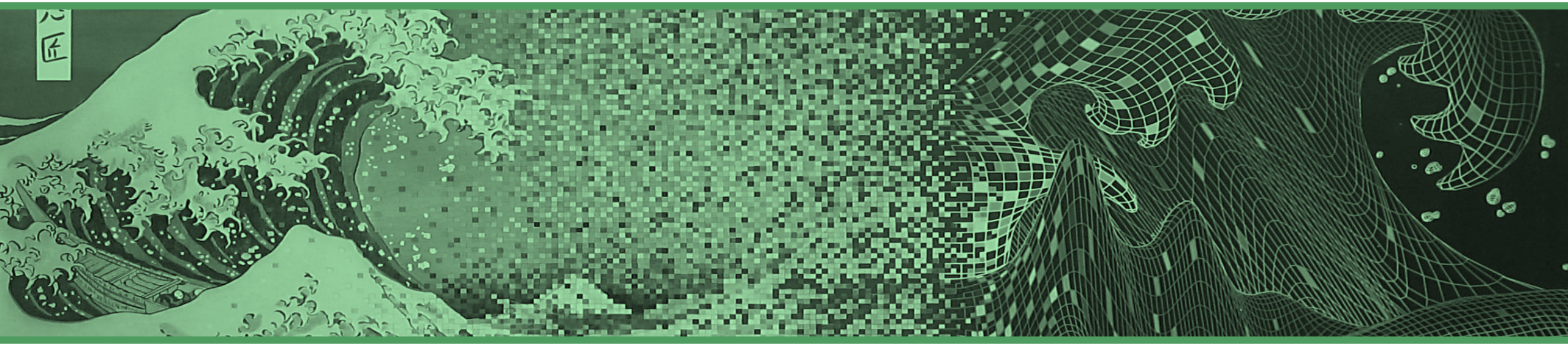
TÉCNICO LISBOA

Model Evaluation: Part II

Assessing regression models, generalization ability



Aprendizagem 2024

Miguel Couceiro miguel.j.couceiro@tecnico.ulisboa.pt
Andreas Wischert andreas.wichert@tecnico.ulisboa.pt

Outline

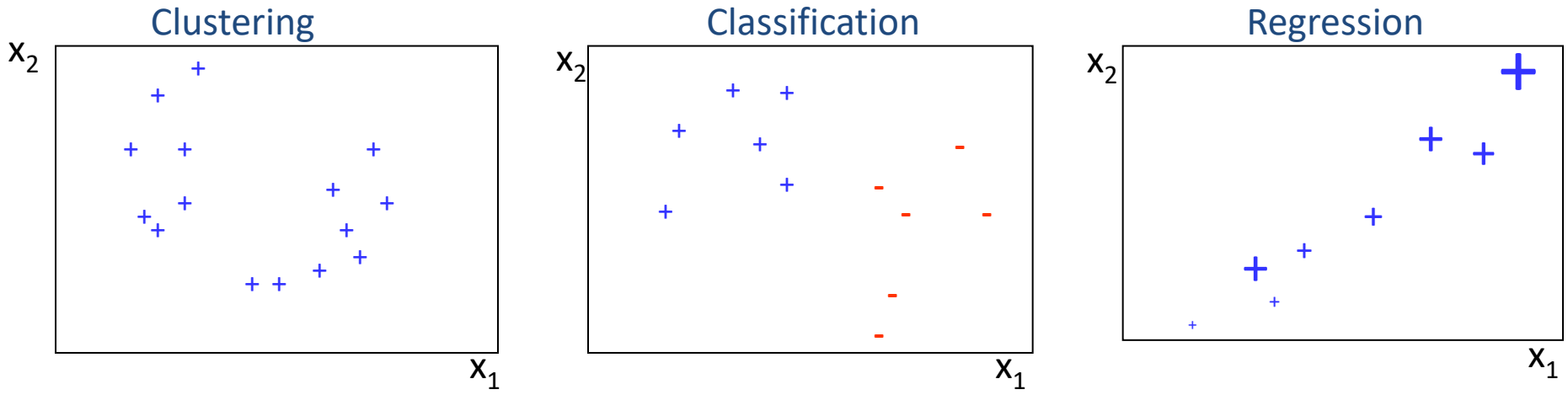
- **Evaluating regression models**
  - loss functions
  - residue analysis
- **Generalization ability**
  - bias and variance
  - model complexity: VC dimension
- **Statistically testing performance differences**
- **Advanced notes on classifier evaluation**
  - ROC curves and AUC

Outline

- **Evaluating regression models**
  - **loss functions**
  - **residue analysis**
- Generalization ability
  - bias and variance
  - model complexity: VC dimension
- Statistically testing performance differences
- Advanced notes on classifier evaluation
  - ROC curves and AUC

Regression models

- *descriptive/unsupervised setting*
  - given a set of observations, describe relation between (explanatory) variables and a numeric target
  - evaluation using training observations
- *predictive/supervised setting*
  - given a set of observations with a real-valued outcome, learn a mapping to estimate the outcome of new observations
  - Evaluation using testing observations



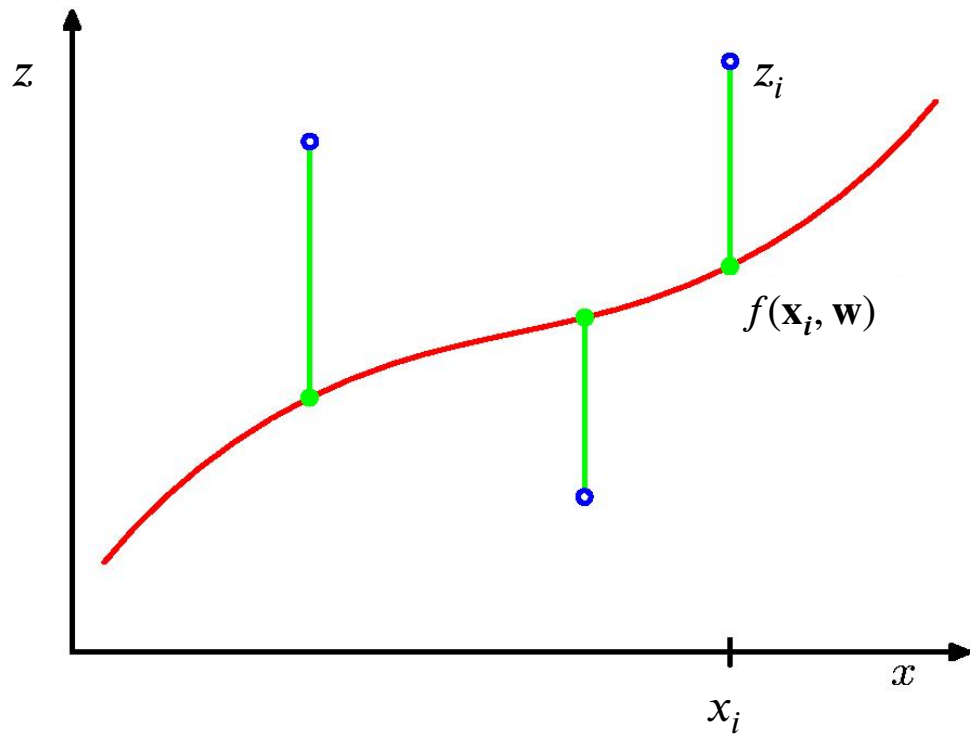
Recall: hold-out and CV approaches

- Regressors can yield good performance on **training data** yet poorly perform on new observations
  - problem known as **overfitting**
  - we need to be able to assess learning adequacy outside training set!
- Solution: set aside a separate **testing set** of observations (**hold-out**)
    - we can estimate the empirical risk on this set

$$\int_{D} L\big(z, f(\mathbf{x})\big) \approx \sum_{(\mathbf{x},z)\in D_{test}} L\big(z, f(\mathbf{x})\big)$$

