Supervised Learning

Johanni Brea

Introduction to Machine Learning

EPFL

Table of Contents

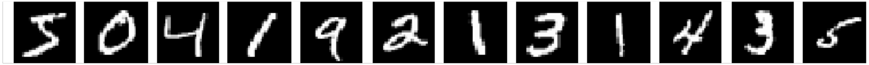
1. Our Datasets for Supervised Learning

2. Data Generating Process

3. How Does Supervised Learning Work?

4. Application of Linear Regression

Handwritten Digit Classification (MNIST)



our goal: assign the correct digit class to images

5 0 4 1 9 2 1 3 1 4 3 5

input X: 28x28 = 784 pixels with values between 0 (black) and 1 (white)
output Y: digit class 0, 1, . . ., 9

Spam Detection with the Enron Dataset

spam

Subject: follow up
here ' s a question i ' ve been wanting to ask you , are you feeling down but too embarrassed to go to the doc to get your m / ed ' s ?
here ' s the answer , forget about your local p harm . acy and the long waits , visits and em- barassments . . do it all in the privacy of your own home , right now . http : / / chopin . manil- amana . com / p / test / duet it ' s simply the best and most private way to obtain the stuff you need without all the red tape .

ham

Subject: darrin presto
amy :
please follow up as soon as possible with dar- rin presto regarding a real time interview . i for- warded his resume to you last week . he can be reached at 509 - 946 - 7879
thanks
greg

Our goal: classify new emails as spam or “ham“ (not spam).

input X: sequences of characters (emails), output Y: label spam or ham

Wind Speed Prediction

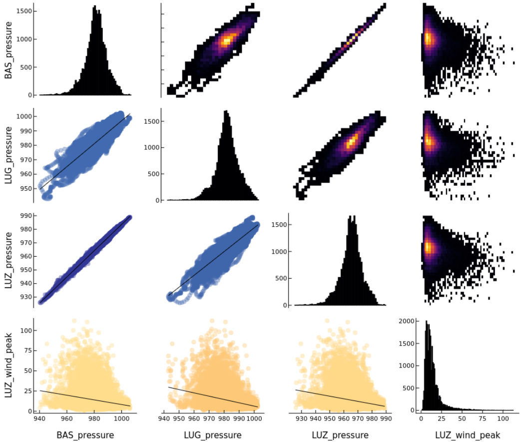
- SwissMeteo data: hourly measurements for 5 years from different stations (Bern, Basel, Luzern, Lugano, etc.).
- Our goal: given measurements at different stations, predict wind speed in Luzern 5 hours later.

Wind Speed Prediction

time	BAS_pressure	LUG_pressure	...	LUZ_pressure	LUZ_wind_peak_in5h
$x_{11} = 2015010100$	$x_{12} = 997.1$	$x_{13} = 998.6$	$\ldots$	$x_{1p} = 980.0$	$y_1 = 15.5$
$x_{21} = 2015010101$	$x_{22} = 997.3$	$x_{23} = 998.8$	$\ldots$	$x_{2p} = 979.9$	$y_2 = 13.0$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$x_{n1} = 2017123123$	$x_{n2} = 972.7$	$x_{n3} = 981.5$	$\ldots$	$x_{np} = 957.5$	$y_n = 11.9$

- p **input** variables $X = (X_1, X_2, \ldots, X_p)$
  e.g. X_1 time, X_2 BAS_pressure, X_3 LUG_pressure
  also called: **predictors, independent variables, features**
- **output** variable Y e.g. LUZ_wind_peak_in5h
  also called: **response, dependent variable**
- n **measurements** or **data points**

Always Look at Raw Data!

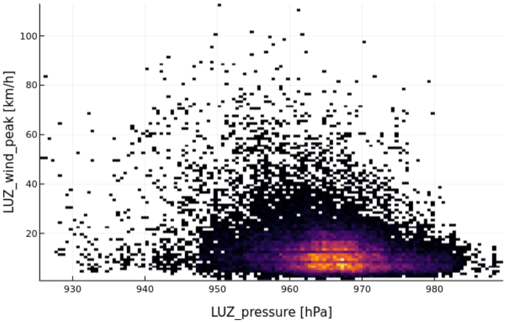


- **on diagonal**: 1D histogram
- **lower triangle**: scatter plot & trend line
- **upper triangle**: 2D histogram

Observations

1. LUZ_wind_peak_in5h has a long tail.
2. For low pressures there are outliers of strong wind.
3. Pressure in Basel and Luzern is highly correlated.
4. ...

