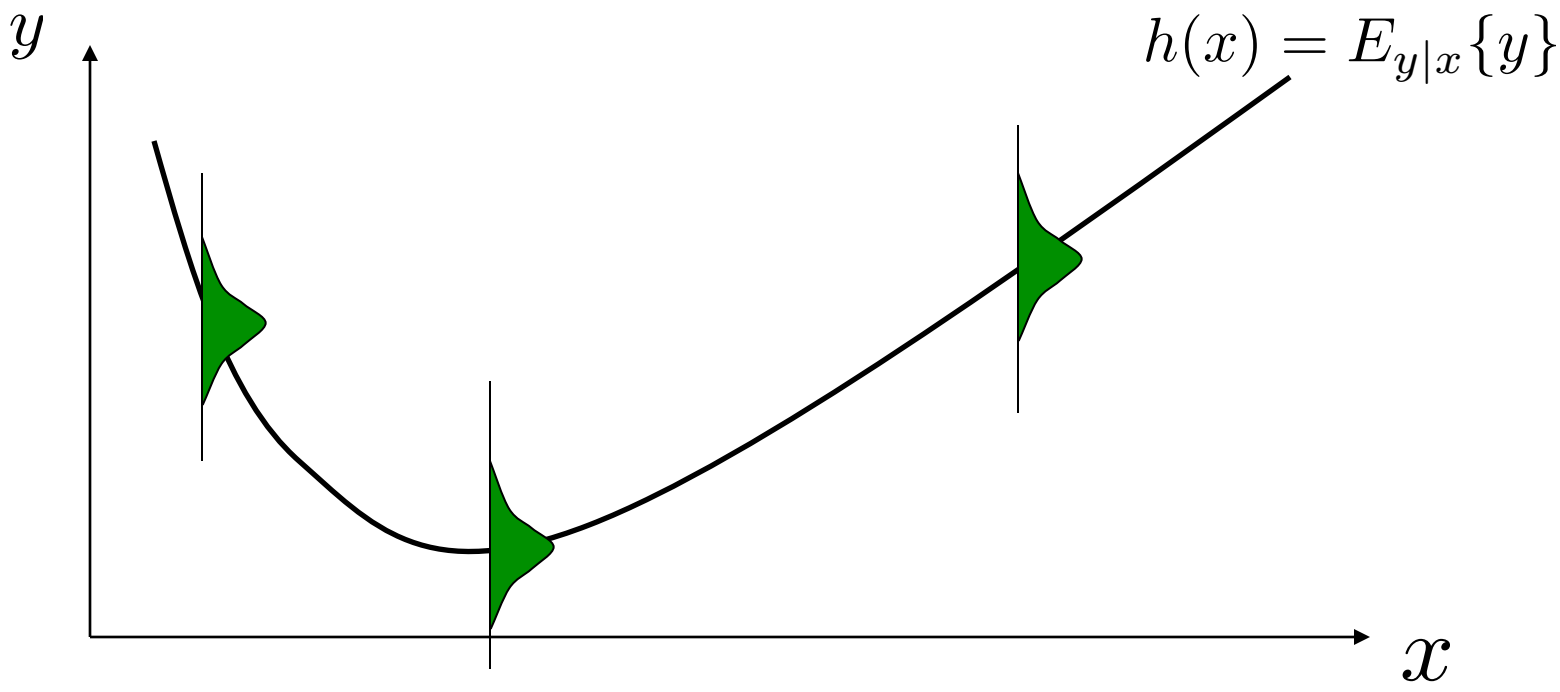


Illustration (2)

- Low variance, high bias method ⇒ underfitting

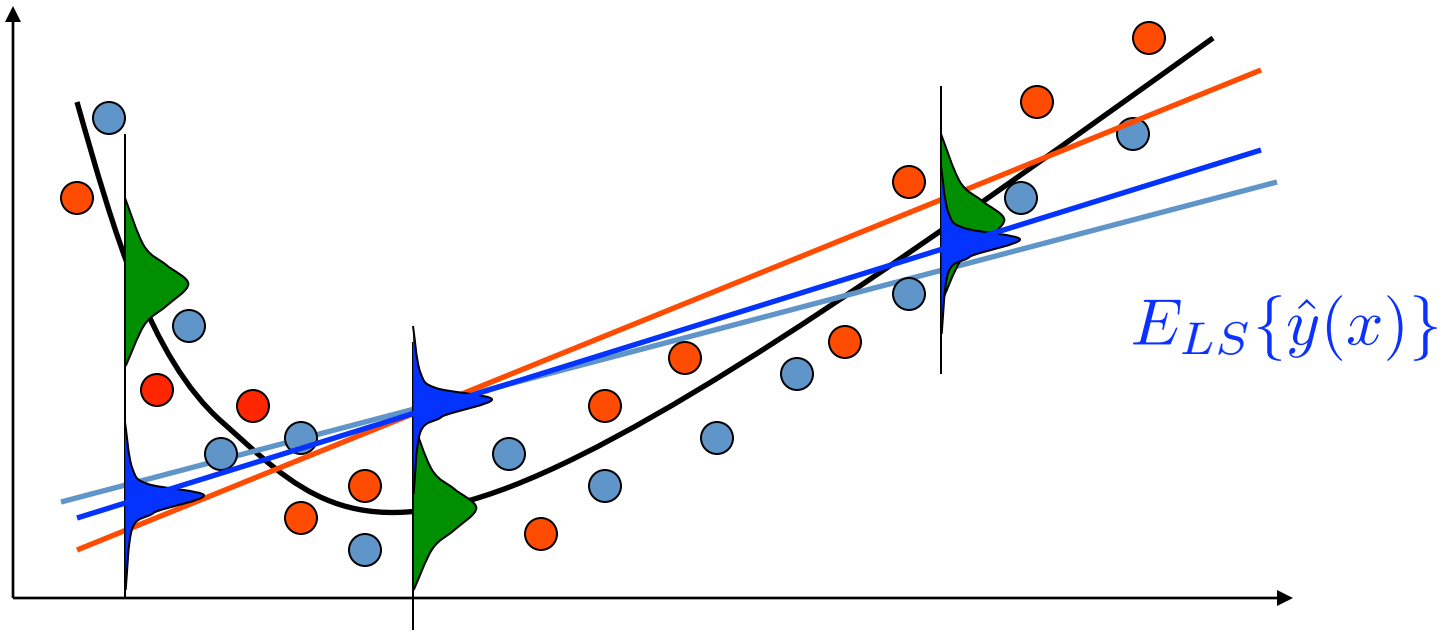


Illustration (3)

- Low bias, high variance method ⇒ overfitting

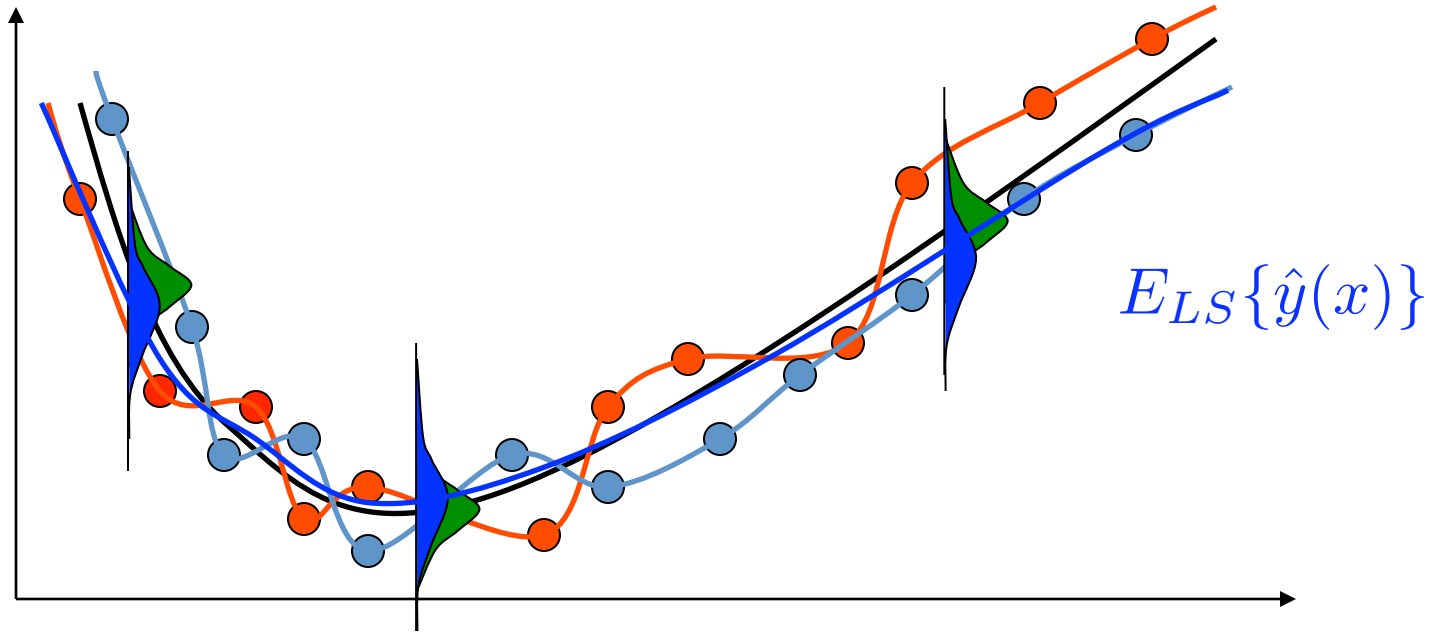
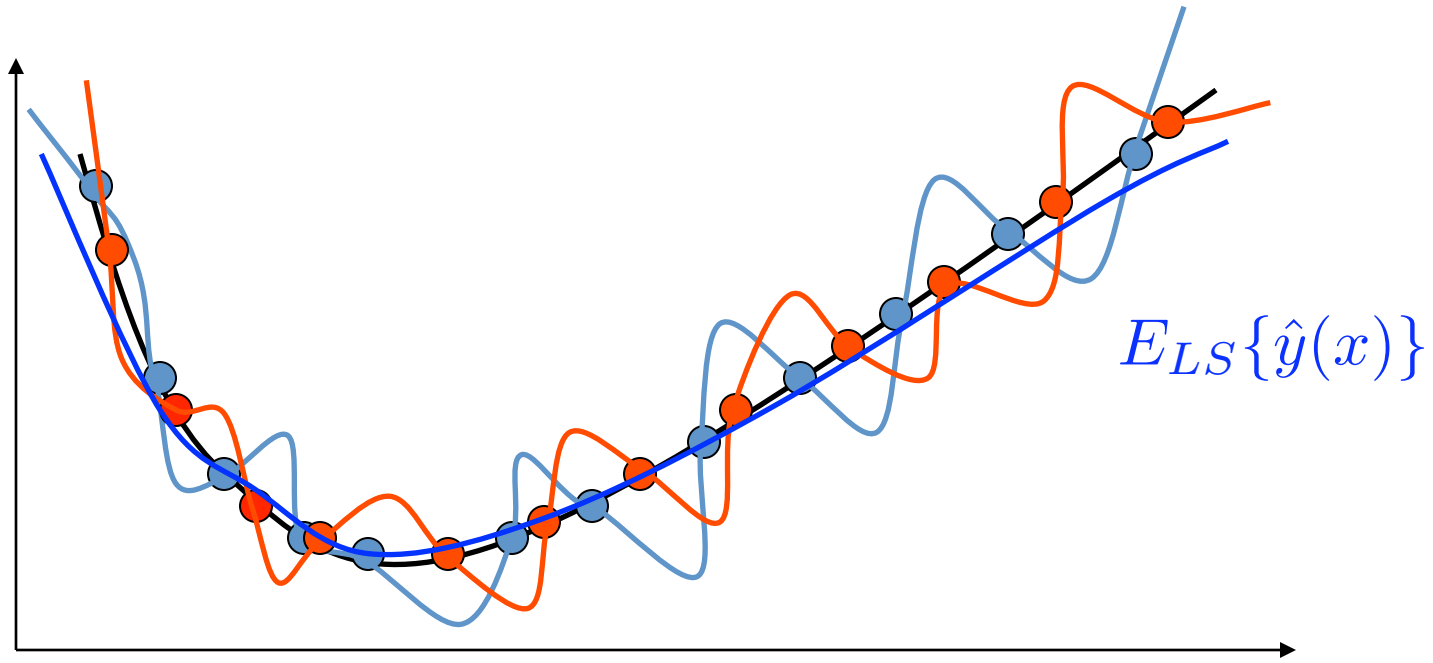


