Modern regression and classification techniques in computational biology

Michael G. Schimek

Medical University of Graz
Institute for Medical Informatics, Statistics and Documentation
8036 Graz, Austria, Europe

Lectures 2009
IBA, Masaryk University

What are typical tasks in computational biology and the health sciences?

The two most important ones we focus on in this series of lectures are:
Regression of quantitative (dependent) measurements on quantitative (explanatory) measurements in a designed study (prior information)
Classification of quantitative measurements with respect to a qualitative response in a designed study (prior information)

Both tasks can be interpreted in the context of **supervised statistical learning** in the spirit of Hastie, Tibshirani, and Friedman (2009) [HTF'09]
However, both tasks are associated with **self-contained statistical methodologies**

Statistical methods in computational biology and the health sciences 1

In terms of **regression** we have

- the classical **linear model** under Normal errors
- the parametric class of **generalized linear models** under Exponential Family errors
- the nonparametric class of **generalized additive models** under Exponential Family errors
- several extensions of the above such as **semiparametric models**, **mixed models**, etc.
- adaptations of the above methods to control violations of assumptions (e.g. overdispersion, multicollinearity, high dimensionality)

Statistical methods in computational biology and the health sciences 2

In terms of **classification** we have

- Fisher's linear discriminant analysis
- Linear and quadratic discriminant analysis
- Extensions of linear discriminant analysis (e.g. flexible discriminant analysis, shrinkage methods)
- Nearest neighbor classifiers
- Support vector machines

Fundamental statistical learning goal: Useful information reduction with respect to responses (regression) or classifiers (classification)

Advanced statistical learning goal: Response (regression) or class prediction (classification) for new measurements or objects or cases

Data sources and measurement characteristics

Data sources

- Prospective studies and experiments
- Retrospective studies (e.g. from data bases)
- Ex-post-facto analysis
- Meta analysis (combining study outcomes)
- Genetic analysis
- Image analysis

Measurement characteristics

- Quantitative (metric scale, integer or real valued)
- Qualitative (ordered or unordered units)
- Resolution resp. units on measurement scale
- Distributional features (data sparseness)
- Proportion and distribution of missing observations
- Error sources (systematic or stochastic)
- Signal-to-noise ratio

Data structures

- Sample data (representative for population)
- Reasonable sample size (required for all classical biostatistics methods)
- Very small sample size (unsuitable for most classical biostatistics methods)
- Huge sample size (unsuitable for most classical biostatistics methods)
- $N > p$, where N is sample size and p is number of parameters (required for all classical biostatistics methods)
- $N \ll p$ leading to ill-posed estimation problems (classical biostatistics methods fail)
- Reasonable number of dimensions (parameters; classical biostatistics methods apply)
- Large number of dimensions (parameters; classical biostatistics methods fail)

Medizinische Universität Graz

Conventions and goals

We assume

- outcome measurements y (also called dependent variable, response, target)
- vector of p predictor measurements x (also called independent variables, inputs, covariates, features)
- in the regression problem, y is quantitative
- in the classification problem, y takes values in a finite, unordered set
- empirical (training) data $(x_1, y_1), \ldots, (x_N, y_N)$

Goals are

- decision support (based on inference)
- to understand which input effects output
- model fitting (identification, estimation and verification)
- prediction (of future cases)
- assessment of the quality of inference and prediction

Medizinische Universität Graz

Statistical decision theory basics

- Assume quantitative output Y
- Assume random input vector $X \in \mathbb{R}^p$
- Assume arbitrary function f (e.g. separating two classes)
- Let us have **loss function** $L(Y, f(X))$ for penalizing errors in prediction
- Most common and convenient is **squared error loss** $L(Y, f(X)) = (Y - f(X))^2$
- We can choose f according to **expected** squared **prediction error** $EPE(f) = E(Y - f(X))^2$
- Thus the solution is the **conditional expectation** $f(x) = E(Y \mid X = x)$, also denoted regression function

The elementary statistical method is regression
There are many ways to estimate the regression function from (training) data

Case of categorical response variable 1

- Let us have prediction rule $C(X)$, and Y and $C(X)$ take values in $C = 1, 2, \ldots, K$
- The loss function for penalizing prediction errors is $L(k, l)$, i.e. the price to be paid for classifying an observation belonging to class k as l
- For a zero-one loss function all misclassifications are charged one unit
- The **expected prediction error** is $EPE = E[L(Y, C(X))]$
- The solution is $\hat{C}(x) = argmin_{c \in C} \sum_{k=1}^{K} L(k, c) P(Y = k \mid X = x)$
- Assuming a zero-one loss function we simply get $\hat{C}(x) = k$ if $P(k \mid X) = max_{c \in C} P(c \mid X)$, the so-called **Bayes classifier** (NB: we should decide for the class with maximum probability at input x)

Case of categorical response variable 2

Task is to construct a classifier $C(X)$

- by estimating probabilities $P(c \mid X)$ from the data
- or by estimating class densities $P(X \mid c)$ in combination with Bayes rule

There are many different approaches (for quantitative as well as categorical response variables)

- It is important to understand the ideas behind these approaches
- Otherwise one cannot decide when to use them
- Further it is important to assess their performance
- In contrast to machine learning, statistical approaches (algorithms) suitable for specific responses (output) are not compared with respect to performance only

Linear regression as elementary method

- Regression function $\eta(x) = E(Y \mid X)$ where X is design or data matrix
- A linear form $\eta(x_i) = \beta_0 + \sum_{j=1}^{p-1} x_{ij}\beta_j$ is assumed, which is an approximation to the truth
- In matrix notation $\eta = X\beta$, where η is N-dimensional vector of predicted values, X an $N \times p$ matrix with ones in the first column, and β a p-dimensional vector of parameters
- Fitting (estimation) by least squares
- $\hat{\beta} = argmin \sum_i (y_i - \beta_0 - \sum_{j=1}^{p-1} x_{ij}\beta_j)^2 = argmin(y - X\beta)^T(y - X\beta)$
- Solution is $\hat{\beta} = (X^T X)^{-1} X^T y$ and $\hat{y} = X\hat{\beta}$
- Variance of estimator is $var(\hat{\beta}) = (X^T X)^{-1}\sigma^2$

Medizinische Universität Graz

Linear regression of binary response [HTF'09]

