NMST539

Mnohorozměrná analýza
Multivariate Analysis

Lecture notes O (prerequisites)

Ivan Mizera

Some matrix trix

Eigenvalues and (spectral, Jordan) decomposition

Recall: if $\mathbf{Ax} = \lambda\mathbf{x}$ for a nonzero vector $\mathbf{x} \neq \mathbf{0}$, then λ is called an **eigenvalue** of $\mathbf{A}$ and $\mathbf{x}$ is its corresponding **eigenvector** (that is, one of these eigenvectors, as any $c\mathbf{x}$ for $c \neq 0$ qualifies too)

Every symmetric matrix $\mathbf{A}$ can be written in a form $\mathbf{A} = \mathbf{ULU}^\top$, where $\mathbf{U}$ is an orthogonal matrix (that is, $\mathbf{U}^\top = \mathbf{U}^{-1}$) and $\mathbf{L}$ is a diagonal matrix. It is easy to see then that the diagonal of $\mathbf{L}$ consists of all eigenvalues and $\mathbf{U}$ consists of their corresponding eigenvectors with unit norm.

A $p \times p$ symmetric matrix $\mathbf{A}$ thus has at most p eigenvalues; this is true also in greater generality, but we will deal pretty much exclusively with symmetric matrices, for which all eigenvalues and eigenvectors are real

Matrices $\mathbf{AB}$ and $\mathbf{BA}$ (when both are square matrices) have the same *nonzero* eigenvalues

Eigenvalues determine

The decomposition shows that a symmetric matrix $\mathbf{A}$ is nonnegative definite ($\mathbf{x}^\top \mathbf{A}\mathbf{x} \geqslant 0$ for every $\mathbf{x}$; sometimes they also say positive semidefinite) if all its eigenvalues are nonnegative. In such a case, we can form a square root of matrix: $\mathbf{A}^{1/2} = \mathbf{U}\mathbf{L}^{1/2}\mathbf{U}^\top$

Matrix $\mathbf{A}$ is positive definite ($\mathbf{x}^\top \mathbf{A}\mathbf{x} \geqslant 0$ for every $\mathbf{x} \neq \mathbf{0}$) if all eigenvalues are positive; then it is also invertible, as that is when all eigenvalues are nonzero: the inverse in such a case is $\mathbf{U}\mathbf{L}^{-1}\mathbf{U}^\top$

Trace

The trace of a (square) matrix is the sum of its diagonal elements:

$$\text{tr}(A) = \sum_i a_{ii}$$

A useful property of the trace is

$$\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$$

The "trace trick" uses this property - typically when one of the products has dimension 1×1, as then it is equal to its trace

The trace of a matrix is a sum of its eigenvalues

The eigenvalues of a symmetric and idempotent ($\mathbf{AA} = \mathbf{A}$) matrix are either 1 or 1; its rank is thus equal to its trace

The theorem of Eckart and Young

Let $\mathbf{S}$ be a symmetric nonnegative definite matrix; its best approximation by a symmetric matrix, in the Hilbert-Schmidt norm, that has rank at most m, is the matrix $\mathbf{U}\mathbf{L}_m\mathbf{U}^\top$, where $\mathbf{S} = \mathbf{U}\mathbf{L}\mathbf{U}^\top$ is the eigenvalue decomposition of $\mathbf{S}$, and $\mathbf{L}_m$ is the matrix formed from $\mathbf{L}$ by retaining the m largest eigenvalues, and replacing everything else by zero.

