High-Dimensional Data Analysis

Lecture 3 - Linear Regressions

Fall semester - 2024

Dr. Eng. Valentin Leplat

Innopolis University

September 11, 2024

Outline

1. Supervised learning

2. Linear Regression with one variable
   - Main Formulation
   - Direct Methods
   - Iterative Methods

3. Linear Regression with Multiple Variable

4. Probabilistic interpretation

5. Summary

Supervised learning

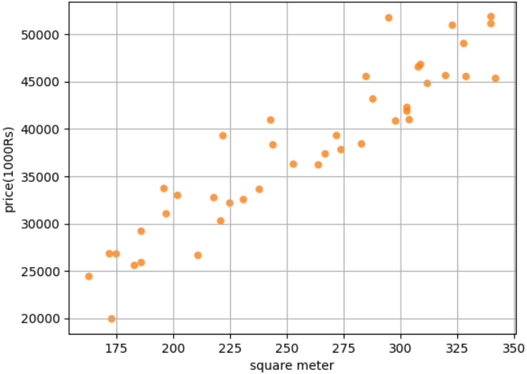
Supervised learning in a nutshell

- Lets start by talking about a few examples of supervised learning problems.
- Suppose we have a dataset giving the living areas and prices of 42 houses from Kazan:

Living area (m^2)	Price (1000 Rs)
238	33634
231	32547
340	51871
320	45639
211	26660
⋮	⋮

Supervised learning in a nutshell

We can plot this data:



Given data like this, how can we learn to predict the prices of other houses in Kazan, as a function of the size of their living areas?

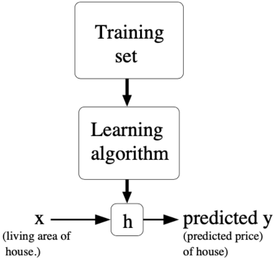
Training set

- To establish notation for future use, we'll use:
  1. $x^{(i)}$ to denote the "input" variables (living area in this example), also called input **features**,
  2. and $y^{(i)}$ to denote the "output" or target variable that we are trying to predict (price).
- A pair $(x^{(i)}, y^{(i)})$ is called a **training example/sample** .
- and the dataset that we'll be using to learn—a list of m training examples $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$—is called a **training set**. [1]
- We will also use $\mathcal{X}$ denote the space of input values, and $\mathcal{Y}$ the space of output values. In this example, $\mathcal{X} = \mathcal{Y} = \mathbb{R}$.

[1] Note that the superscript "(i)" in the notation is simply an index into the training set, and has nothing to do with exponentiation.

h hypothesis

- To describe the supervised learning problem slightly more formally, our goal is: *given a training set, to learn a function*
$$h : \mathcal{X} \to \mathcal{Y}$$
so that $h(x)$ is a "good" predictor for the corresponding value of y.
- For historical reasons, this function h is called a **hypothesis**.
- Seen pictorially, the process is therefore like this:



Terminology: Regression and Classification

- When the target variable that we're trying to predict is continuous, such as in our housing example, we call the learning problem a **regression problem**.
- When y can take on only a small number of discrete values [2], we call it a **classification problem**.

[2] such as if, given the living area, we wanted to predict if it is a house or an apartment, say

Linear Regression with one variable

Outline

1 Supervised learning

2 Linear Regression with one variable
- Main Formulation
- Direct Methods
- Iterative Methods

3 Linear Regression with Multiple Variable

4 Probabilistic interpretation

5 Summary

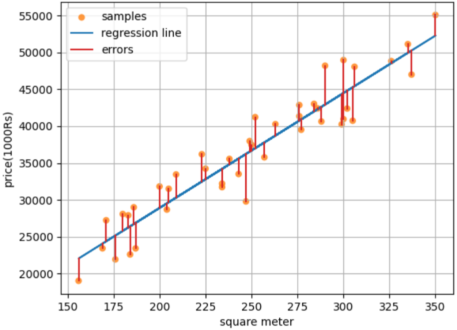
One-dimensional linear regression

We have a set of m points in the plane

$$p^{(i)} = (x^{(i)}, y^{(i)}) \in \mathbb{R}^2 \qquad i = 1, 2, \dots, m,$$

and we want to approximate them with a line[3]

$$d = \{(x, y) \in \mathbb{R}^2 \mid y = \theta_1 x + \theta_0\} \subset \mathbb{R}^2.$$



[3] We assume that the data is linearly related

One-dimensional linear regression

In the exact case, we would have

$$y^{(i)} = \theta_1 x^{(i)} + \theta_0 \text{ for all } i.$$

In matrix form, this is equivalent to solving a linear system (with two variables: θ_1 and θ_0)

$$\begin{pmatrix} x^{(1)} & 1 \\ x^{(2)} & 1 \\ \vdots & \vdots \\ x^{(m)} & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_0 \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}.$$

Few words on linear system of equations

- Many physical systems may be represented as a linear system of equations:

  $$A\theta = b$$

  where the constraint matrix A and vector b are known, and the vector θ is unknown.
- If A is a square, invertible matrix (i.e., A has nonzero determinant), then there exists a unique solution θ for every b.
- However, when A is either singular or rectangular, there may be one, none, or infinitely many solutions, depending on the specific b and the column and row spaces of A.

Few words on linear system of equations

The system is *underdetermined* and *full row* rank [4]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \ll n$, i.e., A is a short-fat matrix, so that there are fewer equations than unknowns.