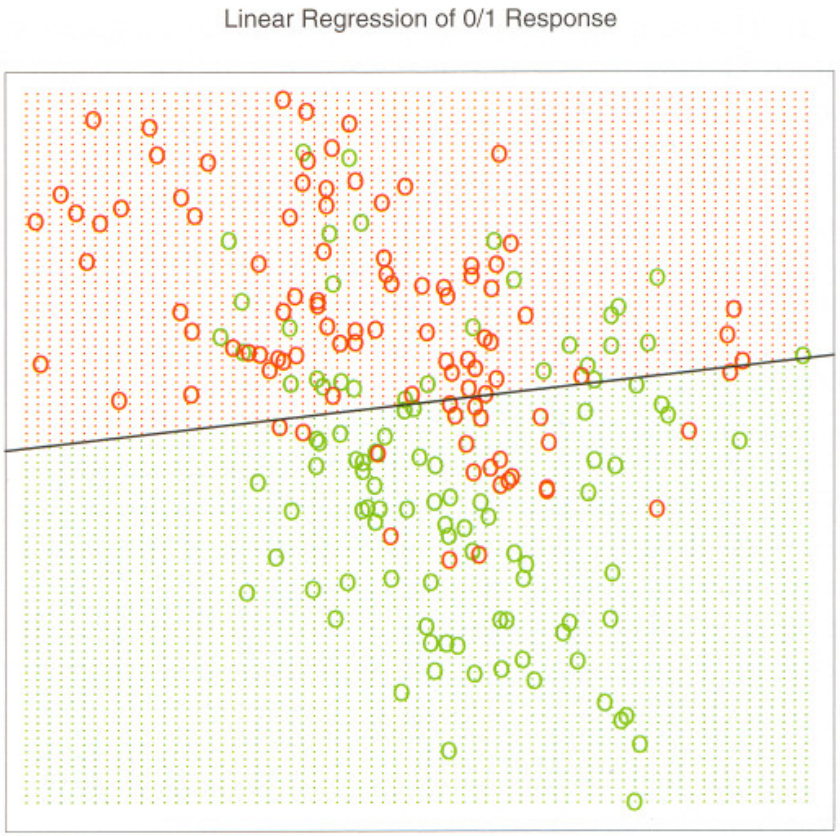


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable—*GREEN $= 0$, RED $= 1$*—and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$. *The red shaded region denotes that part of input space classified as* RED, *while the green region is classified as* GREEN.

Nearest-neighbor methods

- No linear decision boundary as in least-squares regression
- No stringent assumption about the data, they act data-driven (adaptive)
- Those observations in the training data are used that are closest in input space to x to form $\hat{Y}$
- The k-nearest neighbor (k-NN) fit for $\hat{Y}$ is defined $\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$, where $N_k(x)$ is the neighborhood of x defined by the closest points x_i in the training sample
- Metric for closeness: e.g. Euclidean distance
- Effective number of parameters in k-NN is not k but $\frac{N}{k}$ (decreases with increasing k due to overlapping neighborhoods)
- A sum-of-squared-error criterion does not work here (would always pick $k = 1$!)

Medizinische Universität Graz

15-NN classifier of binary response [HTF'09]

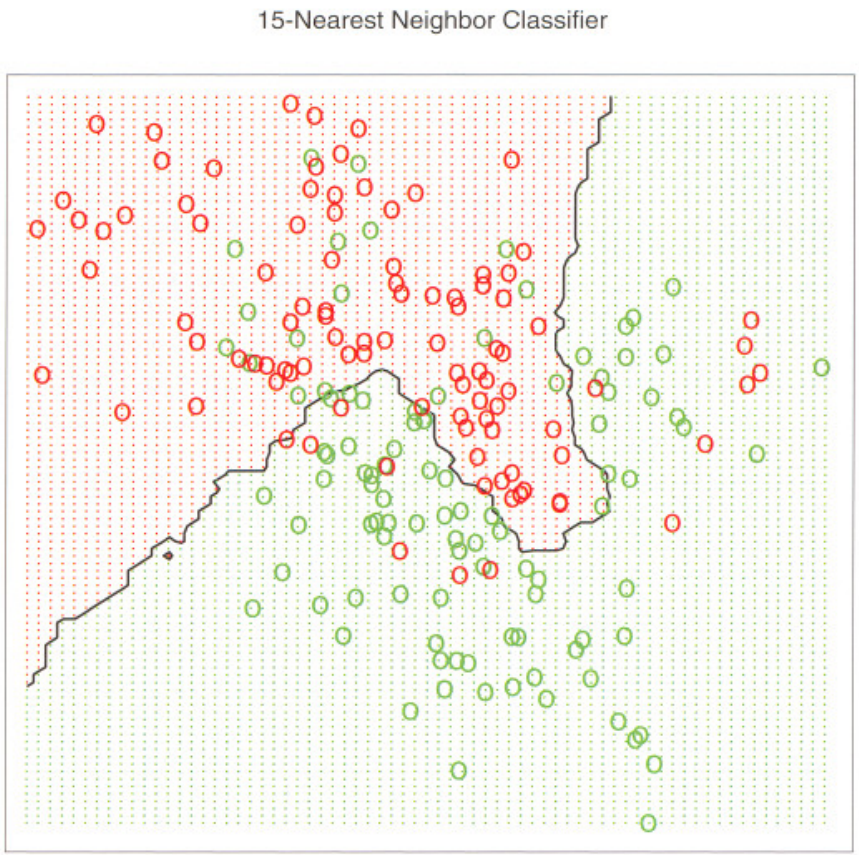


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (GREEN $= 0$, RED $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

1-NN classifier of binary response [HTF'09]

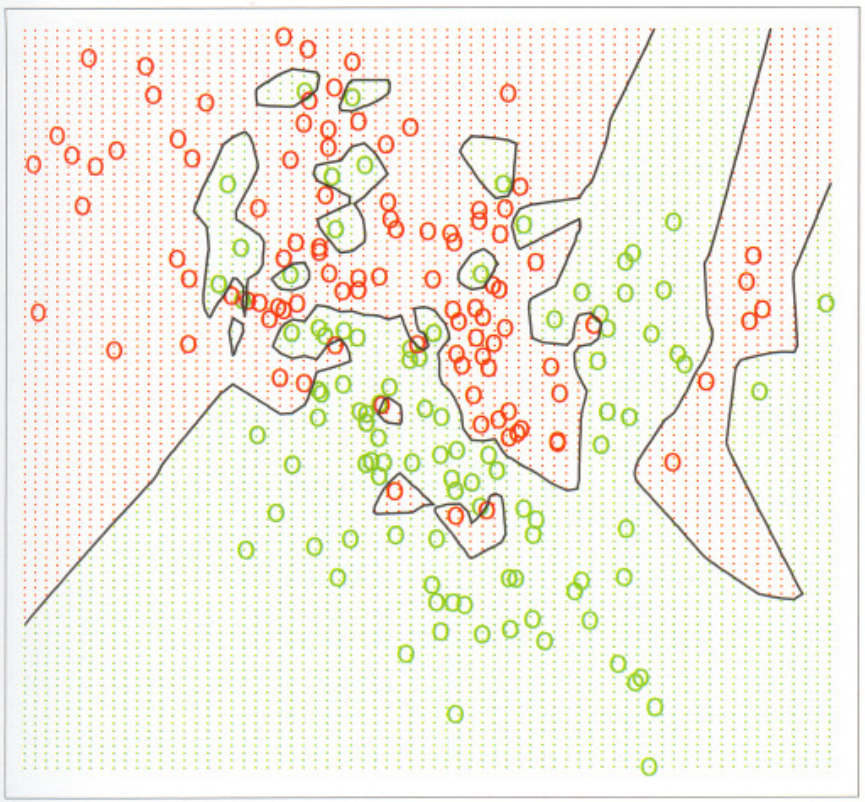


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (GREEN $= 0$, RED $= 1$), *and then predicted by* 1*-nearest-neighbor classification.*

Simple extensions of least squares and k-NN methods

- Kernel methods use weights that decrease smoothly to zero with distance from the target point (in contrast to 0/1-weights in k-NN)
  - k-NN methods are the simplest form of smoothing methods
  - Kernel methods belong also to the class of smoothing methods (they also comprise smoothing splines yet not motivated by weighting schemes)
- For high-dimensional problems kernel methods can be modified: some variables obtain higher weights than others
- Local regression, another smoothing method, fits linear models by locally weighted least squares
- Linear models fit to a basis expansion of the original inputs can cope with rather complex data structures

Medizinische Universität Graz

Some more decision theory 1

How can we best predict Y at any point $X = x$?