Illustration (4)

- No noise doesn't imply no variance (but less variance)



Classification problems (1)

- The mean misclassification error is:

$$E = E_{LS}\{E_{\underline{x},y}\{1(y \neq \hat{y}(\underline{x}))\}\}$$

- The best possible model is the Bayes model:

$$h_B(\underline{x}) = \arg\max_c P(y = c|\underline{x})$$

- The "average" model is:

$$\arg\max_c P(\hat{y}(\underline{x}) = c|\underline{x})$$

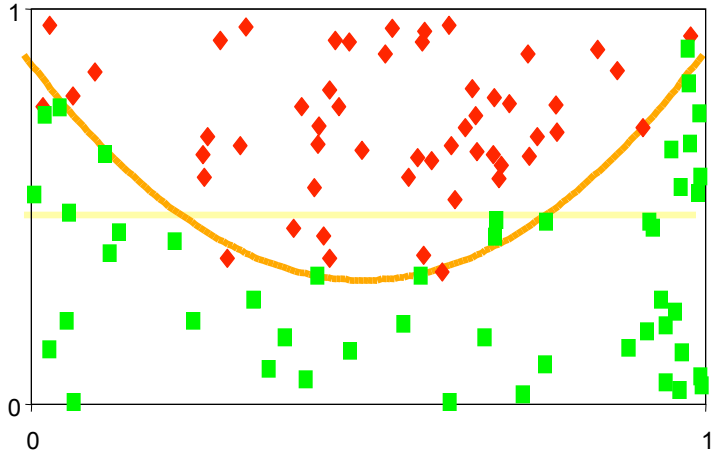
- Unfortunately, there is no such decomposition of the mean misclassification error into a bias and a variance terms.
- Nevertheless, we observe the same phenomena

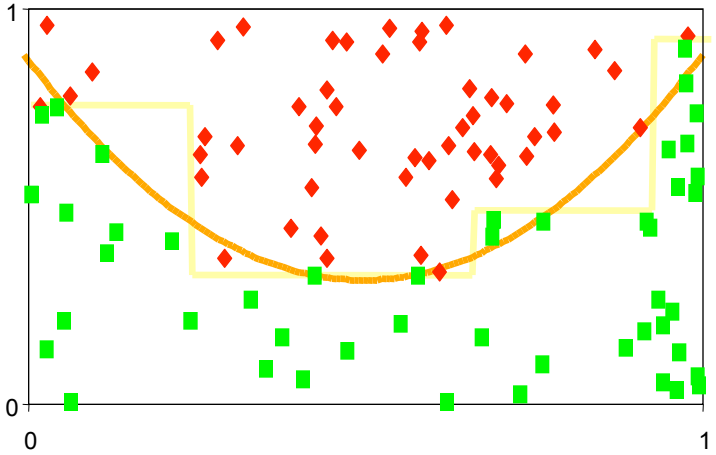
Classification problems (2)

One test node

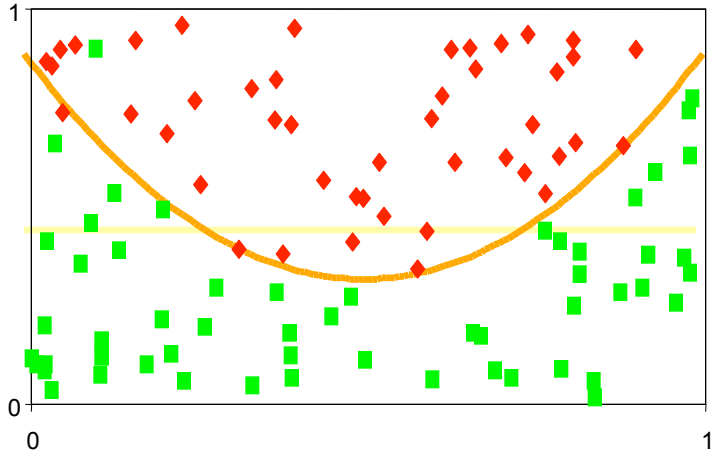
A full decision tree

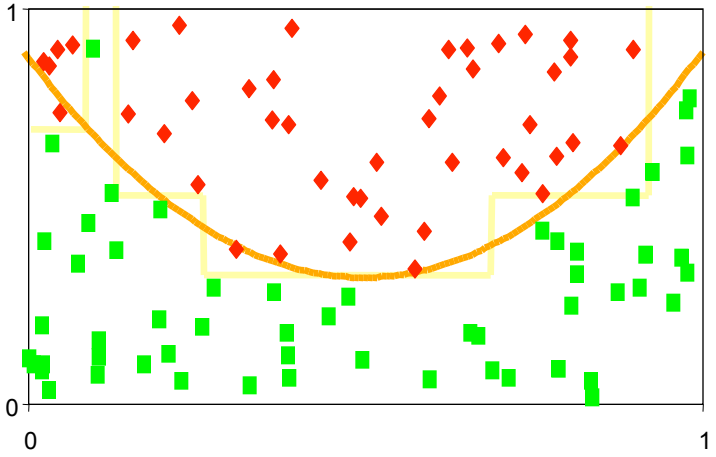
LS_1





LS_2

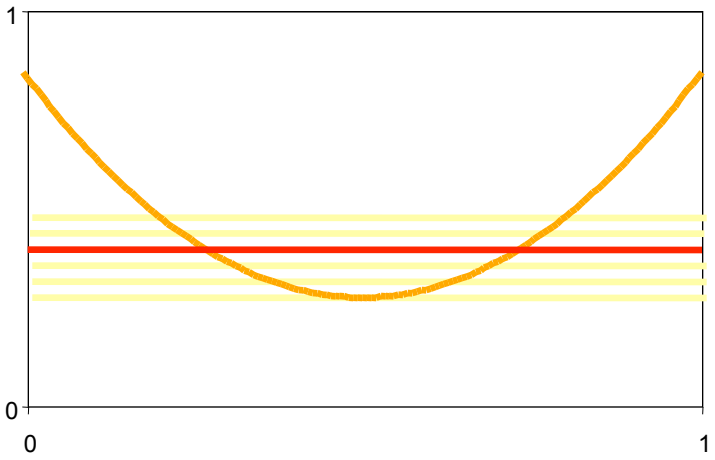




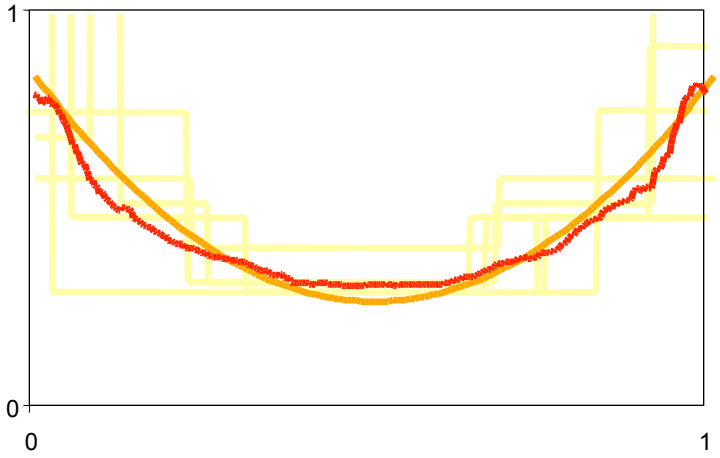
Classification problems (3)

- Bias ≈ systematic error component (independent of the learning sample)
- Variance ≈ error due to the variability of the model with respect to the learning sample randomness
- There are errors due to bias and errors due to variance

One test node



Full decision tree



Content of the presentation

- Bias and variance definitions
- Parameters that influence bias and variance
  - Complexity of the model
  - Complexity of the Bayes model
  - Noise
  - Learning sample size
  - Learning algorithm
- Bias and variance reduction techniques

Illustrative problem

- Artificial problem with 10 inputs, all uniform random variables in [0,1]
- The true function depends only on 5 inputs:

