Lecture 1 - Linear Regression

Rodrigo Loza[1]

[1]Computational Biologist - pfm Medical
Biomedical Engineer - Universidad Católica Boliviana "San Pablo" La Paz, Bolivia

August, 2017

Outline

Resources

Recommended books

- Pattern Recognition and Machine Learning [Bis06]
- Machine Learning: A Probabilistic Perspective [Mur12]
- Machine Learning: The Art and Science of Algorithms That Make Sense of Data [Fla12]

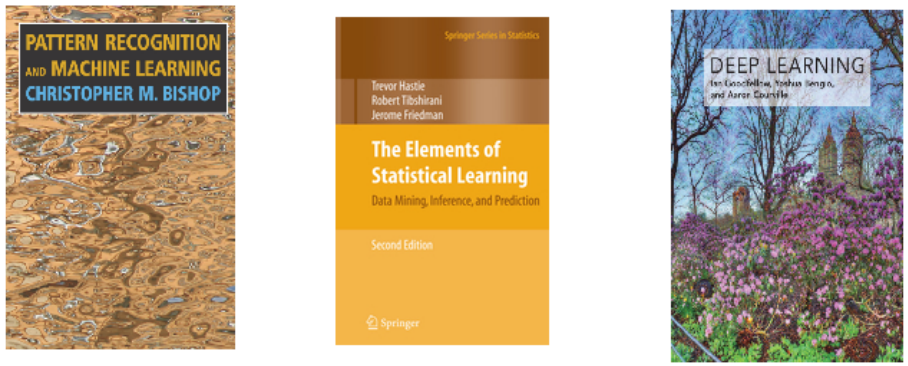


Figure 1: Recommended books

Data Sets

Figure 2: Examples datasets

Problem example

Figure 3: Netflix competition - recommendation systems

Problem visualization

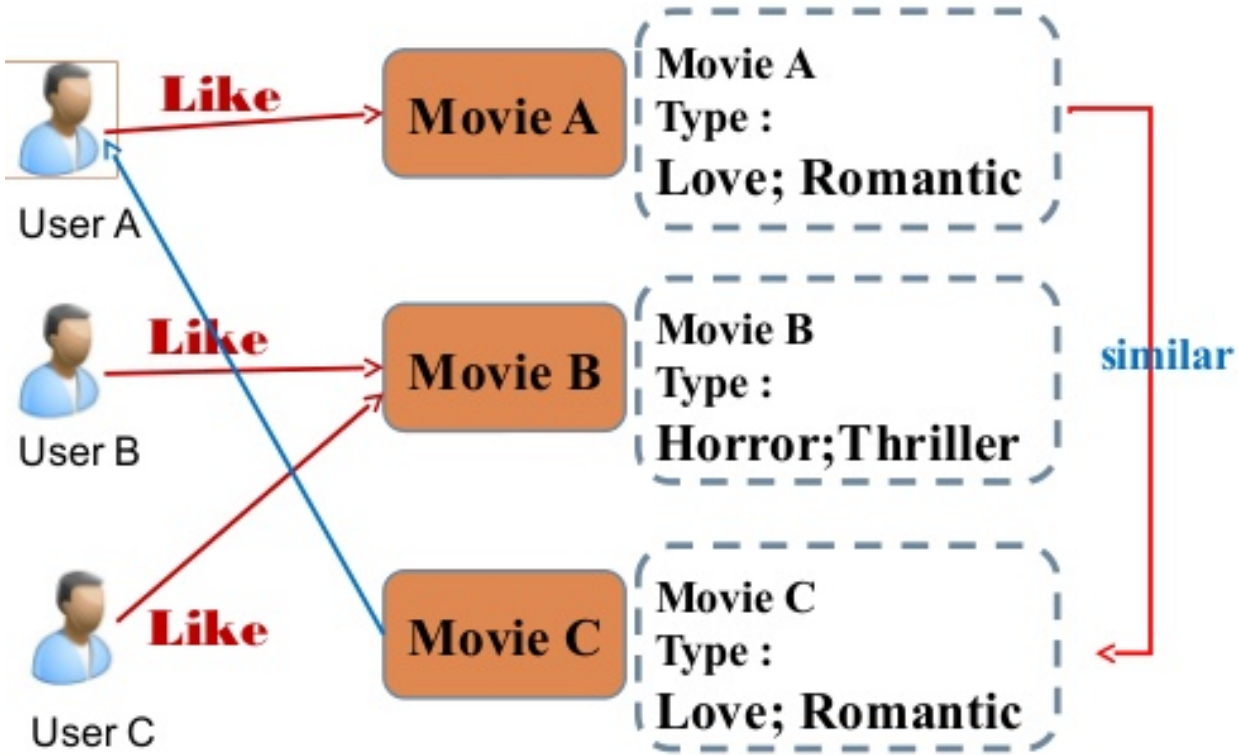


Figure 4: Visual way to describe the problem

Problem: Predicting how a viewer will rate a movie
10% improvement - 1 million dollars
The essence of machine learning:

1. A pattern exists

Problem: Predicting how a viewer will rate a movie
10% improvement - 1 million dollars
The essence of machine learning:

1. A pattern exists
2. There is no way to pin it down mathematically

The components of learning

Formalization:

1. Input x: customer behaviour

The components of learning

Formalization:

1. Input x: customer behaviour
2. Output y: what movies does the customer like?

The components of learning

Formalization:

1. Input x: customer behaviour
2. Output y: what movies does the customer like?
3. Target function: f: x $->$ y

The components of learning

Formalization:

1. Input x: customer behaviour
2. Output y: what movies does the customer like?
3. Target function: f: x $->$ y
4. Data: (x1,y1), (x2,y2), (x3,y3) ... (xn,yn)

The components of learning

Formalization:

1. Input x: customer behaviour
2. Output y: what movies does the customer like?
3. Target function: f: x $->$ y
4. Data: (x1,y1), (x2,y2), (x3,y3) ... (xn,yn)
5. Hypothesis: g: x $->$ y

Learning model

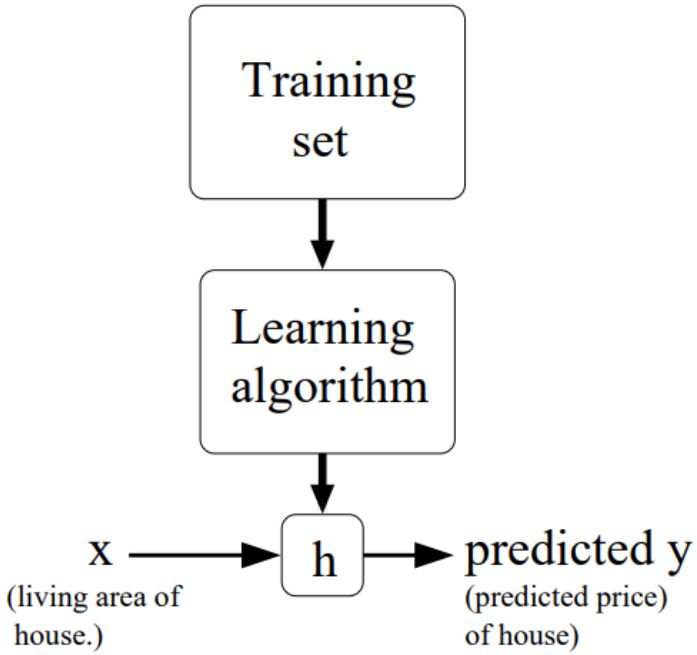


Figure 5: Learning model

Differences between a task and a learning model

...tasks are addressed by models, whereas learning problems are solved by learning algorithms that produce models.[Fla12]

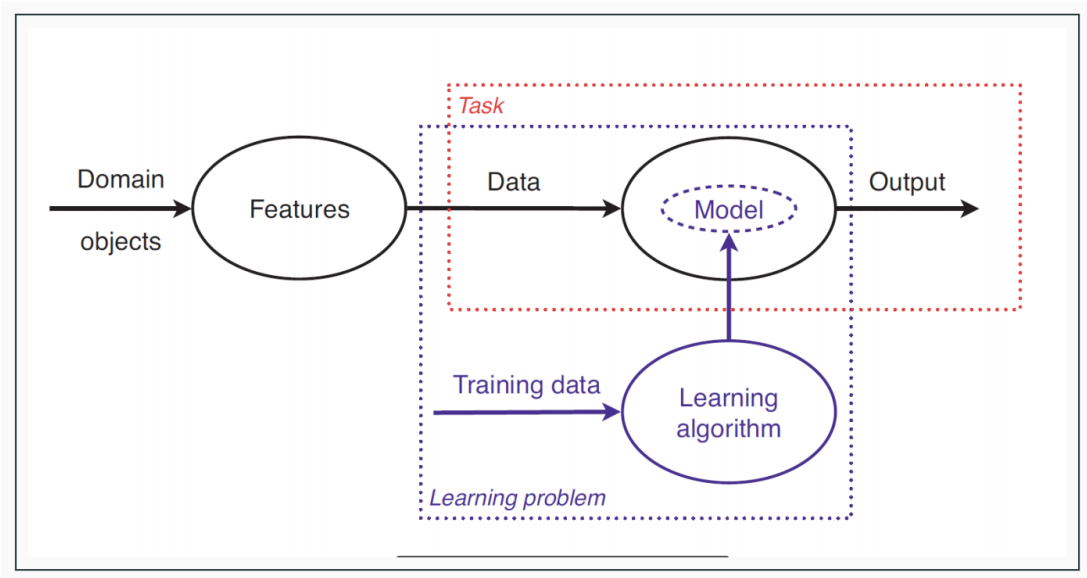
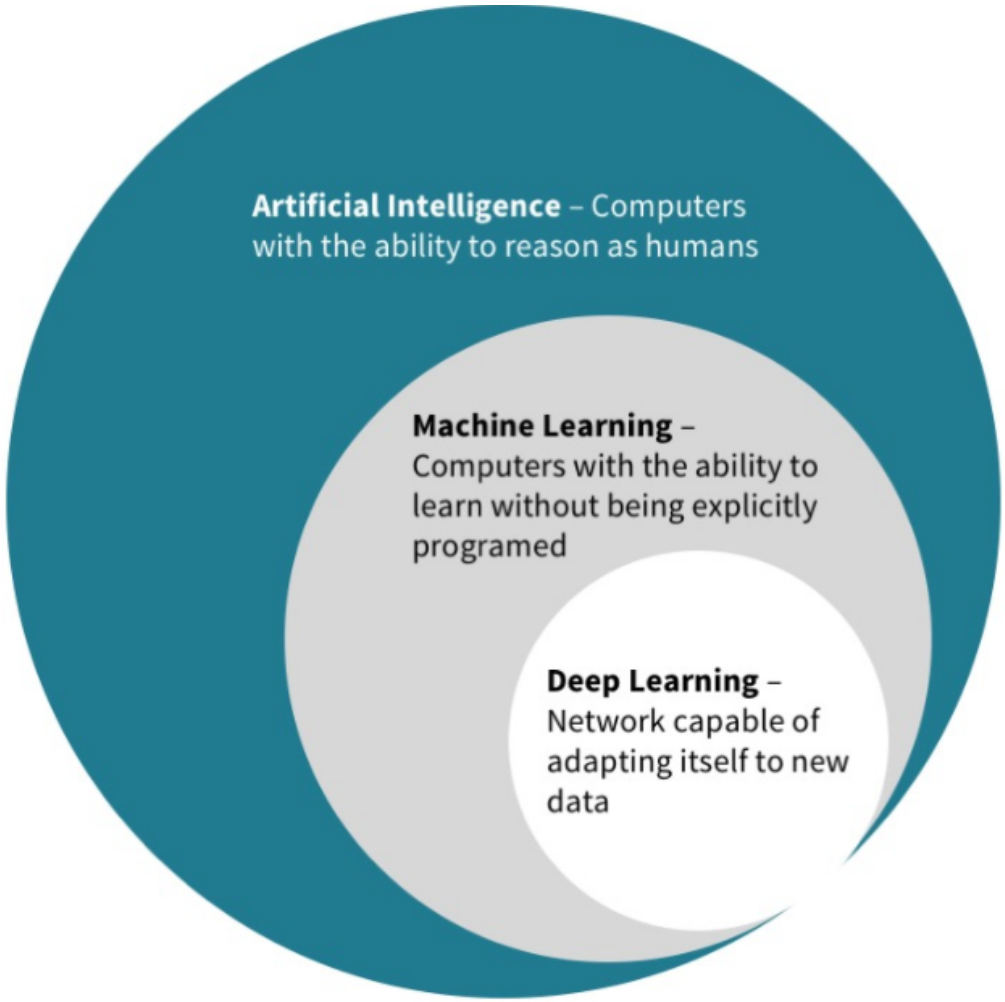