Let $\mathbf{A}$ be an arbitrary matrix. The Hilbert-Schmidt distance of $\mathbf{S}$ and $\mathbf{A}$ is

$$\begin{aligned}
\operatorname{tr}\left((\mathbf{S}-\mathbf{A})(\mathbf{S}-\mathbf{A})^\top\right) &= \operatorname{tr}\left(\mathbf{U}\mathbf{U}^\top(\mathbf{S}-\mathbf{A})\mathbf{U}\mathbf{U}^\top(\mathbf{S}-\mathbf{A})^\top\right)\\
&= \operatorname{tr}\left(\mathbf{U}^\top(\mathbf{S}-\mathbf{A})\mathbf{U}\,\mathbf{U}^\top(\mathbf{S}-\mathbf{A})^\top\mathbf{U}\right)\\
&= \operatorname{tr}\left((\mathbf{U}^\top\mathbf{S}\mathbf{U}-\mathbf{U}^\top\mathbf{A}\mathbf{U})(\mathbf{U}^\top\mathbf{S}\mathbf{U}-\mathbf{U}^\top\mathbf{A}\mathbf{U})^\top\right)\\
&= \operatorname{tr}\left((\mathbf{L}-\mathbf{U}^\top\mathbf{A}\mathbf{U})(\mathbf{L}-\mathbf{U}^\top\mathbf{A}\mathbf{U})^\top\right) \text{ etc.}
\end{aligned}$$

Derivatives of functions with matrices

For a function F defined on $p \times q$ matrices we define:

$\frac{\partial F(\mathbf{X})}{\partial \mathbf{X}}$ - a matrix with $\frac{\partial F(\mathbf{X})}{\partial X_{ij}}$ in i-th row and j-th column

We have:

$$\frac{\partial \mathbf{a}^\top \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a} \qquad \frac{\partial \mathbf{x}^\top \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$$

$$\frac{\partial \mathbf{a}^\top \mathbf{X} \mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^\top \qquad \frac{\partial \log \det(\mathbf{X})}{\partial \mathbf{X}} = (\mathbf{X}^{-1})^\top$$

Mathematical leftovers

Convexity

Probability tidbits

Transformation of a density

Suppose that $\mathbf{X}$ is a p-dimensional random vector with density $g(\mathbf{x})$, and let $\mathbf{Y} = T(\mathbf{X})$, where T is a mapping from $\mathbb{R}^p$ to $\mathbb{R}^p$ possessing an inverse T^{-1}. If $\mathbf{X}$ has a probability density (with respect to the Lebesgue measure on $\mathbb{R}^p$) $h(\mathbf{x})$, then Y has a density

$$h(T^{-1}(\mathbf{x}))|\det(J_{T^{-1}}(\mathbf{x}))| = \frac{h(T^{-1}(\mathbf{x}))}{|\det(J_T(\mathbf{x}))|}$$

where J_T denotes the Jacobi matrix consisting of partial

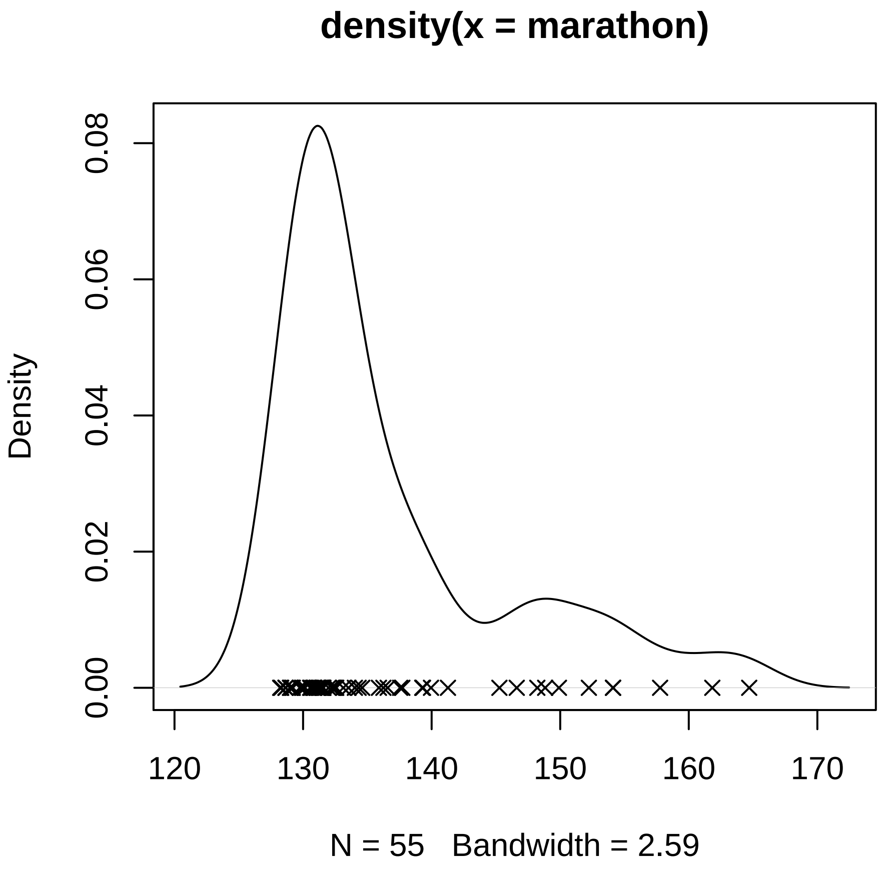
derivatives $\dfrac{\partial T_i(\mathbf{x})}{\partial x_j}$