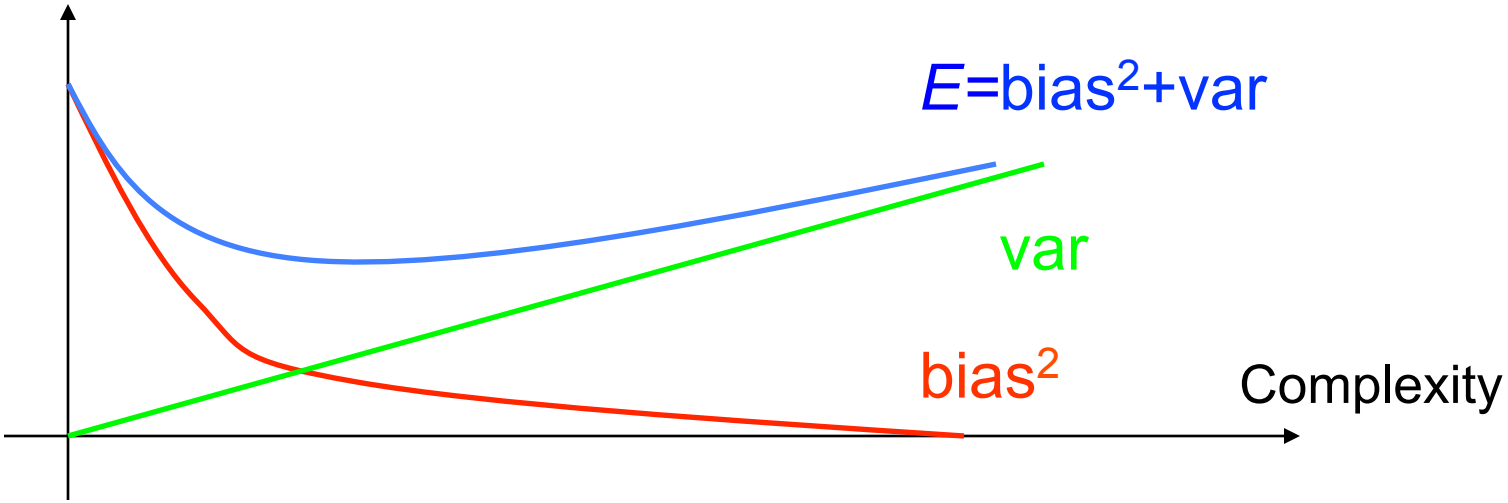
$$y(\underline{x}) = 10sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 6x_5 + \epsilon$$

where ϵ is a $N(0,1)$ random variable

- Experimentations:
  - $E_{LS} \Rightarrow$ average over 50 learning sets of size 500
  - $E_{\underline{x},y} \Rightarrow$ average over 2000 cases

  $\Rightarrow$ Estimate variance and bias (+ residual error)

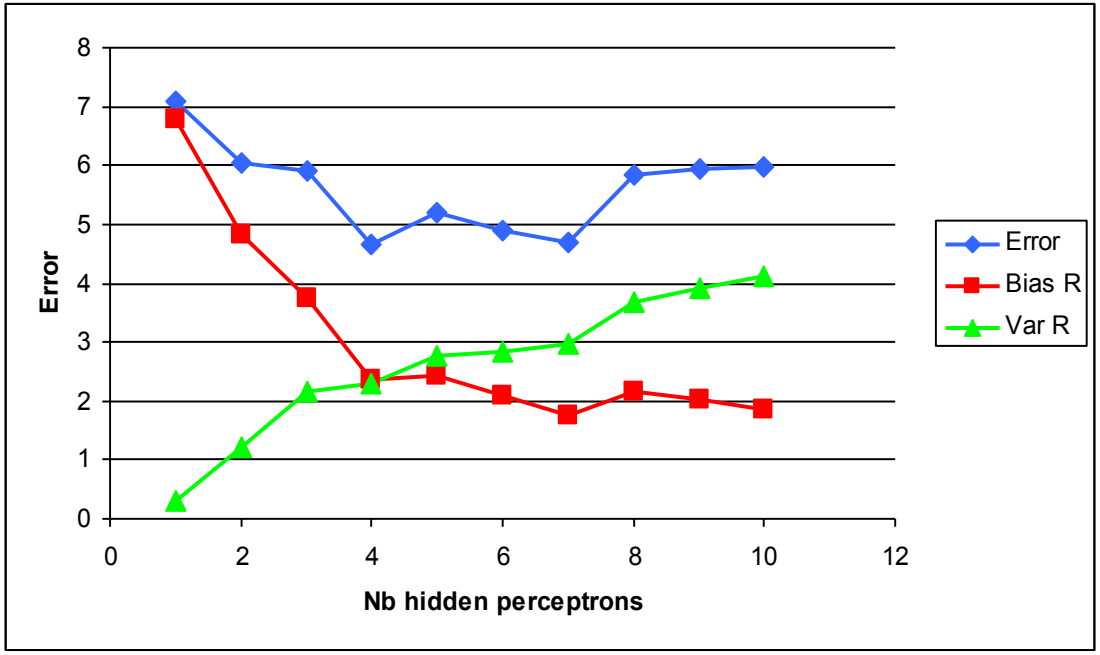
Complexity of the model



Usually, the bias is a decreasing function of the complexity, while variance is an increasing function of the complexity.

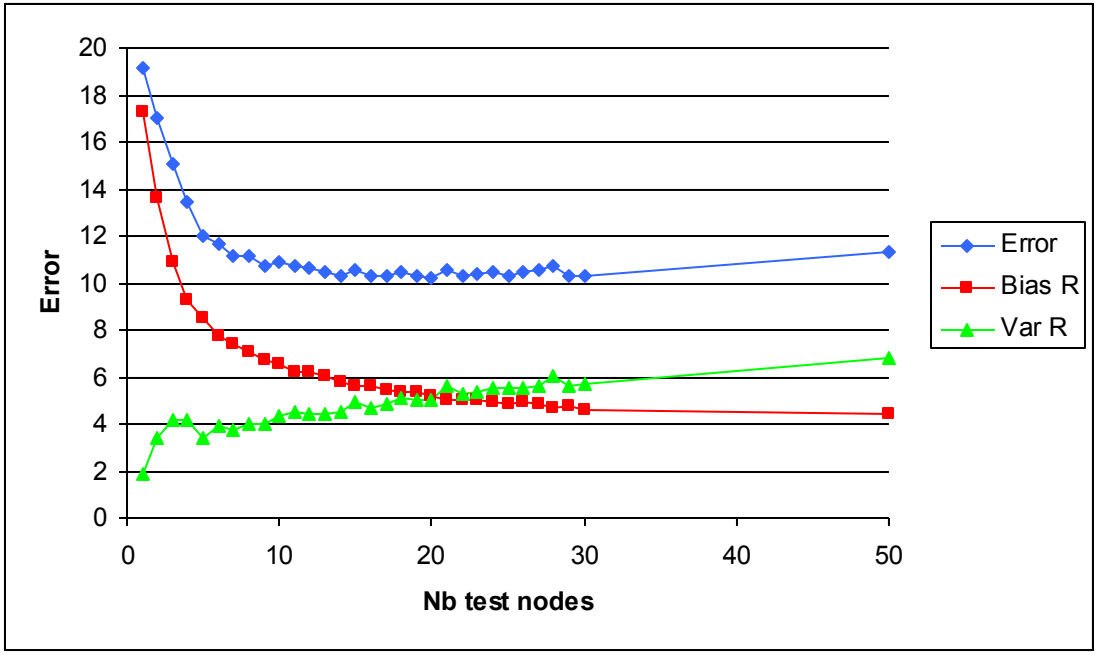
Complexity of the model – neural networks

- Error, bias, and variance w.r.t. the number of neurons in the hidden layer



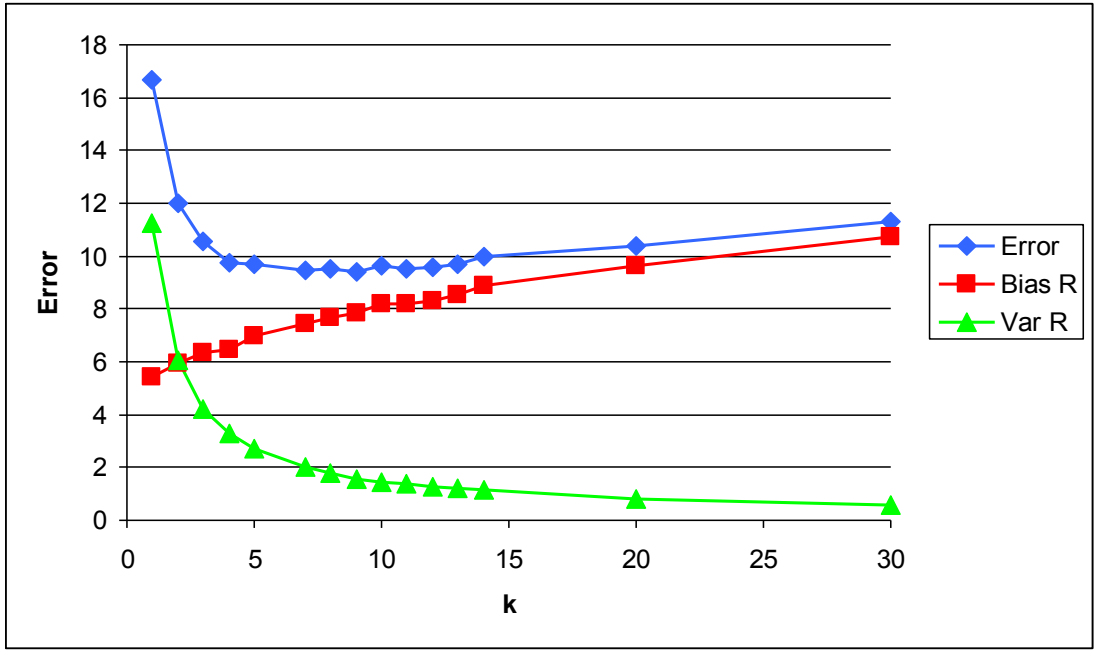
Complexity of the model – regression trees