Figure 6: Difference between a task and learning problems

What is Machine Learning?

- Field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel, IBM, Stanford University, 1959
- A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. Tom Mitchell [Mit97]
- Instead of writing a program by hand, we collect lots of examples that specify the correct output for a given input. A machine learning algorithm then takes these examples and produces a program that does the job. Geoffrey Hinton [Hin14]

ML is a subset



ML solves problems with the following shape

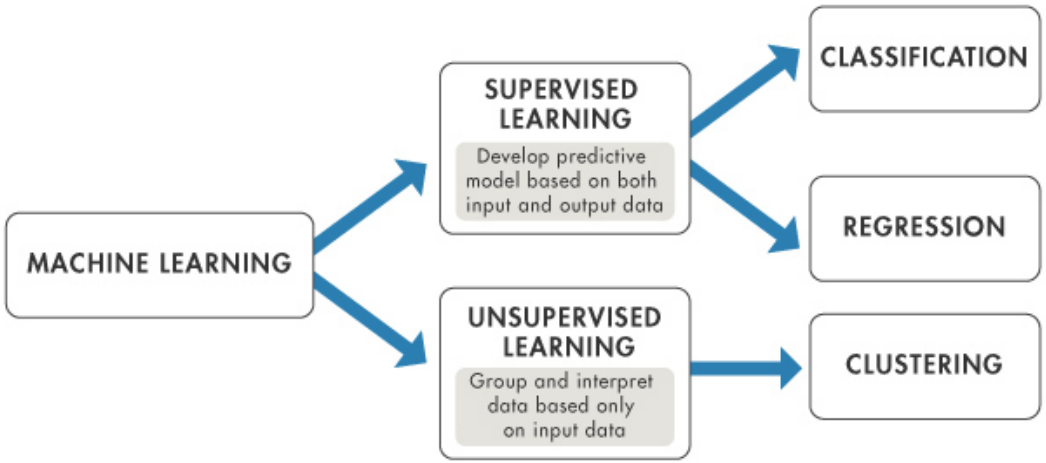
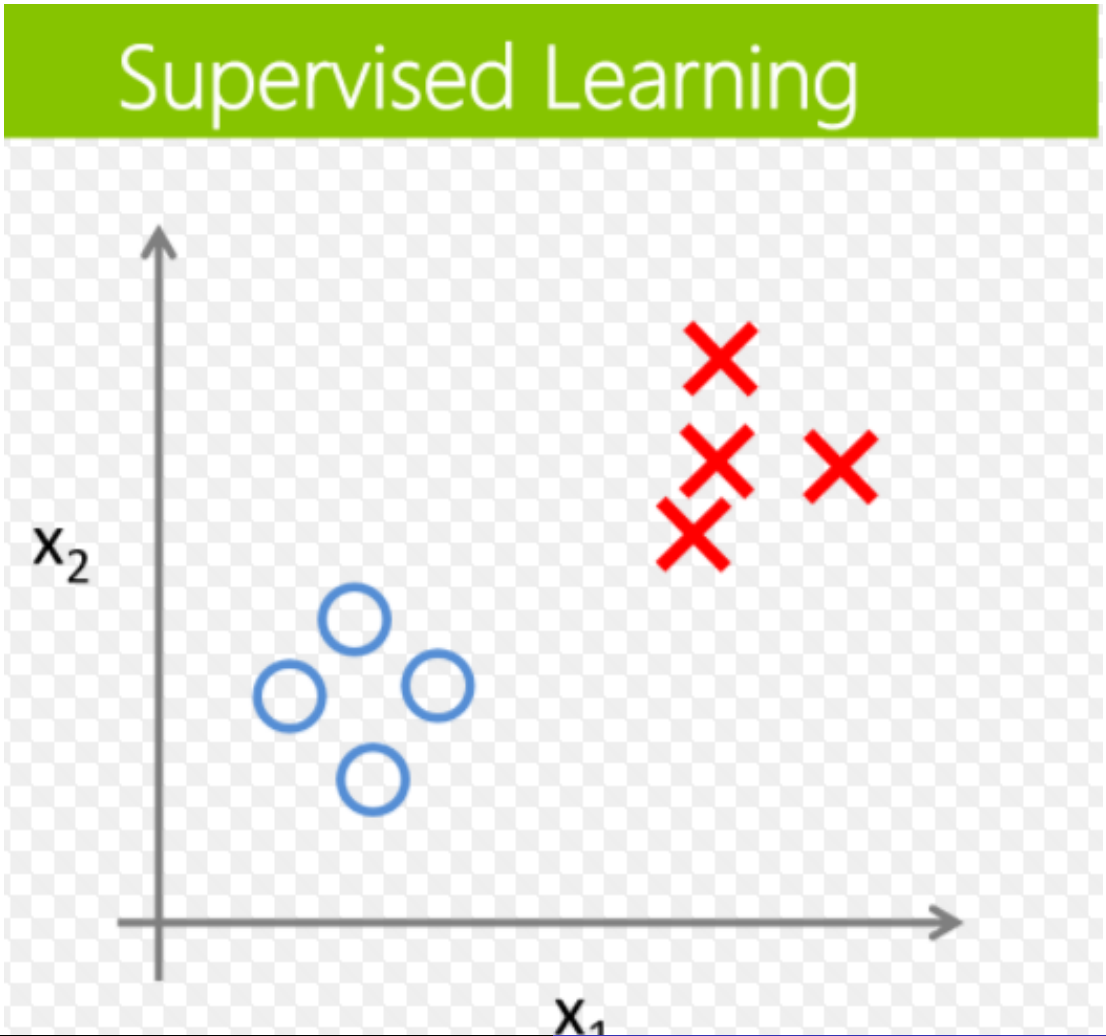
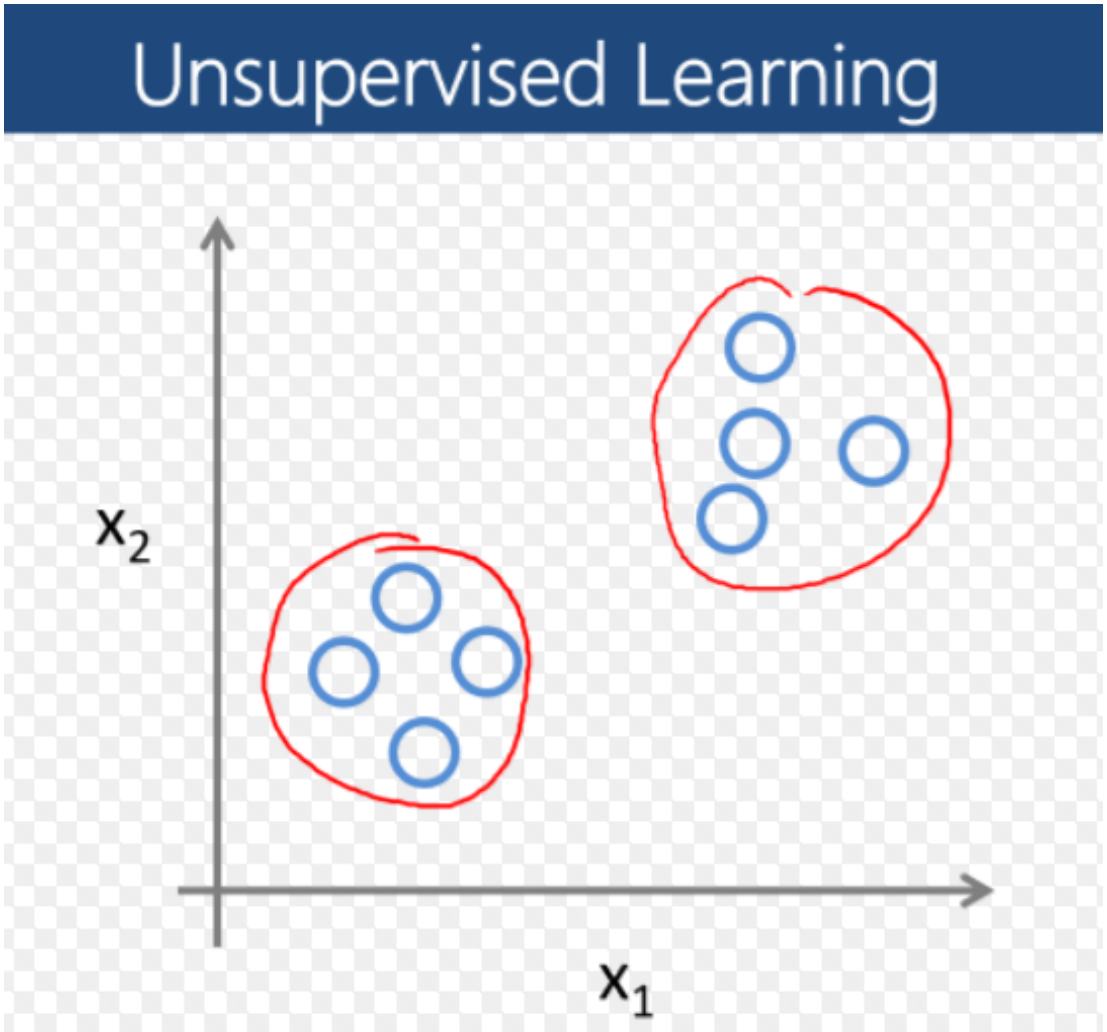


