Week 3a:

Linear models for regression and classification

G6061: Fundamentals of Machine Learning [23/24]

Dr. Johanna Senk

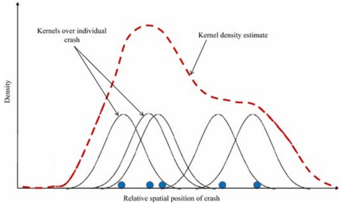
UNIVERSITY OF SUSSEX

Recap of previous lecture

- Application of **Bayes' theorem** to interpret results
  - Classifiers, odds ratio for tests

- **Probability density estimation**
  - **Non-parametric approach**:
    histograms, kernel density estimation
    (smoothed-out histogram, density estimate in a region around each datapoint)
  - **Parametric approach**:
    assuming a distribution (often Gaussian),
    finding the best fit parameters
    - usually mean, standard deviation
    (plus covariances if multi-dimensional)

$p(x)$



x

Recap on instances and instance spaces

- Consider data to consist of a set of **instances**
  - an **instance** represents an object of interest
- Set of all possible instances is the **instance space**:
  - e.g., set of all email messages written in English
  - or, set of human faces

Notation

- We will denote the **instance space** by $\mathcal{X}$
- An individual **instance** will be denoted by x
- $x \in \mathcal{X}$

UNIVERSITY OF SUSSEX

Labels and label spaces

- In supervised problems, each **instance** is associated with a known **label**. In unsupervised tasks the label is unknown.
- Set of all **labels** for a task is called the **label space**.

  Class labels $\mathcal{C}$ in **classification** tasks, e.g., $\mathcal{C} = \{\text{frogs}, \text{badgers}\}$
  Real numbers $\subseteq \mathcal{R}$ in **regression** tasks, e.g., temperature
  Cluster indices $[0 \ldots N_{\text{max}}]$ in **clustering** tasks

Notation

- We will denote an arbitrary **label space** by $\mathcal{Y}$
- Labelling function $f : \mathcal{X} \rightarrow \mathcal{Y}$ maps from **instances** to **labels**
- **Label** associated with a given **instance** x denoted by $y = f(x)$

Outline

- We'll introduce the regression task in some more detail and talk through some examples.
- We will define what a **linear model** is, and how it can be used for regression and classification.
- We'll analyse the assumptions of linear least squares regression models and understand its probabilistic interpretation.
- We'll relate regression back to probability density estimation.
- We'll see an algorithm for solving linear regression problems.

UNIVERSITY OF SUSSEX

Regression

What is it and when might we use it?

- The regression task predicts a continuous output value, i.e., the label space $\mathcal{Y} \subseteq \mathbb{R}$.
- This can be applied to a wide variety of problems.
  - Today we're going to consider applications where **prediction** of $\mathbf{y}$ is our goal.
  - In the next lecture we'll talk briefly about other uses of regression.
- There are many possible models that have been used for regression
  - Today we'll talk about **linear regression**.
- Let's think about some example tasks, and contrast them with classification.

Example: the hiring problem

- Imagine we are a company looking to hire machine learning experts.
- Maybe all we have to account for is how well they did in the FoML course $\mathcal{X} \subseteq [0, 1]$

- The classification problem is: do we hire?
  hiring approval = discrete output: $\mathcal{Y} = \{\text{yes}, \text{no}\}$
- A **regression problem** is: how much do we pay?
  annual salary (£ amount) = real-valued output: $\mathcal{Y} \subseteq \mathbb{R}^+$

How can we set up a model to predict this?

The training dataset

Input: $\mathbf{x} =$

`FoML Assessment mark`	0.61
`Attendance rate in lectures`	80%
`Attendance rate in labs`	95%
...	...