Least squares regression:
Minimizing the expected prediction error EPE has the solution

$$f(x) = E(Y \mid X = x),$$

thus the best predictor is the conditional mean (in terms of average squared error)

***k*-nearest neighbor method:**
Implements the above idea directly using the training data

$$\hat{f} = average(y_i \mid x_i \in N_k(x)),$$

where $N_k(x)$ is the neighborhood of x defined by the closest points x_i in the training sample

Medizinische Universität Graz

Misclassification curves for simulated data: linear regression vs. k-NN [HTF'09]

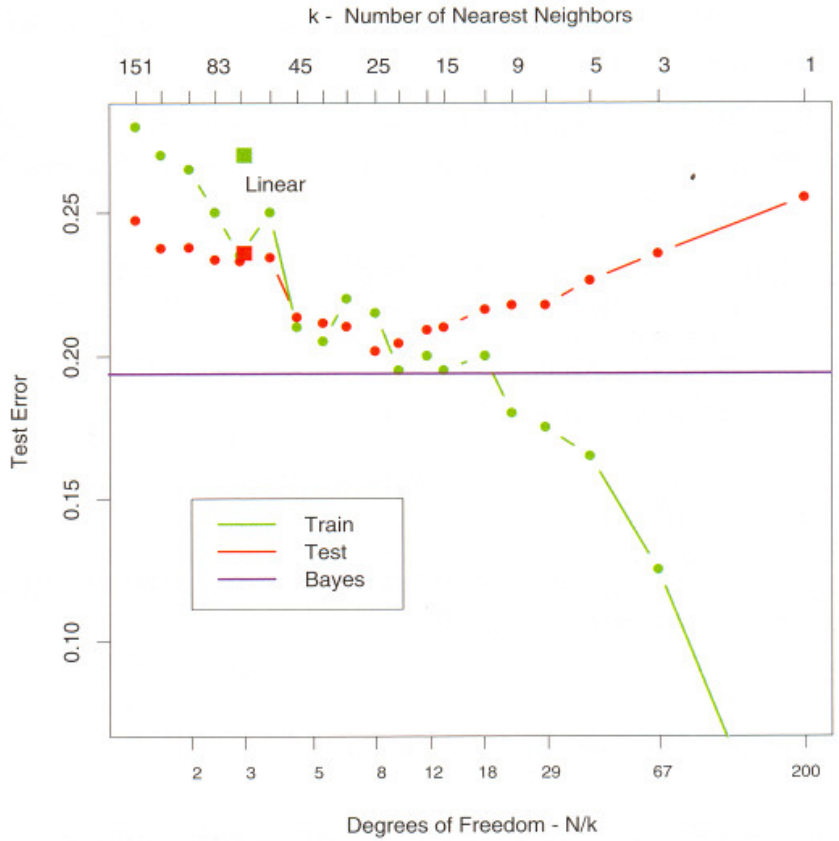


FIGURE 2.4. *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* 10,000. *The red curves are test and the green are training error for k-nearest-neighbor classification. The results for linear regression are the bigger green and red dots at three degrees of freedom. The purple line is the optimal Bayes Error Rate.*

Medizinische Universität Graz

Some more decision theory 2

There are two approximations

- Expectation is approximated by averaging over sample (training) data
- Conditioning at a point is relaxed to conditioning on some region ”close” to the target point

For large N the points in the neighborhood are likely to be close to x, and as k increases the average will stabilize
Under mild regularity conditions on the joint probability distribution $Pr(X, Y)$ **one can show that**
as $N, k \to \infty$ such that $\frac{k}{N} \to 0$, $\hat{f} \to E(Y \mid X = x)$
In practice this does not help as we do not have very large N
Further problem: **in high dimensions the metric size of the** k**-nearest neighborhood gets disproportional large**
The convergence still holds but the **rate of convergence decreases as the dimension increases** (curse of dimensionality)

Measures for quality of an estimator 1

A valuable quality measure for $\hat{\eta}(x)$ is mean squared error
Let $\eta_0(x)$ at point x be the true value of $\hat{\eta}(x)$
Then

$$MSE(\hat{\eta}(x)) = E(\hat{\eta}(x) - \eta_0(x))^2$$

This can be written as variance plus squared bias

$$MSE(\hat{\eta}(x)) = var(\hat{\eta}(x)) + (E\hat{\eta}(x) - \eta_0(x))^2$$

We have the following relationship: low bias with high variance, and vice versa

Consequences

- Selecting an estimator involves a **tradeoff between bias and variance**
- Even for the simplest data structure resp. model this is true
- For ill-posed problems (e.g. data with $N \ll p$), where regularization is required, the selection problem becomes even more delicate

Measures for quality of an estimator 2

Predictive ability of k-NN regression fit $\hat{f}_k(x_0)$
Assume data from $Y = f(X) + \epsilon$ with $E(\epsilon) = 0$ and $var(\epsilon) = \sigma^2$
Let x_i being fixed (non-random)
The **expected prediction error** at x_0, also called **generalization** (test) **error** is