  - challenges of hold-out approach
  - too few test examples? how to assess performance variability?
- Solution: **resampling** to produce multiple train-test data partitions
  - **cross-validation**: the data is divided into *k* folds
    - at each iteration, one different fold is reserved for testing while others used for training

Sum of squares error function



$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(f(\mathbf{x}_i, \mathbf{w}) - \hat{z}_i)^2 = \frac{1}{2}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2$$

Mean squared error:

$$MSE(w) = MSE(\hat{z}, z) = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2$$

MSE can be used to guide the training as well as post-assess regressors

Evaluating regression models

- Loss measures over the quantity estimates:
  - mean absolute error

$$\mathrm{MAE}(\hat{z}, z) = \frac{1}{n}\sum_{i=1}^{n} |z_i - \hat{z}_i|$$

  - root mean square error

$$\mathrm{RMSE}(\hat{z}, z) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (z_i - \hat{z}_i)^2}$$

  - mean absolute percentage error

$$\mathrm{MAPE}(\hat{z}, z) = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{z_i - \hat{z}_i}{z_i}\right|$$

- A loss estimate can be collected per test fold
  - compute average and deviation across folds or the error distribution (e.g. boxplot)

Evaluating regression models

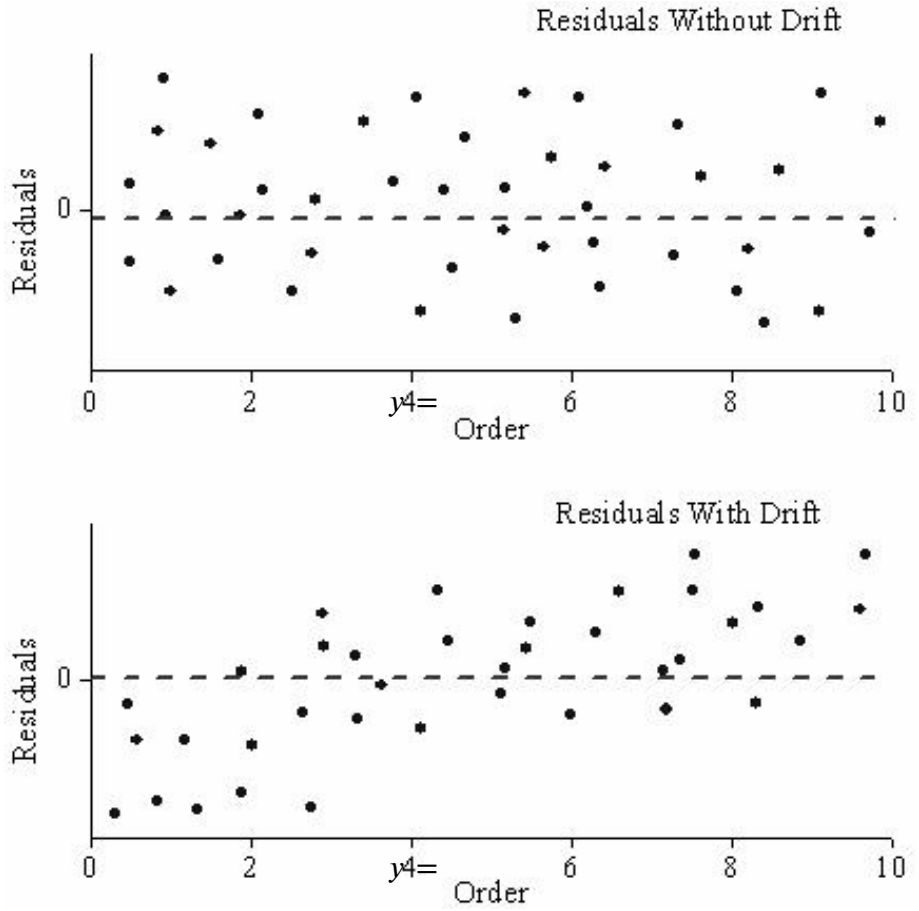
Analysis of the **residuals** $(z_i - \hat{z}_i)$

– quantitative examination of residuals (e.g. serial correlation)

– scatter plot residuals against *input variables*

– visual detection of deviations from randomness

  – signal the presence of learning biases



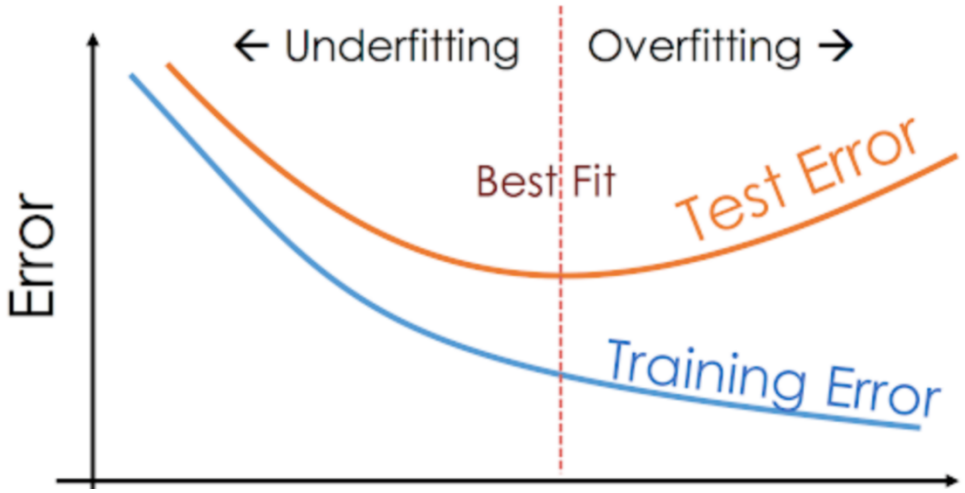
Outline

- Evaluating regression models
  - loss functions
  - residue analysis
- **Generalization ability**
  - **bias and variance**
  - **model complexity: VC dimension**
- Statistically testing performance differences
- Advanced notes on classifier evaluation
  - ROC curves and AUC