If $T(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$, then $J_T = \mathbf{A}$

Nonparametric univariate statistics recalled: kernel density estimation

The probability density can be estimated?

```
> attach(Trackmen)
> plot(density(marathon))
> points(marathon,rep(0,length(marathon)),pch=4)
```

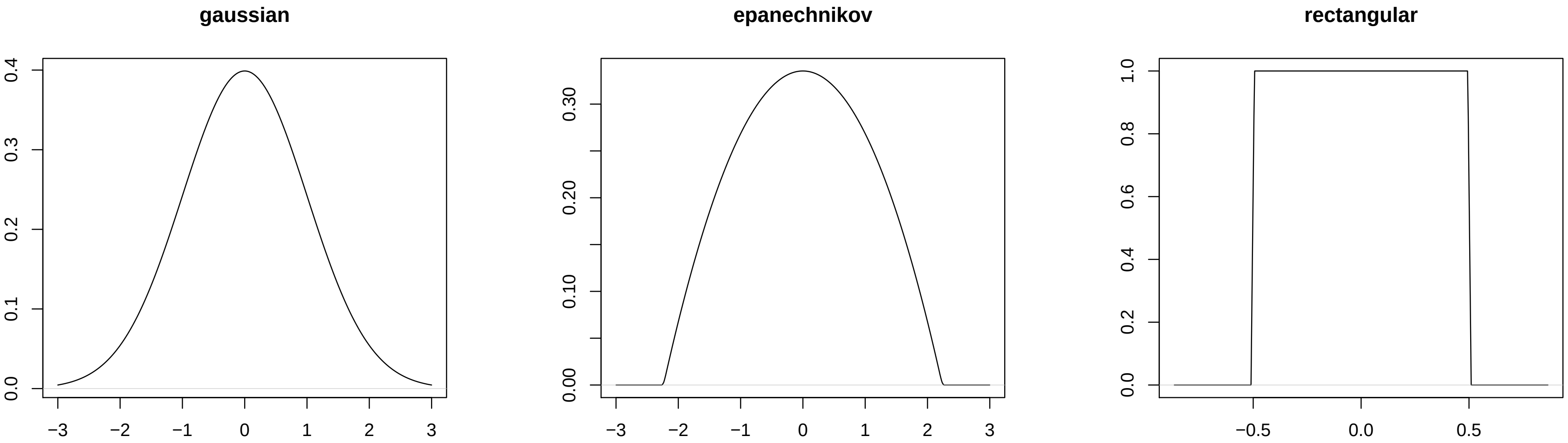


Kernel density estimator

$$\hat{f}(x) = \frac{1}{nb} \sum_i K\left(\frac{x_i - x}{b}\right)$$

kernel: $\int K(u)\,du = 1$ and also $K(u) \geqslant 0$

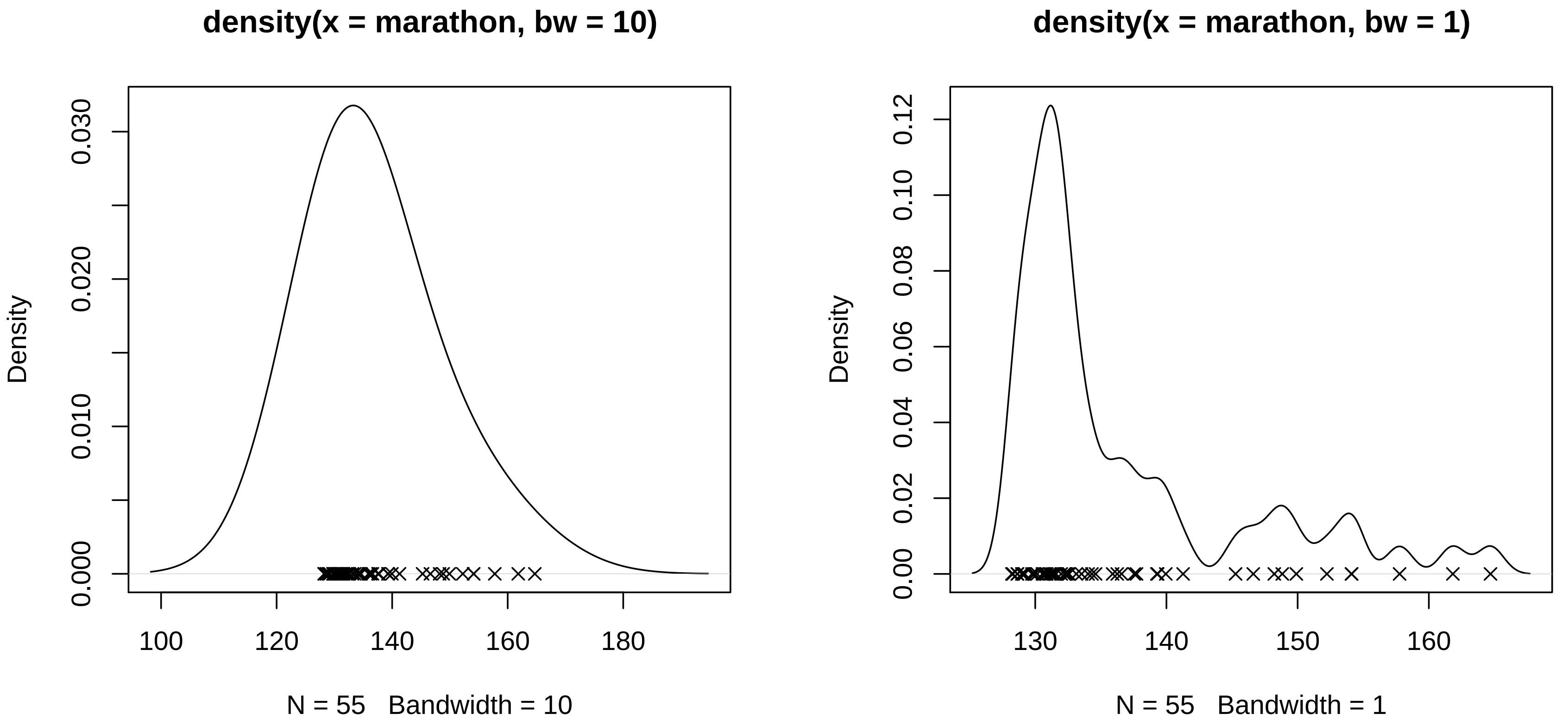
Examples: Gaussian (standard normal density), Epanechnikov, Rectangular (Parzen), and others