Figure 8: Types of problems machine learning solves

Supervised learning - probability interpretation



Unsupervised learning - probability interpretation



Definition

- Model a single response **(dependent, continuous, outcome)** variable based on one or more input **(independent, predictor)** variables

Linear Regression example

- Suppose that we are given a training set comprising N observations of x, written **x = (x1, ..., xn)** together with corresponding observations of the values of y, denoted **y = (y1, ..., yn)**
- The input data set x in Figure 1.2 was generated by choosing values of xn, for n = 1,...,N, spaced uniformly in range [0, 1], and the target data set t was obtained by first computing the corresponding values of the function sin(2pix)

Linear Regression example

Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t. The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the green curve.

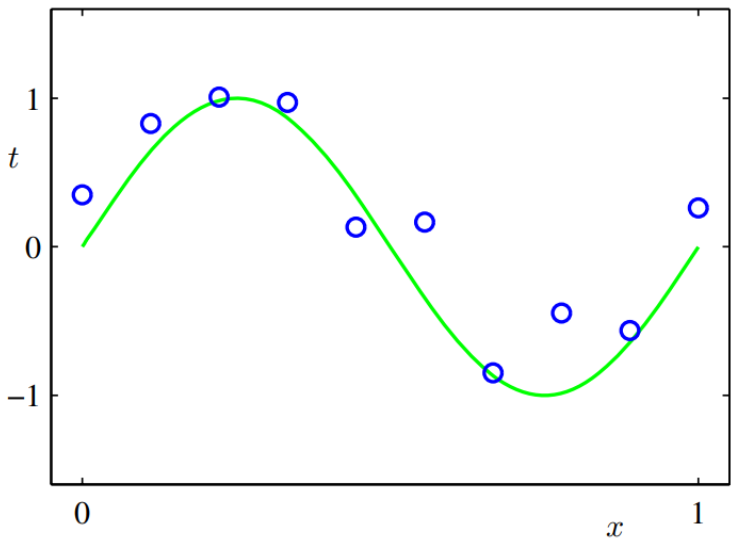


Figure 11: Target function

Curve fitting approach - Hypothesis

We shall fit the data using a polynomial function of the form:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Figure 12: Linear combination of learning parameters and features

Curve fitting approach - Cost function

The values of the coefficients will be determined by fitting the polynomial to the training data. This can be done by minimizing an **error function** that meaures the misfit between the function **y(x, w)**, for any given value of w, and the training set data points. The simplest error function is the **sum of squares** of the errors between the predictions **y(xn, w) for each data point xn and the corresponding target values yn, so that we minimize:**

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

Figure 13: Cost function for continuous type variables

Curve fitting approach - Cost function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

The error function (1.2) corresponds to (one half of) the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function $y(x, \mathbf{w})$.

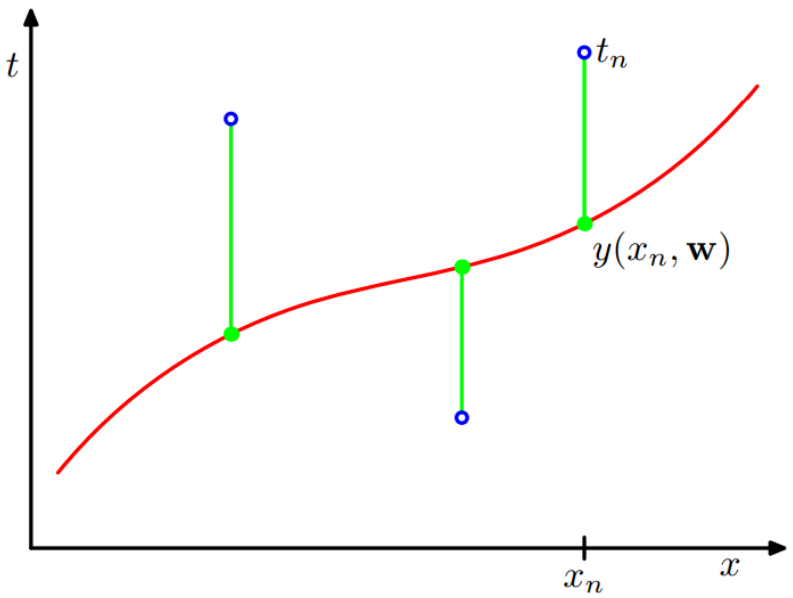
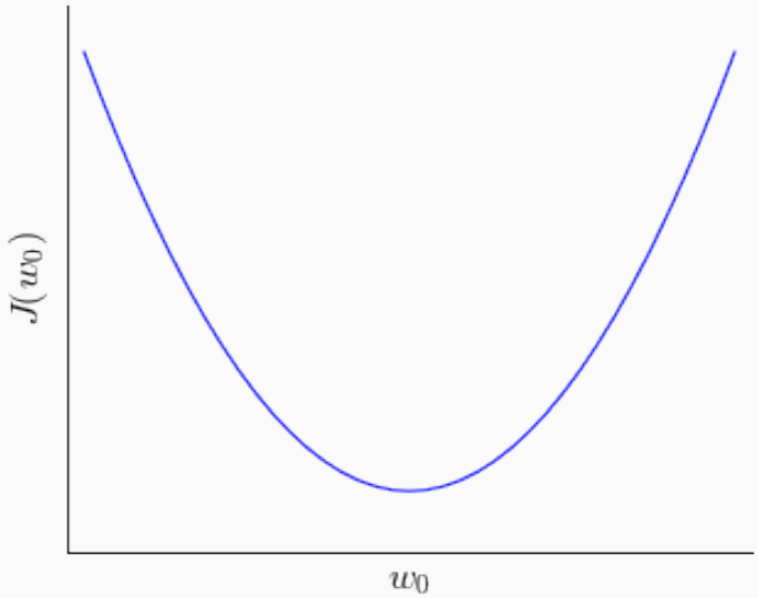


