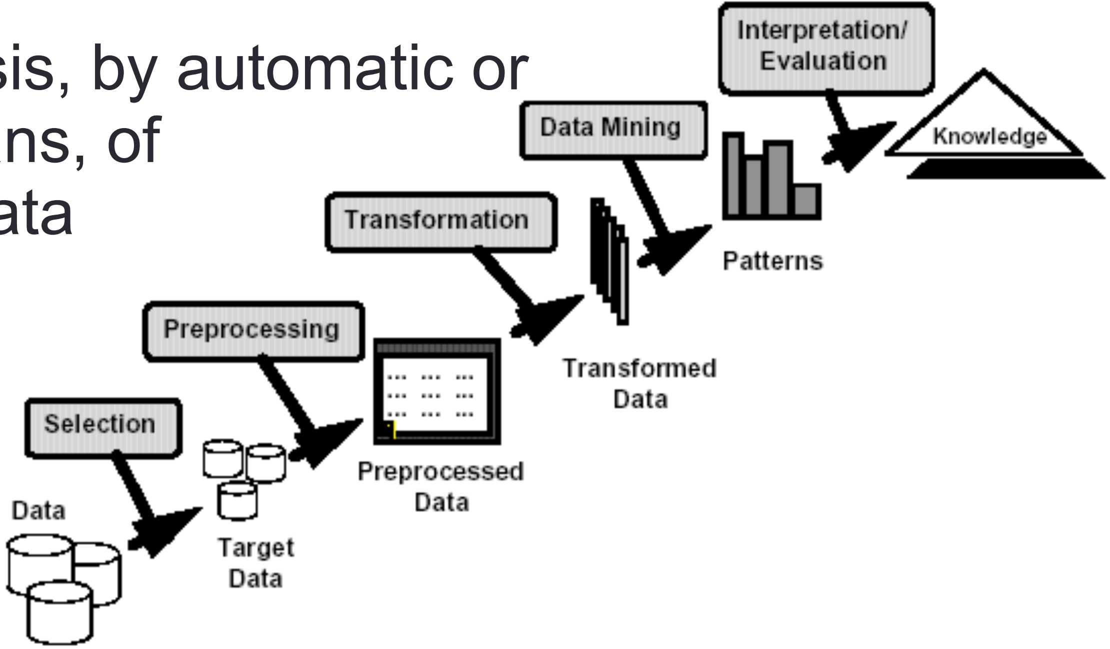
MINERÍA DE DATOS: PREPROCESAMIENTO Y CLASIFICACIÓN

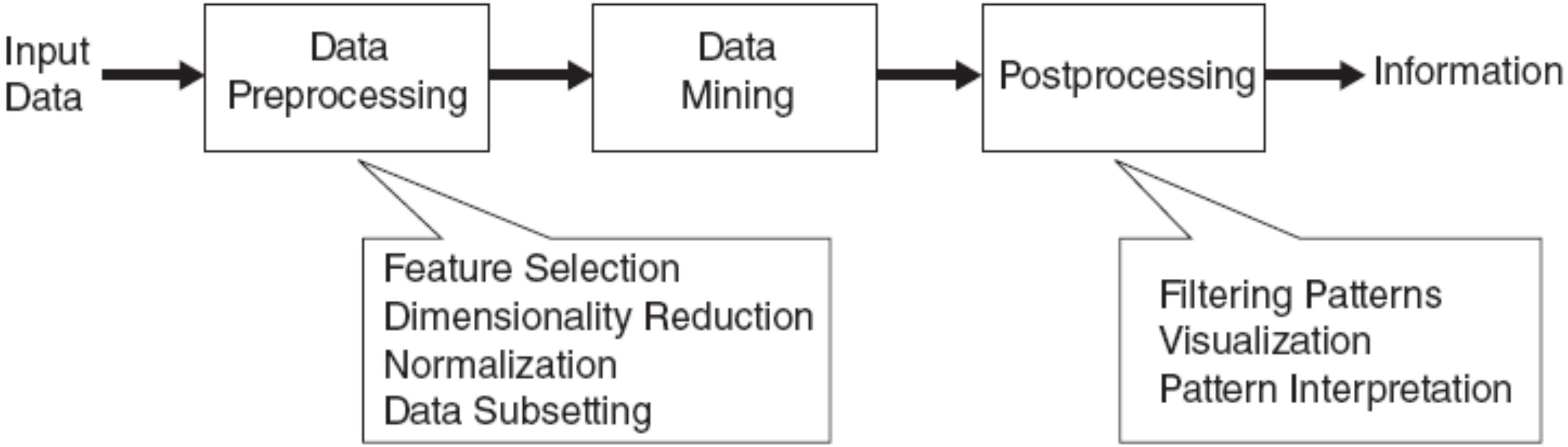
What is Data Mining?

- Many Definitions
  - Non-trivial extraction of implicit, previously unknown and potentially useful information from data
  - Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns



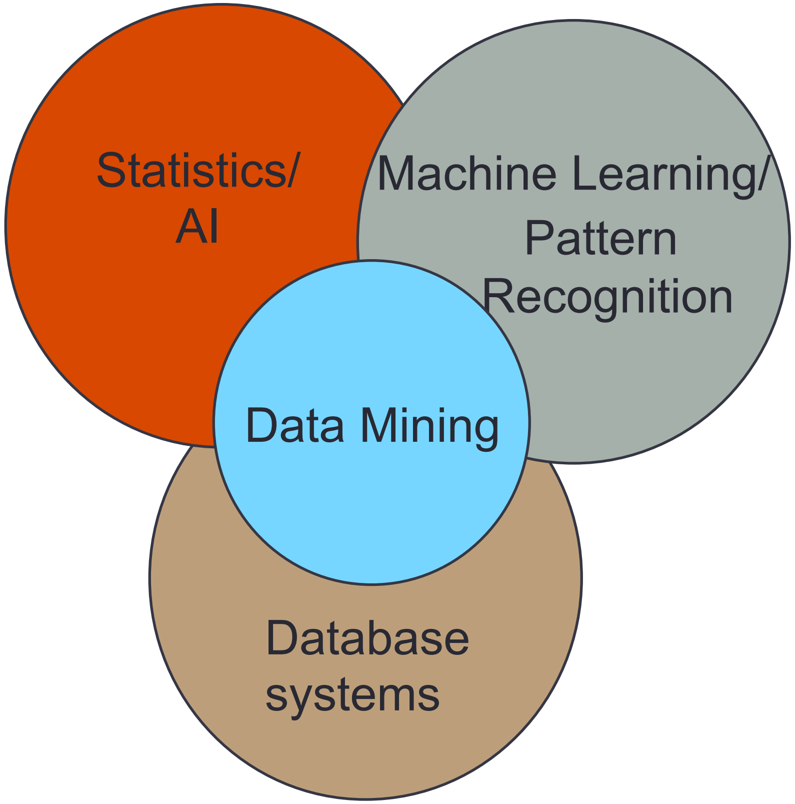
"Introduction to Data Mining", Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson, 2014

What is Data Mining?