Hiring managers decide on salary:

$$(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^N, y^N)$$

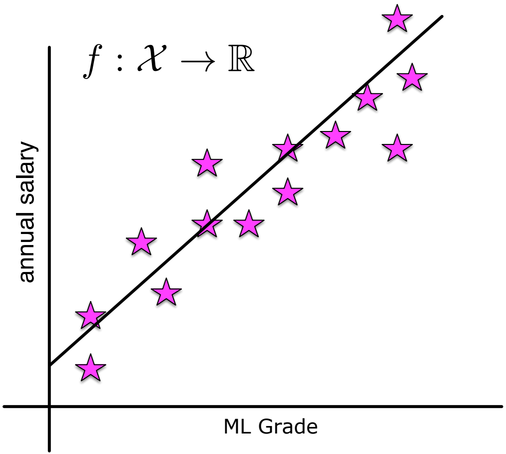
$y^n \in \mathbb{R}^+$ is the annual salary for applicant $\mathbf{x}^n$.

Linear regression tries to replicate that.

UNIVERSITY OF SUSSEX

Modelling the relationship – simplified

- An applicant only has one feature: **FoML grade**
- What is the relationship between **FoML grade** and **annual salary**?
- What is our hypothesis about the relationship?



Hypothesis space

Hypothesis

The relationship is modelled by a straight line:

$$\text{annual salary} \approx \hat{f}(\text{FoML Grade}) = w_0 + w_1 \times \text{FoML Grade}$$

- Why is this called **linear regression**?
  - The hypothesis space consists of straight lines
  - The model is linear in its parameters (w_0 and w_1)

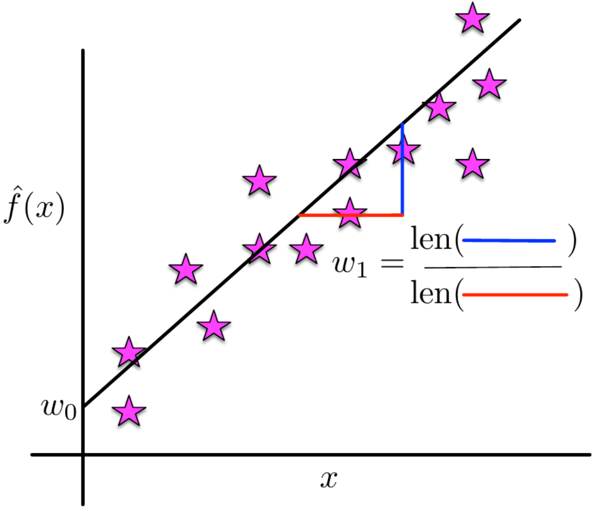
- **Linear classification** implements:

$$\text{hire decision} \approx \hat{f}(\text{FoML Grade}) = \text{sign}(w_0 + w_1 \times \text{FoML Grade})$$

  - This is a binary classifier, predicting 1 or 0 for a given input.
  - The classification boundary is perpendicular to the line given by w_0, w_1

UNIVERSITY OF SUSSEX

Fitting a line to data

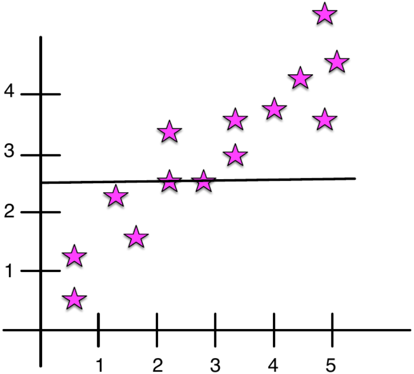


- Form a hypothesis: $\hat{f}(x) = w_0 + w_1 x$
- Need to choose the parameter values w_0 and w_1 to get best fit of line to training data
- Need to find some way of evaluating fit of line to data

UNIVERSITY OF SUSSEX

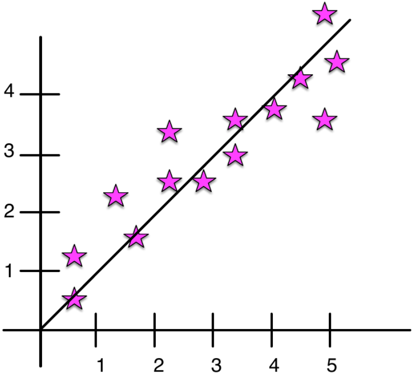
Evaluating hypotheses

$$\hat{f}(x) = w_0 + w_1 x$$



$w_0 = 2.5$ and $w_1 = 0$

$\hat{f}(x) = w_0 + w_1 x$



$w_0 = 0$ and $w_1 = 1$

$\hat{f}(x) = w_0 + w_1 x$

Linear regression – more generally

Formalisation:

- Input: $\mathbf{x}$ (student application)
  - where $\mathbf{x}$ contains several variables this is **multiple** linear regression.
- Output: y (£ amount)
- Labelled data: $(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)$
- Target function: $f : \mathcal{X} \to \mathbb{R}^+$ (ideal annual salary formula)
- Hypothesis space:
  - Given applicant's features (FoML Grade, attendance rates, ...)
  - Find weights $\mathbf{w}$: $y \approx \hat{f}(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i = \langle \mathbf{w}, \mathbf{x} \rangle$ with $x_0 = 1$
    Why is $x_0 = 1$?
    Offset for when $x = 0$, in this case minimum salary if you got 0% on the assignment and 0% attendance.

Matrix notation for linear regression

$$\hat{y}^n = w_0 \cdot 1 + \sum_{i=1}^{d} w_i x_i^n \quad \rightarrow \quad \hat{\mathbf{y}} = X\mathbf{w}$$

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \vdots \\ \hat{y}^N \end{bmatrix} \quad X = \underbrace{\begin{bmatrix} 1 & x_1^1 \dots x_d^1 \\ 1 & x_1^2 \dots x_d^2 \\ & \vdots \\ 1 & x_1^N \dots x_d^N \end{bmatrix}}_{d+1 \text{ features}} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

x superscript: instance n, subscript: feature i

Python:

```
ones = np.ones((data.shape[1],1)) # make a col of ones
X = np.concatenate([ones, data], axis=1) # concat left
y_hat = np.matmul(X, w) # predict
```

UNIVERSITY OF SUSSEX

Think Break

- I will give you a few minutes to reflect on what we've talked about.
- In this time, get some paper, and draw some simple examples of regression.
  - Try and identify what the parameters w_0 and w_1 are: what can you interpret them as in your example?
- Have a think if you can reinterpret it as a classification problem.

How to measure the error (cost function)

How well does $\hat{f}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ approximate our corresponding labelled data, y.

In linear least squares regression, sometimes called ordianry least squares (OLS), we used squared error: $\frac{1}{2}(\hat{f}(\mathbf{x}) - y)^2$

$$\text{residual error} : E_{\text{res}} = \frac{1}{2N} \sum_{n=1}^{N} (\hat{f}(\mathbf{x}^n) - y^n)^2$$

The residual error in matrix notation

$$E_{\text{res}} = \frac{1}{2N} \sum_{n=1}^{N} (\langle \mathbf{w}, \mathbf{x}^n \rangle - y^n)^2$$
$$= \frac{1}{2N} \|X\mathbf{w} - \mathbf{y}\|^2$$

Python:

```
ones = np.ones((data.shape[1],1)) # make a col of ones
X = np.concatenate([ones, data], axis=1) # concat left
y_hat = np.matmul(X, w) # predict
residual = y_hat - y
mse = np.mean(np.square(residual)) / 2
```

Question: Why do we use the squared error?

Gaussian error

- **Answer**: Gaussians, Gaussians, and Gaussians, . . .
- In a probabilistic sense, our prediction is a linear function of the data **plus Gaussian noise**:
  $y^n = \hat{f}(\mathbf{x}^n) + \epsilon_n;\ \epsilon_n \sim \mathcal{N}(0, \sigma^2)$
- Therefore

$$\begin{aligned} p(y^n|\mathbf{x}^n, \mathbf{w}) &= \textbf{Gaussian}(\hat{f}(\mathbf{x}^n), \sigma^2) \\ &\propto \exp(-\frac{1}{2\sigma^2}(\hat{f}(\mathbf{x}^n) - y^n)^2) \end{aligned}$$

- So the sum of squares maximises the **log probability** of the data under a Gaussian distribution.