Wind Speed Prediction



- The higher the pressure in Luzern, the less probable it is to have strong winds.
- No function `LUZ_wind_peak_in5h` = f(`LUZ_pressure`) can describe this data.

See also `https://bio322.epfl.ch/notebooks/supervised_learning.html`

Your Observations

What are the commonalities and the differences between these three datasets?

Table of Contents

1. Our Datasets for Supervised Learning

2. Data Generating Process

3. How Does Supervised Learning Work?

4. Application of Linear Regression

Probabilities & Expectations

- random variable/vector
- probability
- probability density function
- joint probability
- conditional probability
- expectation
- conditional expectation
- average

Data Generating Processes

It is useful to think of our datasets as samples from **data generating processes** for the input X and the conditional output Y|X.

$$\underbrace{P(X, Y)}_{\text{joint}} = \underbrace{P(Y|X)}_{\text{conditional}} \underbrace{P(X)}_{\text{input}}$$

- **MNIST** X: people write digits $\rightarrow$ people take standardized photos thereof.
  Y|X: different people label the same photo X.
- **Spam** X: people write emails.
  Y|X: different people classify the same email X as spam or not.
- **Weather** X: the weather acts on sensors in weather stations.
  Y|X: the weather evolves from X and is measured again 5 hours later.

Using samples from these data generating processes, supervised learning aims at learning something about the conditional processes, i.e how Y depends on X.

Table of Contents

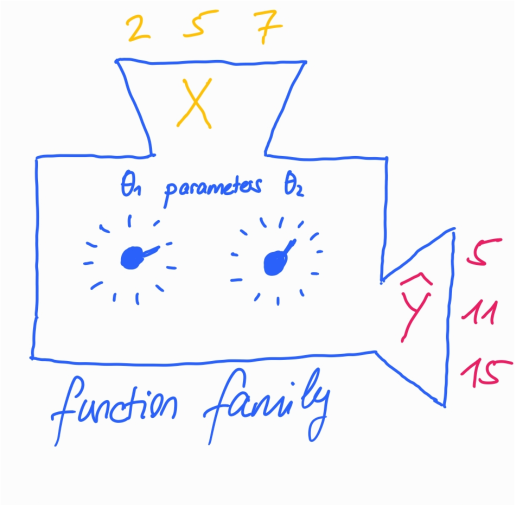
1. Our Datasets for Supervised Learning

2. Data Generating Process

3. How Does Supervised Learning Work?

4. Application of Linear Regression

How Does Supervised Learning Work?



Function Family

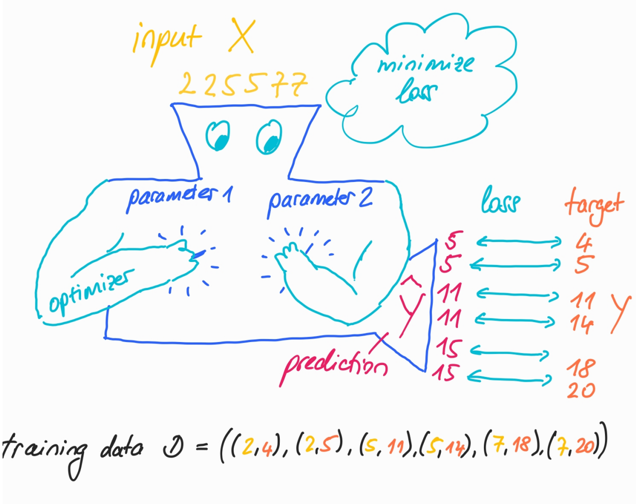
- We change the parameters.
- The machine computes $\hat{y}$ given parameters θ and x.

For example

$$\hat{y} = f_\theta(x) = \theta_0 + \theta_1 x$$

When we change the parameters θ_0 and θ_1, we change the way $\hat{y}$ depends on x.

