Regression

Data compression

X
12
12
12
12

Memorizing the equation X = 12 we make data compression. Representing the table needs less space.

Coding the equation we can come back finding exactly same data.

Compression without loss of information.

X
12
13
10
12

Memorizing the equation X = 12 we make data compression again. But this coding is not perfectly invertible. Decoding, we find only an approximation of original data.

Compression with loss of information.

coding: (12, 13, 10, 12) → “X = 12”

decoding: “X = 12” → (12, 12, 12, 12)

$$decode(code(original)) \neq original$$

There is some **distortion**.

Distortion depends on the sample and the constant number we use as code for all elements of the original table.

Problem: given a sample $\{x_1, \ldots, x_n\}$ find a constant representing it in the best way

Intuitive answer: the mean.

$$code(\{2, 9, 10\}) = 7$$

Neither 38 nor 2.

The intuition is right and has a mathematical counterpart.

Using a constant k as code, the distortion can be defined as the sum of squared errors:

$$distortion = \sum_{i=1}^{n} (x_i - k)^2$$

The distortion of estimate 5 for sample {2, 9, 10} is

$(2 - 5)^2 + (9 - 5)^2 + (10 – 5)^2 = 9 + 16 + 25 = 50.$

The distortion of estimate 8 is

$(2 - 8)^2 + (9 - 8)^2 + (10 – 8)^2 = 36 + 1 + 4 = 41.$

Equation $x_i = 8$ codes (represents, approximates) the sample better than equation $x_i = 5$.

The distortion of estimate 7 is

$(2 - 7)^2 + (9 - 7)^2 + (10 – 7)^2 = 25 + 4 + 9 = 38$.

No other constant can do better.

Equation $x_i = 7$ codes in the optimal way.

More rigorously:

Equation $x_i = 7$ is the equation of form $x_i = k$,

being k a constant,

that minimizes the distortion for sample {2, 9, 10}, being distortion defined as the sum of squared errors.

We are circumscribing ourselves to equations of form

$$x_i = k$$

We have chosen a *model*, a form of equation giving us an estimate for each data point (each number in the data set).

The equation $x_i = 7$ is not the best possible equation, only the best possible equation following this model.

Modeling a data set means:

1. Choosing a model, i.e. an equational form containing parameters like *k*;
2. Choosing optimal parameters values, i.e. values that minimize the distortion.

Our model

$$x_i = k$$

is a case of the general model

$$x_i = f(x_i)$$

where $f(x_i, \theta)$, the estimation, is a function of the data point itself and a parameter θ, chosen by the analyst.

In our case

$$x_i = f(x_i, k) = k$$

Here the parameter is k and the estimation function f is independent on x_i.

Strange estimation function: it ignores data.

But only apparently: the parameter k really exploits data, not point by point but as a whole set.

If *k* is the mean μ

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Then the average distortion is the variance σ^2

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$$

The best choice for parameter *k* in the model

$$x_i = k$$

is the mean μ (i.e. 7 for the sample {2, 9, 10}).

Our model

$$x_i = k$$

is a case of the model family

$$x_i = a_n x_i^n + a_{n-1} x_i^{n-1} + \ldots + a_1 x_i^1 + a_0 x_i^0$$

This is the family of polynomial models of degree n.

Set $n = 0$ and $a_0 = k$. You obtain our equation.

Indeed, the mean is the best choice of parameter value for a polynomial model of degree 0.

If you want to ignore data points, the best estimation equation you can build up is polynomial of degree 0 with the mean of the sample as parameter.

Why to ignore data points?

They could be not accessible, or too expensive to collect.

Maybe you are not allowed to use them for privacy reasons.

If you know their mean, you can build up a meaningful model, which sounds:

Assume each point of the sample is equal to the sample mean.

You do not really have to believe this statement is true.

A model is a tool: you use it as an estimation machine.

Your are happy if it gives you useful estimations.

Useful does not mean exact, only close enough to reality.

Models bring additional value. They enable you to play games in a more profitable way.

A model is better than another (for you and for a certain game) if playing rationally it gives you a greater expected additional profit.

It brings added revenue and added cost.

This concept is related to the concept of expected information value.

X	*Y*
2	7
5	13
6	15
11	25

If data is this and we memorize

$$Y = 2X + 3$$

then we have data compression without information loss.