Origins of Data Mining

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems
- Traditional Techniques may be unsuitable due to
  - Enormity of data
  - High dimensionality of data
  - Heterogeneous, distributed nature of data



Data Mining Tasks

- Prediction Methods
  - Use some variables to predict unknown or future values of other variables
- Description Methods
  - Find human-interpretable patterns that describe the data

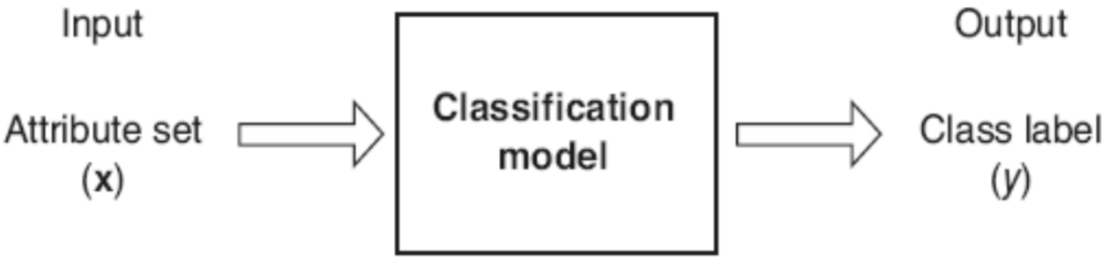
Data Mining Tasks...

- Classification [Predictive]
- Regression [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Sequential Pattern Discovery [Descriptive]
- Deviation Detection [Predictive]

Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification: Definition



- Descriptive Modeling
- Predictive Modeling

- Accuracy
- Interpretability

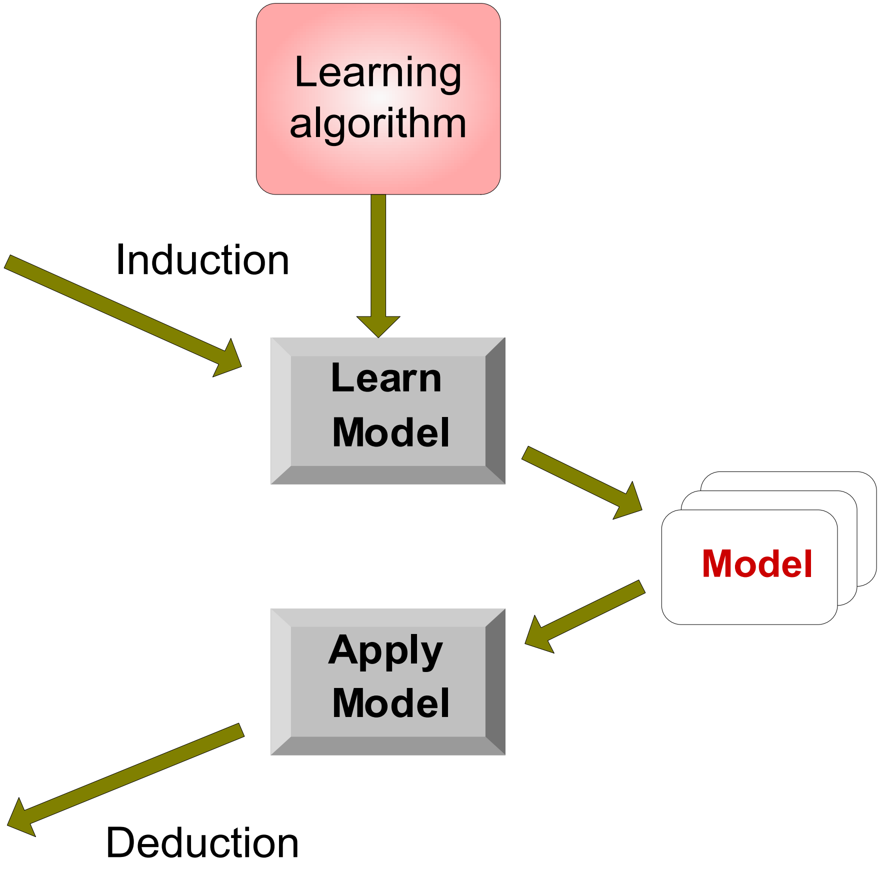
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

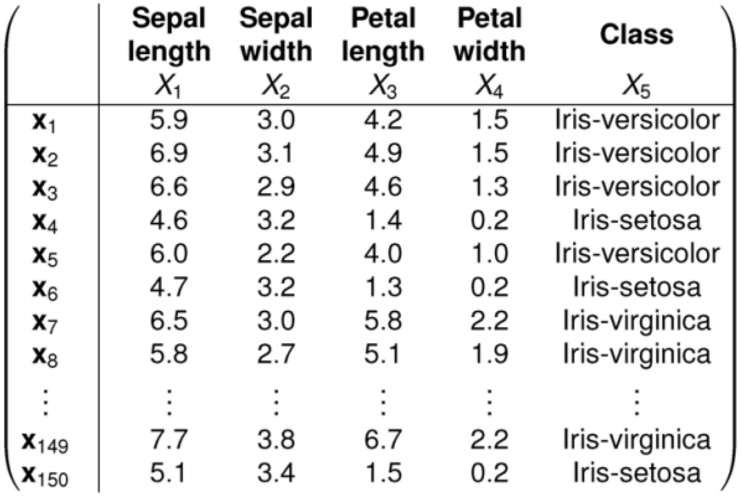
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Ejemplo de problema de clasificación

- Ejemplo: el problema de clasificación de la flor de lirios (IRIS)
- Tres clases de lirios: setosa, versicolor y virginica
- Cuatro atributos: longitud y anchura del pétalo y sépalo respectivamente.
- 150 ejemplos, 50 de cada clase (disponible en UC Irvine Machine Learning Repository http://archive.ics.uci.edu/ml/index.php)