- Error, bias, and variance w.r.t. the number of test nodes



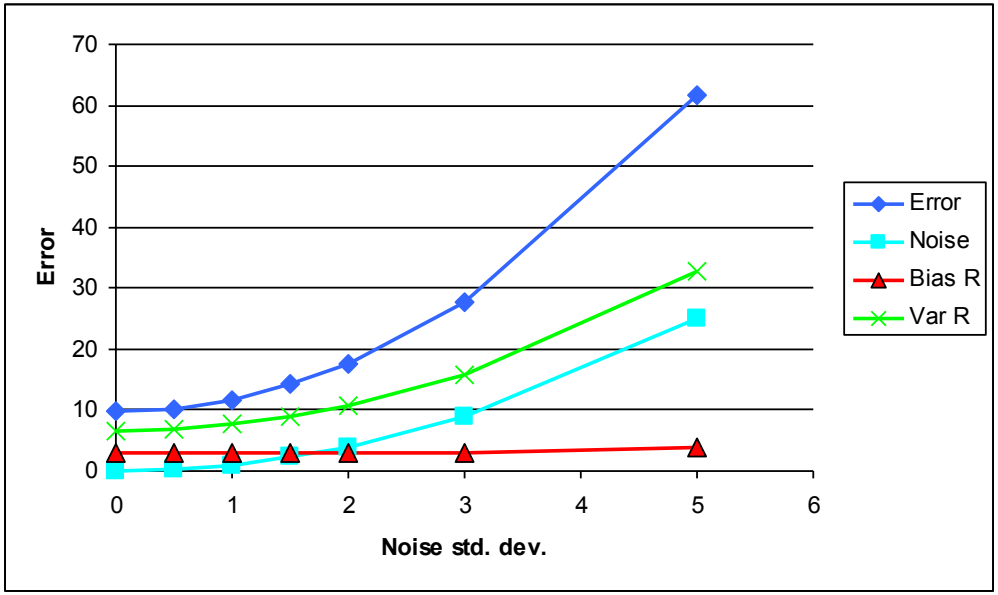
Complexity of the model – k-NN

- Error, bias, and variance w.r.t. k, the number of neighbors



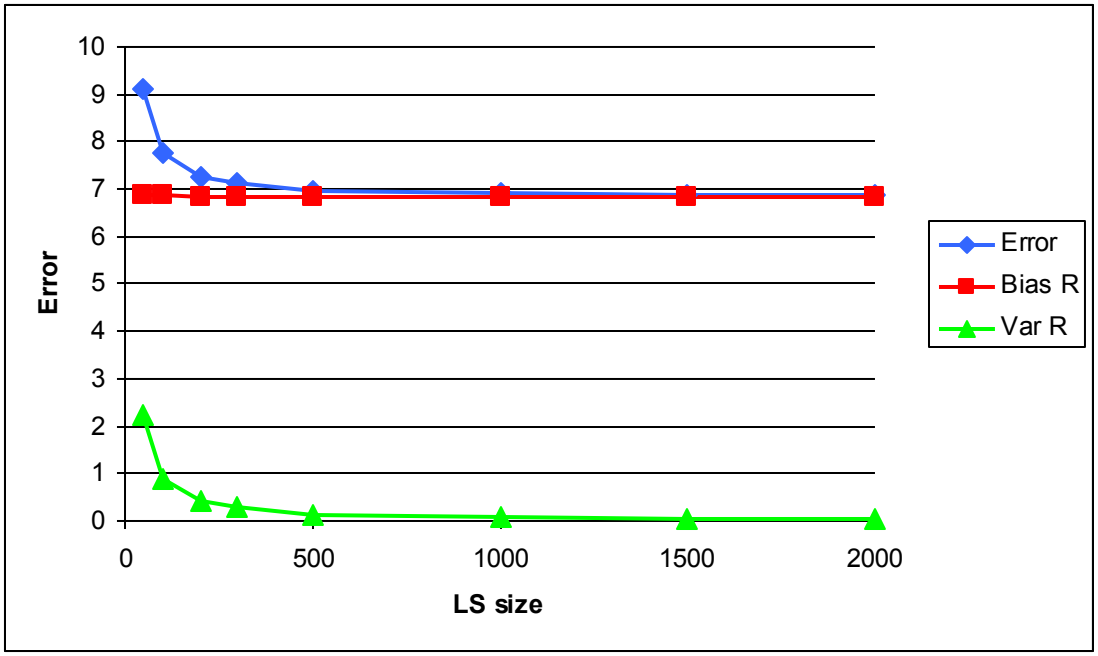
Learning problem

- Complexity of the Bayes model:
  - At fixed model complexity, bias increases with the complexity of the Bayes model. However, the effect on variance is difficult to predict.
- Noise:
  - Variance increases with noise and bias is mainly unaffected.
  - E.g. with (full) regression trees



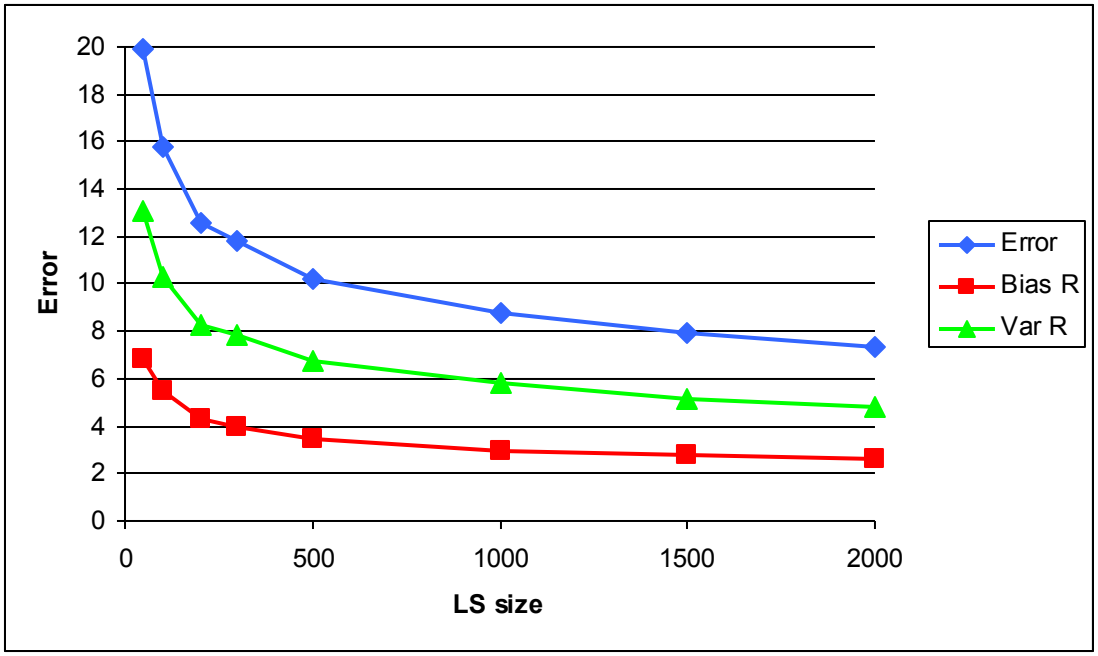
Learning sample size (1)

- At fixed model complexity, bias remains constant and variance decreases with the learning sample size. E.g. linear regression



Learning sample size (2)

- When the complexity of the model is dependant on the learning sample size, both bias and variance decrease with the learning sample size. E.g. regression trees



Learning algorithms – linear regression

Method	Err^2	$Bias^2$+Noise	Variance
Linear regr.	7.0	6.8	0.2
k-NN (k=1)	15.4	5	10.4
k-NN (k=10)	8.5	7.2	1.3
MLP (10)	2.0	1.2	0.8
MLP (10 – 10)	4.6	1.4	3.2
Regr. Tree	10.2	3.5	6.7

- Very few parameters : small variance
- Goal function is not linear : high bias

Learning algorithms – k-NN

Method	Err^2	$Bias^2$+Noise	Variance
Linear regr.	7.0	6.8	0.2
k-NN (k=1)	15.4	5	10.4
k-NN (k=10)	8.5	7.2	1.3
MLP (10)	2.0	1.2	0.8
MLP (10 – 10)	4.6	1.4	3.2
Regr. Tree	10.2	3.5	6.7

- Small k : high variance and moderate bias
- High k : smaller variance but higher bias

Learning algorithms - MLP

Method	Err^2	$Bias^2$+Noise	Variance
Linear regr.	7.0	6.8	0.2
k-NN (k=1)	15.4	5	10.4
k-NN (k=10)	8.5	7.2	1.3
MLP (10)	2.0	1.2	0.8
MLP (10 – 10)	4.6	1.4	3.2
Regr. Tree	10.2	3.5	6.7

- Small bias
- Variance increases with the model complexity

Learning algorithms – regression trees

Method	Err^2	$Bias^2$+Noise	Variance
Linear regr.	7.0	6.8	0.2
k-NN (k=1)	15.4	5	10.4
k-NN (k=10)	8.5	7.2	1.3
MLP (10)	2.0	1.2	0.8
MLP (10 – 10)	4.6	1.4	3.2
Regr. Tree	10.2	3.5	6.7

- Small bias, a (complex enough) tree can approximate any non linear function
- High variance

Content of the presentation

- Bias and variance definition
- Parameters that influence bias and variance
- Bias and variance reduction techniques
  - Introduction
  - Dealing with the bias/variance tradeoff of one algorithm
  - Ensemble methods

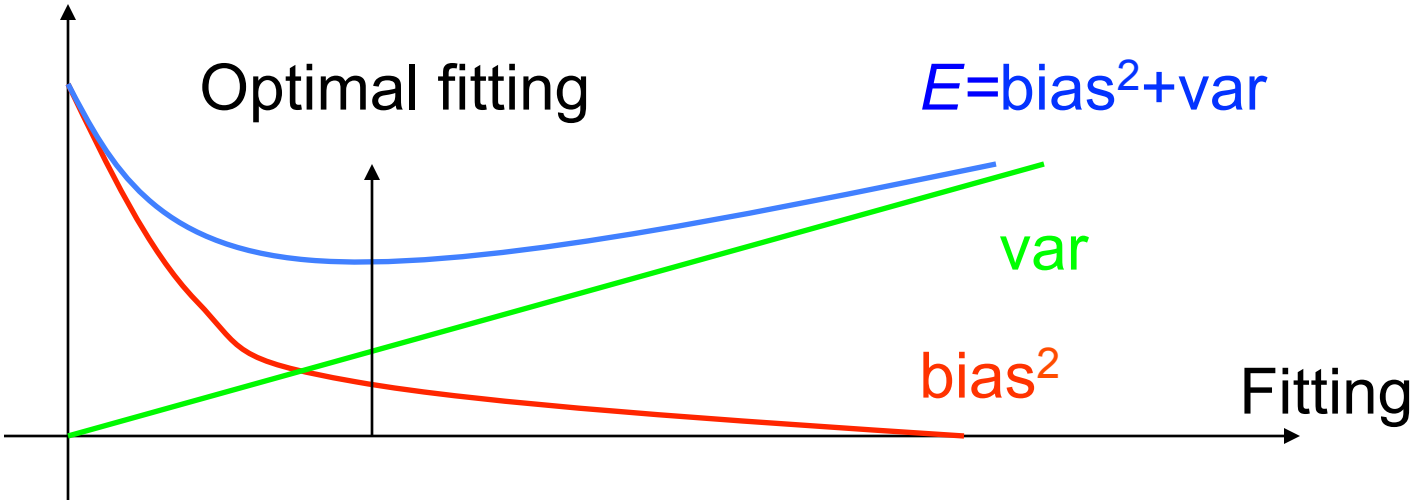
Bias and variance reduction techniques

- In the context of a given method:
  - Adapt the learning algorithm to find the best trade-off between bias and variance.
  - Not a panacea but the least we can do.
  - Example: pruning, weight decay.
- Ensemble methods:
  - Change the bias/variance trade-off.
  - Universal but destroys some features of the initial method.
  - Example: bagging, boosting.