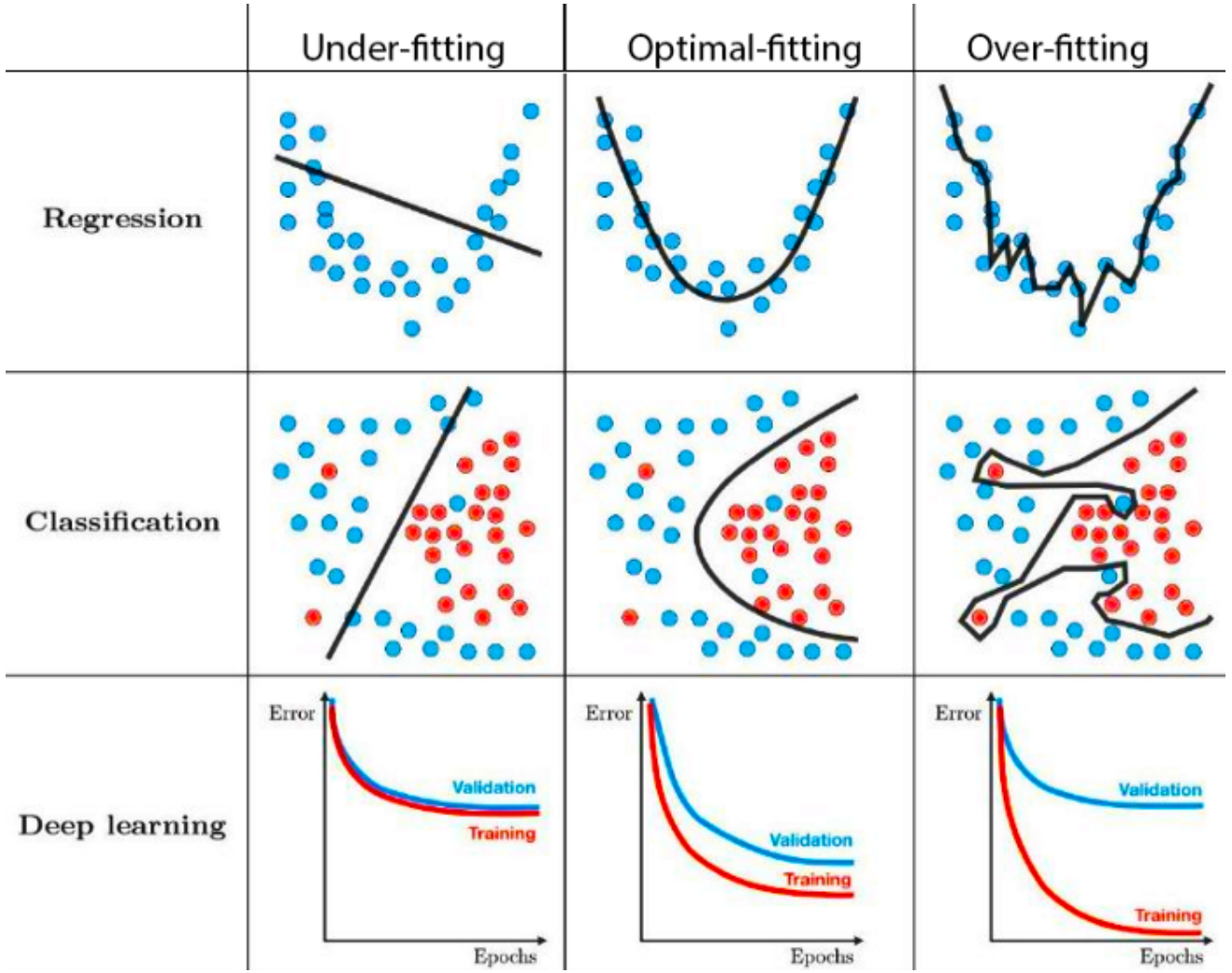
Generalization ability

- Learning should be able to generalize descriptions and predictions towards new observations
  - "first-order" generalization: new observations drawn from the same population
  - "second-order" generalization: observations drawn from external populations (e.g. other geographies, monitoring protocols, sensors)

- Goal: minimizing...
  - **overfitting** risks
  - **underfitting** risks



Generalization ability



https://i.pinimg.com/originals/72/e2/22/72e222c1542539754df1d914cb671bd7.png

Overfitting

Let us recover our previous definition of overfitting

- Consider the error of hypothesis h over
  - training data, $error_{train}(h)$
  - entire data, $error_D(h)$, or test data alone

- Hypothesis $h \in H$ *overfits* training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_D(h) > error_D(h')$$

- Why learning in high-dimensional data spaces is particularly susceptible to overfitting risks?

How many observations needed?

Avoiding overfitting

- **decision trees**
  - stop growing when data split not statistically significant
  - limiting depth and post-pruning
- **Bayes learning**
  - apply naïve assumption
  - prefer theoretical probability functions over empirical ones
- ***k*NN**
  - increasing *k* and prefer uniform against non-uniform weigths
- **regression approaches**
  - regularization terms (e.g., Ridge and Lasso variants)

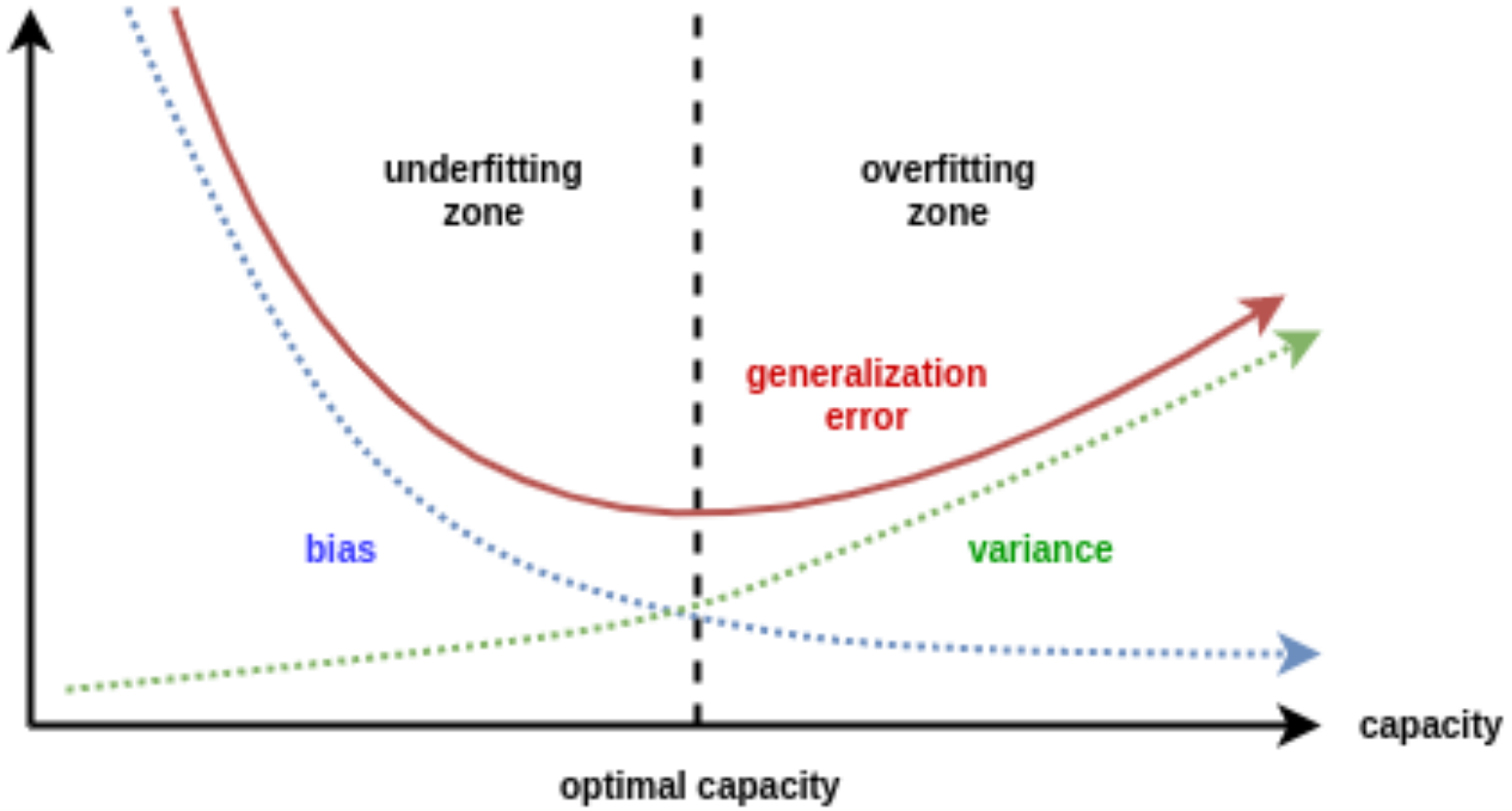
Bias and variance

- It can be shown that the loss between a learning function and the targets is a sum of:
  - a (squared) **bias**, **variance**, and constant **noise** terms