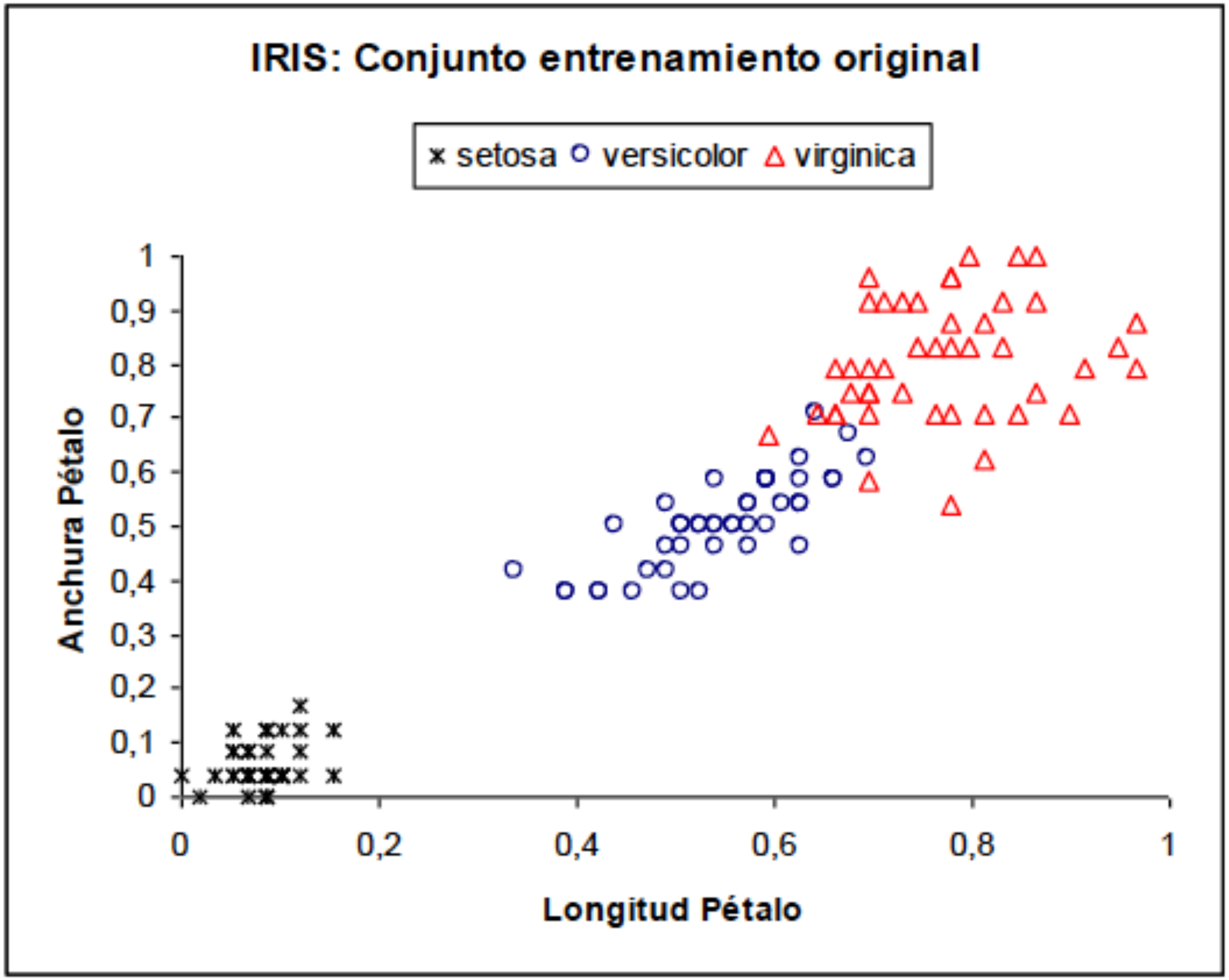
	Sepal length X_1	**Sepal width** X_2	**Petal length** X_3	**Petal width** X_4	**Class** X_5
$\mathbf{x}_1$	5.9	3.0	4.2	1.5	Iris-versicolor
$\mathbf{x}_2$	6.9	3.1	4.9	1.5	Iris-versicolor
$\mathbf{x}_3$	6.6	2.9	4.6	1.3	Iris-versicolor
$\mathbf{x}_4$	4.6	3.2	1.4	0.2	Iris-setosa
$\mathbf{x}_5$	6.0	2.2	4.0	1.0	Iris-versicolor
$\mathbf{x}_6$	4.7	3.2	1.3	0.2	Iris-setosa
$\mathbf{x}_7$	6.5	3.0	5.8	2.2	Iris-virginica
$\mathbf{x}_8$	5.8	2.7	5.1	1.9	Iris-virginica
⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{x}_{149}$	7.7	3.8	6.7	2.2	Iris-virginica
$\mathbf{x}_{150}$	5.1	3.4	1.5	0.2	Iris-setosa

Setosa

Versicolor

Virginica

Ejemplo de problema de clasificación

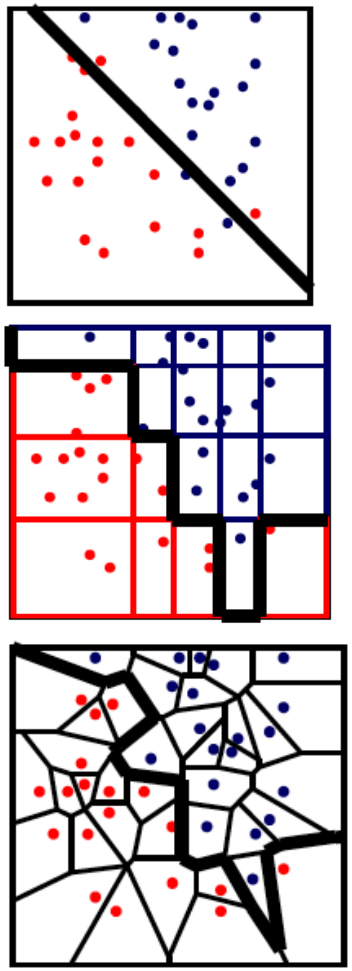


Other example

- Fraud Detection
  - Goal: Predict fraudulent cases in credit card transactions.
  - Approach:
    - Use credit card transactions and the information on its account-holder as attributes.
      - When does a customer buy, what does he buy, how often he pays on time, etc
    - Label past transactions as fraud or fair transactions. This forms the class attribute.
    - Learn a model for the class of the transactions.
    - Use this model to detect fraud by observing credit card transactions on an account.

Classification Techniques

- Nonlinear models
- Decision Tree based Methods
- Rule-based Methods
- Support Vector Machines
- …



Descripción

- **Temario**:
  - Modelos no lineales.
  - Árboles de Decisión. Multiclasificadores.
  - Descomposición de problemas multiclase.
  - Aprendizaje de Reglas.
  - Máquinas soporte vectorial (SVM).
  - Preprocesamiento de Datos.
- **Bibliografía**:
  - "An Introduction to Statistical Learning with Applications in R", Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, Springer, 2013.
  - "Introduction to Data Mining", Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson, 2013.
  - "Foundations of Rule Learning", Johannes Fürnkranz, Dragan Gambergerm Nada Lavrac, Springer, 2012.
  - "Data Preprocessing in Data Mining". Salvador García, Julián Luengo, Francisco Herrera, Springer, 2015.
- **Relacionado con**: Introducción a la Programación para Ciencia de Datos e Introducción a la Ciencia de Datos

Descripción

- **Temario**:
  - Modelos no lineales.
  - Árboles de Decisión. Multiclasificadores.
  - Descomposición de problemas multiclase.
  - Aprendizaje de Reglas.
  - Máquinas soporte vectorial (SVM).
  - Preprocesamiento de Datos.

- **Bibliografía**:
  - “An Introduction to Statistical Learning with Applications in R”, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, Springer, 2013.
  - “Introduction to Data Mining”, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson, 2013.
  - “Foundations of Rule Learning”, Johannes Fürnkranz, Dragan Gambergerm Nada Lavrac, Springer, 2012.
  - “Data Preprocessing in Data Mining”. Salvador García, Julián Luengo, Francisco Herrera, Springer, 2015.

- **Relacionado con**: Introducción a la Programación para Ciencia de Datos e Introducción a la Ciencia de Datos

MOVING BEYOND LINEARITY

Minería de Datos: Preprocesamiento y clasificación

The truth is never linear!

The truth is never linear!

Or almost never!

The truth is never linear!

Or almost never!

But often the linearity assumption is good enough.

The truth is never linear!
Or almost never!

But often the linearity assumption is good enough.

When its not …

✓ polynomials,
✓ step functions,
✓ splines,
✓ local regression, and
✓ generalized additive models

offer a lot of flexibility, without losing the ease and interpretability of linear models.

Logistic regression

- Standard lineal model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

- The output variable is discrete rather than continuous.
- Special case with binary outcomes (snows in Granada on a given day).
- How could we model and analyze such data?
  - Simply guessing “yes” or “not” is pretty crude especially if there is no perfect rule.
  - We need to fit better a stochastic model P(Y|X).
  - There’s a 25% chance of snow.

Logistic regression

- Let’s pick one of the classes and call it “1” and the other “0”.

$$P(Y=1)=E[Y]$$

$$P(Y=1|X)=E[Y|X=x]$$

- To sum up: we have a binary output variable Y, and we want to model the conditional probability

$$P(Y = 1|X = x)$$

as a function of x.

Logistic regression

How can we use linear regression to solve this?

- The most obvious idea is to let p(x) be a linear function of x.
- The conceptual problem here is that p must be between 0 and 1, and linear functions are unbounded.
- Moreover, in many situations we empirically see that changing p by the same amount requires a bigger change in x when p is already large (or small) than when p is close to 1/2. Linear models can't do this.
- An interesting proposal has been the use of the logistic (or logit) transformation

$$\log \frac{p}{1-p}$$

- This last alternative is logistic regression.