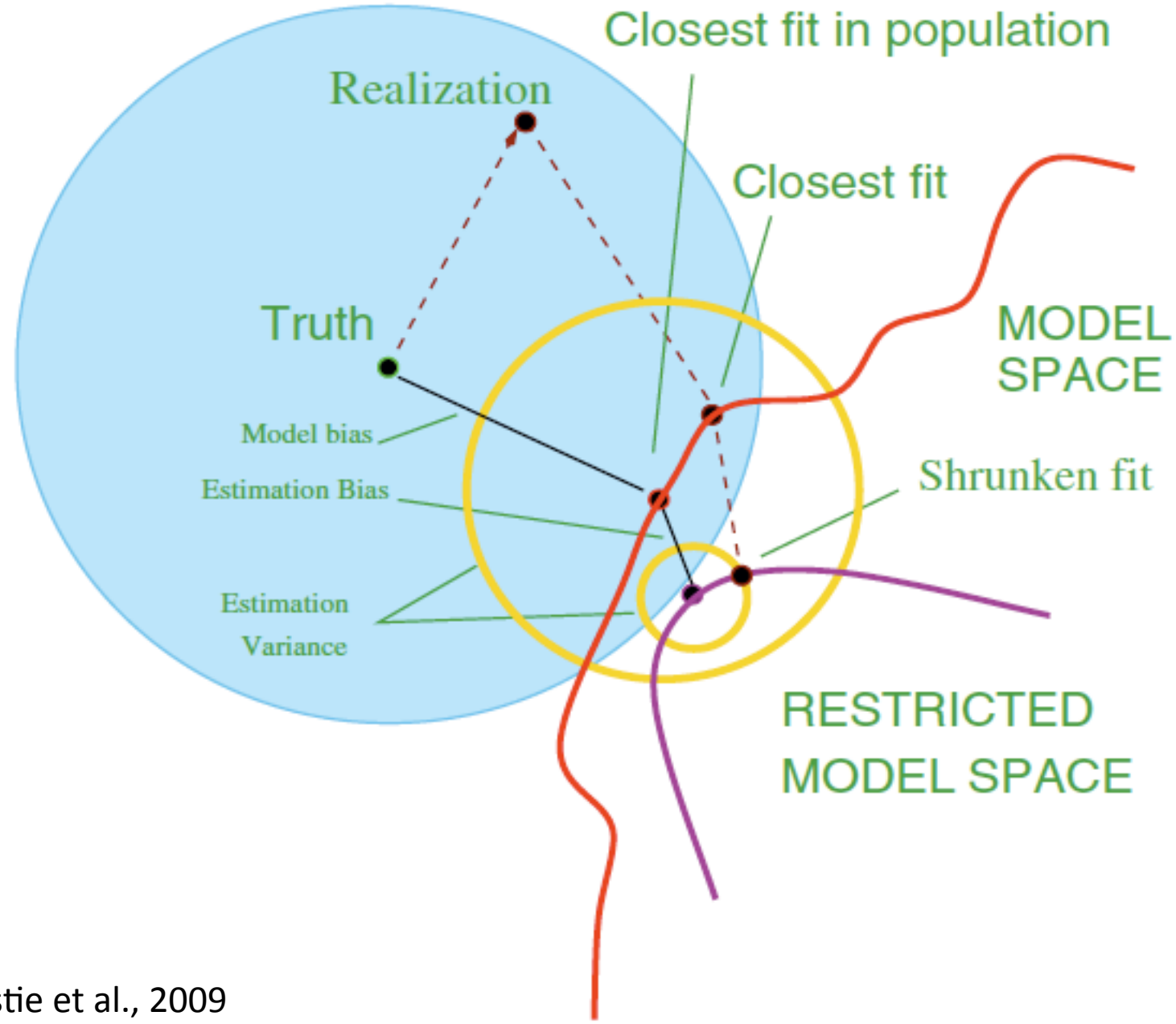
Variance reduction: 1 model (1)

- General idea: reduce the ability of the learning algorithm to fit the *LS*
  - Pruning
    - reduces the model complexity explicitly
  - Early stopping
    - reduces the amount of search
  - Regularization
    - reduce the size of the hypothesis space
    - Weight decay with neural networks consists in penalizing high weight values

Variance reduction: 1 model (2)



- Selection of the optimal level of fitting
  - a priori (not optimal)
  - by cross-validation (less efficient): Bias² ≈ error on the learning set, *E* ≈ error on an independent test set



Hastie et al., 2009

Variance reduction: 1 model (3)

- Examples:
  - Post-pruning of regression trees
  - Early stopping of MLP by cross-validation

Method	E	Bias	Variance
Full regr. Tree (250)	10.2	3.5	6.7
Pr. regr. Tree (45)	9.1	4.3	4.8
Full learned MLP	4.6	1.4	3.2
Early stopped MLP	3.8	1.5	2.3

- As expected, variance decreases but bias increases

Ensemble methods (1)

- Combine the predictions of several models built with a learning algorithm in order to improve with respect to the use of a single model
- Two main families:
  - Averaging techniques
    - Grow several models independently and simply average their predictions
    - Ex: bagging, random forests
    - Decrease mainly variance
  - Boosting type algorithms
    - Grows several models sequentially
    - Ex: Adaboost, MART
    - Decrease mainly bias

Ensemble methods (2)

- Examples:
  - Bagging, boosting, random forests

Method	E	Bias	Variance
Full regr. Tree	10.2	3.5	6.7
Bagging	5.3	3.8	1.5
Random Forests	4.9	4.0	0.9
Boosting	5.0	3.1	1.9

Discussion

- The notions of bias and variance are very useful to predict how changing the (learning and problem) parameters will affect the accuracy. E.g. this explains why very simple methods can work much better than more complex ones on very difficult tasks
- Variance reduction is a very important topic:
  - To reduce bias is easy, but to keep variance low is not as easy.
  - Especially in the context of new applications of machine learning to very complex domains: temporal data, biological data, Bayesian networks learning, text mining…
- All learning algorithms are not equal in terms of variance. Trees are among the worse methods from this criterion

Outline

- Performance evaluation
  - Model assessment and selection
  - Cross-validation
  - Bootstrap
  - CV based model selection

Estimating the performance of a model

- Question: given a model learned from some dataset (of size *N*), how to estimate its performance from this dataset?
- What for?
  - Model selection
    - to choose the best among several models
    - E.g., to determine the right complexity of a model or to choose between different learning algorithms
  - Model assessment
    - Having chosen a final model, to estimate its performance on new data

When *N* is large: test set method

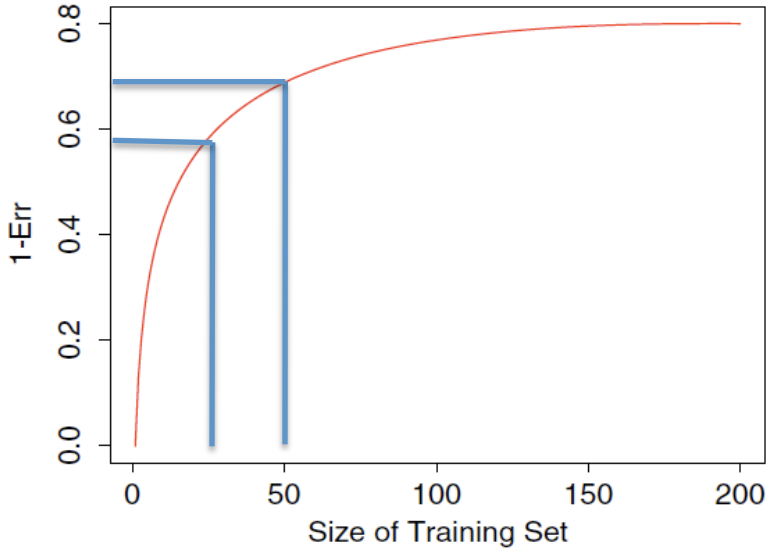
- Randomly divide the dataset into 2 parts: a learning set LS and a test set TS (eg. 70%, 30%)

  - Fit the model on LS
  - Test it on TS
  - The resulting estimate is an estimate of the error of a model learned on the whole dataset (LS+TS)

Small datasets

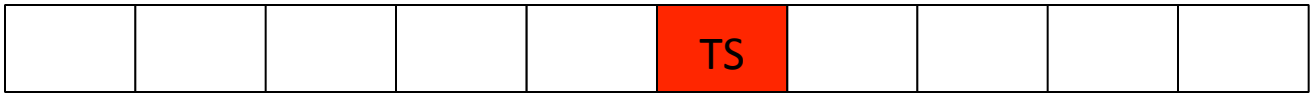
- Test set error is unreliable because it is based on a small sample (30% of an already small dataset)
- It is also pessimistically biased as an estimate of the error of a model built on the whole dataset
  - For small sample sizes, a model learned on 70% of the data is significantly less good than a model learned on the whole data

Learning curve: performance vs the size of the learning set



k-fold Cross Validation

- Randomly divide the dataset into k subsets (eg., k=10)