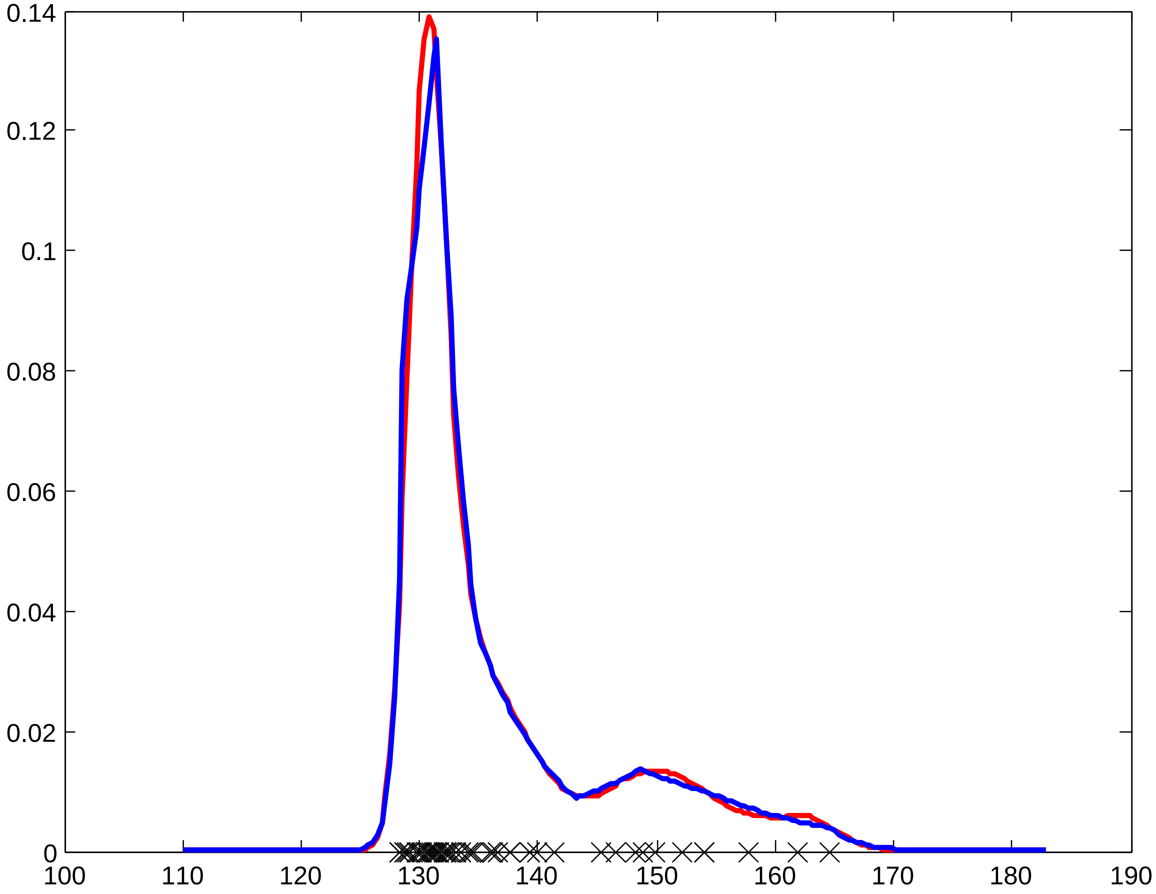
What does rectangular kernel mean? For $b = 1$, $\frac{1}{n} \sum_i K(x_i - x)$ is the relative proportion number of points falling into $[x - 1/2, x + 1/2]$; for general b, we obtain the relative proportion of points falling into $[x - b/2, x + b/2]$, divided by the length b of the interval.

Different bandwidth

The same *bandwidth* $\mathfrak{b}$ may not equally adapt to all parts of the data



Note: there may be better estimators...



Nonparametric bivariate statistics missed: smoothing splines

History

Whittaker (1923), “graduation” of actuarial mortality table

Given $y_1, y_2, \ldots, y_n$, find $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ such that

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum (\Delta^2 \hat{y}_i)^2 \looparrowright \min_{\hat{y}}!$$

Here $\lambda \geqslant 0$. Objective: to rid the original data of fluctuations

General functional fitting

We formulate the initial problem in a *functional*, that is, *infinite-dimensional* space. No splines yet.