Logistic regression

Formally, the model logistic regression model is that

$$\log \frac{p(x)}{1 - p(x)} = \beta_0 + x \cdot \beta$$

Solving for p, this gives

$$p(x; b, w) = \frac{e^{\beta_0 + x \cdot \beta}}{1 + e^{\beta_0 + x \cdot \beta}} = \frac{1}{1 + e^{-(\beta_0 + x \cdot \beta)}}$$

- To minimize the mis-classification rate, we should predict Y = 1 when p ≥ 0.5 and Y = 0 when p < 0.5.

Logistic regression

With several antecedent variables:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_P x_P.$$

$$p = \frac{1}{1 + \exp\left[-(\beta_0 + \beta_1 x_1 + \cdots + \beta_P x_P)\right]}$$

Logistic regression

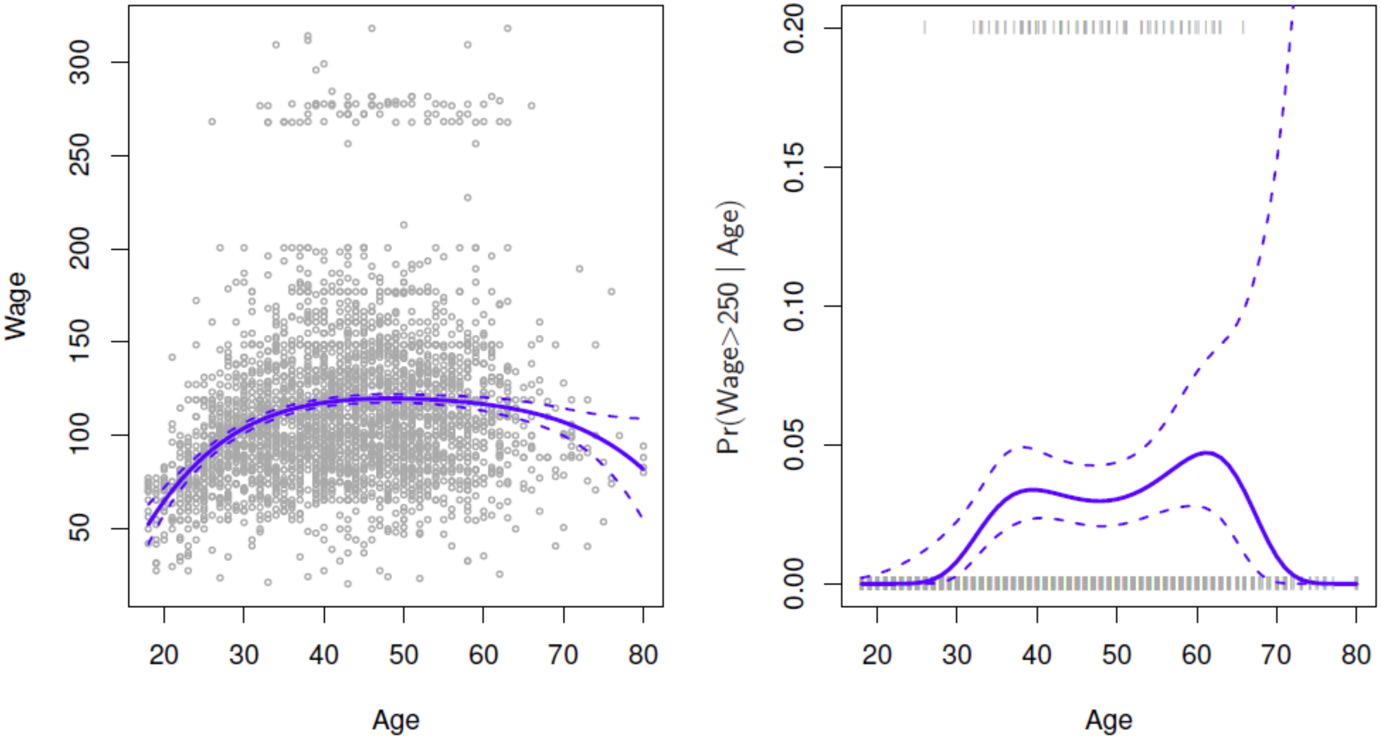
With more than two classes:

$$\Pr\left(Y = c|\vec{X} = x\right) = \frac{e^{\beta_0^{(c)} + x \cdot \beta^{(c)}}}{\sum_c e^{\beta_0^{(c)} + x \cdot \beta^{(c)}}}$$

Polynomial Regression

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$$

Degree–4 Polynomial

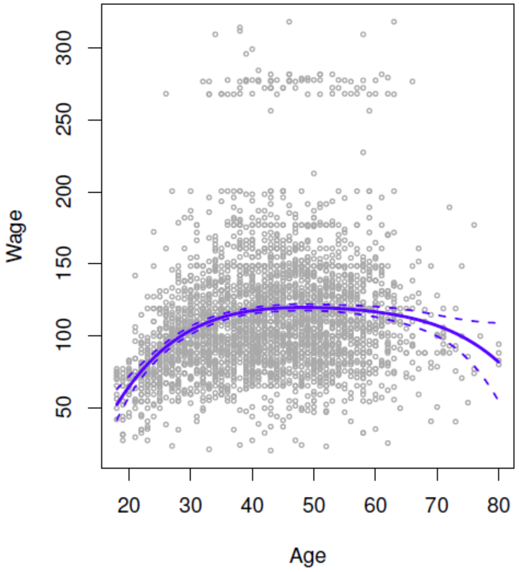


Details

- Create new variables $X_1 = X$, $X_2 = X^2$; etc and then treat as multiple linear regression.
- Not really interested in the coefficients; more interested in the fitted function values at any value x_0:

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4.$$

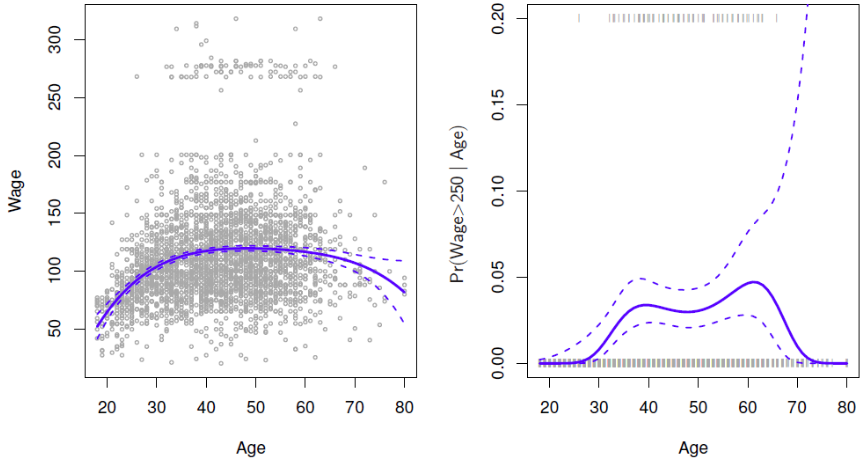
We either fix the degree d at some reasonably low value, else use cross-validation to choose d.