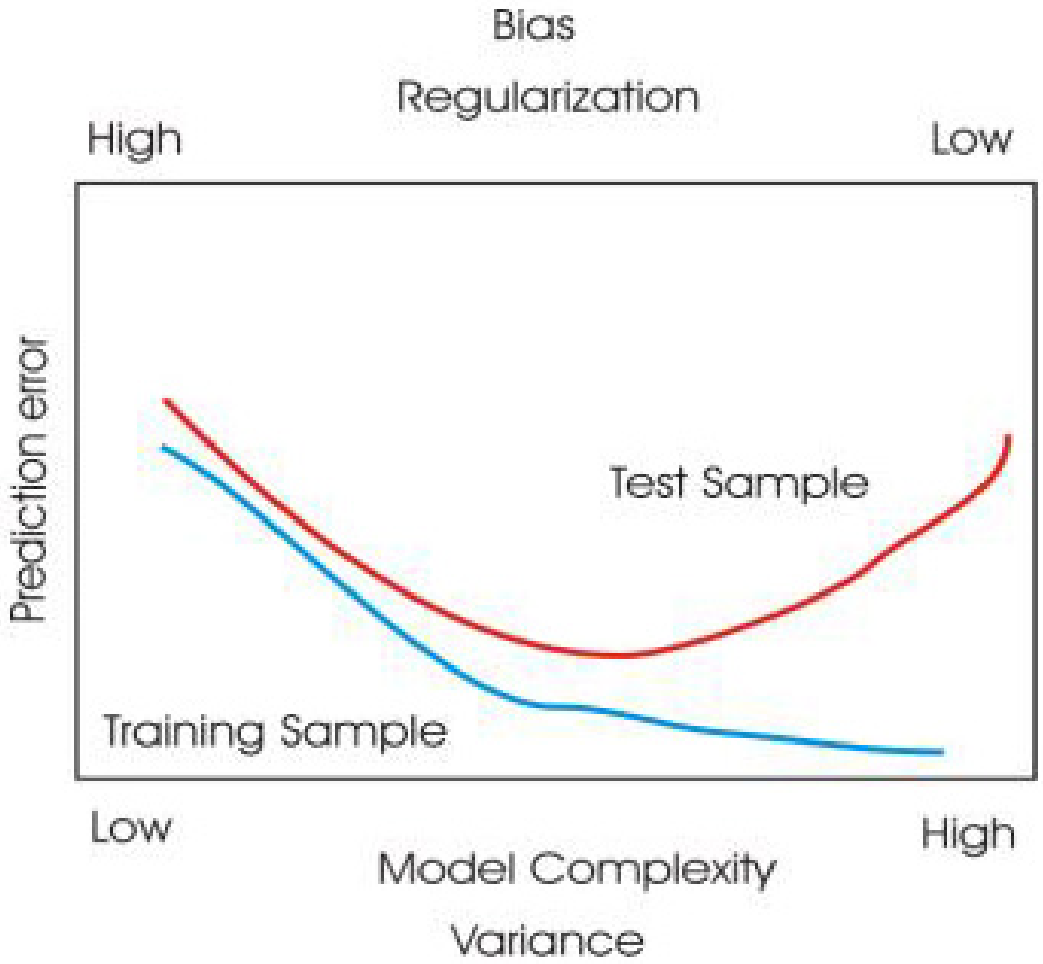
$$EPE_k(x_0) = E[(Y - \hat{f}_k(x_0))^2 \mid X = x_0]$$

$$= \sigma^2 + [bias^2(\hat{f}_k(x_0)) + var(\hat{f}_k(x_0))]$$

$$= \sigma^2 + [f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})]^2 + \frac{\sigma^2}{k}$$

where (l) is the sequence of NN to x_0 (k is model complexity)
σ^2, the irreducible error (variance of new test case) cannot be controlled; mean squared error of $\hat{f}_k(x_0)$ (a bias and a variance component) can be controlled

Bias-variance-tradeoff



Tradeoff between bias and variance

- How about a linear regression model? If it is correct for a given data structure, then the **least squares predictor** $\hat{\eta}$ is **unbiased** and has the **lowest variance among all estimators** that are linear functions of y
- There can be biased estimators with smaller MSE
- When an estimator is regularized (penalized, shrunken), its variance will be reduced
- Examples of regularization: subset selection (forward, backward, all subsets), ridge regression, the lasso
- A further limitation: **empirical models are never correct** which leads to an additional model bias between the closest member of the (e.g. linear) model class and the (unknown) truth

Sources of bias and estimation variance [HTF'09]

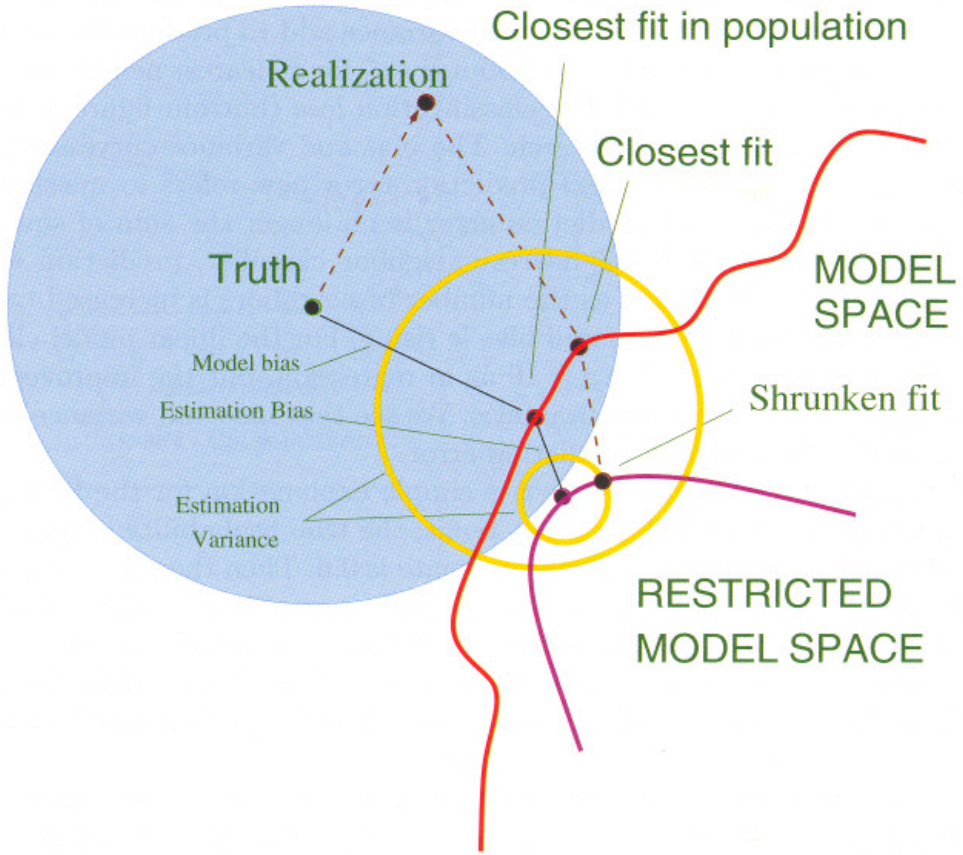


FIGURE 7.2. *Schematic of the behavior of bias and variance. The model space is the set of all possible predictions from the model, with the "closest fit" labeled with a black dot. The model bias from the truth is shown, along with the variance, indicated by the large yellow circle centered at the black dot labelled "closest fit in population". A shrunken or regularized fit is also shown, having additional estimation bias, but smaller prediction error due to its decreased variance.*

Medizinische Universität Graz

Subset selection 1

Subset selection (of predictive variables or features) is a form of shrinkage
Motivations are

- prediction accuracy: least squares estimators can have low bias but large variance (can reduce the variance)
- interpretation: easier with small well-predicting subset

Hence only a subset of parameters (e.g. regression coefficients) is retained and the remaining ones are set to zero
There are several selection strategies
All subset regression: Finds for each $s \in 0, 1, 2, \ldots, p$ the subset of size s that gives the smallest residual sum of squares
This involves the tradeoff between bias and variance (e.g. can use cross-validation)

Subset selection 2

A search strategy can improve the performance

- **Forward stepwise selection:** Sequentially starting with the intercept, variables are added into the model that most improve the fit; typical criterion is

$$F_{1,N-s-2} = \frac{RSS(\hat{\beta}^{old}) - RSS(\hat{\beta}^{new})}{RSS(\hat{\beta}^{new})/(N-s-2)}$$

