IN5526 - Web Intelligence

Lecture 3

Juan Domingo Velasquez Silva
Cristobal Gaspar Ignacio Pizarro Venegas

Departamento de Ingeniería Industrial
Universidad de Chile

September 13, 2016

Contents

Supervised learning

In supervised learning, our data is of the form

$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$$

Where

- $x \in \mathbb{R}^n$ are **feature vectors**, **examples** or **instances**
- Components in x are **attributes** or **features**
- $y \in \mathbb{R}$ are **labels** or **targets**

- if $y \in \mathbb{Z}$, the problem is **Classification**
- if $y \notin \mathbb{Z}$, the problem is **Regression**

Supervised learning

We try to find a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that maps the instances to the targets, in the *best* way possible, and can be used with other unknown instances

Using the data

The ideal case, with a lot of data

Training set

Data used for fitting a model.

Validation set

Data used for choosing the best model, or adjusting hyperparameters of model.

Test set

Data used for testing the model and show final performance.

Cross-validation

When there is not so much data

Training + Validation set

Data used for model fitting and model selection.

Test set

Data used for testing the model and show final performance.

Contents

Linear regression

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p$$

$$\min_{w} ||Xw - y||_2{}^2$$

It has a nice analytical solution

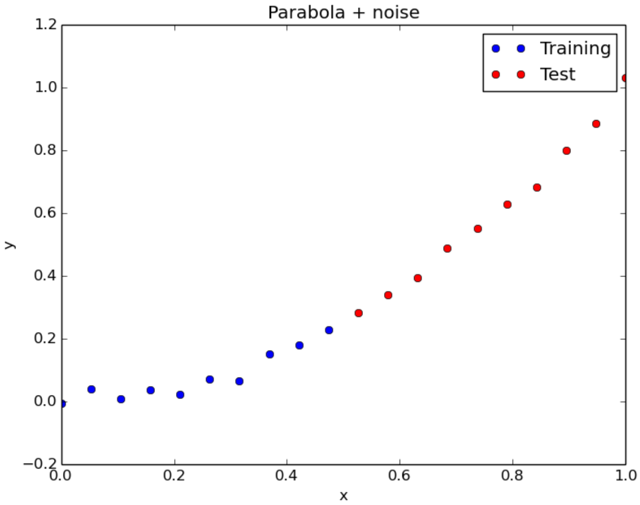
$$w^* = (X^T X)^{-1} X^T y$$

Or we can use numerical methods on it, like gradient descent.
It can be evaluated with the coefficient of determination

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

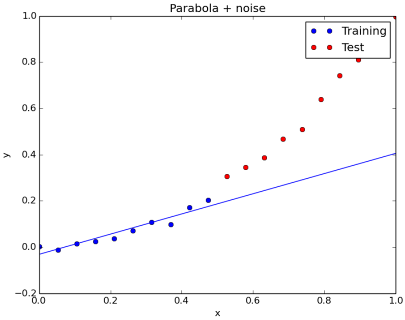
Linear regression

Let's consider data generated with $y = x^2 + \mathcal{N}(0, 0.01)$



Linear regression

We fit a line with the **training** data

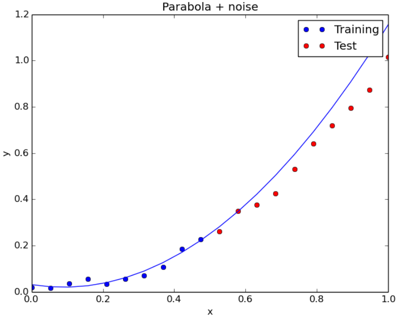


Performance in

- Training: $R^2 = 0.874$
- Test: $R^2 = -1.3$

Linear regression

We fit a parabola with the **training** data (still linear regression)

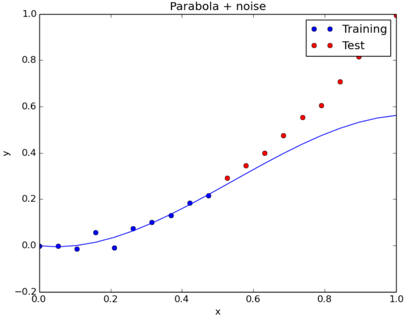


Performance in

- Training: $R^2 = 0.98$
- Test: $R^2 = 0.45$

Linear regression

We fit a cubic polynomial with the **training** data



- Training: $R^2 = 0.99$
- Test: $R^2 = -8.05$

Overfitting

As the model becomes more complex (more terms in the polynomial fit), two things happen:

- Performance in the training set increases
- Performance in the test set increases, then decreases

K-fold cross-validation

Finding the best model

With the training + validation dataset

- Divide data (instances + labels) in K *folds*
- Fit model with K-1 folds
- Evaluate model with the remaining fold
- Repeat with another fold
- Report performance on the model as the average of the performances for each run

With this we select *the model*. That is, the model or its hyperparameters. Finally we train the model with the training + validation dataset, evaluate with the test set and report performance