Details continued

- Logistic regression follows naturally. For example, in figure we model

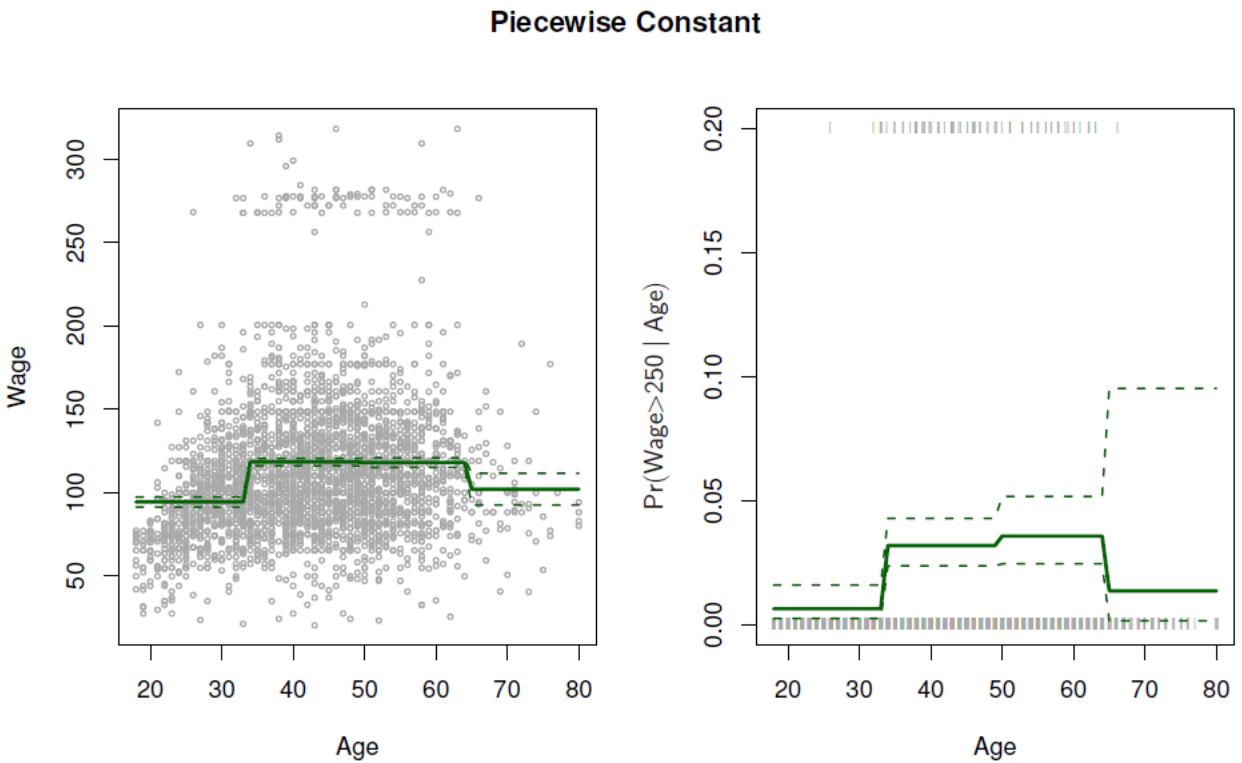
$$\Pr(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_d x_i^d)}{1 + \exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_d x_i^d)}.$$



Step Functions

- Another way of creating transformations of a variable - cut the variable into distinct regions.

$$C_1(X) = I(X < 35), \quad C_2(X) = I(35 \leq X < 50), \ldots, C_3(X) = I(X \geq 65)$$



Step functions continued

- Easy to work with. Creates a series of dummy variables representing each group.

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \ldots + \beta_K C_K(x_i) + \epsilon_i.$$

- Choice of cutpoints or knots can be problematic.

Basis Functions

- Polynomial and piecewise-constant regression models are in fact special cases of a *basis function approach.*

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \ldots + \beta_K b_K(x_i) + \epsilon_i.$$

The basis functions *$b_1(\cdot)$, $b_2(\cdot)$, . . . , $b_K(\cdot)$ are fixed and known*

Regression Splines

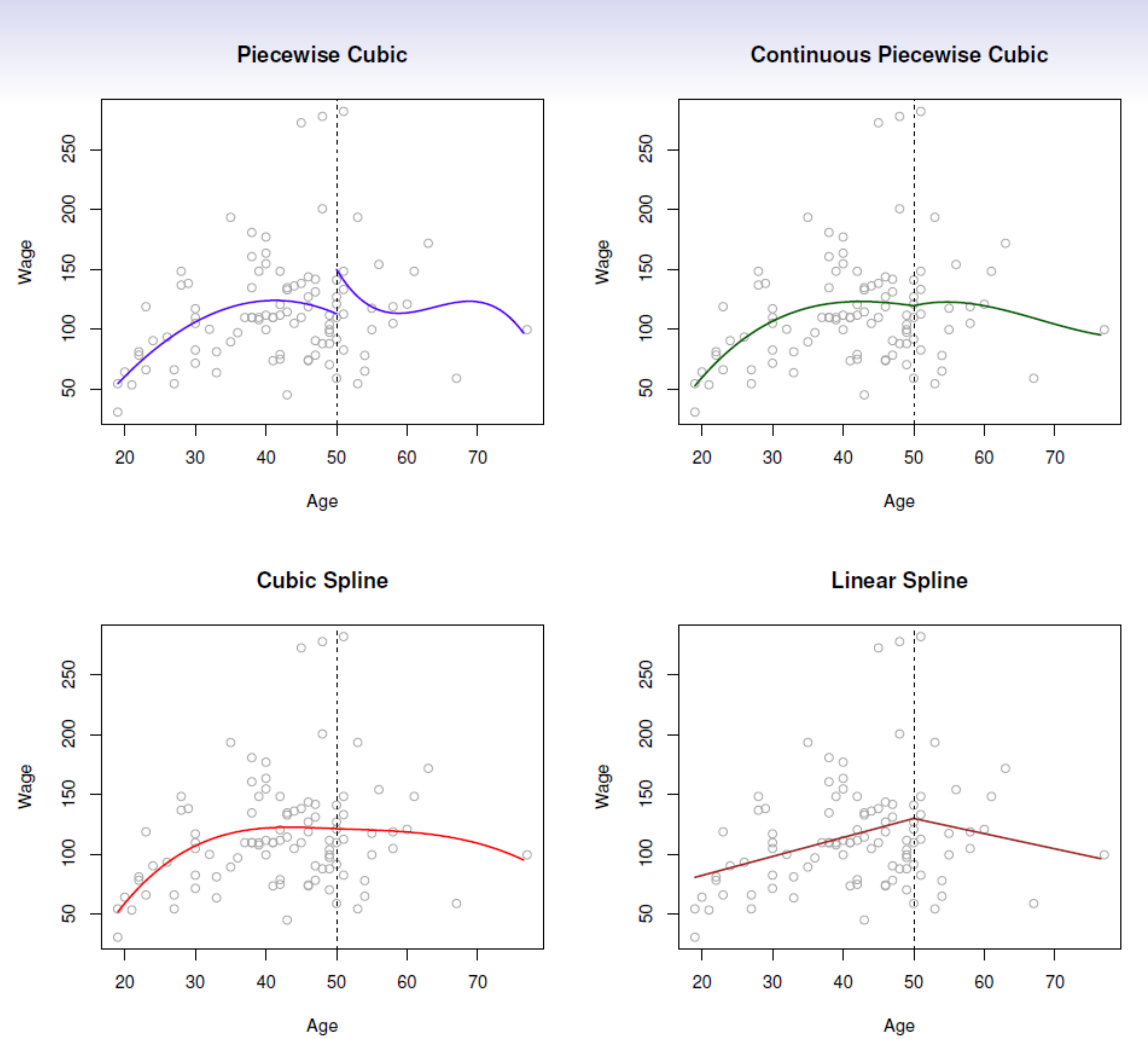
- Now we discuss a flexible class of basis functions that extends upon the polynomial regression and piecewise constant regression approaches that we have just seen.