- **Bias**
  - inability of the learned model $f(\mathbf{x}, D)$ to accurately approximate the reference function $h(\mathbf{x}, \mathbf{w})$
  - can be viewed as an **approximation error**

- **Variance**
  - inadequacy of empirical knowledge contained in training sample D about reference function $h(\mathbf{x}, \mathbf{w})$
  - can be viewed as an **estimation error**

Bias-variance dilemma



Bias-variance dilemma

- **Trade-off** bias and variance
  - in a complex model that learns with a training sample of limited size…
    - the price for achieving a small bias is a large variance ⇒ *overfitting*
  - in contrast, rigid (simple) models generally have high bias and low variance ⇒ *underfitting*
  - The model with the optimal predictive capability is the one that leads to the best balance

- For any model, it is only when the size of the training sample becomes infinitely large that we can hope to eliminate both bias and variance at the same time

Bias and variance: example

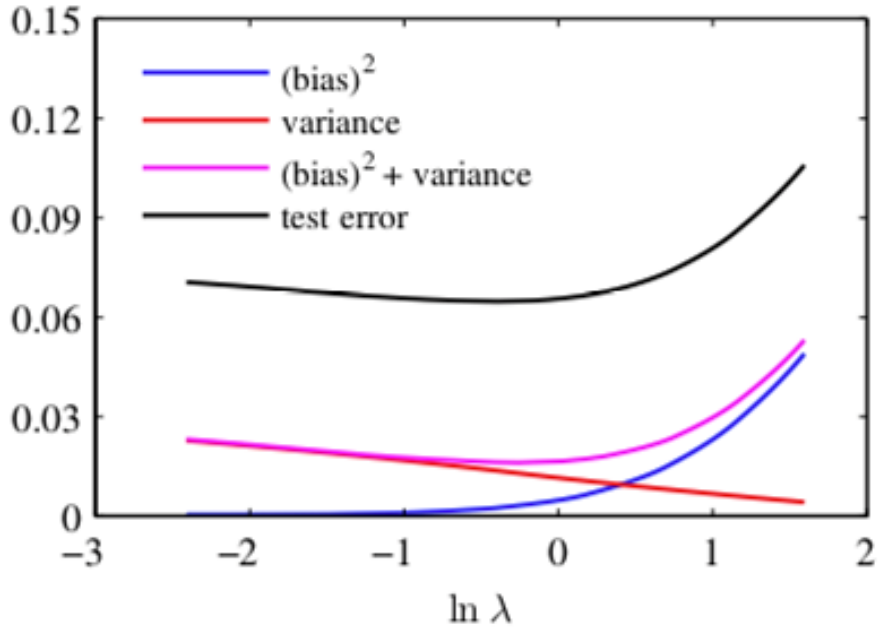
- There are $L = 100$ datasets, each having $n = 25$ data points, and there are 24 Gaussian basis functions in the model so that the total number of parameters, including the *bias*, is $M = 25$

$$f(x, \mathbf{w}) = w_0 + \sum_{j=1}^{24} w_j \cdot x^j = \sum_{j=0}^{25} \phi_j(x)$$

  - let us consider a l_2 regularization term (Ridge)

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \phi(\mathbf{x}_\eta))^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

  - the model with the optimal predictive capability is the one with the best balance between bias and variance

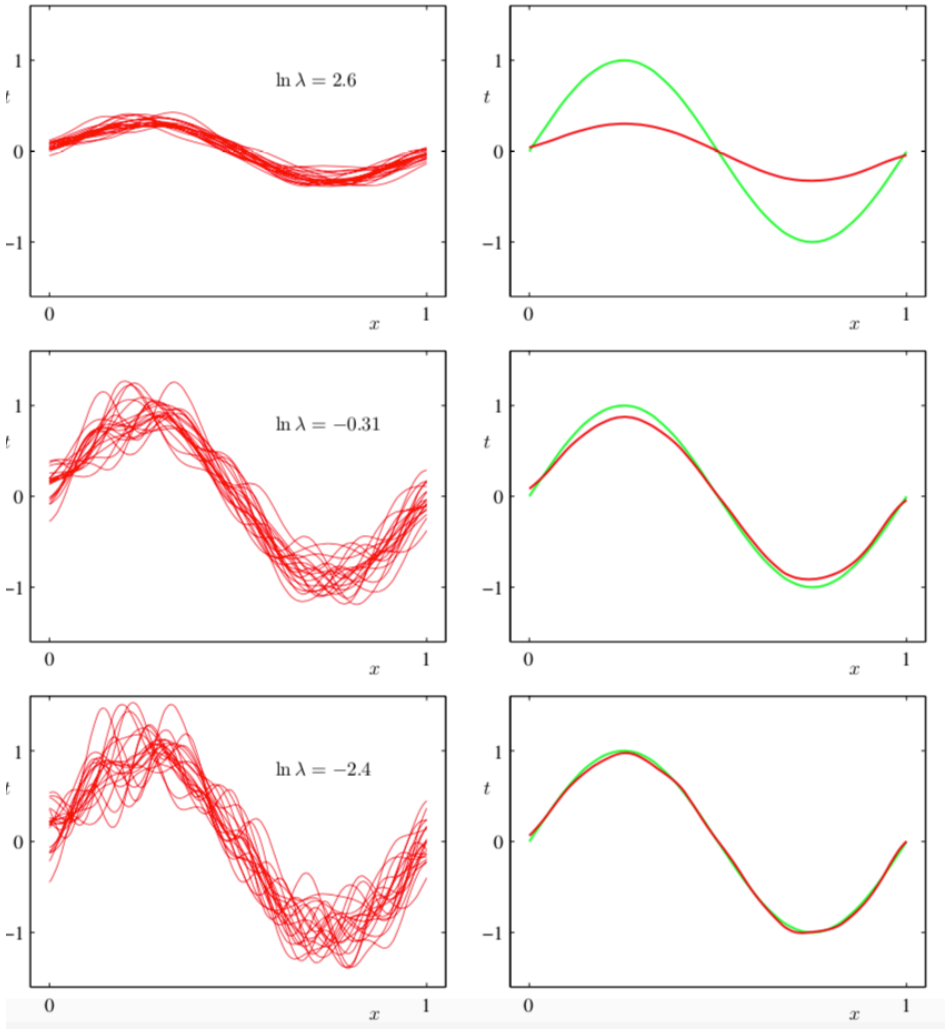


Bias and variance: example

- A large value of the regularization coefficient λ…
  - low variance
    (red curves in the left plot look similar)
  - high bias