How Does Supervised Learning Work?



Loss Minimizing Machine

- We specify
  1. the training data
  2. the function family (model)
  3. the loss function $L(y, \hat{y})$
  4. the optimizer
- The machine changes the parameters with the help of the optimizer until the loss is minimal.

For example: linear regression

Blackboard: Linear Regression as a Loss Minimizing Machine

Data Generating Process

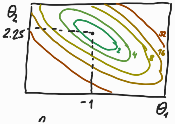
$$y = 2x - 1 + \varepsilon\ , \quad E[\varepsilon] = 0 \quad Var[\varepsilon] = \sigma^2$$

Training Data

$$\big((x_1 = 0,\ y_1 = -1),\ (x_2 = 2,\ y_2 = 4),\ (x_3 = 2,\ y_3 = 3)\big)$$

Function Family

$$\hat{y} = \theta_0 + \theta_1 \cdot x$$



Loss Function

$$L(\theta) = L(\theta_0, \theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2$$

$$= \frac{1}{3}\Big((-1-\theta_0)^2 + (4-\theta_0-2\theta_1)^2 + (3-\theta_0-2\theta_1)^2\Big)$$

Optimizer : Default

Solution : $\hat{\theta}_0 = -1\ ,\ \hat{\theta}_1 = 2.25\ ,\ L(\hat{\theta}) = \frac{2}{3}\cdot 0.5^2$

Test Loss at x_0 :

$$E\Big[\big(\underbrace{2x_0 - \cancel{1} + \varepsilon}_{y} \underbrace{+ \cancel{1} - 2.25x_0}_{-\hat{y}}\big)^2\Big] = (0.25x_0)^2 + \sigma^2$$

Test Data :

$$\big((x_1 = 1,\ y_1 = 0),\ (x_2 = 2,\ y_2 = 3),\ (x_3 = 3,\ y_3 = 5),\ (x_4 = 0,\ y_4 = -1)\big)$$

$\Rightarrow$ (Empirical) Test Loss $= \frac{1}{4}\left(0.25^2 + 0.5^2 + 0.75^2 + 0^2\right)$

Training Loss and Test Loss

- **Training Set** $\mathcal{D}$: Data used by the machine to tune the parameters.
- **Training Loss of Function** f: $\mathcal{L}(f, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))$
- **Test Loss of Function** f **at** x **for a Conditional Data Generating Process**:
  $\mathrm{E}_{Y|x}\left[L(Y, f(x))\right]$ = expected loss under the conditional generating process.
- **Test Loss of Function** f **for a Joint Data Generating Process**:
  $\mathrm{E}_{X,Y}\left[L(Y, f(X))\right]$ = expected loss under the joint generating process.
  We would like to minimize this!
  Usually we do not know $P(X, Y)$, so we want to approximate this with samples:
- **Test Set** $\mathcal{D}_{\text{test}}$: Data from the same generating process as the training set, not used for parameter fitting.
- **Test Loss of Function** f **for a Test Set** $\mathcal{D}_{\text{test}}$: $\mathcal{L}(f, \mathcal{D}_{\text{test}})$ = same computation as for the training loss but for a test set.
  This is an approximation of the test loss for the joint process.

Quiz

Correct or wrong?

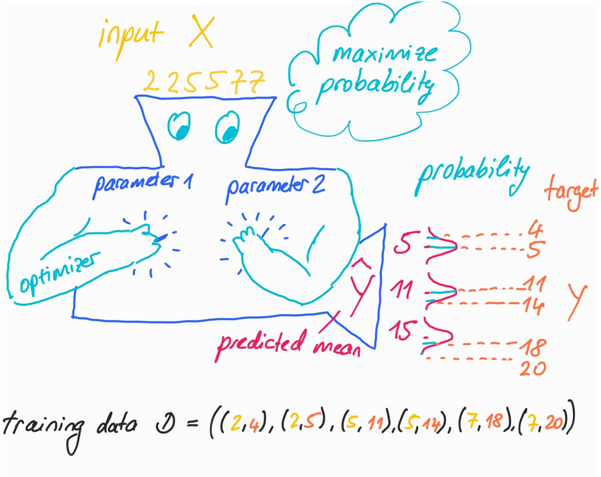
1. Assume we split the MNIST dataset into 60'000 images for training and 10'000 images for testing. The test loss of a fitted function f for this test set is an approximation of
   A) The test loss of f at x for the conditional data generating process
   B) The test loss of f for the joint data generating process
   C) neither nor.
2. For the weather dataset we do not know the data generating process. Therefore it is impossible to compute the test loss of a function f for the joint data generating process.
3. The training loss of a function f is typically
   A) larger than B) equal to C) smaller than
   the test loss of the function f for a test set.