Figure 14: Cost function for continuous type variables

Curve fitting approach - Intuition about solving the problem

We can solve the curve fitting problem by choosing the value of w for which E(w) is as small as possible. Because the error function is a quadratic function of the coefficients w, its derivatives with respect to the coefficients will be linear in the elements of w, and so the minimization of the error function has a unique solution, denoted by w$*$, which can be found in closed form. The resulting polynomial is given by the function y(x, w$*$).

Curve fitting approach - Gradient Descent



Update rule

$$\frac{d}{dw_0}J(w_0)$$

$$w_0 := w_0 - \alpha\frac{d}{dw_0}J(w_0)$$

- A positive $\alpha\frac{d}{dw_0}J(w_0)$ moves w_0 to the left
- A negative $\alpha\frac{d}{dw_0}J(w_0)$ moves w_0 to the right

Figure 15: Gradient descent

BGD, SGD

Batch gradient descent

BGD, SGD

Stochastic gradient descent

Gradient Descent - Multiple Regression problem

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}^T$$

$$\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}^T$$

$$\boldsymbol{w} = \{w_0, w_1, \ldots, w_D\}^T$$

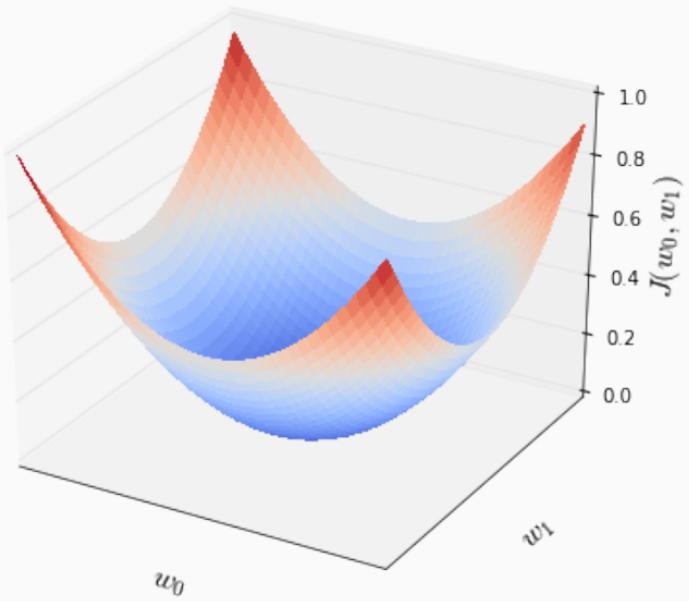
$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} J(\boldsymbol{w})$$

$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = \frac{\partial}{\partial w_j} \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 = \frac{1}{2N} \sum_{i=1}^{N} \frac{\partial}{\partial w_j} (\hat{y}_i - y_i)^2 =$$

$$\frac{1}{2N} \sum_{i=1}^{N} 2(\hat{y}_i - y_i)^2 \frac{\partial}{\partial w_j} (\hat{y}_i - y_i) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i$$

Figure 16: Multiple parameters increase the dimensionality of the parameters' vector

Curve fitting approach - How many features?



Solution

$$\frac{\partial}{\partial w_j} J(w_0, w_1)$$

repeat until convergence {

Simultaneously update for every j:

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} J(\boldsymbol{w})$$

}

Figure 17: Multi-dimensional cost function

Curve fitting approach - How many features?