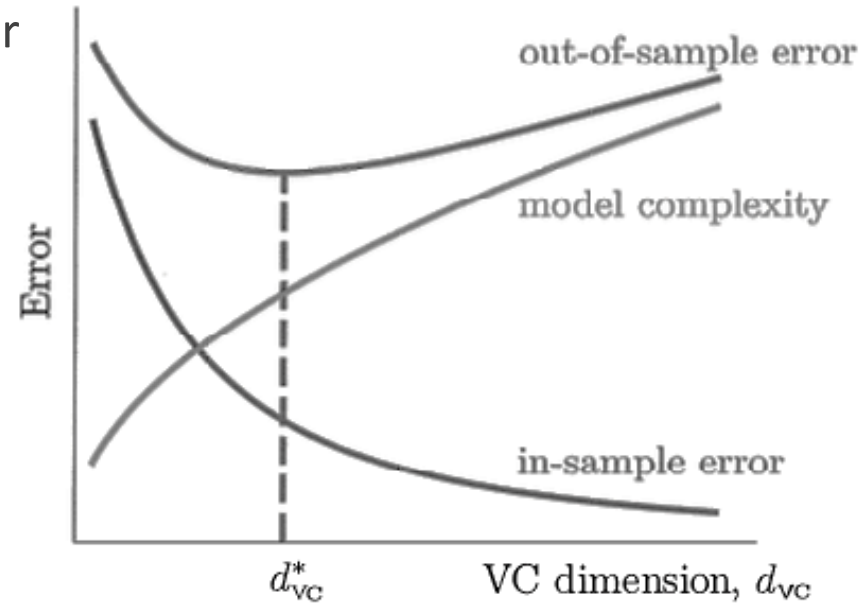
    (2 curves in the right plot are dissimilar)

- A small value of the regularization coefficient λ…
  - large variance (shown by the high variability between the red curves in the left plot)
  - low bias (shown by the good fit between the average model and original sinusoidal function)



VC dimension and generalization

- Vapnik-Chervonenkis (**VC**) **dimension**
  - measures the **complexity** (**capacity term**) of a predictor
- A more complex learning model has higher VC dim…
  - likely to fit the training data: low in-sample-error
  - yet increased **overfitting susceptibility**
- Best performing models have some intermediate optimal capacity, d^*_{VC}
- Learning theory community established formulas to assess the VC dimension of well-known predictors



Parameters and capacity term

- Recall: we have introduced two major types of models
  - **parametric**: Bayesian, regression, neural networks
  - **non-parametric**: kNN, decision trees
  - *parametric* is in reference to the learned parameters that define the model
    - e.g. probability functions and priors in Bayesian approaches or weights in MLPs
    - in contrast: k in kNN, depth in decision trees, learning rate in MLPs are generally fixed (not learned)
      - generally referred as *hyperparameters*

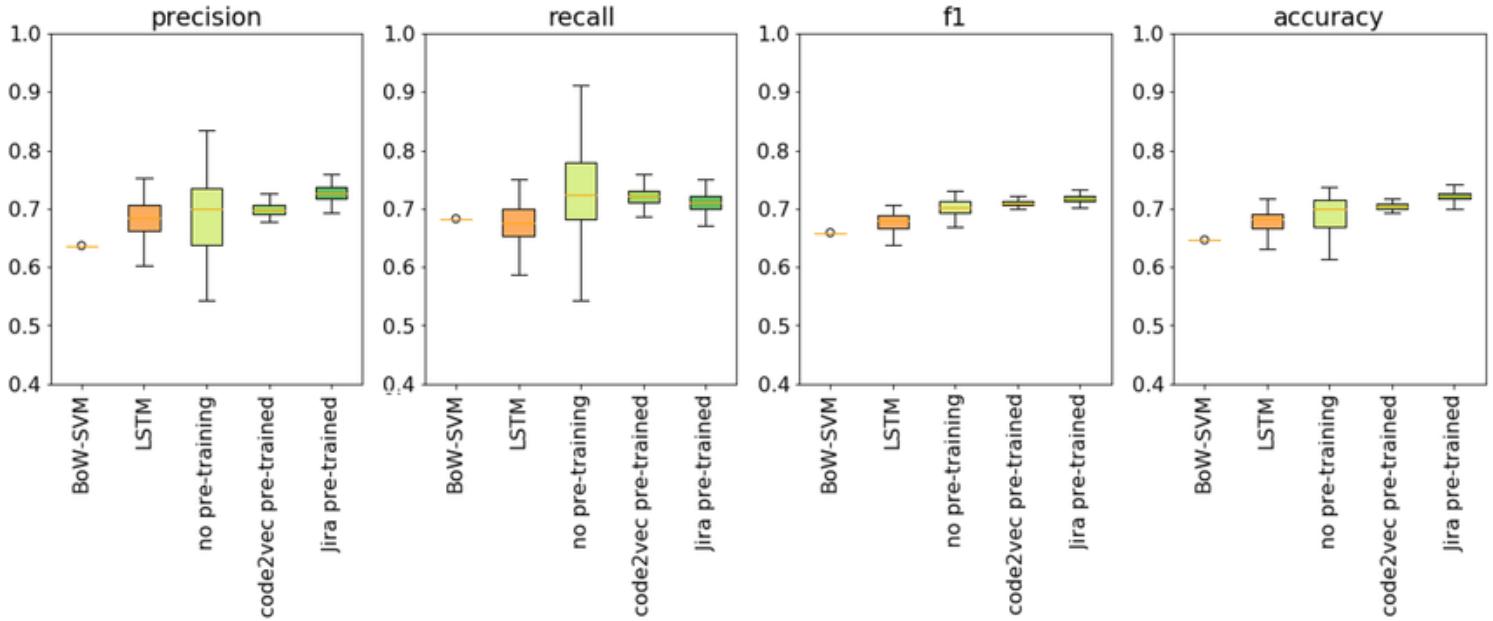
- The number of parameters in a parametric model can be seen as a proxy to estimate its capacity term
  - e.g. recall exercises on the number of parameters of Bayesian and regression approaches

Outline

- Evaluating regression models
  - loss functions
  - residue analysis
- Generalization ability
  - bias and variance
  - model complexity: VC dimension
- **Statistically testing performance differences**
- Advanced notes on classifier evaluation
  - ROC curves and AUC

Performance variability

- **Resampling** allowed the collection of multiple **error estimates**
  - CV allows testing on all observations
- Yet… the average performance across folds does not provide the whole view…
  - … **variability** of error estimates across folds!
  - variability assists us detecting significance differences in performance