[4] recall that in general $\text{rank}(A) \leqslant \min(m, n)$, here we suppose that $\text{rank}(A) = \min(m, n) = m$

Few words on linear system of equations

The system is *underdetermined* and *full row* rank [4]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \ll n$, i.e., A is a short-fat matrix, so that there are fewer equations than unknowns.
2. This type of system is guaranteed to **not** have full column rank, since it has many more columns than are required for a linearly independent basis.

[4] recall that in general $\text{rank}(A) \leqslant \min(m, n)$, here we suppose that $\text{rank}(A) = \min(m, n) = m$

Few words on linear system of equations

The system is *underdetermined* and *full row* rank [4]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \ll n$, i.e., A is a short-fat matrix, so that there are fewer equations than unknowns.
2. This type of system is guaranteed to **not** have full column rank, since it has many more columns than are required for a linearly independent basis.
3. Generically, there are infinitely many solutions θ for every b.
   Indeed, recall slide 28 of Lecture 2: for a solution θ of $A\theta = b$, then $z = \theta + u$ solves also the system with $u \in N(A)$.
   And, here, it is guaranteed that $\dim(N(A)) \neq 0$! Why ?

[4] recall that in general $\text{rank}(A) \leqslant \min(m, n)$, here we suppose that $\text{rank}(A) = \min(m, n) = m$

Few words on linear system of equations

The system is *underdetermined* and *full row* rank [4]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \ll n$, i.e., A is a short-fat matrix, so that there are fewer equations than unknowns.
2. This type of system is guaranteed to **not** have full column rank, since it has many more columns than are required for a linearly independent basis.
3. Generically, there are infinitely many solutions θ for every b.
   Indeed, recall slide 28 of Lecture 2: for a solution θ of $A\theta = b$, then $z = \theta + u$ solves also the system with $u \in N(A)$.
   And, here, it is guaranteed that $\dim(N(A)) \neq 0$! Why ?
4. The system is called *underdetermined* because there are not enough values in b to **uniquely** determine the higher-dimensional θ.

[4] recall that in general $\text{rank}(A) \leqslant \min(m, n)$, here we suppose that $\text{rank}(A) = \min(m, n) = m$

Few words on linear system of equations

The system is *overdetermined* and *full column* rank [5]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \gg n$, i.e., A is a a tall- skinny matrix, so that there are more equations than unknowns.

[5] that is we suppose that $\text{rank}(A) = \min(m, n) = n$

[6] we call that the *rank-deficient* case

Few words on linear system of equations

The system is *overdetermined* and *full column* rank [5]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \gg n$, i.e., A is a a tall- skinny matrix, so that there are more equations than unknowns.
2. There is 1 or 0 solutions to every b, i.e., there are vectors b that have no solution θ.

[5] that is we suppose that $\text{rank}(A) = \min(m, n) = n$

[6] we call that the *rank-deficient* case

Few words on linear system of equations

The system is *overdetermined* and *full column* rank [5]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \gg n$, i.e., A is a a tall- skinny matrix, so that there are more equations than unknowns.
2. There is 1 or 0 solutions to every b, i.e., there are vectors b that have no solution θ.
3. In fact, there will only be a solution θ if b is in the **range** of A, i.e., $b \in R(A)$.

[5] that is we suppose that $\text{rank}(A) = \min(m, n) = n$

[6] we call that the *rank-deficient* case

Few words on linear system of equations

The system is *overdetermined* and *full column* rank [5]

1. $A \in \mathbb{R}^{m \times n}$ is such that $m \gg n$, i.e., A is a a tall- skinny matrix, so that there are more equations than unknowns.
2. There is 1 or 0 solutions to every b, i.e., there are vectors b that have no solution θ.
3. In fact, there will only be a solution θ if b is in the **range** of A, i.e., $b \in R(A)$.
4. Important: if $b \in R(A)$, and if A is **not** full-column rank [6], that is rank$(A)< n$, then $\dim(N(A)) \neq 0$, then there are infinitely many solutions θ.

[5] that is we suppose that $\text{rank}(A) = \min(m, n) = n$

[6] we call that the *rank-deficient* case

Few words on linear system of equations

- There is an extensive literature on random matrix theory, where the previous stereotypes are almost certainly true, meaning that they are true with high probability.
- For example, a system $A\theta = b$ is extremely unlikely to have a solution for a random matrix $A \in \mathbb{R}^{m \times n}$ and random vector $b \in \mathbb{R}^m$ with $m \gg n$,
- since there is little chance that b is in the column space of A.

The *overdetermined* case

In the *overdetermined* case when no solution exists,

- we would often like to find the solution θ that minimizes some error measured by the function

$$\|A\theta - b\|$$

- for some vector norm $\|.\|$ in $\mathbb{R}^m$.

The *overdetermined* case

In the *overdetermined* case when no solution exists,

- we would often like to find the solution θ that minimizes some error measured by the function
$$\|A\theta - b\|$$
- for some vector norm $\|.\|$ in $\mathbb{R}^m$.
- Let us try to derive a lower-bound for this error :).