Solve the problem using different dimensionalities for x

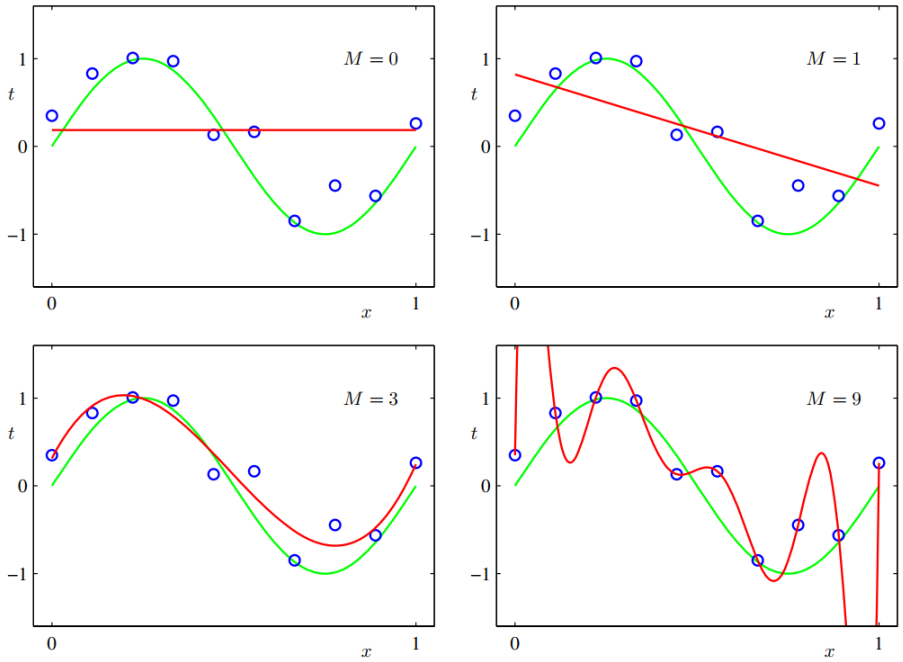


Figure 18: Multi-dimensional cost function

Curve fitting approach - Addressing overfitting (variance) and underfitting (bias)

Which is the correct number of features to be used?

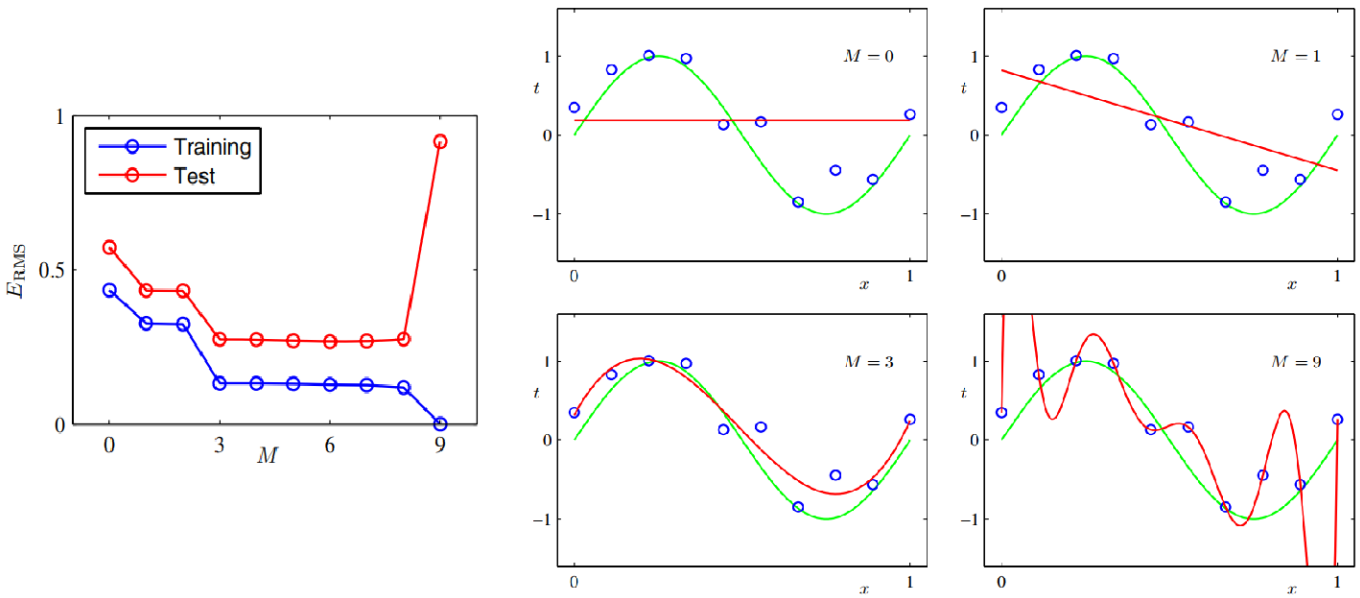


Figure 19: Diagnosing Overfitting and underfitting

Curve fitting approach - Addressing overfitting (variance) and underfitting (bias)

Table of the coefficients $\mathbf{w}^\star$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Figure 20: Why overfitting and underfitting?

Curve fitting approach - Causes of Overfitting

- Extensive search in hypothesis
- Too many features (curse of dimensionality)
- Insufficient examples

Curve fitting approach - Causes of Underfitting

- Not enough data points
- Not enough features
- Not enough polynomial terms to capture non-linearity

Curve fitting approach - Solve overfitting underfitting with regularization

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\text{where } \|\mathbf{w}\|^2 \equiv \mathbf{w}^{\mathrm{T}}\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2.$$

Curve fitting approach - Solve overfitting underfitting with regularization

Table of the coefficients $\mathbf{w}^\star$ for $M = 9$ polynomials with various values for the regularization parameter λ. Note that $\ln \lambda = -\infty$ corresponds to a model with no regularization, i.e., to the graph at the bottom right in Figure 1.4. We see that, as the value of λ increases, the typical magnitude of the coefficients gets smaller.