Which Loss Functions Should We Use?

- Is the mean squared error always a good loss?
- What kind of loss would be good in a classification setting (e.g. MNIST)?
- How should we choose the loss when we know something about the noise distribution?

All these questions have a straight-forward answer, if we use a **family of probability distributions** (instead of a family of functions) and estimate the parameters with a **maximum likelihood approach** (instead of minimizing a hand-picked loss).

How Does Supervised Learning Work?



Likelihood Maximizing Machine

- We specify
  1. the training data
  2. the family of probability distributions (model)
  3. the optimizer
- The machine changes the parameters with the help of the optimizer until the likelihood of the parameters is maximal.

For example: linear regression

The Likelihood Function

For a family of conditional probability distributions $P(y|x,\theta)$ and training data $\mathcal{D} = ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ the **likelihood function** is defined as

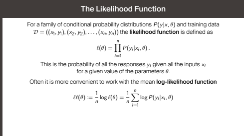
$$\ell(\theta) = \prod_{i=1}^{n} P(y_i|x_i,\theta)\,.$$

This is the probability of all the responses y_i given all the inputs x_i for a given value of the parameters θ.

Often it is more convenient to work with the mean **log-likelihood function**

$$\ell\ell(\theta) := \frac{1}{n}\log\ell(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log P(y_i|x_i,\theta)$$

Notes



The Likelihood Function

For a family of conditional probability distributions $P(y|x,\theta)$ and training data $\mathcal{D} = ((x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n))$ the **likelihood function** is defined as

$$\ell(\theta) = \prod_{i=1}^{n} P(y_i|x_i,\theta)\,.$$

This is the probability of all the responses y_i given all the inputs x_i for a given value of the parameters θ.

Often it is more convenient to work with the mean **log-likelihood function**

$$\ell\ell(\theta) := \frac{1}{n}\log\ell(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log P(y_i|x_i,\theta)$$

The log-likelihood is more convenient, because (math argument:) some proofs are easier with the log, as you can see in exercise 1 of this week, and (numerics argument:) products of small numbers (as in $\ell(\theta)$) become quickly smaller than the smallest representable 64-bit floating point number, whereas the sum of the log of small numbers remains a reasonable 64-bit floating point number.
Note that the likelihood function and the log-likelihood function have the maximum at the same argument value, i.e. $\arg\max_\theta \ell(\theta) = \arg\max_\theta \log(\ell(\theta))$, because the $\log$ is a monotonic function.

Linear Regression from the Maximum-Likelihood Perspective

- There are many possibilities how to parametrize the conditional distribution.
- If one chooses a linear function for the mean of a normal distribution,

$$P(y_i|x_i, \beta_0, \beta_1, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i - \beta_0 - \beta_1 x)^2}{2\sigma^2}}$$

the maximum-likelihood solution for β_0 and β_1 is the same as the least squared error solution of linear regression, i.e. $\hat{\beta}_1 = \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\langle x^2\rangle - \langle x\rangle^2}$, $\hat{\beta}_0 = \langle y\rangle - \hat{\beta}_1\langle x\rangle$.

- The maximum likelihood estimate of σ is $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x)^2$
- See example on the website and proof in exercise 1.