Piecewise Polynomials

- Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

- Using more knots leads to a more flexible piecewise polynomial.
- Better to add constraints to the polynomials, e.g. continuity.
- Splines have the “maximum" amount of continuity.


Piecewise Cubic
Continuous Piecewise Cubic
Cubic Spline
Linear Spline
Wage
Age
50
100
150
200
250
20
30
40
50
60
70

Degree-d Spline

- The general definition of a degree-*d spline is that it is a piecewise* degree-*d polynomial, with continuity in derivatives up to degree d − 1 at* each knot.

Linear Splines

- A linear spline with knots at ξ_k; k = 1…K is a piecewise linear polynomial continuous at each knot.
- We can represent this model as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{K+1} b_{K+1}(x_i) + \varepsilon_i,$$

where the b_k are basis functions.

$$\begin{aligned} b_1(x_i) &= x_i \\ b_{k+1}(x_i) &= (x_i - \xi_k)_+, \quad k = 1, \ldots, K \end{aligned}$$

Here the $()_+$ means positive part; i.e.

$$(x_i - \xi_k)_+ = \begin{cases} x_i - \xi_k & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}$$

Cubic Splines

- A cubic spline with knots at ξ $_k$; k = 1,…,K is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.
- Again we can represent this model with truncated power basis functions

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i,$$

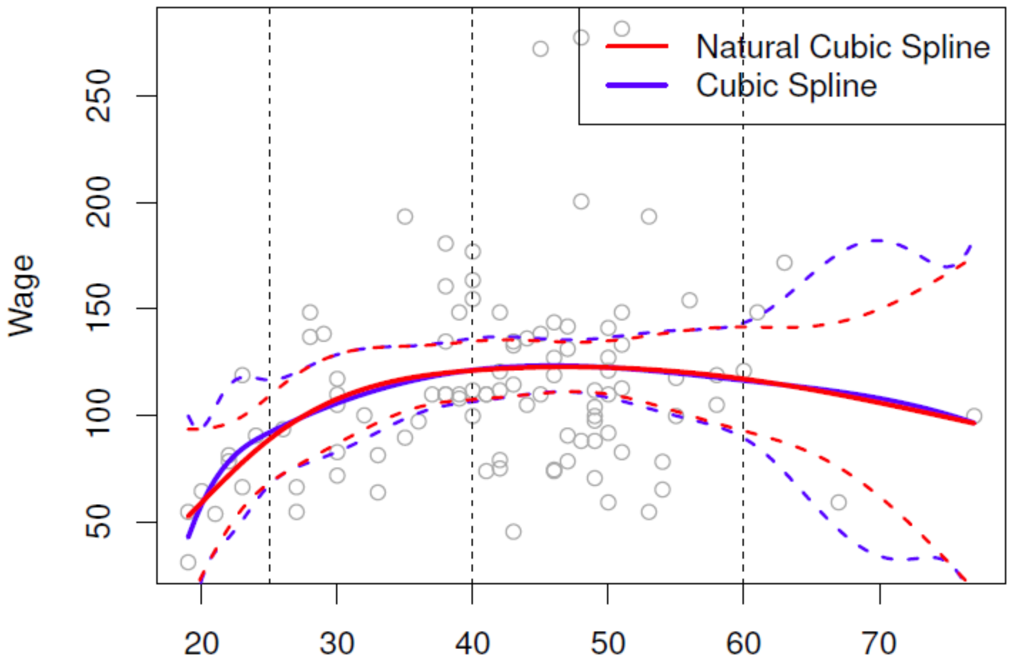
$$\begin{aligned} b_1(x_i) &= x_i \\ b_2(x_i) &= x_i^2 \\ b_3(x_i) &= x_i^3 \\ b_{k+3}(x_i) &= (x_i - \xi_k)_+^3, \quad k = 1, \ldots, K \end{aligned}$$

where

$$(x_i - \xi_k)_+^3 = \begin{cases} (x_i - \xi_k)^3 & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}$$

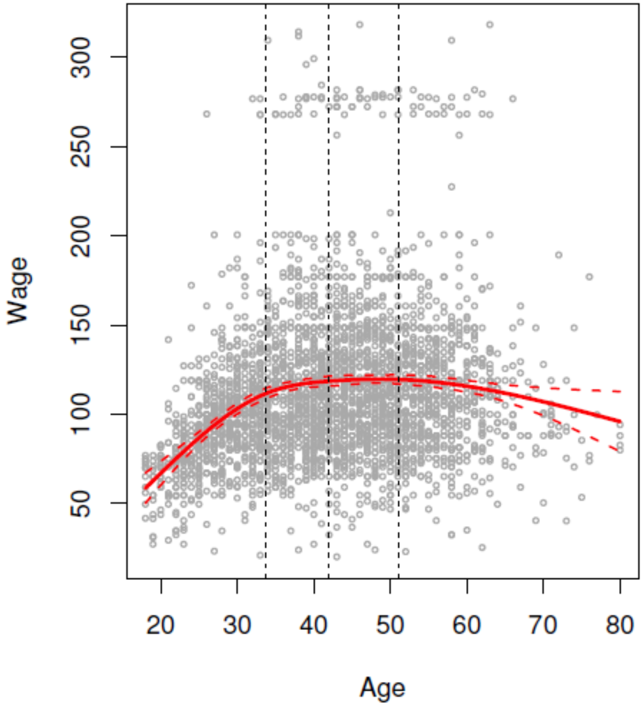
Natural Cubic Splines

- A natural cubic spline extrapolates linearly beyond the boundary knots. This adds 4 = 2x2 extra constraints, and allows us to put more internal knots for the same degrees of freedom as a regular cubic spline.



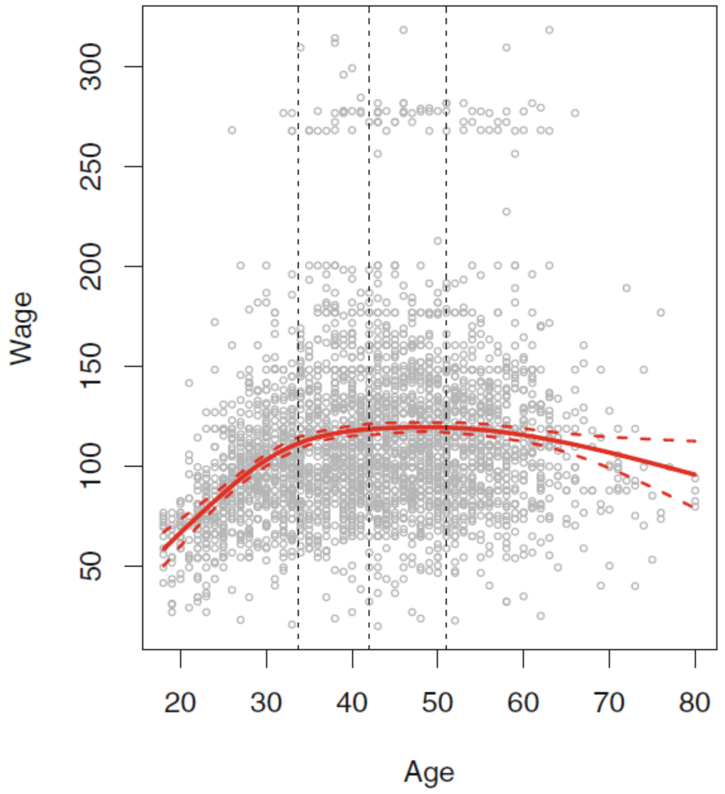
Splines in R

- Fitting splines in R is easy: bs(x, ...) for any degree splines, and ns(x, ...) for natural cubic splines, in package splines.