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Curve fitting approach - Solve overfitting underfitting with regularization

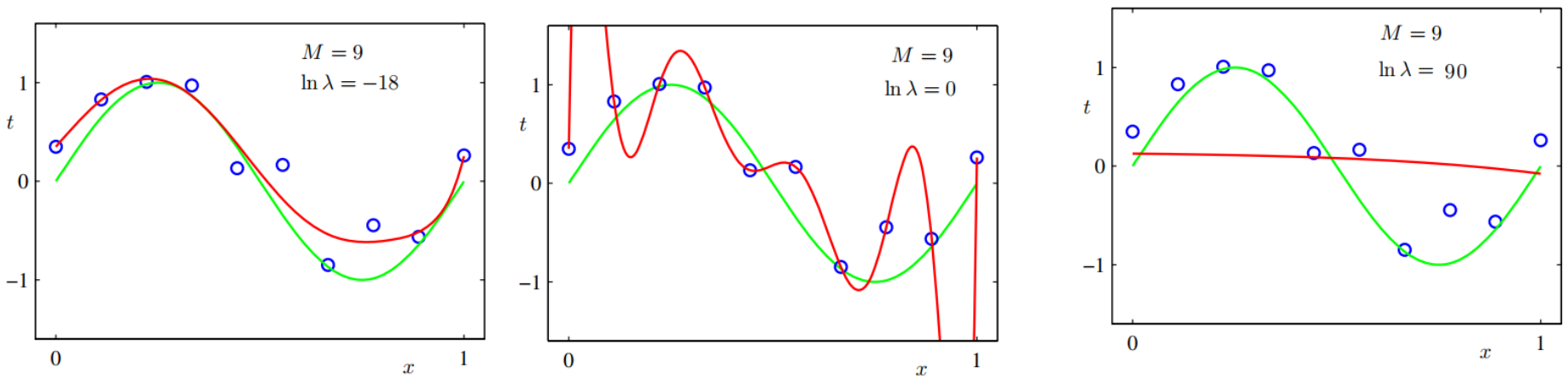


Figure 23: Apllying regularization

Curve fitting approach - Solve overfitting underfitting adding more data (always works)

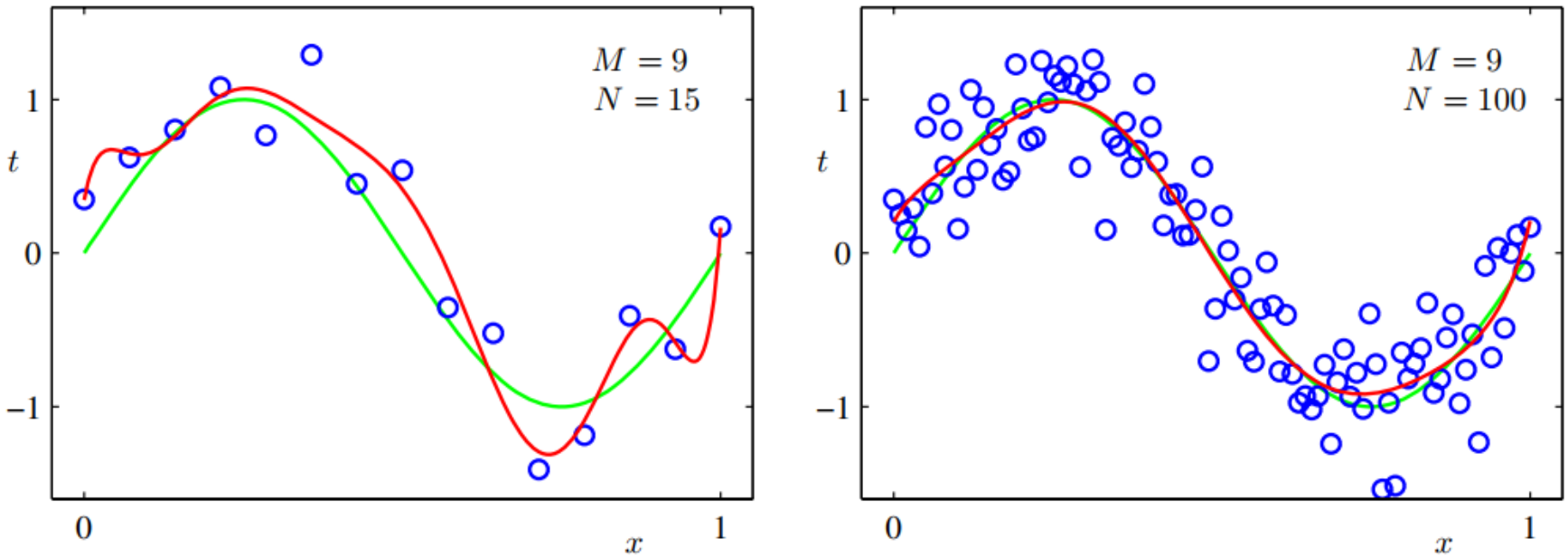


Figure 24: The more data, the better

Summary

- Hypothesis
- Cost function
- Learning algorithm

Summary - Hyperparameters

Hyperparameters:

References I

- Olivier Bousquet and Leon Bottou, The tradeoffs of large scale learning, Advances in Neural Information Processing Systems 20 (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), Curran Associates, Inc., 2008, pp. 161168.
- Christopher M. Bishop, Pattern recognition and machine learning (information science and statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- Peter Flach, Machine learning: The art and science of algorithms that make sense of data, Cambridge University Press, New York, NY, USA, 2012.
- Geoffrey Hinton, Csc321: Introduction to neural networks and machine learning, University Lecture, 2014.

References II

- Thomas M. Mitchell, Machine learning, 1 ed., McGraw-Hill, Inc., New York, NY, USA, 1997.
- Kevin P. Murphy, Machine learning: A probabilistic perspective, The MIT Press, 2012.
- Andrew Ng, Note 1, Lecture notes in CS 229 Machine Learning, 2012.