Decoding comes back to original values of Y.

X	*Y*
2	8
5	13
6	13
11	26

If data is this and we memorize

$$Y = 2X + 3$$

then we have data compression with information loss.

Decoding gives an approximation of original values of Y.

E.g. the third row assigns Y = 2 * 6 + 3 = 15 instead of 13.

We code the attribute *Y* with a function *f* of another attribute *X*.

In general, we define

$$distortion = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

where

$$\hat{x}_i = decoding(coding(x_i))$$

or simply

$$\hat{x}_i = f(x_i)$$

f(x) is called *regression function*.

The regression function gives an estimate of the value of attribute *Y* for each observation in the sample, using the value of attribute *X*.

The sample is called the *training set*.

The regression function is built choosing the general model, then computing the “best” parameters exploiting information in the training set.

The regression function is an estimator of known data, or a predictor of already observed data.

The idea is: if this equation can reconstruct the training set with good approximation, then it is likely to reconstruct unknown samples as well.

The real goal is to estimate or predict new unknown data.

Interpolation

We are in search for a function *f(X)* estimating *Y* while minimizing distortion.

In these terms the problem is not well-posed. There are infinite functions doing what we want.

Il problema non è ben posto. Ce ne sono infinite. One is the *interpolation polynomial*.

Given a set of points

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

You can always get a polynomial in *X* touching all points: $P(x_i) = y_i$.

The interpolation polynomial generally has degree $n - 1$, i.e. the biggest power is x_i^{n-1}.

A first problem is that substituting *n* values of *Y* with *n* coefficients of the polynomial gives no real benefit (tehre is no compression).

Note: a polynomial of degree *n – 1* has *n* coefficients, including the constant

$$a_{n-1}x^{n-1} + ... + a_1x^1 + a_0$$

A second problem (deriving form the first) is that a polynomial of high degree is a geometric curve with wild oscillations. See the next figure for an example.

Yet, real phenomena (including economic ones) generally do not oscillate in such a way.

If we get a slightly different sample, the polynomial can change greatly. Thus. Our predictions about observations outside the sample change radically.

Such an unstable regression function is not useful.

The figure shows how polynomials of increasing degree are closer and closer to data, but also more and more oscillating.

Some graphs are represented for a small part only, because too big.