Table of Contents

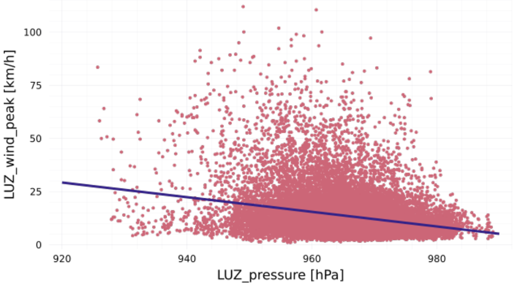
1. Our Datasets for Supervised Learning

2. Data Generating Process

3. How Does Supervised Learning Work?

4. Application of Linear Regression

Wind Speed Prediction



$$\hat{y} = \theta_0 + \theta_1 x$$

$$\theta_0 = 346 \text{ km/h}, \theta_1 = -0.344 \text{ (km/h)/hPa}$$

- **Training Set**: Hourly data 2015-2018
- **Training Loss (rmse)**: 10.0 km/h
- **Test Set**: Hourly data 2019-2020
- **Test Loss (rmse)**: 11.5 km/h

root-mean-squared error:

$$\text{rmse} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

Application of Linear Regression

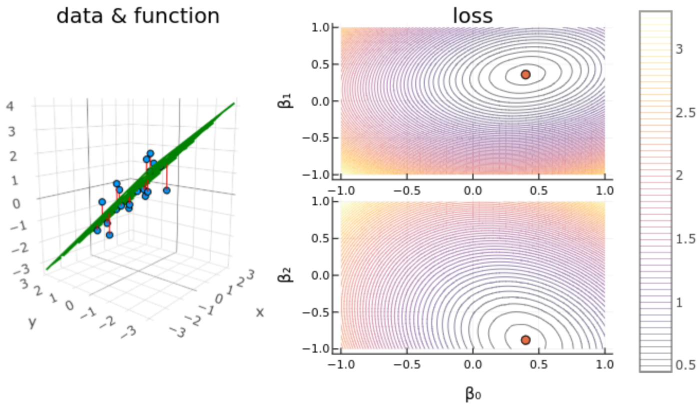
$$\hat{y} = f(x) = f(x_1, x_2, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

$$\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}}_{\boldsymbol{X}} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

Often the output correlates with multiple factors.

For example:
x_1: pressure in Luzern
x_2: temperature in Luzern
x_3: pressure in Basel
x_4: pressure in Lugano
etc.

Application of Linear Regression Example: $p = 2, n = 20$



Multiple Linear Regression finds the plane closest to the data.
Closeness is measured by the sum (or the mean) of the square of the red vertical distances between the plane and the data.

Multiple Linear Regression for Wind Speed Prediction

predictor name	fitted parameter
LUZ_pressure	-2.79 (km/h)/hPa
PUY_pressure	-2.39 (km/h)/hPa
BAS_precipitation	-0.66 (km/(h)/mm
⋮	⋮
LUZ_temperature	0.87 (km/h)/C
GVE_pressure	3.95 (km/h)/hPa

Interpretation

An increase of one hPa of LUZ_pressure correlates with a decrease of the expected wind speed by 2.79 km/h, if all other measurements remain the same.

Evaluation

- **Training Set**: Hourly data 2015-2018
- **Training Loss (rmse)**: 8.1 km/h
- **Test Set**: Hourly data 2019-2020
- **Test Loss (rmse)**: 8.9 km/h

Summary

We use a training set to find a conditional distribution that captures some regularities of the conditional data generation process. The goal is to find a conditional distribution that minimizes the test loss of the joint data generation process. With a test set we can assess how close we are at reaching this goal.

Supervised Learning as Loss Minimization	Supervised Learning as Likelihood Maximization
We provide 1. training data 2. function family 3. loss function 4. optimizer	We provide 1. training data 2. probability distribution family 3. optimizer The negative log-likelihood function of the parameters implicitly defines a loss function.
It is not (always) obvious what kind of loss function to take for classification problems or regression problems with a specific noise distributions	We will see that we take the binomial for binary classification problems and the categorical for other classification problems. Regression with other noise distribution is also possible.

Suggested Reading

The following chapters from “An Introduction to Statistical Learning” (second edition, `https://www.statlearning.com`) are complementary to the material presented in this lecture. It is not mandatory to read them, but maybe it helps to better understand the material of this lecture.

- 3.1 Simple Linear Regression
- 3.2 Multiple Linear Regression