The *overdetermined* case

Property

Let $A \in \mathbb{R}^{m \times n}$ with $m \gg n$, and the linear system $A\theta = b$. Pose $y = A\theta \in R(A)$, we have

$$\begin{aligned} \|\text{proj}_{R(A)} b - b\| &= \min_{y \in R(A)} \|y - b\| \\ &\leqslant \|y - b\|, \quad \forall y \in R(A) \end{aligned}$$

with $\text{proj}_{R(A)} b$ the orthogonal projection of y onto the range of A. Moreover the optimal $x^\star$ satisfies $A\theta^\star = \text{proj}_{R(A)} b$, and the minimal error $\|\text{proj}_{R(A)} b - b\| = 0$ as soon as $b \in R(A)$.

Proof: Let $y = A\theta \in R(A)$, then

$$\begin{aligned} \|y - b\|^2 &= \|y - \text{proj}_{R(A)} b + \text{proj}_{R(A)} b - b\|^2 \\ &\underset{\text{Pyth. Theo.}}{=} \|y - \text{proj}_{R(A)} b\|^2 + \|\text{proj}_{R(A)} b - b\|^2 \\ &\geqslant \|\text{proj}_{R(A)} b - b\|^2 \end{aligned}$$

The lower-bound is reached as soon as $y = \text{proj}_{R(A)} b = A\theta$, and equal to zero if $b \in R(A)$.

The *overdetermined* case

- Although this result is elegant, it does not tell us yet how to compute $\text{proj}_{R(A)} b$.
- This orthogonal projection depends on the choice of the inner product.
- Moreover:
  1. the Pythagorean Theorem (used in the proof) holds for any vector spaces with an inner product $\langle ., . \rangle$ and the induced norm $\|.\| = \sqrt{\langle ., . \rangle}$.
  2. But: in the case our vector space is endowed with a norm induced by none inner product, such as the ℓ_1 and ℓ_∞ norms, these results dot not hold.
- Roughly speaking: depending on the norm used to measure $\|A\theta - b\|$, different methods can be used or not to solve $\min_\theta \|A\theta - b\|$.
- Let us divide them in two categories:
  1. The *direct* methods - relying on the tools of Linear Algebra: orthognal projections and SVD :) !
  2. The *iterative* methods - relying on the tools of (numerical) optimization.

  It is interesting to note that for some choice norms, methods from both categories can be considered.

Outline

1. Supervised learning

2. Linear Regression with one variable
   - Main Formulation
   - Direct Methods
   - Iterative Methods

3. Linear Regression with Multiple Variable

4. Probabilistic interpretation

5. Summary

The tools of Linear Algebra

In this section, we focus on

- the *overdetermined* case, that is $m \gg n$,
- We consider the squared ℓ_2-norm of $A\theta - b$, that is $\|A\theta - b\|_2^2$. [7]
- We will solve that problem by using the *Moore-Penrose pseudo-inverse* of matrix A, and
- show how the SVD of A can be useful for solving *general* least squares problems.

[7] it's *equivalent* to consider $\|A\theta - b\|_2$, in the sense that it doesn't change the optimal solutions.

The least-squares (linear) regression

- We are interesting in solving:

$$\min_{\theta \in \mathbb{R}^2} \quad \|A\theta - b\|_2^2$$

- It is equivalent to solve:

$$\min_{\theta_1, \theta_0} \quad \frac{1}{m} \sum_{i=1}^{m} (y^{(i)} - (\theta_1 x^{(i)} + \theta_0))^2$$

- This is called a *least-squares* (linear) regression.

The least-squares (linear) regression

- Starting with solving:

$$\min_{\theta \in \mathbb{R}^2} \quad \|A\theta - b\|_2^2 \tag{1}$$

- Let us pose $f(\theta) := \|A\theta - b\|_2^2$ to ease the notations.
- Our goal is then to minimize a differentiable function $f(.) : \mathbb{R}^2 \to \mathbb{R}$ over its entire domain, that is, there are no *constraints*.

[8] A is then *full-column* rank

[9] this is a standard result in convex optimization, more details in the Optimization course.

The least-squares (linear) regression

- Starting with solving:

$$\min_{\theta \in \mathbb{R}^2} \quad \|A\theta - b\|_2^2 \tag{1}$$

- Let us pose $f(\theta) := \|A\theta - b\|_2^2$ to ease the notations.
- Our goal is then to minimize a differentiable function $f(.) : \mathbb{R}^2 \to \mathbb{R}$ over its entire domain, that is, there are no *constraints*.
- **Key observation**: the function $f(.)$ is *convex*, and **strongly convex** if $\text{rank}(A) = n$ [8]. Hence, every local minimum is a global minimum, and potentially unique ! [9]

[8] A is then *full-column* rank

[9] this is a standard result in convex optimization, more details in the Optimization course.

The least-squares (linear) regression

- Starting with solving:

$$\min_{\theta \in \mathbb{R}^2} \quad \|A\theta - b\|_2^2 \tag{1}$$

- Let us pose $f(\theta) := \|A\theta - b\|_2^2$ to ease the notations.
- Our goal is then to minimize a differentiable function $f(.) : \mathbb{R}^2 \to \mathbb{R}$ over its entire domain, that is, there are no *constraints*.
- **Key observation**: the function $f(.)$ is *convex*, and **strongly convex** if $\text{rank}(A) = n$ [8]. Hence, every local minimum is a global minimum, and potentially unique ! [9]
- How to find a local minimum ?

[8] A is then *full-column* rank

[9] this is a standard result in convex optimization, more details in the Optimization course.