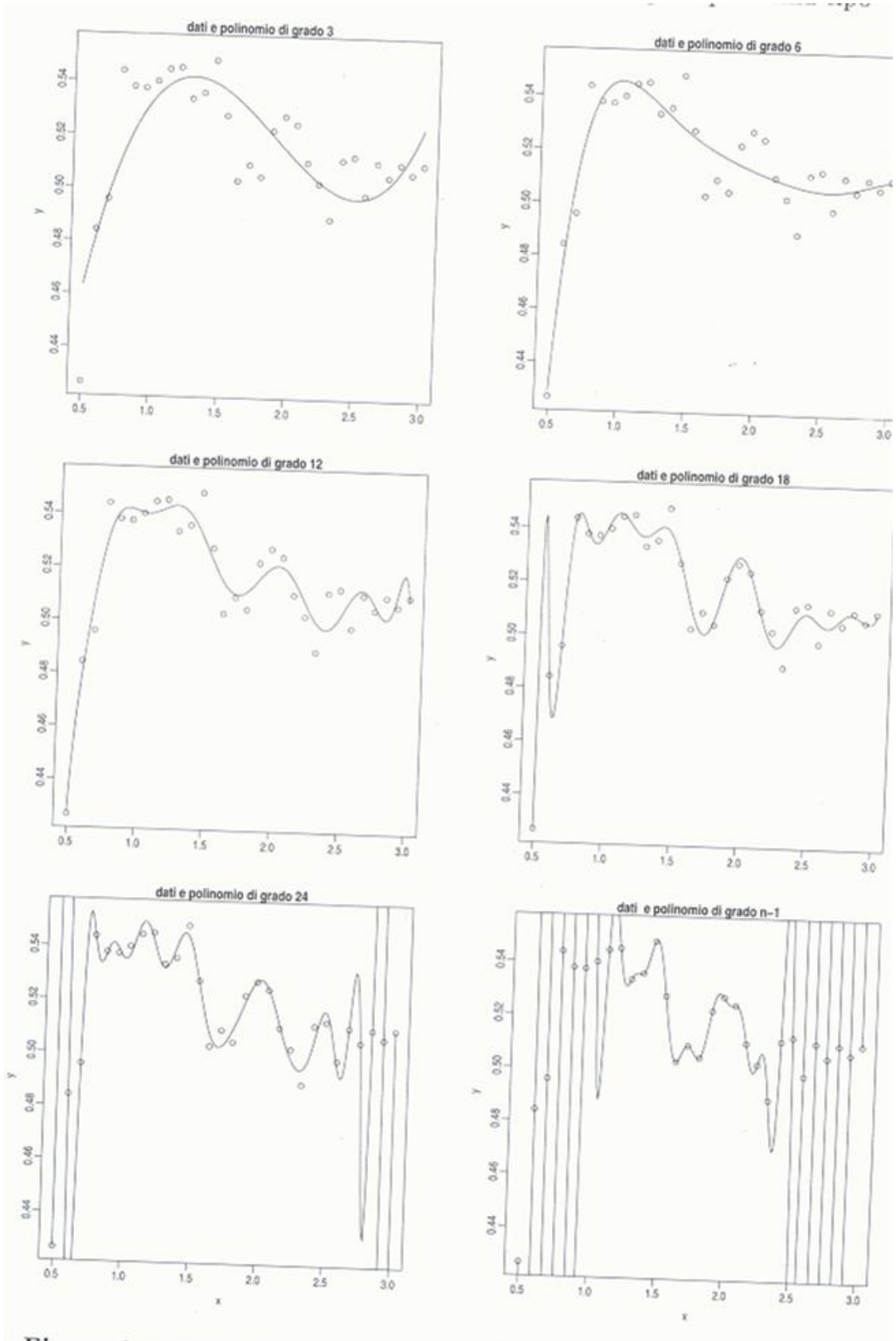

Figura 3.2 Dati (di

We have two contrasting needs:

• f(X) must fit well to data in training set, i.e. it must have small distortion (property of *accuracy*);

• f(X) must not oscillate very much and not change much if the training set change slightly, i.e. it must have small variance (property of *stability*).

There is a trade-off between the distortion with respect to the training set and the intrinsic variance.

When the degree of the regression polynomial increases, generally the distortion decreases but the variance increases.

The polynomial tracks random oscillations of data, often increasing the variance with negligible gain in distortion.

This phenomenon is named *overfitting*: the polynomial learns “too well”. It adapts to features of the sample which are not real features of the population which the sample is extracted from. It learns form “noise”.

So, it is not very generalizable outside the sample and not very useful for prediction.

Measuring the fitness

Fitness (accuracy, low distortion) is good thing, but it is about already known data. We also need generalization, capability to be appleid to unknown data. The regression function is required to predict the future, not only to explain the past.

We accept some distortion in order to avoid overfitting.

We saw how to measure absolute distortion, We can also measure relative distortion, using a benchmark.

The benchmark is classical variance, which is the distortion of this polynomial function of degree zero:

$$f(x_i) = \mu$$

where μ is the mean of the training dataset.

The prediction rule is: given any x_i value, predict $y = \mu$, without any consideration of x_i features.

This is the best possible constant (blind) prediction rule.

Let δ^2 be the average distortion (sum of square errors divided by the number of points).

A regression function *f* is really useful if $\delta^2 < \sigma^2$.

The measure δ^2 is named *residual variance* and σ^2- δ^2 explained variance (or absorbed variance).

The explained variance measures the effectiveness of *f(X)* when predicting the value of *Y* for an observed x_i.

For example, if $\delta^2 = ½ \sigma^2$ means that the oscillation of true y_i values around the predicted values is half the oscillation around the mean.

Classes of regression functions

We do not search “the best regression function”. This would be an ill-posed problem: infinite functions are the best ones.

First, we design a model for regression functions, a structure, a formula with parameters to be assigned.

Then we search for the optimal values of these parameters. After assigning these values, we have the best funciton in the chosen class.

If we choose the model $y = \theta$, then the best assignment for parameter θ is the mean of the training dataset. This is a limit case of polynomial regression functions (here with degree zero).

We can choose polynomial models of degree n or any other functional form.