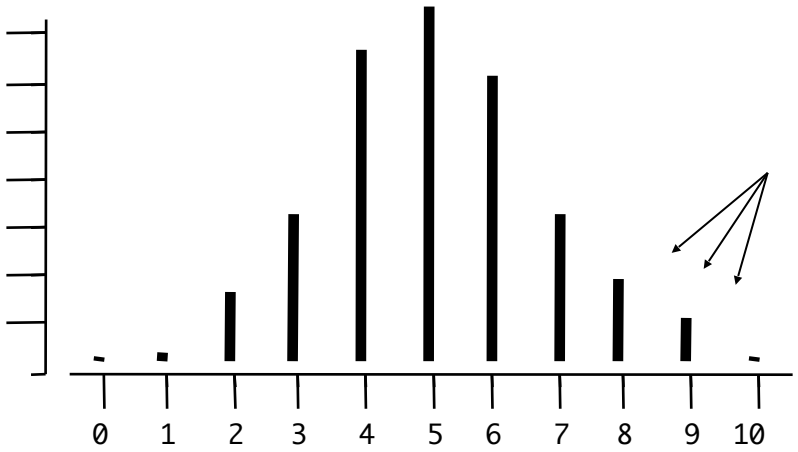


Statistical significance testing

- Is the performance of one predictor statistically superior than another?
  - can the difference be attributed to real characteristics of the algorithms?
  - can the observed difference(s) be merely coincidental concordances?
- Compare different
  - **learning approaches**
  - **parameterizations** under the same learning approach
- We primarily focus on *Null Hypothesis Significance Testing* (**NHST**)

Hypothesis testing

- Establish a null hypothesis
  - $H_0 : p = 0.5$, the coin is fair
- Establish a statistic
  - r: number of heads in n tosses
- Sampling distribution of r given H_0



  - sampling distribution tells us the probability *p* of a result as extreme as our sample result, e.g. $r = 8$
  - if this probability is very low, reject H_0 the null hypothesis

Statistical significance testing

- State a **null hypothesis**
  - usually the opposite of what we wish to test (e.g. classifiers A and B perform equivalently)
- Choose a **suitable statistical test** and statistic that will be used to (possibly) reject the null hypothesis
- Choose a **critical region** for the statistic to lie in/reject null hypothesis (commonly significance $\alpha = 0.05$)
  - if the test statistic lies in the critical region: 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 that our observations did not occur merely by chance
  - confidence *p* does not signify that the performance-difference holds with probability 1-p

- Statistical tests considered can be categorized by the **task** they address:
  - 2 algorithms on a single domain
  - 2 algorithms on several domains
  - multiple algorithms on multiple domains

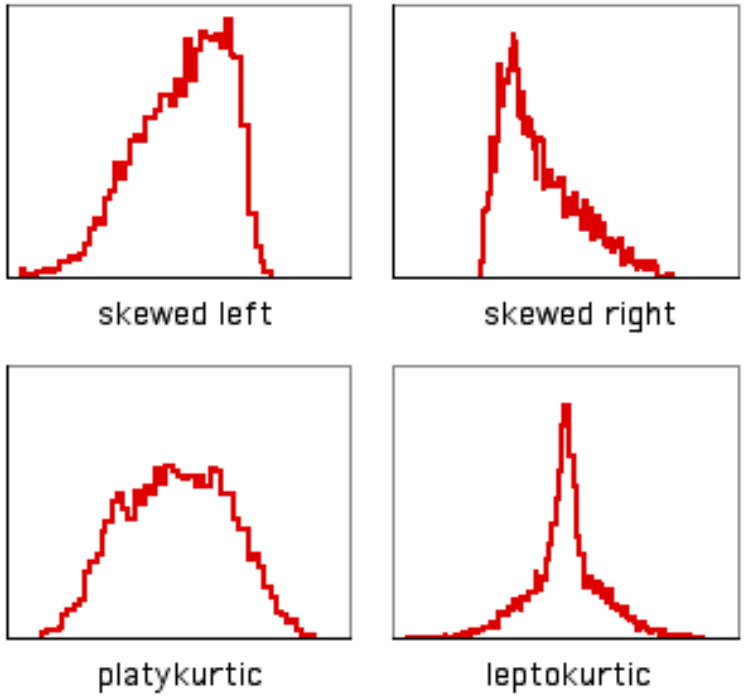
Comparing 2 models: *t*-test

- Arguably, one of the most widely used statistical tests
- Measures if the performance difference between the mean of two models is meaningful
  - can be applied with accuracy, recall, precision, F1
- When fixing folds with a seed: **paired sampled *t*-test**
  - estimates are drawn from the same testing folds
  - *single tail* if unidirectional superiority, $f_1 > f_2$
- Null hypothesis: the two samples (estimates from two predictors) have comparable mean

- Example:
  - C4.5 and Naïve Bayes (NB) algorithms on the *Labor* dataset
  - 10-fold CV (maintaining same folds across models)
  - t-statistic calculus gives us $t = 8.0845$, by checking the t-Student test table, for $k - 1 = 9$ degrees:
    - we can reject the null hypothesis at 0.001 significance level
    - i.e. the classifiers have distinct performance

Statistical significance testing

- Limitations of the *t*-test
  - assume error estimates are **normally distributed**
  - assume **equal variance** between populations
  - these conditions may not hold!
- Good practice
  - assess whether estimates follow normal
    - e.g. Shapiro-Wilk test
- If not! Solution? **Non-parametric testing**
  - McNemar's test
  - Wilcoxon rank test



Comparing multiple models

- Comparing **multiple algorithms** on multiple datasets (**omnibus** tests)
  - null hypotheses: all classifiers perform similarly
  - rejection means that there exists at least one pair of models with significantly different performance
- Two possibilities
  - parametric: ANOVA
  - non-parametric: Friedman's Test
- In case of rejection of this null hypothesis, the omnibus test is followed by a **post-hoc test**
  - to identify the significantly different pairs of classifiers

- *Exercise*:
  - collect accuracy and recall estimates of classifiers on the *iris* dataset using CV with a fixed seed
  - statistically test their differences

Outline

- Evaluating regression models
  - loss functions
  - residue analysis
- Generalization ability
  - bias and variance
  - model complexity: VC dimension
- Statistically testing performance differences
- **Advanced notes on classifier evaluation**
  - **ROC curves and AUC**