The least-squares (linear) regression

- Starting with solving:

$$\min_{\theta \in \mathbb{R}^2} \quad \|A\theta - b\|_2^2 \tag{1}$$

- Let us pose $f(\theta) := \|A\theta - b\|_2^2$ to ease the notations.
- Our goal is then to minimize a differentiable function $f(.) : \mathbb{R}^2 \to \mathbb{R}$ over its entire domain, that is, there are no *constraints*.
- **Key observation**: the function $f(.)$ is *convex*, and **strongly convex** if $\text{rank}(A) = n$ [8]. Hence, every local minimum is a global minimum, and potentially unique ! [9]
- How to find a local minimum ?
- $\theta^\star$ is said to be a local minimum of Problem (1) if it satisfies the equation:

$$\nabla_\theta f(\theta^\star) = 0,$$

  where $\nabla_\theta f(\theta^\star)$ denotes the *gradient* of f evaluated at $\theta^\star$.

[8] A is then *full-column* rank

[9] this is a standard result in convex optimization, more details in the Optimization course.

The least-squares (linear) regression: computing the gradient

- We know that the gradient is the vector of all the partial derivatives. Hence, we can compute $\frac{\partial f}{\partial \theta_i}(\theta)$ for all i and reconstruct the vector $\nabla f(\theta) := (\frac{\partial f}{\partial \theta_1}(\theta), ..., \frac{\partial f}{\partial \theta_n}(\theta))^T$.
- Consider $f(\theta) = \|A\theta - b\|_2^2$ with $A \in \mathbb{R}^{m \times n}$, we can write:

$$f(\theta) = \sum_{i=1}^{m} (\sum_{j=1}^{n} A_{i,j}\theta_j - b_i)^2,$$

- and so:

$$\frac{\partial f}{\partial \theta_k}(\theta) = 2\sum_{i=1}^{m} A_{i,k}(\sum_{j=1}^{n} A_{i,j}\theta_j - b_i).$$

- We recognise the components of the vector: $\nabla f(\theta) = 2A^T(A\theta - b)$.

The least-squares (linear) regression: computing the gradient

Using the definition

- We compute $f(\theta + h)$ and try to isolate $f(\theta)$, a linear term in h and a negligible term.
- Consider $f(\theta) = \|A\theta - b\|_2^2$.

The least-squares (linear) regression: computing the gradient

Using the definition

- We compute $f(\theta + h)$ and try to isolate $f(\theta)$, a linear term in h and a negligible term.
- Consider $f(\theta) = \|A\theta - b\|_2^2$.
- We write:

$$\begin{aligned} f(\theta + h) &= \|A(\theta + h) - b\|_2^2 = \|A\theta - b\|_2^2 + 2\langle A\theta - b, Ah\rangle + \|Ah\|_2^2 \\ &= f(\theta) + 2\langle A^T(A\theta - b), h\rangle + o(h) \end{aligned}$$

The least-squares (linear) regression: solution

- Hence, finding θ such that $\nabla f(\theta) = 0$ is equivalent to solve:

$$\begin{aligned} \nabla f(\theta) &= 2A^T A\theta - 2A^T b = 0 \\ \Leftrightarrow A^T A\theta &= A^T b \end{aligned} \tag{2}$$

  This is called the *normal equations* (funny, another linear system to solve) !

- Assuming A is full-column rank, then Equation (2) becomes

$$\begin{aligned} A^T A\theta &= A^T b \\ \Leftrightarrow \quad \theta &= (A^T A)^{-1} A^T b \\ \Leftrightarrow \quad \theta &= A^{\dagger} b \end{aligned} \tag{3}$$

  where $A^{\dagger}$ is the *Moore-Penrose pseudo-inverse.*

- Demo : ▸ colab file

Puzzle: what is the error we make with such as solution ?

General least squares problems using SVD

- When $\operatorname{rank}(A) < \min(m, n)$, we call the problem *rank-deficient*.
- As a gift of nature, SVD enables us to solve general least squares problems, whether $\operatorname{rank}(A) = \min(m, n)$ or $\operatorname{rank}(A) < \min(m, n)$.

General least squares problems using SVD

- When $\text{rank}(A) < \min(m, n)$, we call the problem *rank-deficient*.
- As a gift of nature, SVD enables us to solve general least squares problems, whether $\text{rank}(A) = \min(m, n)$ or $\text{rank}(A) < \min(m, n)$.
- Let $A = U\Sigma V^T$ be an SVD of A. Note that

$$\|A\theta - b\|_2 = \|U\Sigma V^T\theta - b\|_2 = \|\Sigma V^T\theta - U^T b\|_2$$

why ?