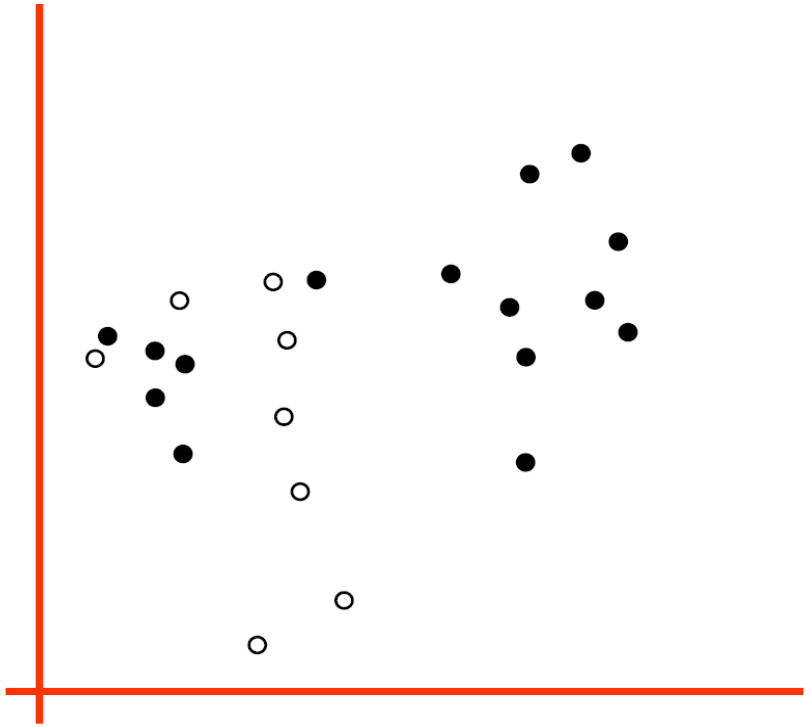


  - For each subset:
    - Learn the model on the objects that are not in the subset
    - Compute a prediction with this model for the points in the subset
  - Report the mean error over these predictions
- When *k*=*N*, the method is called *leave-one-out* cross-validation

Which value of *k*?

- *k*=*N*:
  - Unbiased: removing one object does not change much the size of the learning sample
  - High variance: highly dataset dependant
  - Slow: require to train N models
- *k*=5,10:
  - Lower variance and faster: only 5-10 models on fewer data
  - Potentially biased: see learning curve

Small exercise

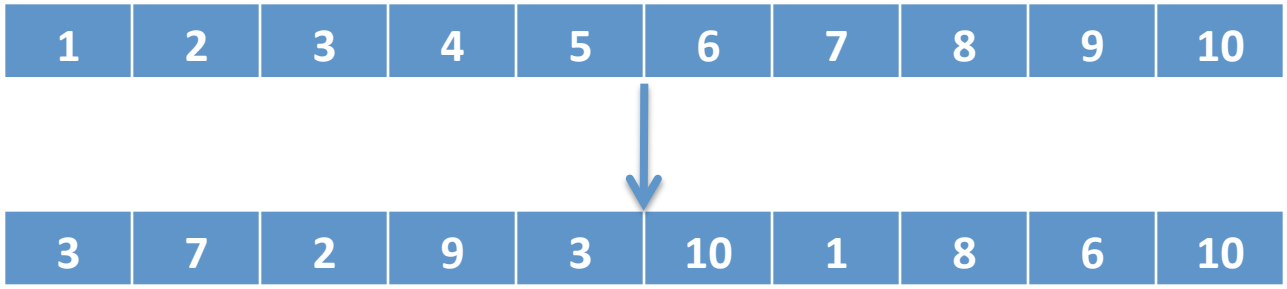


- In this classification problem with two inputs:
  - What it the resubstitution error (LS error) of 1-NN?
  - What is the LOO error of 1-NN?
  - What is the LOO error of 3-NN?
  - What is the LOO error of 22-NN?

Andrew Moore

Bootstrap

- Bootstrap sampling=sampling with replacement



- Some objects do not appear, some objects appear several times
- $P(o_i \in \text{bootstrap}) = 1 - (1 - \frac{1}{N})^N \approx 1 - \frac{1}{e} = 0.632$

Bootstrap

- Bootstrap error estimate:
  - For i=1 to B
    - Take a bootstrap sample B_i from the dataset
    - Learn a model f_i on it
  - For each object, compute the expected error of all models that were built without it (about 30%)
  - Average over all objects
- Improvements:
  - “.632 bootstrap” corrects for the learning curve
  - “.632+ bootstrap” corrects for overfitting

Conditional vs expected test errors

- Conditional test error (for a given *model* $\hat{f}_{LS}$):

$$\mathrm{Err}_{LS} = E_{x,y}\{L(y, \hat{f}_{LS}(x))\}$$

- Expected test error:

$$E_{LS}\{\mathrm{Err}_{LS}\} = E_{LS}\{E_{x,y}\{L(y, \hat{f}_{LS}(x))\}\}$$

- Only the test set method estimates the first
- Cross-validation estimates the second (even leave-one-out)

Model selection: typical scenario

- Given a dataset with N objects (input-output pairs), how to best exploit this dataset to obtain:
  - The best possible model (eg. among regression trees and k-NN) *(model selection)*
  - An estimate of its prediction error *(model assessment)*
- Again, the solution depends on the size N of the dataset

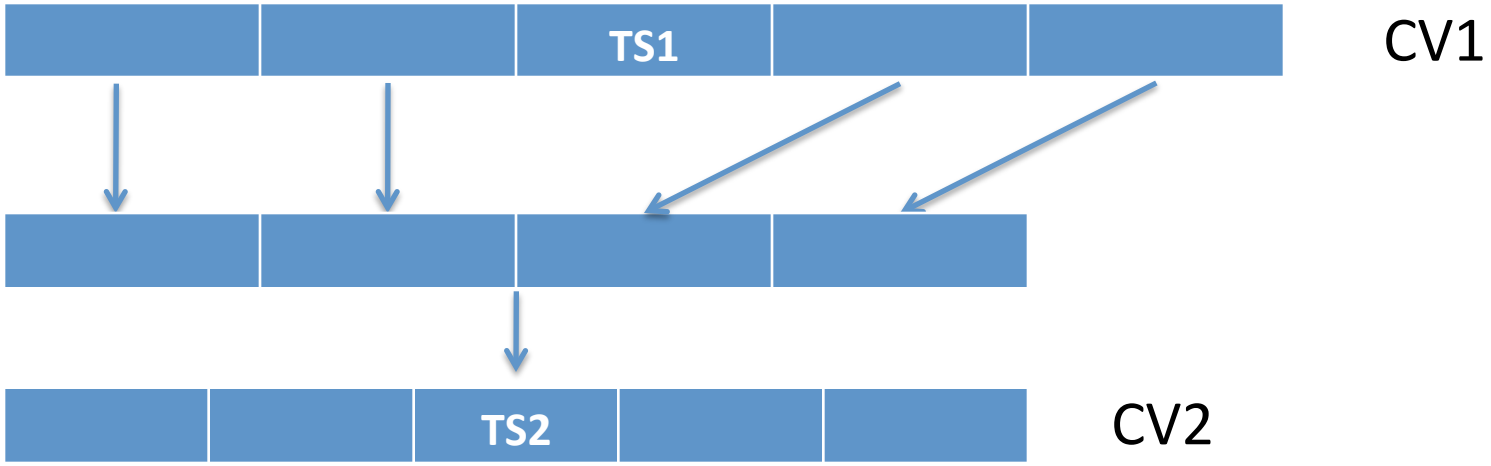
When N is large: test set method

- Randomly divide the dataset into 3 parts: a learning set LS, a validation set VS, and a test set TS (eg. 50%, 25%, 25%)

  - Fit the models to compare on LS (using different learning algorithms or different complexity values)
  - Select the best one based on its performance on VS
  - Retrain it on LS+VS
  - Test it on TS => the performance estimate
  - Retrain it on LS+VS+TS => the finally chosen model

When N is small: cross-validation

- Use two stages of k-fold cross-validation



- CV1 is used for the assessment of the final model, CV2 is used for model selection
- We could also combine test set and CV

Do we really need two stages?

- How well does the CV2 error (and VS error) estimate the true error?
- If you compare many (complex) models, the probability that you will find a good one by chance on your data increases

$\Rightarrow$The VS or CV2 errors are overly optimistic

Illustration

- Dataset: N=50, 1000 inputs variables, all unrelated to the class (values are randomly drawn from $N(0,1)$)

  => any model should have a 50% generalization error rate
- Comparison of 1000 learning algorithms: *i*th algorithm learns a decision tree on the *i*th feature only
- 10-fold CV error of the best model: 16%
- Its error on a test sample of 5000 cases: 48%

Selection bias

- General rule:
  - Any choice made using the output should be inside a cross-validation loop
- Another example: on the same dataset:
  - Select the 10 attributes that are the most correlated with the output on the LS
  - Estimate the error rate of a tree built with these 10 attributes using 10-fold CV on the same LS => 20%
  - Error of this model on a sample of 5000 cases => 51%

  (this problem is called the selection bias)

Analytical methods for model selection

- Find the model that minimizes a criterion typically of the form:

$$Err(LS) + G(\text{Complexity})$$

  where *G* is a monotonically increasing function
- The criterion is derived from theoretical arguments (eg., the Minimum Description Length approach is motivated from coding theory)
- Advantage:
  - Cheap: no retraining
- Drawbacks:
  - Ok for model selection but not for model assessment
  - May miss the true optimum in the finite sample case

Outline

- Performance measures
  - Classification: error rate, sensitivity, specificity, ROC curve, precision, recall
  - Regression: square error, absolute error, correlation
  - Loss functions for learning

Performance criteria

	True class	Model 1	Model 2
1	Negative	Positive	Negative
2	Negative	Negative	Negative
3	Negative	Positive	Positive
4	Negative	Positive	Negative
5	Negative	Negative	Negative
6	Negative	Negative	Negative
7	Negative	Negative	Positive
8	Negative	Negative	Negative
9	Negative	Negative	Negative
10	Positive	Positive	Positive
11	Positive	Positive	Negative
12	Positive	Positive	Positive
13	Positive	Positive	Positive
14	Positive	Negative	Negative
15	Positive	Positive	Negative

- Which of these two models is the best?
- The choice of an error or quality measure is highly application dependent.