- **Backward stepwise selection:** A full least squares model is fitted, then variables are sequentially eliminated based on F-statistic (requires $N > p$)
- **Hybrid stepwise selection:** Sequentially for each (best) variable added the least important variable is deleted

Subset selection and shrinkage

The subset selection procedures need one or more tuning parameters

- Subset size
- Position along stepwise path

Shrinkage methods, related to subset selection, have a **penalty or ridge parameter** (also true for certain smoothers, e.g. smoothing splines)
Subset selection does not automatically reduce the prediction error (the discrete selection process can retain a high variance)
Shrinkage methods are a continuous alternative
Ridge regression shrinks the coefficients by imposing a penalty on their size

Ridge regression 1

The ridge coefficients minimize a penalized residual sum of squares

$$\hat{\beta}^{ridge} = argmin_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

where $\lambda \geq 0$ is the ridge (complexity) parameter weighting the penalty – the coefficients are shrunk towards 0
Equivalently we can write

$$\hat{\beta}^{ridge} = argmin_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

subject to

$$\sum_{j=1}^{p} \beta_j^2 \leq s,$$

a constraint on the parameters

Ridge regression 2

Ridge solutions are not equivariant under scaling of inputs (input has to be standardized)
Further there is no intercept β_0 (X has p rather than $p+1$ columns)
In **matrix notation** we have

$$RSS(\lambda) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$$

The solutions are

$$\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T y,$$

where I is $p \times p$
Because of a **quadratic L_2 ridge penalty** $\beta^T\beta$ the solution is a linear function of y
One might chose other penalties too

The lasso 1

Lasso is another **shrinkage method** with important differences
The estimate is

$$\hat{\beta}^{lasso} = argmin_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

subject to

$$\sum_{j=1}^{p} \mid \beta_j \mid \leq s$$

The solution for β_0 is $\bar{y}$ (no intercept needed)
The **lasso penalty** is L_1, leading to a non-linear estimation problem
A quadratic programming algorithm is required

The lasso 2

The **standardized tuning parameter** s is

$$s = \frac{t}{\sum_{j=1}^{p} \mid \hat{\beta}_j \mid},$$

where t is the penalty parameter

Relationship between lasso and ridge coefficients:

If t is chosen larger than $t_0 = \sum_{j=1}^{p} \mid \beta_j^{ls} \mid$ then $\hat{\beta}^{lasso} = \hat{\beta}_j^{ls}$

For instance for $t = t_0/2$ the least squares (*ls*) estimates are shrunken by around 50%

For $s \rightarrow 0$ the *ls* estimates tend to zero

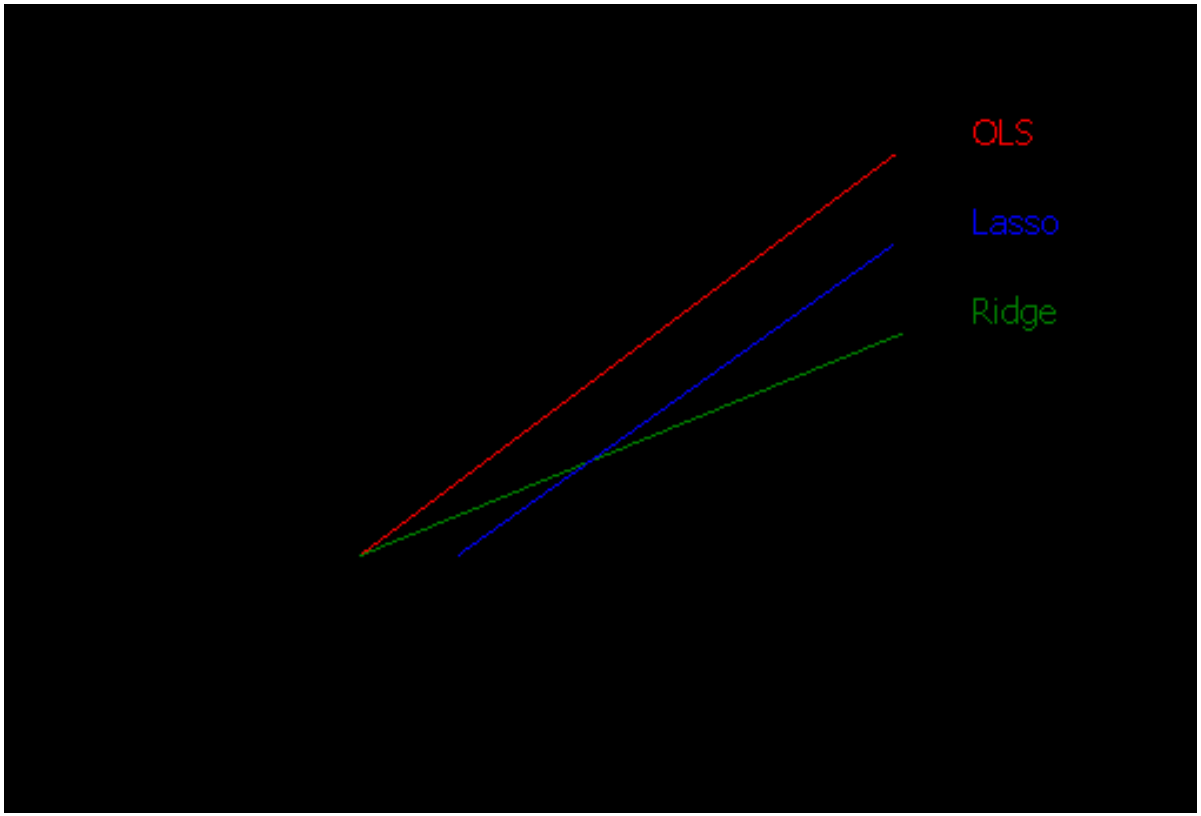
Lasso translates each *ls* coefficient by a constant factor, truncating at zero

Ridge however does a proportional shrinkage on the *ls* coefficients

t can be chosen by cross-validation

Medizinische Universität Graz

Transformation of least squares coefficients in ridge regression and lasso



Medizinische Universität Graz

The curse of dimensionality 1

Multidimensional smoothers work well for moderate numbers of predictors, but the curse of dimensionality hinders them in higher dimensions

Problems are

- local neighborhoods are empty
- nearest-neighborhoods are not local
- all points are close to the boundary
- samples size needs to grow exponentially

Moreover without some structure in the models high-dimensional functions are hard to represent and interpret

Illustration:

- Uniformly distributed data in p-dimensional unit cube
- Construct subcube from origin to capture fraction f of the data
- What distance do we have to reach out on each axis?