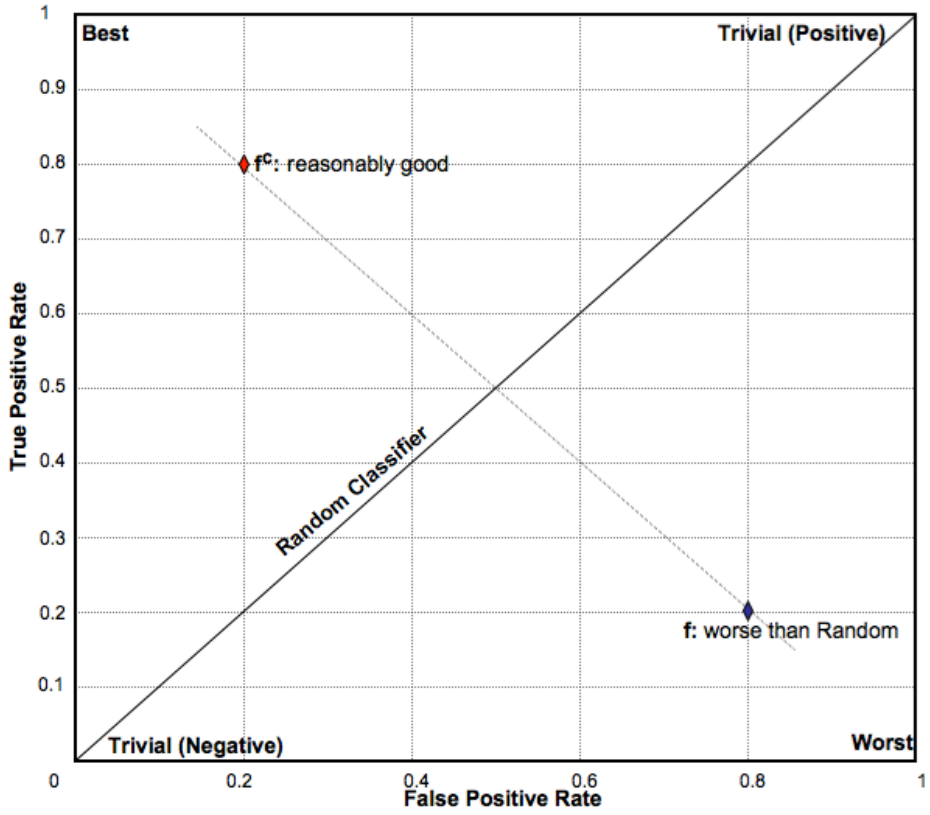
Graphical measures: ROC analysis

- Many classifiers are probabilistic in nature
  - $p(A) = 0.4$ and $p(B) = 0.6$
  - *k*NN can be as well probabilistic by returning the percentage of neighbors with concordant class!
- However, the default classification threshold $\theta = 0.5$ may not be the ideal one
  - at times biased towards the majority class
- Solution: use ROC analysis
  - more robust than accuracy in class imbalanced situations
  - takes the class distribution into consideration and, therefore, gives more weight to correct classification of the minority class

ROC curves

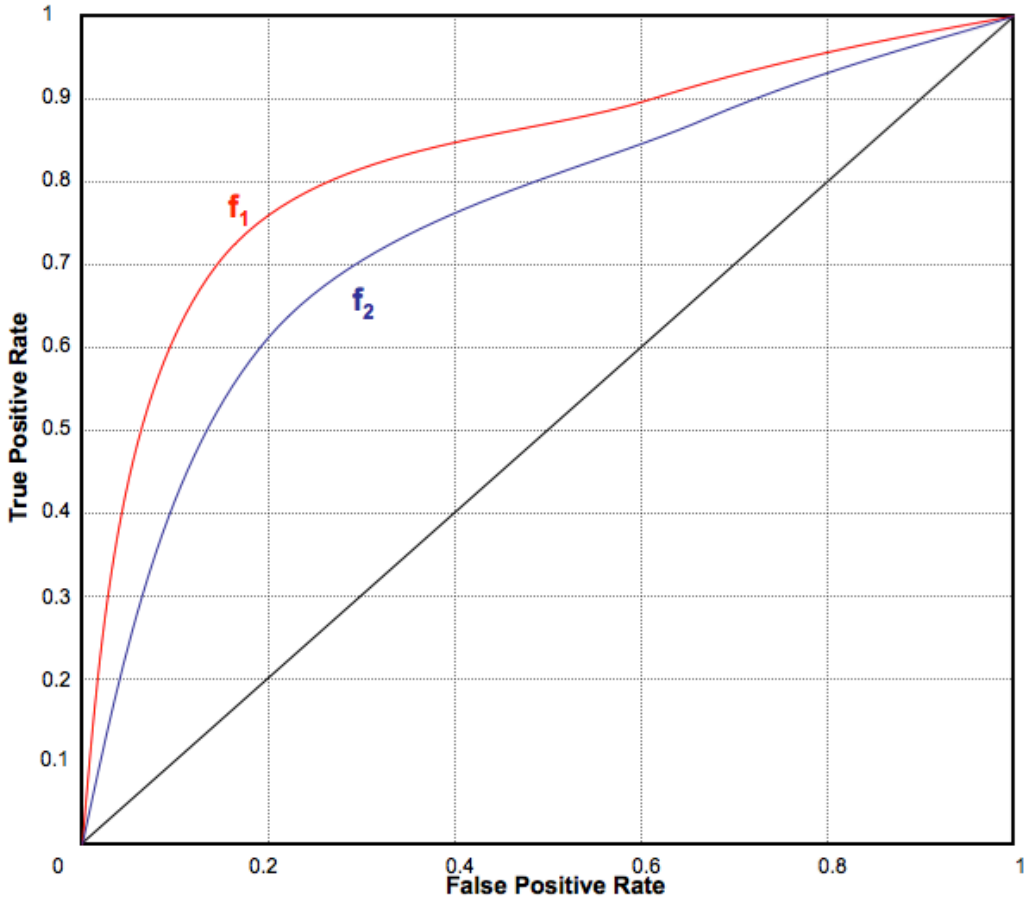
- ROC (receiver operating characteristics) analysis has origins in signal detection theory to set an operating point for desired signal detection rate

- ROC maps FPR on horizontal axis and TPR on the vertical axis. Recall:
  - FPR = 1 – Specificity
  - TPR = Sensitivity



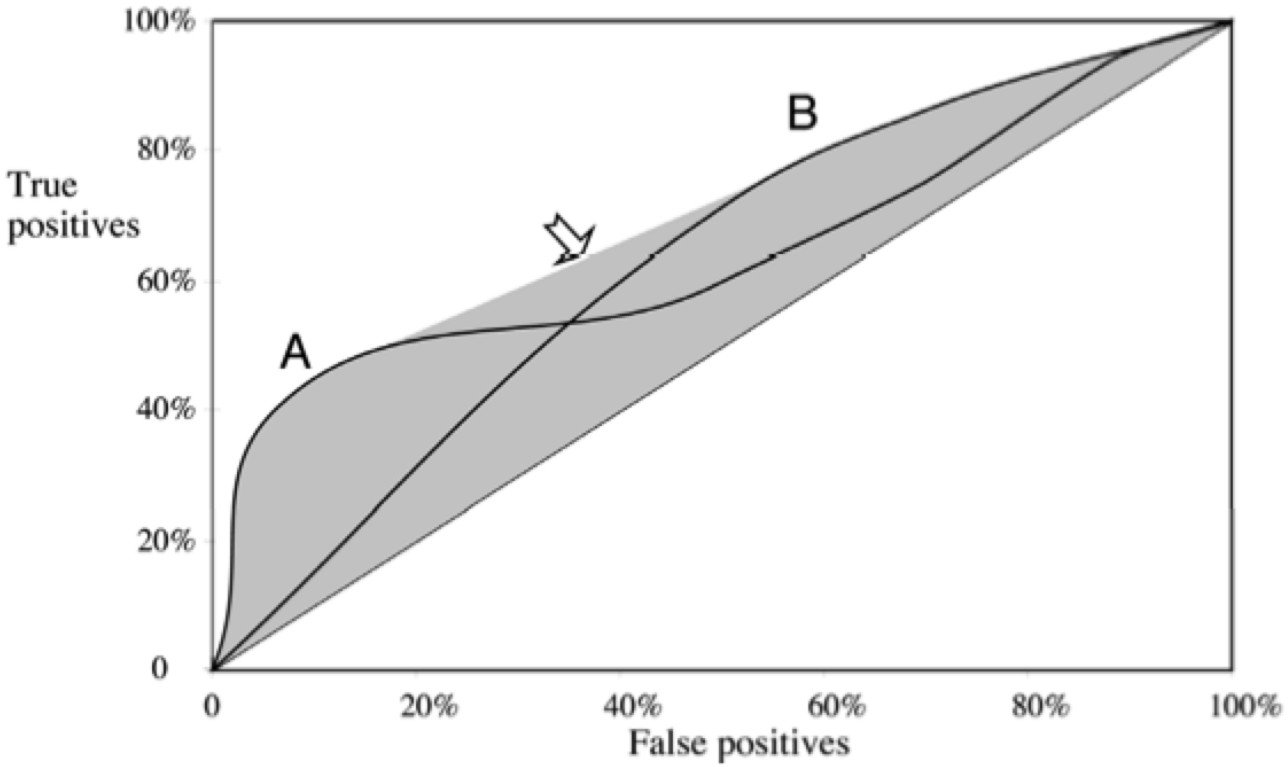
ROC curves

- By varying the decision threshold we can assess how a classifier performs in terms of sensitivity and specificity
  - f_1 better than f_2
- The area under the ROC curve is termed AUC, a common evaluation criterion
  - ideal classifier has an AUC of 1.0
  - random classifier has an AUC of 0.5