Given $y_1, y_2, \ldots, y_n$, and $x_1 < x_2 < \cdots < x_n$, find f such that

- $f(x_1), \ldots, f(x_n)$ fit well $y_1, y_2, \ldots, y_n$
- but at the same time, f is not too "wiggly", not too "rough"

How to do it? First, we have to propose some measure of "wiggliness". We may

- take some derivative f the fitted function: f', or f'', or f'''
- then take its absolute value or square (only size of interest)
- and finally make it a global measure via integration

For instance, $J(f) = \int (f''(x))^2 dx;$ or $J(f) = \int |f''(x)| dx$

Such $J(f)$ will be referred to as (roughness) *penalty*.

To gain some small, partial insight about such a penalty, it may be instructive to investigate for which f is $J(f) = 0$; for both examples of J given above, it means that f is linear, $f(x) = \alpha + \beta x$.

Penalized fits

We seek a fit with guaranteed wiggliness

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 \looparrowright \min_f ! \qquad J(f) \leqslant \Lambda \qquad \text{(a tuning constant)}$$

Via Lagrange multiplier theory, this equivalent task is

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda J(f) \looparrowright \min_f !$$

Here, $\lambda > 0$ is another tuning constant, with unambiguous (but typically not explicit) relationship to Λ

Schoenberg (1964): smoothing splines

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int (f''(x))^2 dx \looparrowright \min_f !$$

There are mathematical details here, which we omit. However: how come we can speak about splines?

Because the solutions are splines

The solution of the smoothing spline problem is a *natural* cubic spline, with knots at x_i (and only x_i)

Also, “natural”: it just says that outside of knots it continues linearly. The first two derivatives (for a cubic spline) are to be matched: that is, at the extremal knots the first derivative, and also the second one, which is zero (second derivative of a linear function)

Note: once $f(x_i)$ given, the solution is found by minimizing $J(f)$

That gives the linearity of f outside the knots; inside of the knots, some further mathematics (it may be simply integration by parts) shows that...

... the solution to the smoothing spline problem, exists within the class of natural cubic splines, with knots at x_i (and only at x_i)

Finitary perspective

The original problem acted in the general functional spaces; now, however, we are in the finite dimensional space: all natural splines with given knots (finite number of knots, right?) can be written as linear combination of some (finite) basis functions

$$f(x) = \sum_j b_j g_j(x)$$

With some skill, we rewrite everything as a finite-dimensional problem in b_j - and in fact a quadratic one, as

- we are doing least-squares fitting) problem

- and the penalty has some square in it too, so it can be written as a quadratic form in b_j

And finally it is easy

So the original $$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\int_c^d (f''(x))^2 dx \looparrowright \min_f !$$

becomes $$(y - Lb)^\top(y - Lb) + \lambda b^\top G b \looparrowright \min_b !$$

where L is a linear operator (=matrix) yielding the functional values at the x_i's in terms of the b_j's, and G defines a quadratic form related to the penalty

(the solution in fact solves the system $L^\top L + \lambda G b = L^\top y$)

Remarks

The selection of the basis does not play a role, as long as the bases are equivalent (they generate the same linear spaces, any function that is a linear combination in one base, is a linear combination in another one)

Thus, we have something more general than just bases here...

Technical issue in this particular case: if x_i have duplicate values among them, we should take some care; there is no problem in the first, lack-of-fit part of the objective function, but the second, penalty part, should involve only “cleaned” x_i, with duplicates removed.

The tuning knob

So, we construct fits by trading off between the lack-of-fit criterion and penalty. The extent of this trade-off is controled by *smoothing, regularization parameter* λ.

It is a *tuning* parameter: looking at the original formulation, we notice

- for large λ the penalty prevails: the fit is linear

- for $\lambda \to 0$ ($\lambda = 0$ won't fly!) the lack-of-fit prevails: the fit, if there are no duplicates in the x_i's, is just the spline interpolation of the data

Note also the analogy in tree-based methods: $R + \alpha$ size

- the lack-of-fit criterion here is R

- the complexity measure (penalty) is size

- only λ is named α

So what did we arrive to?

Originally, we faced problem of selecting the *right* spline: how many knots, where to place them, ...