Medizinische Universität Graz

The curse of dimensionality 2 [HTF'09]

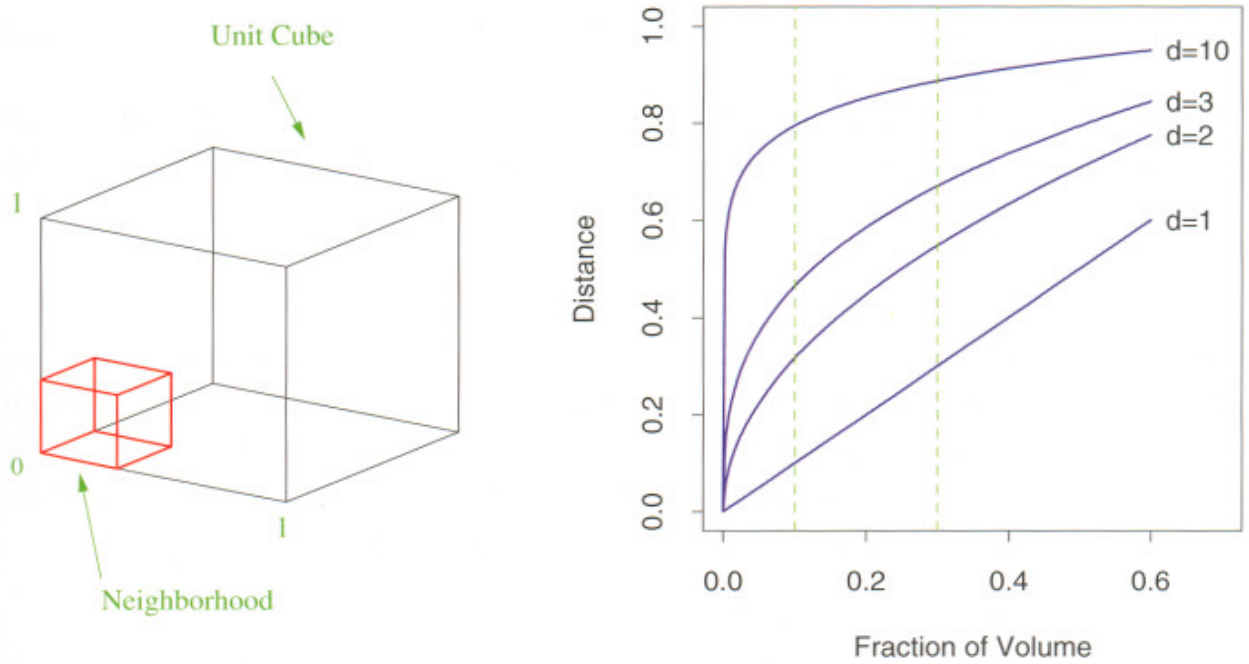


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

Medizinische Universität Graz

Generalized linear models 1

Generalized linear models (GLM) extend linear (regression) models to **accommodate both non-normal response and transformations to linearity**
Characteristics of a GLM:

- Response y observed independently at fixed values of explanatory variables $x_1, x_2, \ldots, x_p$
- The explanatory variables x_j may only influence the distribution of y through a single linear function called **linear predictor** $\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$
- The probability density function of y is assumed to follow the form of the **exponential family distributions**
- Mean μ is a smooth invertible function of the linear predictor $\mu = m(\eta), \eta = m^{-1}(\mu) = G(\mu)$, where the inverse function G is called the **link function**

Generalized linear models 2

Probability density function of **exponential family distributions**

$$f(y_i;\theta,\varphi) = \exp\left\{\frac{A_i[y_i\theta_i - \gamma(\theta_i)]}{\varphi + \tau(y_i,\varphi/A_i)}\right\},$$

where φ is a scale parameter (possibly known), A_i a known prior weight, and θ_i depends upon the linear predictor
θ is an invertible function of μ, $\theta = (\gamma\prime)^{-1}(\mu)$
For known φ the distribution of y is a one-parameter canonical exponential family

The Gaussian case

For Normal distribution $\varphi = \sigma^2$ and

$$\log f(y) = \frac{1}{\varphi}\left[y\mu - \frac{1}{2}\mu^2 - \frac{1}{2}y^2\right] - \frac{1}{2}\log(2\pi\varphi)$$

so $\theta = \mu$ and $\gamma(\theta) = \frac{\theta^2}{2}$

The Poisson case

For a Poisson distribution with mean μ we have

$$\log f(y) = y \log \mu - \mu - \log(y!)$$

so $\theta = \log \mu$, $\varphi = 1$ and $\gamma(\theta) = \mu = \exp\{\theta\}$

Each response (error) distribution allows one or more link functions to connect the mean μ to the linear predictor

Table: Canonical default links and variance functions

Family	Canonical link	Name	Variance
Binomial	$\log(\mu/(1-\mu))$	logit	$\mu(1-\mu)$
Gamma	$-1/\mu$	inverse	μ^2
Gaussian	μ	identity	1
Inverse Gaussian	$-2/\mu^2$	$1/\mu^2$	μ^3
Poisson	$\log \mu$	log	μ

Generalized linear models 4

The log-likelihood of a GLM is

$$l(\theta, \varphi; Y) = \sum_i \left\{ \frac{A_i[y_i\theta_i - \gamma(\theta_i)]}{\varphi} + \tau[y_i, \frac{\varphi}{A_i}] \right\}$$

The score function for θ is

$$U(\theta) = \frac{A_i[y_i - \gamma\prime(\theta_i)]}{\varphi}$$

From this it follows that

$$\mathbf{E}(y_i) = \mu_i = \gamma\prime(\theta_i)$$

and

$$\mathbf{Var}(y_i) = \frac{\varphi}{A_i}\gamma\prime\prime(\theta_i)$$

Generalized linear models 5

Further it follows that

$$\mathbf{E}\left(\frac{\partial^2 l(\theta,\varphi;y}{\partial\theta\partial\varphi}\right)=0$$

Because of that θ and φ, or more generally β and φ are **orthogonal parameters**
The function $V(\mu)=\gamma''(\theta(\mu))$ is called **variance function**
For each response (error) distribution the link function $L=(\gamma')^{-1}$ for which $\theta\equiv\eta$ is called the **canonical link**
For $\eta=X\beta$, φ known, $A=\mathrm{diag}A_i$, and the canonical link it can be seen that X^TAy is a minimum sufficient statistic for β
Finally the score equations for β reduce to

$$X^TAy=X^TA\hat{\mu}$$

Estimation technique: **Iterative weighted least squares**
Hessian matrix is replaced by its expectation (Fisher scoring)

Medizinische Universität Graz

Smoothing 1

What is a 'smoother'?
A statistical tool for summarizing a response measurement Y as a function of one or more predictor measurements $X_1, \ldots, X_p$
Produces an estimate ('average', 'trend') of Y that is less variable (wiggly) than Y itself
Principal application:

- As exploratory tool for the visualization of scatterplots
- As method for the estimation of the dependence of the mean of Y on the predictors
- As a building block in various modelling approaches

What are the primary **statistical applications**?