Binary classification

- Results can be summarized in a contingency table (aka confusion matrix)

Actual class	Predicted class: p	Predicted class: n	Total
p	**T**rue **P**ositive	**F**alse **N**egative	P
n	**F**alse **P**ositive	**T**rue **N**egative	N

Error rate = *FP+FN/N+P*

Accuracy = *TP+TN/N+P*
= 1-*Error rate*

- Simplest criterion: error rate or accuracy

Model 1

Actual class	Predicted class: p	Predicted class: n	Total
p	5	1	6
n	3	6	9

Error rate=4/15=27%
Accuracy=11/15=73%

Model 2

Actual class	Prediction class: p	Prediction class: n	Total
p	3	3	6
n	2	7	7

Error rate=5/15=33%
Accuracy=10/15=66%

Limitation of error rate

Model 1

Actual class	Predicted class: p	Predicted class: n	Total
p	0	10	10
n	0	90	90

Error rate=10%

Model 2

Actual class	Predicted class: p	Predicted class: n	Total
p	10	0	10
n	10	80	90

Error rate=10%

Model 2

Actual class	Predicted class: p	Predicted class: n	Total
p	0	50	50
n	0	50	50

Error rate=50%

- Does not convey information about how errors are distributed across classes
- Sensitive to changes in class distribution in the test sample

Sensitivity/Specificity

- For medical diagnosis, more appropriate measures are:
  - sensitivity = TP/P (also called recall, TP rate)
  - specificity = TN/(TN+FP) = 1- FP/N

Model 1

Actual class	Predicted class p	Predicted class n	Total
p	0	10	10
n	0	90	90

Error rate=10%

Sensitivity=0/10=0%

Specificity=90/90=100%

Model 2

Actual class	Predicted class p	Predicted class n	Total
p	10	0	10
n	10	80	90

Error rate=10%

Sensitivity=10/10=100%

Specificity=80/90=89%

Model 2

Actual class	Predicted class p	Predicted class n	Total
p	0	50	50
n	0	50	50

Error rate=50%

Sensitivity=0/50=0%

Specificity=90/90=100%

Sensitivity/Specificity

	True class	Model 1	Model 2
1	Negative	Positive	Negative
2	Negative	Negative	Negative
3	Negative	Positive	Positive
4	Negative	Positive	Negative
5	Negative	Negative	Negative
6	Negative	Negative	Negative
7	Negative	Negative	Positive
8	Negative	Negative	Negative
9	Negative	Negative	Negative
10	Positive	Positive	Positive
11	Positive	Positive	Negative
12	Positive	Positive	Positive
13	Positive	Positive	Positive
14	Positive	Negative	Negative
15	Positive	Positive	Negative

Model 1

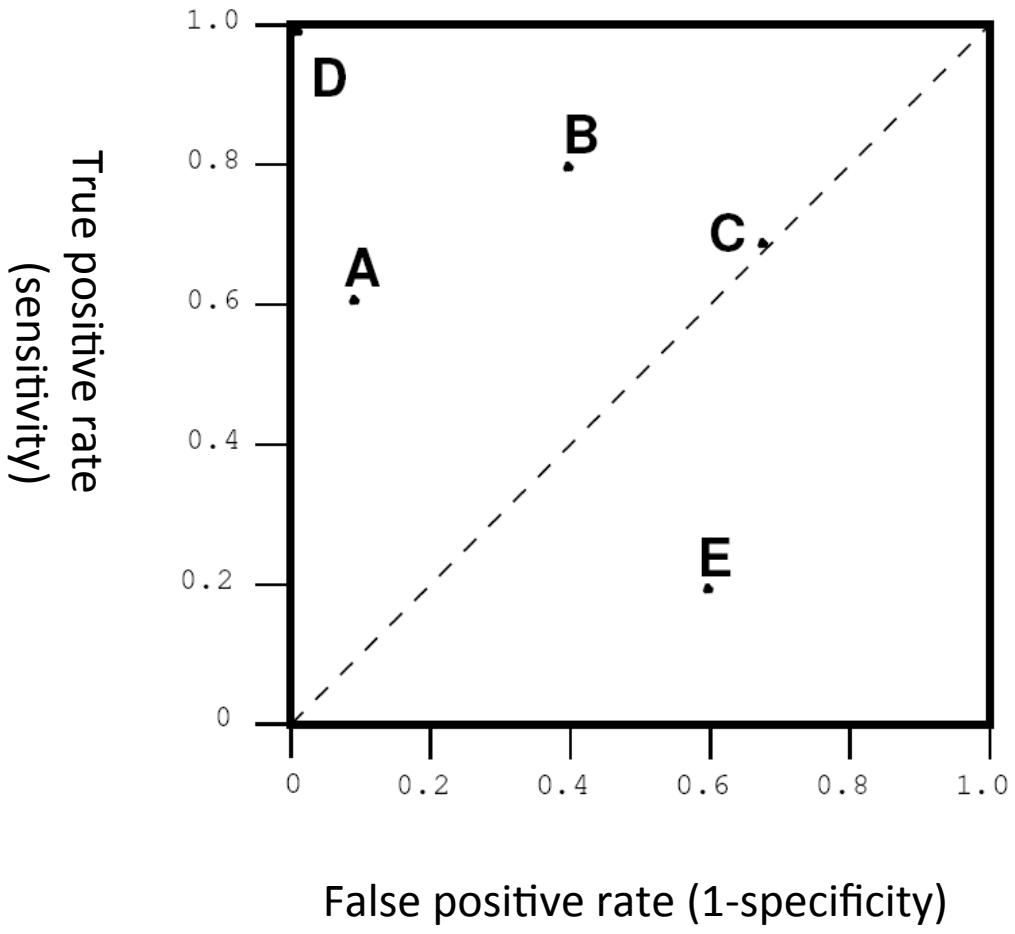
Actual class \ Predicted class	p	n	Total	
p	5	1	6	Sensitivity=5/6=83%
n	3	6	9	Specificity=6/9=66%

Model 2

Actual class \ Prediction class	p	n	Total	
p	3	3	6	Sensitivity=3/6=50%
n	2	7	7	Specificity=7/9=78%

- Which one of the models is better depends on the application

ROC Plot



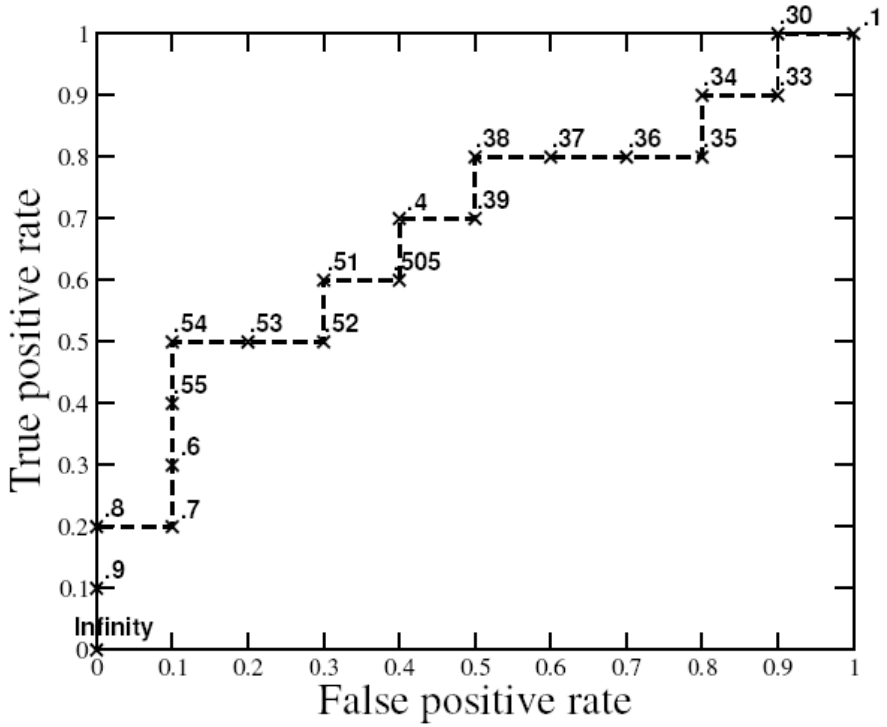
- Where are
- the best classifier?
- a classifier that always says positive?
- a classifier that always says negative?
- a classifier that randomly guesses the class?

ROC curve

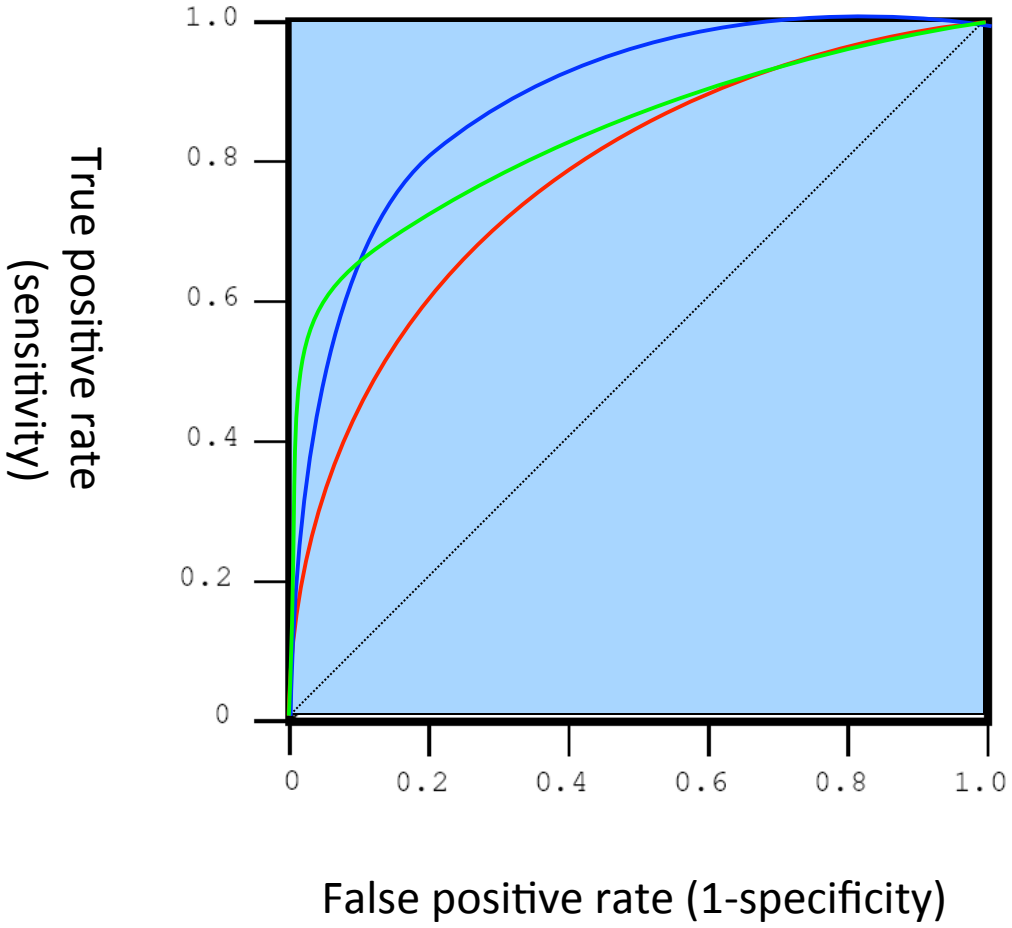
- Often the output of a learning algorithm is a number (e.g., class probability)
- In this case, a threshold may be chosen in order to balance sensitivity and specificity
- A ROC curve plots sensitivity versus 1-specificity for different threshold values