General least squares problems using SVD

- When $\text{rank}(A) < \min(m, n)$, we call the problem *rank-deficient*.
- As a gift of nature, SVD enables us to solve general least squares problems, whether $\text{rank}(A) = \min(m, n)$ or $\text{rank}(A) < \min(m, n)$.
- Let $A = U\Sigma V^T$ be an SVD of A. Note that

$$\|A\theta - b\|_2 = \|U\Sigma V^T\theta - b\|_2 = \|\Sigma V^T\theta - U^T b\|_2$$

  why ?
- Let us look for

$$y = V^T\theta \in \mathbb{R}^n$$

General least squares problems using SVD

- When $\text{rank}(A) < \min(m, n)$, we call the problem *rank-deficient.*
- As a gift of nature, SVD enables us to solve general least squares problems, whether $\text{rank}(A) = \min(m, n)$ or $\text{rank}(A) < \min(m, n)$.
- Let $A = U\Sigma V^T$ be an SVD of A. Note that

$$\|A\theta - b\|_2 = \|U\Sigma V^T\theta - b\|_2 = \|\Sigma V^T\theta - U^T b\|_2$$

  why ?
- Let us look for

$$y = V^T\theta \in \mathbb{R}^n$$

  then. Writing

$$c = U^T b \in \mathbb{R}^m$$

General least squares problems using SVD

- When $\operatorname{rank}(A) < \min(m, n)$, we call the problem *rank-deficient*.
- As a gift of nature, SVD enables us to solve general least squares problems, whether $\operatorname{rank}(A) = \min(m, n)$ or $\operatorname{rank}(A) < \min(m, n)$.
- Let $A = U\Sigma V^T$ be an SVD of A. Note that

$$\|A\theta - b\|_2 = \|U\Sigma V^T\theta - b\|_2 = \|\Sigma V^T\theta - U^T b\|_2$$

  why ?
- Let us look for

$$y = V^T\theta \in \mathbb{R}^n$$

  then. Writing

$$c = U^T b \in \mathbb{R}^m$$

  we have $\|A\theta - b\|_2 = \|\Sigma y - c\|_2$!

General least squares problems using SVD

If we wan to minimize $\|\Sigma y - c\|_2$, then for those i with $\sigma_i > 0$, we should set $y_i = c_i/\sigma_i$. For all other i, the value of y_i has no impact on the of $\|\Sigma y - c\|_2 = \|A\theta - b\|_2$. In summary

Theorem 1

Let $A \in \mathbb{R}^{m\times n}$, and suppose that $A = U\Sigma V^T$ is a singular value decomposition. Let $b \in \mathbb{R}^m$. The following construction yields all $\theta \in \mathbb{R}^n$ which minimize $\|A\theta - b\|$:

1. Define $c = U^T b \in \mathbb{R}^m$.
2. Define $y \in \mathbb{R}^n$ by

$$y_i = \begin{cases} c_i/\sigma_i, & \text{if } 1 \leqslant i \leqslant \min(m,n) \text{ and } \sigma_i > 0, \\ \text{whatever you like} & \text{otherwise} \end{cases}$$

3. Let $\theta = Vy$.

On the "whatever you like"

- Since $\theta = Vy$ and V is orthogonal, we have:

$$\|\theta\|_2 = \|y\|_2.$$

[10] we will discuss these aspects in more details another time

On the ”whatever you like”

- Since $\theta = Vy$ and V is orthogonal, we have:

$$\|\theta\|_2 = \|y\|_2.$$

- If we wanted to find, among all least squares solutions, the one with smallest $\|\theta\|_2$ (there are applications in which this is desirable [10]), we'd therefore have to choose $y_i = 0$ for those i that we are free to choose.

[10] we will discuss these aspects in more details another time

On the "whatever you like"

- Since $\theta = Vy$ and V is orthogonal, we have:

$$\|\theta\|_2 = \|y\|_2.$$

- If we wanted to find, among all least squares solutions, the one with smallest $\|\theta\|_2$ (there are applications in which this is desirable [10]), we'd therefore have to choose $y_i = 0$ for those i that we are free to choose.

Exercise: Let $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $b = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

1. Find two orthonormal vectors $v_1, v_2 \in \mathbb{R}^2$ such that Av_1 and Av_2 are orthogonal.
2. Find a SVD of A. (Do it by hand, using previous step, for better understanding).
3. Using the SVD found, find all least squares solutions of $A\theta \approx b$, then the one with minimal Euclidean norm.

[10] we will discuss these aspects in more details another time

Computation of $A^\dagger$ using SVD

The two previous slides (Theorem and remarks) can also be summarized as follows:

The Moore-Penrose pseudo-inverse and SVD

Let $A \in \mathbb{R}^{m\times n}$, and suppose that $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal, and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix with non-negative diagonal entries, and

$$A = U\Sigma V^T.$$

Let $b \in \mathbb{R}^m$. Then among those $\theta \in \mathbb{R}^n$ which minimize $\|A\theta - b\|_2$, the one with minimal Euclidean norm $\|\theta\|_2$ is

$$\theta = V\Sigma^\dagger U^T b, \tag{4}$$

where $\Sigma^\dagger$ is the $n \times m$ matrix with diagonal entries

$$\begin{cases} 1/\sigma_i & \text{if } \sigma_i > 0, \\ 0 & \text{if } \sigma_i = 0. \end{cases}$$

Puzzle continued what is the error here ?

Computation of $A^\dagger$ using SVD

- Equation (4) has an interesting consequence: The minimum-norm least square solution of $A\theta \approx b$ depends on b *linearly* – it is a matrix times b. That doesn't seem obvious a priori.

Definition

The matrix

$$A^\dagger = V\Sigma^\dagger U^T$$

is called the *Moore-Penrose pseudo-inverse* of A.

- With this notation, the minimum-norm least squares solution of $A\theta \approx b$ is **always**:

$$\theta = A^\dagger b.$$

Computation of $A^\dagger$ using SVD

- Equation (4) has an interesting consequence: The minimum-norm least square solution of $A\theta \approx b$ depends on b *linearly* – it is a matrix times b. That doesn't seem obvious a priori.