1. For density estimation
2. For regression analysis

What characterizes a smoother?
For estimation at a point x weighted averaging across a centered neighborhood of x takes place
The simplest example is k-NN (no weighting) regression (smoothing)
For **kernel-based smoothing** (and related procedures) a neighborhood (i.e. bandwidth) needs to be specified
For **spline-based smoothing** a choice of basis functions and of the number and placement of knots resp. of the tuning (smoothing) parameter is required

- Both approaches are nonparametric strategies
- In clear contrast to rigid parametric approaches
- Here dependence of Y's on $X_1, \ldots, X_p$ is highly flexible

Kernel smoothing 1

Popular example: **Nadaraya-Watson kernel** (Nadaraya, 1964, Watson, 1964)

$$\hat{m}_{NW}(x) = \frac{\sum_{i=1}^{n} Y_i K((x - X_i)/h)}{\sum_{i=1}^{n} K((x - X_i)/h)},$$

where n is the sample size and h the bandwidth ($h > 0$)
The **bandwidth controls the smoothness** of the function estimate
Other kernels are Gasser-Müller and Parzen-Rosenblatt (Parzen, 1962, Rosenblatt, 1956)
Such **kernels produce a continuous function**
K is a bounded and integrable real-valued function such that

$$\lim_{x \to \infty} |x| K(x) = 0$$

Kernel smoothing 2

Usually K is taken to be a compactly supported symmetric probability density function
Examples:

$$K_1(x) = I(x \in [-0.5, +0.5])),$$

$$K_2(x) = \frac{3}{4}(1 - x^2)I(x \in [-1, +0.1])),$$

$$K_3(x) = \frac{15}{32}(3 - 10x^2 + 7x^4)I(x \in [-1, +0.1])),$$

$$K_4(x) = \frac{1}{\sqrt{2\pi}} exp(-x^2/2)$$

Explanation: (a) Uniform kernel K_1; (b) Epanechnikov kernel K_2; (c) 4th order kernel K_3; (d) Normal kernel K_4

Kernel smoothing 3

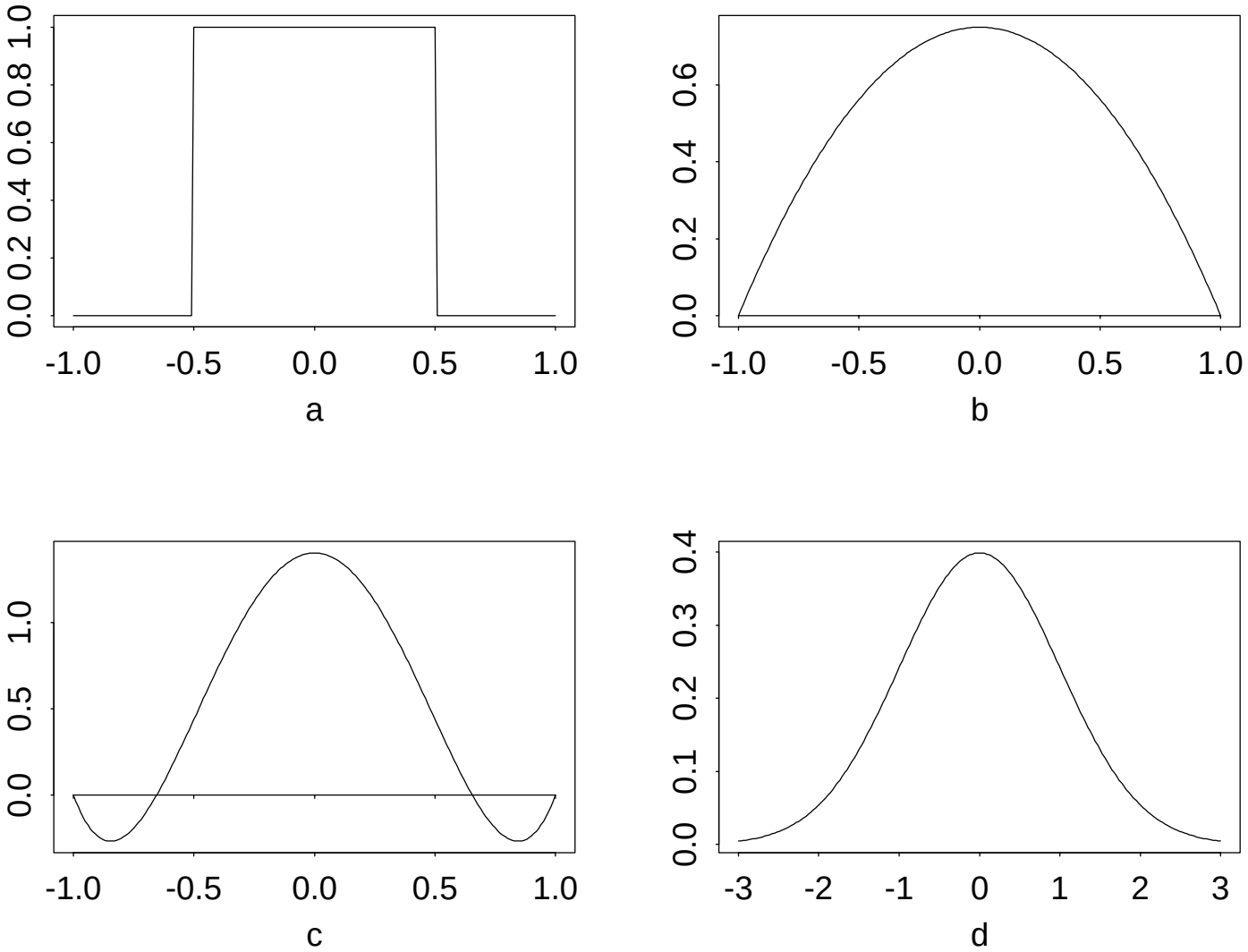


Figure: a: Uniformer Kern K_1; b: Epanechnikov Kern K_2; c: Kern 4. Ordnung K_3; d: Normaler Kern K_4. [S'00]