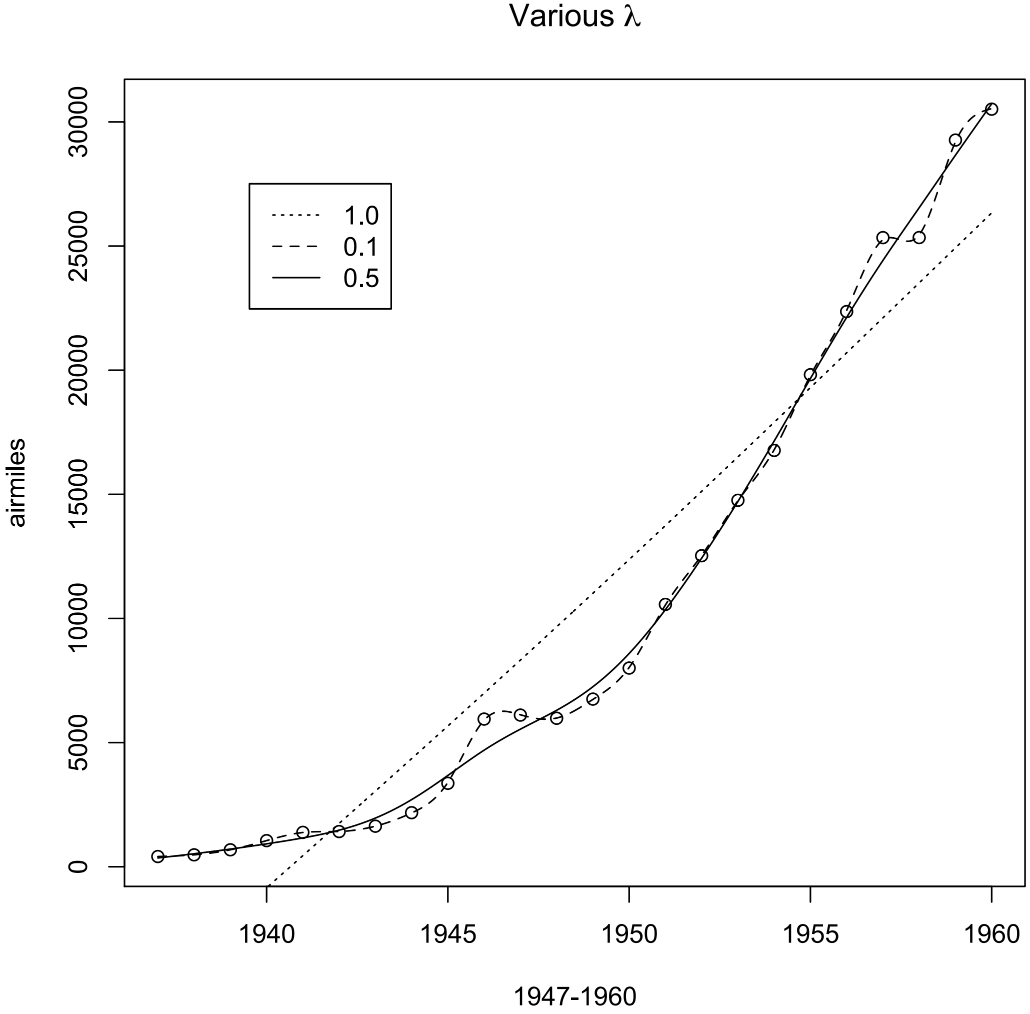
We somewhat mitigated the problem by the approach which

- put in many knots (in every x_i; do we need more?)

- introduced a reasonable criterion (penalty) to distinguish among various fits

- via regularization (fitting with penalty), we reduced the problem with many loose ends to a problem with just one loose end: λ

Revenue passenger airmiles flown by US airlines



```
> legend(locator(),lty=c(3,2,1),legend=c('1.0','0.1','0.5'))
```

The `legend` command does not show λ but `spar`. A closer look at `help(smooth.spline)` reveals that `spar` is a monotonous function of λ, normed so that `spar` lies between 0 and 1.