ROC curve

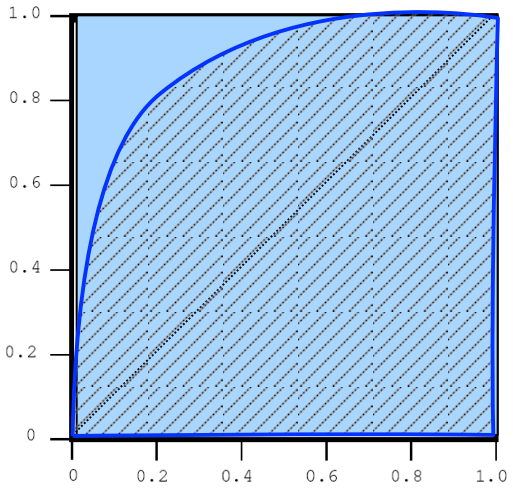
Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



ROC curve


True positive rate
(sensitivity)
1.0
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1.0
False positive rate (1-specificity)

Area under the ROC curve



- Summarize a ROC curve by a single number
- can be interpreted as the probability that two objects randomly drawn from the sample are well ordered by the model, ie., the positive has a higher score than the negative
- Warning: does not tell the whole story

Precision and recall

- Other frequently used measures:
  - Precision = TP/(TP+FP) = proportion of good predictions among positive predictions
  - Recall = TP/(TP+FN) = proportion of positives that are detected (= sensitivity)
  - F-measure=2*Precision*Recall/(Precision+Recall)

Model 1

	Predicted class		
Actual class	p	n	Total
p	10	0	10
n	50	950	1000

Sensitivity=10/10=100%

Specificity=950/1000=95%

Precision=10/60=17%

Recall =10/10=100%

F-measure=29%

Model 2

	Predicted class		
Actual class	p	n	Total
p	10	0	10
n	10	990	1000

Sensitivity=10/10=100%

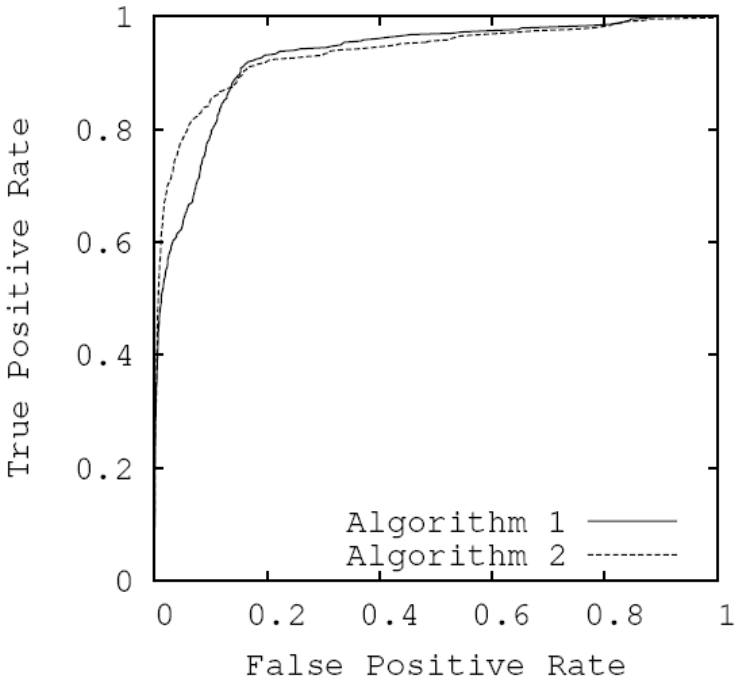
Specificity=990/1000=99%

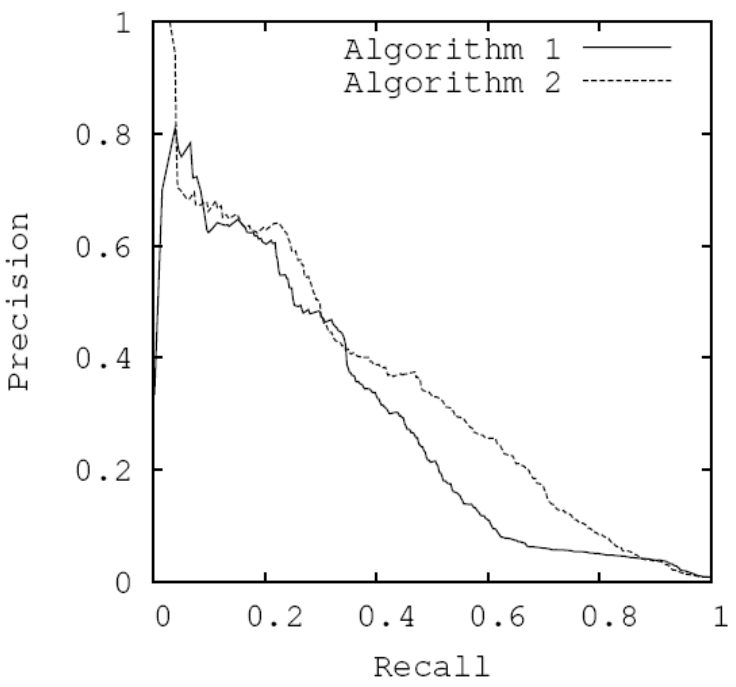
Precision=10/20=50%

Recall =10/10=100%

F-measure=66%

Precision/Recall vs ROC curve





Regression performance

	True Y	Model 1	Model 2
1	59.43	63.08	78.12
2	33.15	36.66	37.28
3	11.8	13.28	19.62
4	77.11	78.38	90.01
5	40.6	42.92	60.19
6	81.25	85	86.13
7	83.01	86.37	102.32
8	0.55	1.67	16.6
9	76.72	80.92	94.74
10	29.18	33.28	52.54
11	20.37	22.43	36.94
12	95.13	98.86	104.24
13	11.66	15.97	28.16
14	42.95	46.12	65.82
15	94.55	94.75	104.02

MSE1= 53.38 MSE2= 249.6

MAE1=4.3 MAE2=14.6

- Mean squared error

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Mean absolute error (more tolerant to sporadic large errors)

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Regression performance

	True Y	Model 1	Model 2
1	59.43	63.08	78.12
2	33.15	36.66	37.28
3	11.8	13.28	19.62
4	77.11	78.38	90.01
5	40.6	42.92	60.19
6	81.25	85	86.13
7	83.01	86.37	102.32
8	0.55	1.67	16.6
9	76.72	80.92	94.74
10	29.18	33.28	52.54
11	20.37	22.43	36.94
12	95.13	98.86	104.24
13	11.66	15.97	28.16
14	42.95	46.12	65.82
15	94.55	94.75	104.02

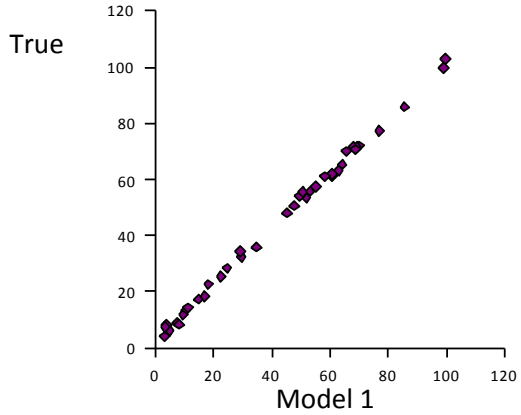
- Pearson correlation

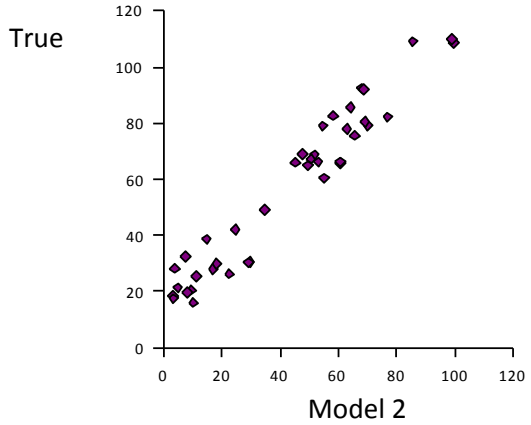
$$\frac{\sum_i (y_i - \frac{1}{N}\sum_j y_j)(\hat{y}_i - \frac{1}{N}\sum_j \hat{y}_j)}{(N-1)s_y s_{\hat{y}}}$$

- Spearman rank correlation

$$1 - \frac{6\sum_i d_i^2}{N(N^2-1))}$$

with d_i the difference of rank of y_i and $\hat{y}_i$





Performance measures for training

- Performance measures for training can be different from performance measures for testing
- Several reasons:
  - Algorithmic:
    - A derivable measure is amenable to gradient optimization
    - Eg., error rate and MAE are not derivable, AUC is not decomposable
  - Overfitting:
    - For training, the loss often incorporates a penalty term for model complexity (which is irrelevant at test time)
    - Some measures are less prone to overfitting (eg., margin)

References

- Hastie et al., the elements of statistical learning, 2009:
  - Bias/variance tradeoff 2.5, 2.9, 7.2, 7.3
  - Model assessment and selection: 7.1, 7.2, 7.3, 7.10, 7.11