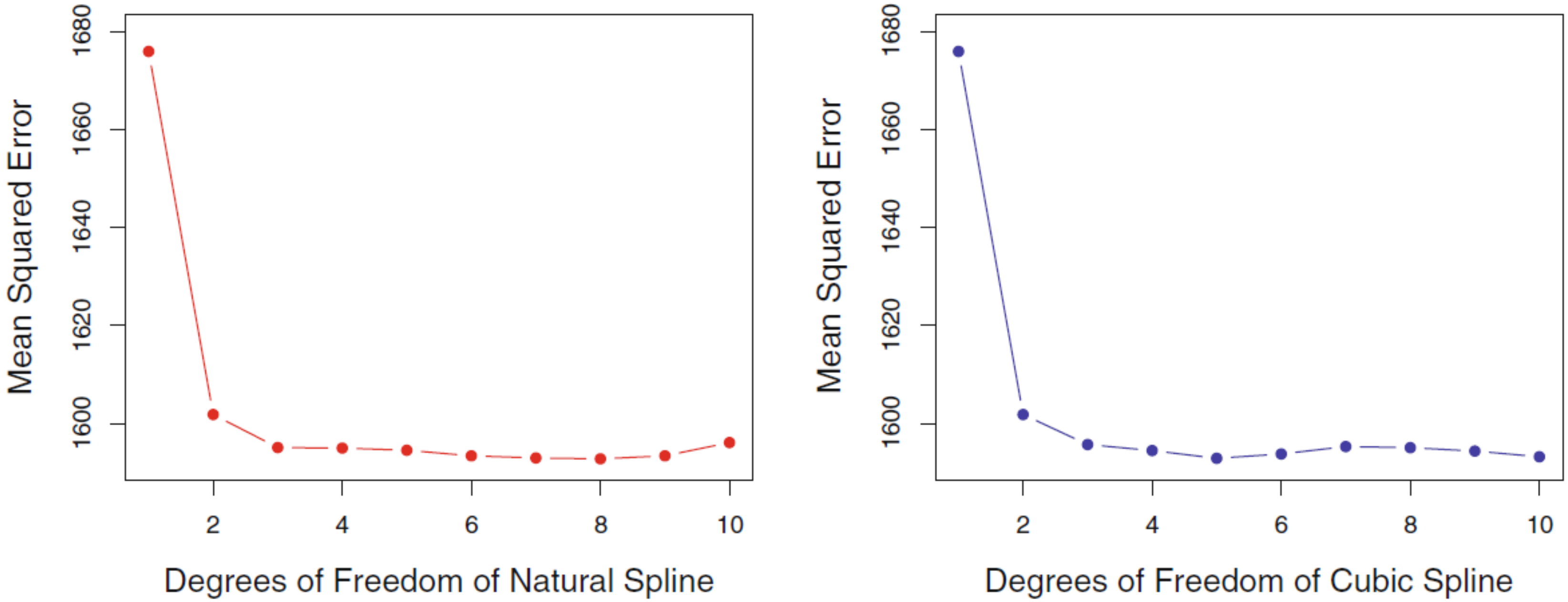


Knot placement

- One strategy is to decide K, the number of knots, and then place them at appropriate quartiles of the observed X.
- A cubic spline with K knots has K + 4 parameters or degrees of freedom.
- A natural spline with K knots has K degrees of freedom.



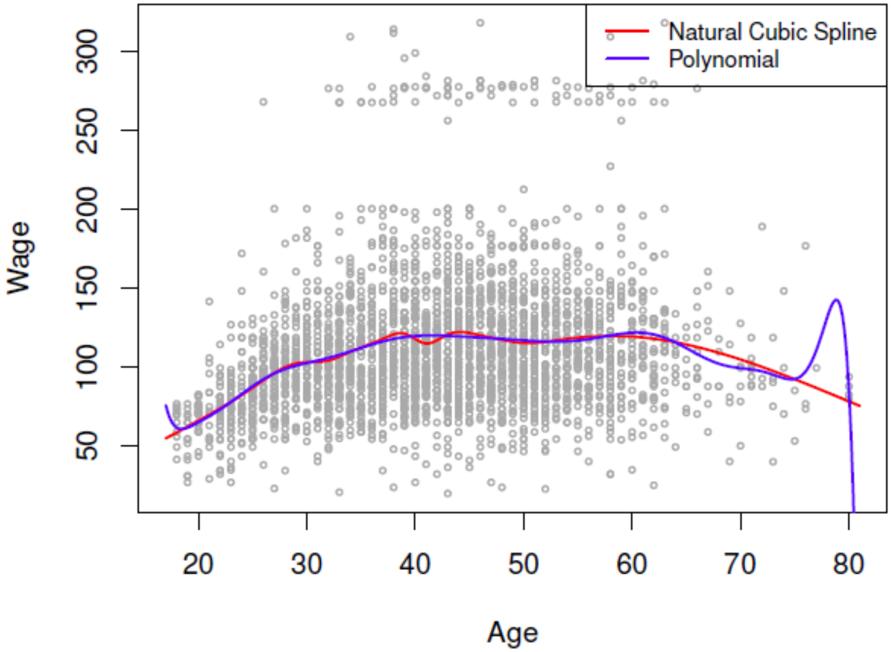
Number of Knots



Comparison

- Comparison of a degree-15 polynomial and a natural cubic spline, each with 15df.

```
ns(age, df=15)
poly(age, deg=15)
```



Smoothing Splines

- Consider this criterion for fitting a smooth function g(x) to some data:

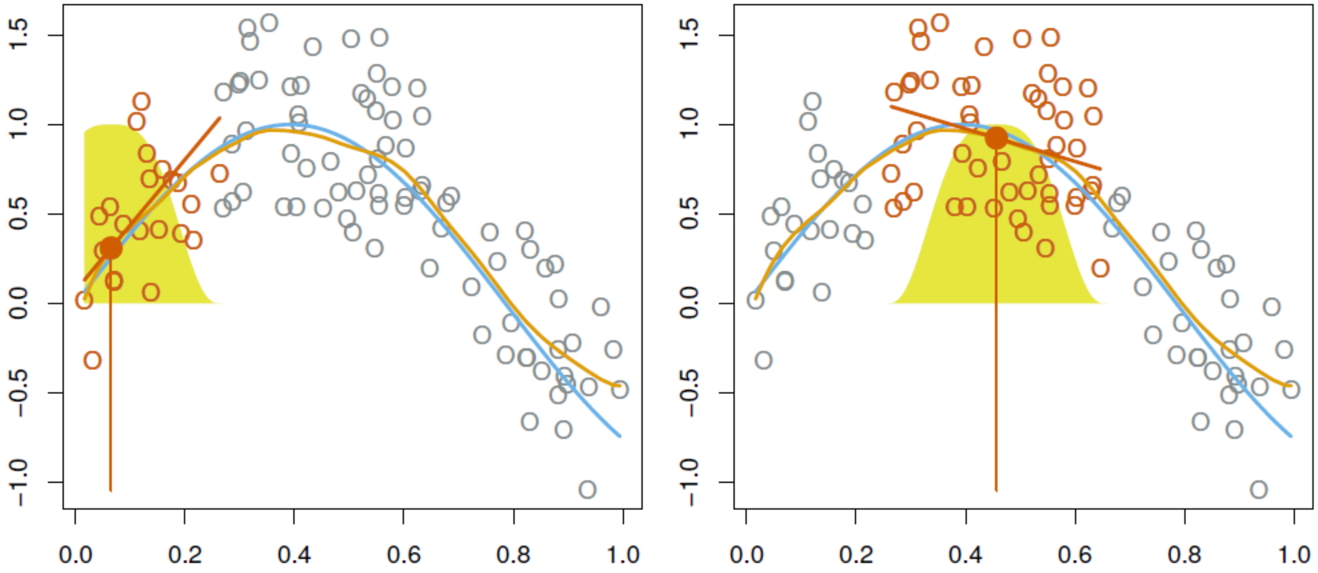
$$\underset{g \in \mathcal{S}}{\text{minimize}} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

- The first term is RSS, and tries to make g(x) match the data at each x_i.
- The second term is a roughness penalty and controls how wiggly g(x) is. It is modulated by the tuning parameter λ ≥ 0.
  - The smaller λ, the more wiggly the function, eventually interpolating y_i when = 0.
  - As λ→∞, the function g(x) becomes linear.