Finesses of the R implementation I

```
> plot(1937:1960,airmiles,xlab=’1947-1960’)
> title(expression(paste("Various ",lambda)))
> xx=seq(1937,1960,len=400)
> smsp=smooth.spline(1937:1960,airmiles,spar=1)
> lines(xx,predict(smsp,xx)$y,lty=3)
> smsp
Call:
smooth.spline(x = 1937:1960, y = airmiles, spar = 1)
Smoothing Parameter  spar= 1  lambda= 0.9681153
Equivalent Degrees of Freedom (Df): 2.063613
Penalized Criterion: 197298405
GCV: 9840216
```

Finesses of the R implementation II

```
> smsp=smooth.spline(1937:1960,airmiles,spar=.1)
> lines(xx,predict(smsp,xx)$y,lty=2)
> smsp
Call:
smooth.spline(x = 1937:1960, y = airmiles, spar = 0.1)
Smoothing Parameter  spar= 0.1  lambda= 3.045644e-07
Equivalent Degrees of Freedom (Df): 22.97617
Penalized Criterion: 34624.87
GCV: 792768.6

> smsp=smooth.spline(1937:1960,airmiles,spar=.5)
> lines(xx,predict(smsp,xx)$y)
> smsp
Call:
smooth.spline(x = 1937:1960, y = airmiles, spar = 0.5)
Smoothing Parameter  spar= 0.5  lambda= 0.0002363563
Equivalent Degrees of Freedom (Df): 7.460656
Penalized Criterion: 6128442
GCV: 537681.1
```

Some new statistics: regression

Regularization: not only for splines, but general

Penalization techniques are nowadays widely used in many branches of statistics and statistical machine learning. They come back to techniques developed by Tikhonov, and others, under the name *regularization*. Nowadays, “regularization” is understood in a general sense, where the objective functions does not have to be necessarily quadratic.

For instance, still for splines, one can consider an example of *penalized* splines: splines with knots at x_i penalized by basis coefficients:

$$\sum_{i=1}^{n}(y_i - \sum_j \beta_j g_j(x_i))^2 + \lambda \sum_j \beta_j^2 \looparrowright \min_\beta !$$

that is $$\|y - L\beta\|_2^2 + \lambda \beta^\top \beta \looparrowright \min_\beta !$$

Note that the penalty is different here!

Regularization in regression

In classical least squares regression estimation, when $\boldsymbol{\beta}$ minimizing

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

is sought, the necessary condition to obtain a solution is that $\mathbf{X}$ has full rank, or, equivalently $\mathbf{X}^\mathsf{T}\mathbf{X}$ is invertible. If this is violated, either exactly or approximately (for instance, when the number of predictors exceeds the number of datapoints: $p > n$), alternative estimation strategies have been proposed, most of these usually referred to as **regularization**. An instance of these, the penalized approach, amends the minimized function by a so-called *penalty*; in an **ridge regression** (originally proposed just for the problems when the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$ has problems with invertibility) the penalty is the square of the ℓ^2 norm, resulting in the minimized function

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta}$$

where $\lambda > 0$ is a *tuning parameter*, quantifying the "strength" of the regularization; for every $\lambda > 0$, there exists the minimizer $\boldsymbol{\beta}$ that can be obtained as a solution of

$$(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = \mathbf{X}^\mathsf{T}\mathbf{y}$$

- note that the matrix $(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})$ is invertible for any $\lambda > 0$

Atomic pursuit (LASSO)

A variation of the ridge regression is the *atomic pursuit*, also known as LASSO, which instead of the ℓ^2 one uses the ℓ^1 penalty; if $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^\top$, the minimized function is

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_j |\beta_j|$$

What makes this version attractive is that while for $\lambda > 0$ it still handles situations when $\mathbf{X}$ is not of full rank (for instance when $p > n$, the number of variables used as predictors is greater than the number of all observations), the absolute value in the penalty causes the resulting vector $\boldsymbol{\beta}$ of estimates to be *sparse* - to contain only few nonzero elements

This is unlike the ridge regression, which returns solutions that are rather nonzero; even for the regressors that do not have predictive value for the response, it tends to return estimates that are small in magnitude, but still not exactly zero