Definition

The matrix

$$A^\dagger = V\Sigma^\dagger U^T$$

is called the *Moore-Penrose pseudo-inverse* of A.

- With this notation, the minimum-norm least squares solution of $A\theta \approx b$ is **always**:

$$\theta = A^\dagger b.$$

- $A^\dagger$ is defined for *any* $A \in \mathbb{R}^{m\times n}$.
- However, there can be different SVD for the same matrix.
- **Question**: could there be different Moore-Penrose pseudo-inverses ?

Outline

1 Supervised learning

2 Linear Regression with one variable

- Main Formulation
- Direct Methods
- Iterative Methods

3 Linear Regression with Multiple Variable

4 Probabilistic interpretation

5 Summary

The ℓ_1-norm and Linear Programming

For solving the regression problem in the non-exact (more general) case, that includes the cases where *noise* is present within the data [11], and the *overdetermined* case,

- we want to minimize the norm of the errors vector $A\theta - b$.
- One common evaluation metric for regression problems is the MAE (Mean Absolute Error), which minimizes the ℓ_1-norm of $A\theta - b$.
- Note that $\|A\theta - b\|_1 = \sum_{i=1}^{m} |y^{(i)} - (\theta_1 x^{(i)} + \theta_0)|$.

[11] think about the case the samples have been acquired by sensors, noise is inevitable

The ℓ_1-norm and Linear Programming

We want to solve:

$$\min_{\theta_1,\theta_0} \quad \sum_{i=1}^{m} |y^{(i)} - (\theta_1 x^{(i)} + \theta_0)|$$

The ℓ_1-norm and Linear Programming

We want to solve:

$$\min_{\theta_1,\theta_0} \quad \sum_{i=1}^{m} |y^{(i)} - (\theta_1 x^{(i)} + \theta_0)|$$

which is equivalent to

$$\begin{aligned} \min_{\theta_1,\theta_0\in\mathbb{R},t\in\mathbb{R}^n} \quad & \sum_{i=1}^{m} t_i \\ \text{such that} \quad & t_i \geqslant \theta_1 x^{(i)} + \theta_0 - y^{(i)},\ 1 \leqslant i \leqslant m, \\ & t_i \geqslant -\theta_1 x^{(i)} - \theta_0 + y^{(i)},\ 1 \leqslant i \leqslant m. \end{aligned}$$

Remarks:

- This is a *linear* problem that can be solved with the *simplex algorithm*.
- See the Optimization course for the details.

The ℓ_∞-norm and Linear Programming

We want to solve:

$$\min_{\theta_1,\theta_0} \quad \max_{i\in\{1,..,m\}} |y^{(i)} - (\theta_1 x^{(i)} + \theta_0)|$$

The ℓ_∞-norm and Linear Programming

We want to solve:

$$\min_{\theta_1,\theta_0} \quad \max_{i\in\{1,..,m\}} |y^{(i)} - (\theta_1 x^{(i)} + \theta_0)|$$

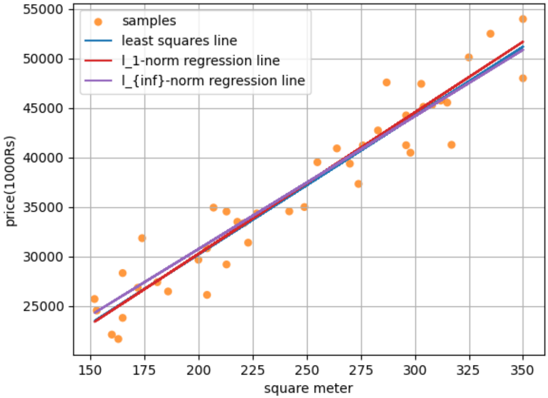
which is equivalent to

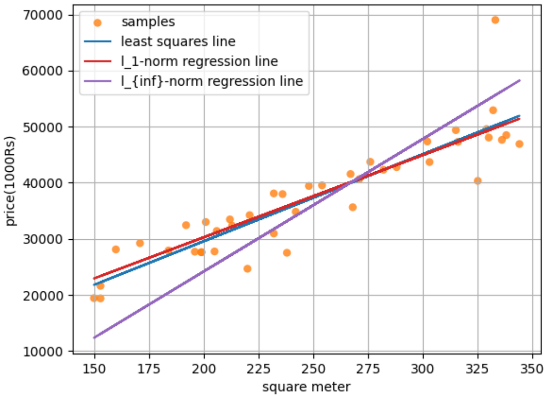
$$\begin{aligned} \min_{\theta_1,\theta_0\in\mathbb{R},t\in\mathbb{R}^n} \quad & t \\ \text{such that} \quad & t \geqslant \theta_1 x^{(i)} + \theta_0 - y^{(i)},\ 1 \leqslant i \leqslant m, \\ & t \geqslant -\theta_1 x^{(i)} - \theta_0 + y^{(i)},\ 1 \leqslant i \leqslant m. \end{aligned}$$

Remark:

- This is *again* a linear problem that can be solved with the *simplex algorithm*.

Il buono, il brutto, il cattivo





- **Left**: the data has not outliers and the three linear models, although different, produce approximately the same model.
- **Right**: With outliers, the predictions are significantly different.