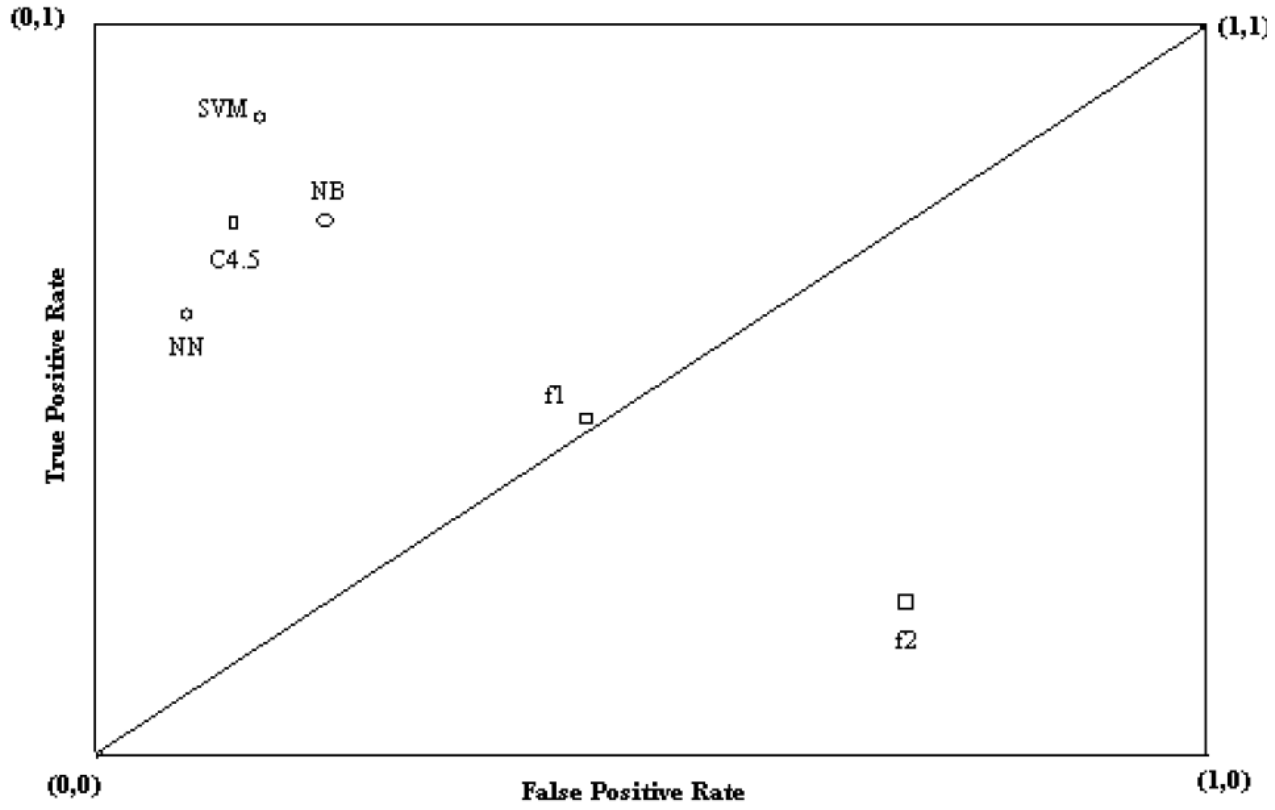
ROC curves

- Functions A and B with same AUC
- For a small, focused sample, use A
  For a larger one, use function B
  - Why?
- In between: choose according to the desirable characteristics



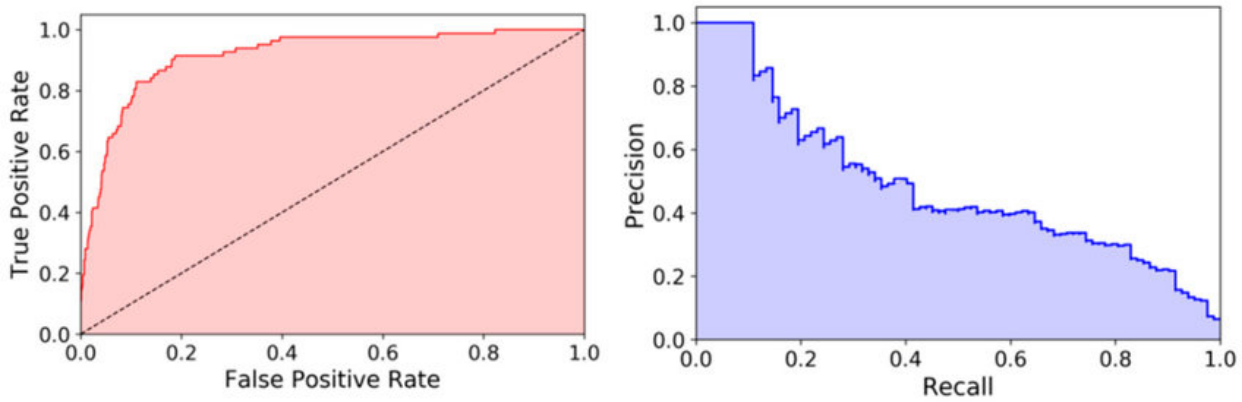
ROC curves

- We can select a desirable point along the curve for different classifiers
  - … and compare them with regards to TPR (sensitivity) and FPR (1-specificity)



Other curves

- Other research communities have produced and used graphical evaluation similar to ROC
  - **Precision-recall** curves: precision as a function of its recall



  - **Lift charts** plot the number of true positives versus the overall number of examples in the data sets that was considered for the specific true positive number on the vertical axis
  - **Detection Error Trade-off (DET)** curves: FNR rather than TPR on the vertical axis and log-scaled
  - **Relative Superiority Graphs** are more akin to Cost-Curves as they consider costs
    - ratios of costs into the [0,1] interval

Outline

- **Evaluating regression models**
  - loss functions
  - residue analysis
- **Generalization ability**
  - bias and variance
  - model complexity: VC dimension
- **Statistically testing performance differences**
- **Advanced notes on classifier evaluation**
  - ROC curves and AUC

Thank You

miguel.j.couceiro@tecnico.ulisboa.pt
andreas.wichert@tecnico.ulisboa.pt