Kernel smoothing 4

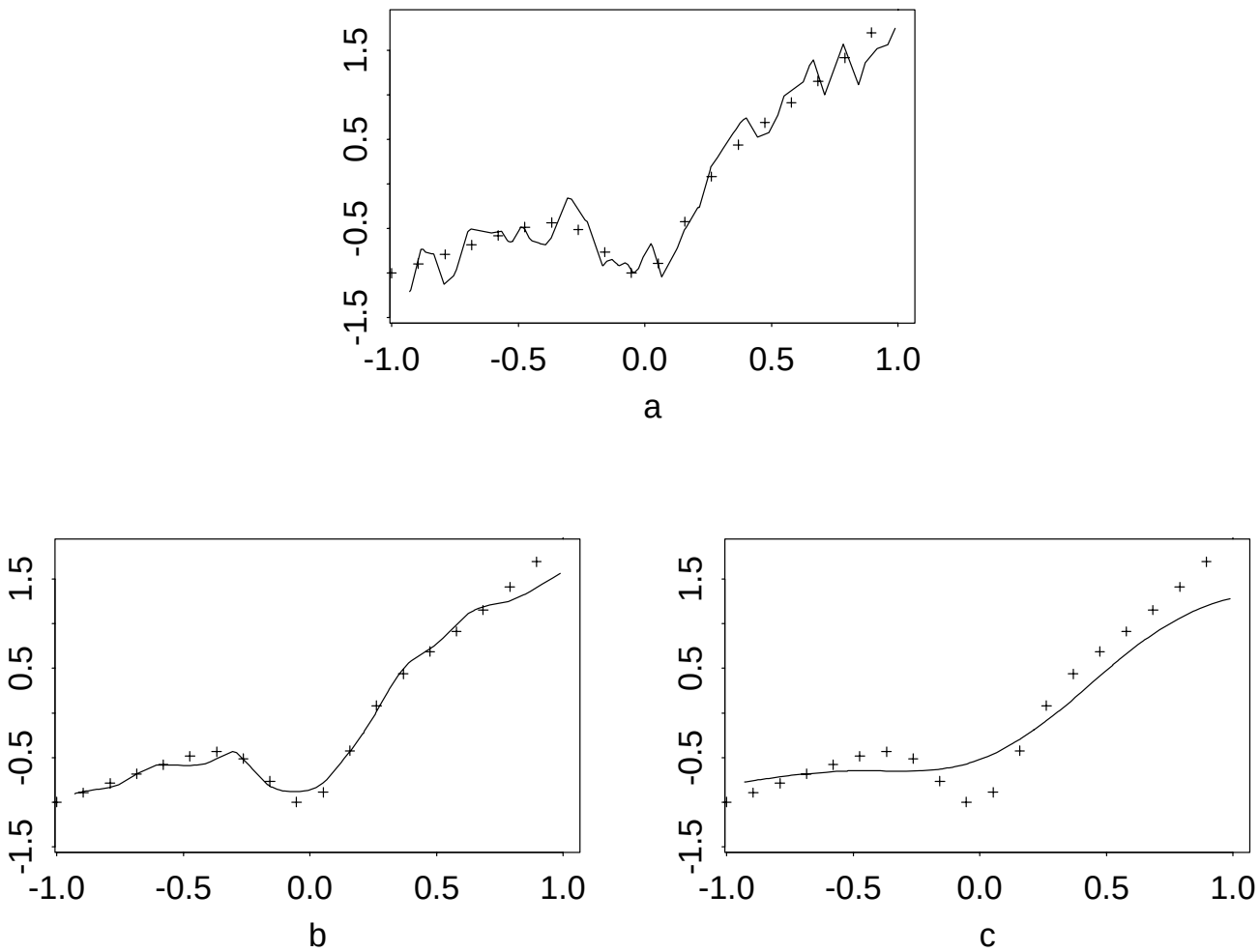


Figure: Curve estimates for bandwidth (a): $h = 0.05$; (b): $h = 0.2$; (c): $h = 0.5$. [S'00]

Motivation for smoothing splines

Suppose responses $y_1, \ldots, y_n$ observed at (nonstochastic) ordered design points $t_1 < \cdots < t_n$

Regression model

$$y_i = f(t_i) + \epsilon_i, \qquad i = 1, \ldots, n,$$

where $f(\cdot)$ is an unknown regression function and $\epsilon_1, \ldots, \epsilon_n$ are zero mean, uncorrelated random errors

Task: Estimation of f from the observed data

Classic approach: linear regression

$\hat{f}(t) = \hat{a} + \hat{b}t$ with $\hat{a}$ and $\hat{b}$ the least squares intercept and slope estimators obtained by minimizing the residual sum of squares

$$\text{RSS}(g) = \sum_{i=1}^{n} (y_i - g(t_i))^2$$

over all functions of the form $g(t) = a + bt$

Smoothing splines 1

Linear parametric model only has acceptable **model bias** if f is approximately linear

Ultimate slope flexibility: every two y_i's are connected by lines with their own individual slopes (for this estimator $\text{RSS}(g) = 0$), resulting in a summary not being useful

Motivation behind spline smoothing:

Need to compromise between fits with constant and completely flexible slopes

Can be achieved by **penalizing functions whose slopes vary too rapidly**

Rate of change in the slope of a function g is given by g'' and hence **an overall measure of the change** in slope of a potential fitted function is

$$J(g) = \int_{t_1}^{t_n} g''(t)^2 dt$$

Smoothing splines 2

Modified version of original estimation criterion comprises a penalty function

$$\text{RSS}(g) + \lambda J(g), \qquad \lambda \geq 0,$$

and can be minimized over, e.g., all functions with two continuous derivatives

λ is the smoothing (tuning) parameter and can be viewed as a measure of the importance we place on the flexibility for the slope of a fit

Behavior of λ:

$\lambda = \infty$ produces constant slope (i.e. the straight line)

λ tending to zero produce completely flexible slopes (i.e. interpolation).

The **solution of the above optimization problem** is a smoothing spline function (Thompson and Tapia, 1978, were the first to study such penalized problems)

The idea of penalization goes back to Whittaker (1923)

Local polynomial smoothing

Local polynomials have another motivation than splines
Were introduced by Stone (1977) and Cleveland (1979)
The idea is that a smooth function can be well approximated by a low order polynomial in the neighborhood of an arbitrary point x
Very popular is the **local linear approximation**

$$\mu(x_i) \approx a_0 + a_1(x_i - x)$$

for $x - h \leq x_i \leq x + h$ (h is the bandwidth)
The fitting can be obtained from **locally weighted least squares**
The weights are defined via kernels K (as in kernel smoothing)
The coefficients a_0 and a_1 are obtained by minimizing

$$\sum_{i=1}^{n} K((x - x_i)/h)(y_i - (a_0 + a_1(x_i - x)))^2$$

Medizinische Universität Graz

Linear smoothers

Smoothers that can be represented as

$$\hat{y} = Sy$$

are called **linear smoothers**
$y = (y_1, y_2, \ldots, y_n)$ and
$\hat{y} = (\hat{f}(x_1), (\hat{f}(x_2), \ldots, (\hat{f}(x_n))$, and
S an $n \times n$ **smoother matrix** depending on $x = (x_1, x_2, \ldots, x_n)$ and the smoothing (tuning) parameter λ, but not on y
S has nearly banded shape and is often called $\hat{h}at$ matrix (compare with projection matrix in linear models)
Examples for linear smoothers: kernels, local polynomials, smoothing splines
An example for a non-linear smoother: adaptive smoothers

Degrees of freedom and λ of linear smoothers

The (equivalent) degrees of freedom (equivalent number of parameters; effective dimension) of a linear smoother is

$$df(S) = trace(S)$$

the dimension of the space of fits
Compare with number of regressors in a linear model
If $\epsilon \sim N(0, \sigma^2 I)$

$$var(\hat{y}) = S\sigma^2 IS^T = \sigma^2 SS^T$$

and the diagonals give the pointwise variances
CV for λ selection estimates the mean predictive error
$PSE = MSE + \sigma^2$, where

$$MSE(\hat{f}) = E_X \left[var(\hat{f}(x)) + (f(x) - E_Y(\hat{f}(x))^2 \right]$$

Medizinische Universität Graz

Eigen-analysis of S

If S is symmetric we can write

$$Se_i = \theta_i e_i$$

for $i = 1, 2, \ldots, n$ (the e_i can be seen as response vectors y)
θ_i takes values $0 \leq \theta_i \leq 1$
Smoothers with $0 < \theta < 1$ are called **shrinking smoothers**
If all θ_i are 0 or 1, the smoother is called **regression smoother**

For the popular case of **cubic smoothing splines** the e_i are approximately orthogonal polynomials of increasing order, and

$$\theta_i = 1/(1 + \lambda\rho_i)$$

with $\rho_1 \leq \rho_2 \ldots \leq \rho_n$, the eigenvalues of the penalty matrix.

Medizinische Universität Graz