Linear Regression with Multiple Variable

Some context and motivation

To make our housing example more interesting, lets consider a slightly richer dataset in which we also know the number of bedrooms in each house:

Living area (m^2)	#bedrooms	Price (1000 Rs)
238	3	33634
231	3	32547
340	4	51871
320	4	45639
211	2	26660
$\vdots$	$\vdots$	$\vdots$

Some context and motivation

- Here, the x's are two-dimensional vectors in $\mathbb{R}^2$.
- For instance: $x_1^{(i)}$ is the living area of the i-th house in the training set, and $x_2^{(i)}$ is its number of bedrooms.
- **In general**: when designing a learning problem, it will be up to you to decide what features to choose.
- So, if you are out in Kazan gathering housing data, you might also decide to include other features such whether each house has a fireplace, the number of bathrooms, and so on.
- As for the one-dimensional case, to perform supervised learning, we must decide how we're going to represent functions/hypotheses h in a computer.
- We decide to approximate y as a linear function of x:

$$h_\theta(x) := \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Some context and motivation

- the θ's are the **parameters** (also called **weights**) parameterizing the space of linear functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- When there is no risk of confusion, we will drop the θ subscript in $h_\theta(x)$, and write it more simply as $h(x)$.
- To simplify our notation, we also introduce the convention of letting $x_0 = 1$ (this is the intercept term), so that

$$h(x) := \sum_{i=0}^{n} \theta_i x_i = \langle \theta, x \rangle$$

where on the right-hand side above we are viewing θ and x both as vectors, and here n is the number of input variables (not counting x_0).

The most interesting slide in the universe

- **Q**: How can we solve the Linear Regression with Multiple Variable ?

The most interesting slide in the universe

- **Q**: How can we solve the Linear Regression with Multiple Variable ?
- **A**: the same way as for one variable... All the formulas are the same, just consider n general, and that is it !

Example : Cement heat generation data

- we begin with a dataset that describes the heat generation for various cement mixtures comprised of four basic ingredients.
- In this problem, we are considering the least squares (linear) regression where $A \in \mathbb{R}^{13\times4}$ since there are four ingredients and heat measurements for 13 unique mixtures. [12]
- The goal is to determine the weighting $\theta \in \mathbb{R}^4$ that relates the proportions of the four ingredients to the heat generation.

[12] i.e., we do not consider θ_0.

Example : Cement heat generation data

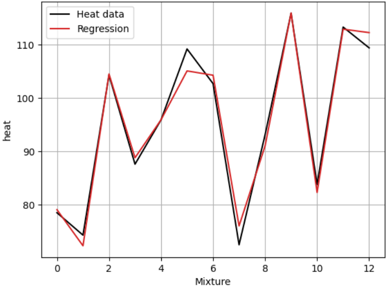


Figure: Heat data for cement mixtures containing four basic ingredients.

	Ingredient	θ's
1	Tricalcium aluminate	2.193
2	Tricalcium silicate	1.153
3	Tetracalcium alumiferrite	0.758
4	Beta-dicalcium silicate	0.486

Probabilistic interpretation

Some context and motivation

- When faced with a regression problem, why might linear regression, and specifically why might the least-squares be a reasonable choice?
- In this section, we will give a set of probabilistic assumptions, under which least-squares regression is derived as a very natural algorithm.

Probabilistic generative model

- Let us assume that the target variables and the inputs are related via the equation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

where $\epsilon^{(i)}$ is an error term that captures either unmodeled effects such as

1. if there are some features very pertinent to predicting housing price, but that we'd left out of the regression,
2. or random noise.

Probabilistic generative model

- Let us assume that the target variables and the inputs are related via the equation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

  where $\epsilon^{(i)}$ is an error term that captures either unmodeled effects such as

  1. if there are some features very pertinent to predicting housing price, but that we'd left out of the regression,
  2. or random noise.

- Let us further assume that the $\epsilon^{(i)}$ are distributed IID (independently and identically distributed) according to a Gaussian distribution (also called a Normal distribution) with mean zero and some variance σ^2.

Distribution

- We can write this assumption as $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$.
- I.e., the *pdf* (probability density function) of $\epsilon^{(i)}$ is given by

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

[13] We can also write the distribution of $y^{(i)}$ as $y^{(i)}|x^{(i)}; \theta \sim \mathcal{N}(\theta^T x^{(i)}, \sigma^2)$

Distribution

- We can write this assumption as $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$.
- I.e., the *pdf* (probability density function) of $\epsilon^{(i)}$ is given by

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

- This implies that

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- The notation "$p(y^{(i)}|x^{(i)};\theta)$" indicates that this is the "distribution of $y^{(i)}$ given $x^{(i)}$ and parameterized by θ.
- Note that we should not condition on θ ($p(y^{(i)}|x^{(i)},\theta)$), since θ is not a random variable. [13]

[13] We can also write the distribution of $y^{(i)}$ as $y^{(i)}|x^{(i)};\theta \sim \mathcal{N}(\theta^T x^{(i)}, \sigma^2)$

The likelihood function

- Given X (the design matrix, which contains all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?