Smoothing Splines continued

- The solution is a natural cubic spline, with a knot at every unique value of x_i. The roughness penalty still controls the roughness via λ.
- Some details
  - Smoothing splines avoid the knot-selection issue, leaving a single λ to be chosen.
  - The algorithmic details are too complex to describe here. In R, the function smooth.spline() will fit a smoothing spline.

Local Regression



- With a sliding weight function, we fit separate linear fits over the range of X by weighted least squares.
- See loess() function in R.

Local Regression

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0.

2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.

3. Fit a *weighted least squares regression* of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2.$$

4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

Generalized Additive Models

- Allows for flexible nonlinearities in several variables, but retains the additive structure of linear models.

- A natural way to extend the multiple linear regression model

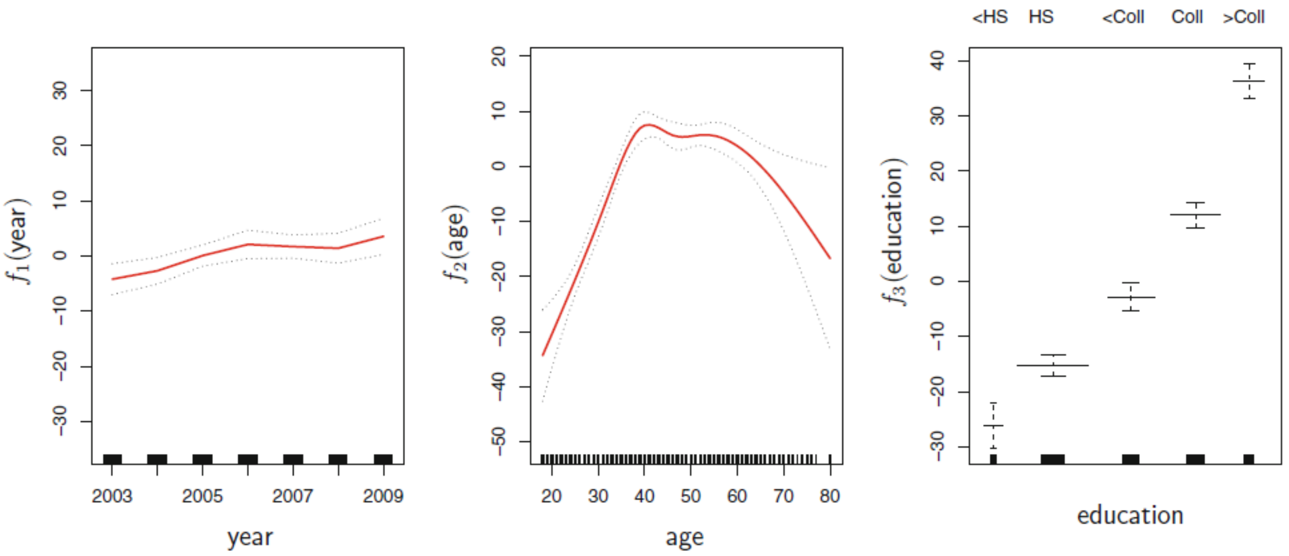
$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$$

is

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \\ &= \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i . \end{aligned}$$

Generalized Additive Models

$$\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$$



wage tends to increase slightly with year; this may be due to inflation,
wage tends to be highest for intermediate values of age, and lowest for the very young and very old,
wage tends to increase with education: the more educated a person is, the higher their salary, on average.

Pros and Cons of GAMs

- GAMs allow us to fit a non-linear function *to each variable, so that we can* automatically model non-linear relationships that standard linear regression will miss.
- The non-linear fits can potentially make more accurate predictions for the response *Y.*
- Because the model is additive, we can still examine the effect of each *variable on Y individually while holding all of the other variables* fixed
- The smoothness of the function *for each variable can be summarized* via degrees of freedom.
- The main limitation of GAMs is that the model is restricted to be additive.

GAMs for Classification Problems

- GAMs can also be used in situations where *Y* is qualitative.
- For simplicity, here we will assume *Y* takes on values zero or one, and let *p*(*X*) = P(*Y* =1|*X*) be the conditional probability (given the predictors) that the response equals one.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p).$$

GAMs for Classification Problems

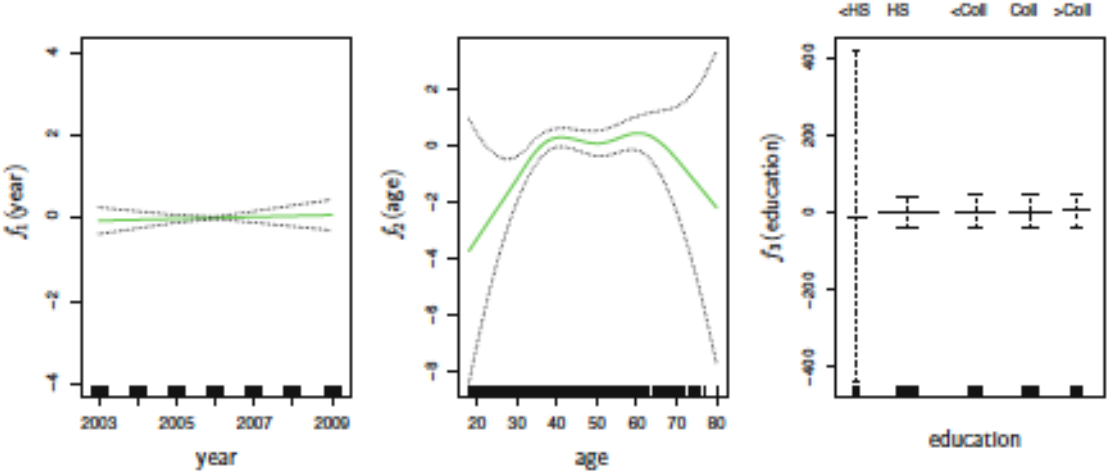
- We fit a GAM to the Wage data in order to predict the probability that an individual's income exceeds $250,000 per year. The GAM that we fit takes the form

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 \times \texttt{year} + f_2(\texttt{age}) + f_3(\texttt{education})$$

Where

$$p(X) = \Pr(\texttt{wage} > 250|\texttt{year}, \texttt{age}, \texttt{education}).$$

- *The first function is linear in* year*, the second function a smoothing spline with five degrees of freedom in* age*, and the third a step function for* education*.*


$f_1(\text{year})$
year
$f_2(\text{age})$
age
<HS
HS
<Coll
Coll
>Coll
$f_3(\text{education})$
education