The likelihood function

- Given X (the design matrix, which contains all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?
- The probability of the data is given by $p(\vec{y}|X;\theta)$.
- This quantity is typically viewed a function of $\vec{y}$ (and perhaps X) for a fixed value of θ.
- When we wish to explicitly view this as a function of θ, we will instead call it the likelihood function:

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X;\theta).$$

The likelihood function

- Given X (the design matrix, which contains all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?
- The probability of the data is given by $p(\vec{y}|X;\theta)$.
- This quantity is typically viewed a function of $\vec{y}$ (and perhaps X) for a fixed value of θ.
- When we wish to explicitly view this as a function of θ, we will instead call it the likelihood function:

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X;\theta).$$

- Note that by the independence assumption on the $\epsilon^{(i)}$'s (and hence also the $y^{(i)}$'s given the $x^{(i)}$'s), this can also be written

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

Maximizing the likelihood

- Now, given this probabilistic model relating the $y^{(i)}$'s and the $x^{(i)}$'s, what is a reasonable way of choosing our best guess of the parameters θ ?

Maximizing the likelihood

- Now, given this probabilistic model relating the $y^{(i)}$'s and the $x^{(i)}$'s, what is a reasonable way of choosing our best guess of the parameters θ ?
- The principle of **maximum likelihood** says that we should choose θ so as to make the data as high probability as possible.
- I.e., we should choose θ to maximize $L(\theta)$.

Maximizing the likelihood

- Now, given this probabilistic model relating the $y^{(i)}$'s and the $x^{(i)}$'s, what is a reasonable way of choosing our best guess of the parameters θ ?
- The principle of **maximum likelihood** says that we should choose θ so as to make the data as high probability as possible.
- I.e., we should choose θ to maximize $L(\theta)$.
- **Ok fine but**: instead of maximizing $L(\theta)$, we can also maximize any strictly increasing function of $L(\theta)$.

Log-likelihood

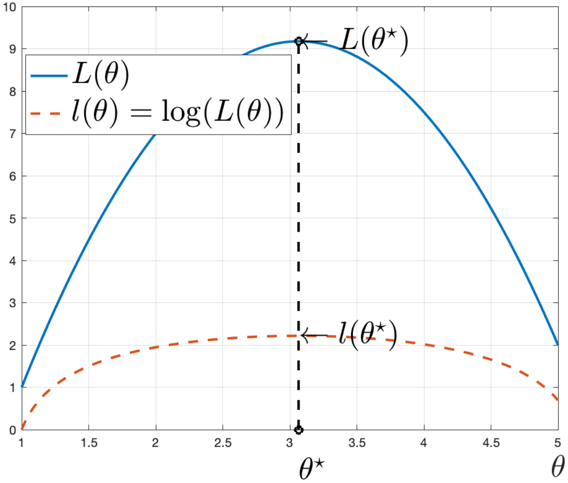


Figure: Maximize a strictly increasing function of $L(\theta)$ gives the same maximizer $\theta^\star$

Maximizing the log-likelihood

- In particular, the derivations will be a bit simpler if we instead maximize the log likelihood $l(\theta)$:

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2 \end{aligned} \tag{5}$$

- **Hence**: maximizing $l(\theta)$ gives the same answer as minimizing

$$\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2$$

which we recognize to be the objective function of a **least squares** (linear) regression problem !!

Equivalency

To summarize:

- Under the previous probabilistic assumptions on the data, least-squares regression corresponds to finding the maximum likelihood estimate of θ.
- This is thus one set of assumptions under which least-squares regression can be justified as a very natural method that's just doing maximum likelihood estimation. [14]
- Note also that, in our previous discussion, our final choice of θ did not depend on what was σ^2, and indeed we'd have arrived at the same result even if σ^2 were unknown.

[14] Note however that the probabilistic assumptions are by no means necessary for least-squares to be a perfectly good and rational procedure, and there may—and indeed there are—other natural assumptions that can also be used to justify it.

Going further

The choice of other standards to measure the errors $(y^{(i)} - \theta^T x^{(i)})$ *to be minimized can also be supported by probabilistic interpretations. In practice, we deal with noisy data, make some assumptions about the noise distribution (that of* $\epsilon^{(i)}$*) and choose the norm on the basis of the corresponding likelihood.*

Puzzles:

- Consider a Laplace distribution for $\epsilon^{(i)}$, what is the linear regression problem corresponding to the maximum likelihood estimate of θ ?
- Can you generalize to polynomial regression (*aka* polynomial fitting problem) ?

Summary

Summary

We have seen :

- A bit of **supervised learning**: given a *training set*, learn the function h so that $h(x)$ is a good predictor.
- **Purpose** of the Linear Regressions problems: predict continuous variables.
- The general **formulation** of a Linear Regression problem, and its link with solving *linear system of equations* $A\theta = b$.
- **Direct** and **Iterative** methods to find approximated solutions in the *overdetermined* case.
- How SVD can be useful for (general) least squares problems.
- Different norms for measuring $A\theta - b \Leftrightarrow$ different methods.
- The **Probabilistic Interpretation** of Linear Regression problems.

Preparation for the lab

- Review the lecture :).
- Practice the linear regression with the hald data set:
  1. ▸ Data set in text format
  2. ▸ colab file - Section 2
- Some useful data sets: ▸ UCI Machine Learning Repository
- A bit of help:
  1. A didactic video for highlighting the robustness to outliers of the ℓ_1-norm linear regression: ▸ Robust Regression with the L1 Norm
  2. Quick recap of the *Maximum Likelihood Estimation* ▸ Link

Goodbye, So Soon

THANKS FOR THE ATTENTION

- v.leplat@innopolis.ru
- sites.google.com/view/valentinleplat/