UNIVERSITY OF SUSSEX

Probabilistic output

- Linear regression gives us a forward model of the data, where we assume that there is some noise on our measurements.
- The simplest model assumes that the level, σ, of noise for all of our outputs is the same.
  - This is referred to as identically distributed or homoscedastic.
  - This can be quite limiting, as it assumes the same variance for small or large outputs.
- If our regression model predicts more than 1 output variable, this is called multivariate linear regression.
  - We usually assume that the errors the model makes for each output are **independent**.
- Overall, we generally assume our errors are i.i.d.

UNIVERSITY OF SUSSEX

Learning in least squares regression

- We've defined our linear regression model as:
  $\hat{\mathbf{y}} = X\mathbf{w}$, where X is our data stored in a matrix with an extra column of 1s, and $\mathbf{w}$ is a vector of weights that we need to choose.
- We've also defined our cost function as the squared error in our predictions:
  $\frac{1}{2N} \|X\mathbf{w} - \mathbf{y}\|^2$, minimising this gives us better predictions.
- How do you think we could choose the best values for w to gives the smallest value for our cost function?

UNIVERSITY OF SUSSEX

Learning in least squares linear regression: 1st approach

Minimising residual error E_{res}

$$E_{\text{res}} = \frac{1}{2N}\left\|X\mathbf{w} - \mathbf{y}\right\|^2 = \frac{1}{2N}\left\langle X\mathbf{w} - \mathbf{y}, X\mathbf{w} - \mathbf{y}\right\rangle$$

Fact: At a minimum of E_{res}, the partial derivative of E_{res} with respect to the vector $\mathbf{w}$ must be zero.
In scalar case of w:

$$\partial_w (xw - y)(xw - y) = 2x(xw - y).$$

In vector case of $\mathbf{w}$:

$$\nabla_{\mathbf{w}}\left\langle X\mathbf{w} - \mathbf{y}, X\mathbf{w} - \mathbf{y}\right\rangle = 2X^\top(X\mathbf{w} - \mathbf{y}).$$

UNIVERSITY OF SUSSEX

Minimising residual error E_{res}

$$E_{\text{res}} = \frac{1}{2N} \left\| X\mathbf{w} - \mathbf{y} \right\|^2 = \frac{1}{2N} \langle X\mathbf{w} - \mathbf{y}, X\mathbf{w} - \mathbf{y} \rangle$$

Fact: At a minimum of E_{res}, the partial derivative of E_{res} with respect to the vector $\mathbf{w}$ must be zero.

$$\begin{aligned} \nabla_{\mathbf{w}} E_{\text{res}}(\mathbf{w}) &= 0 \\ \frac{1}{\cancel{2}N} \left(\cancel{2} X^\top (X\mathbf{w} - \mathbf{y}) \right) &= 0 \\ X^\top X\mathbf{w} - X^\top \mathbf{y} &= 0 \\ X^\top X\mathbf{w} &= X^\top \mathbf{y} \end{aligned}$$

$$\mathbf{w}^* = \underbrace{(X^\top X)^{-1} X^\top}_{\text{pseudo-inverse of } X := X^\dagger} \mathbf{y} = X^\dagger \mathbf{y}$$

UNIVERSITY OF SUSSEX

The linear regression algorithm

1. Construct the matrix X and the vector $\mathbf{y}$ from the data set $(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)$ as follows

$$\underbrace{X = \begin{bmatrix} \text{—}\mathbf{x}^{1,\top}\text{—} \\ \text{—}\mathbf{x}^{2,\top}\text{—} \\ \vdots \\ \text{—}\mathbf{x}^{N,\top}\text{—} \end{bmatrix}}_{\text{input data matrix}} \qquad \underbrace{\mathbf{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix}}_{\text{target vector}}$$

2. Compute the pseudo-inverse $X^\dagger = (X^\top X)^{-1} X^\top$
3. Return $\mathbf{w} = X^\dagger \mathbf{y}$

UNIVERSITY OF SUSSEX

Fitting with pseudo-inverse in Python

```
ones = np.ones((data.shape[1],1)) # make a col of ones
X = np.concatenate([ones, data], axis=1) # concat left
w_hat = np.matmul(np.linalg.pinv(X), y) # fit
y_hat = np.matmul(X, w_hat) # predict
```

UNIVERSITY OF SUSSEX

Summary and outlook

Today we've:

- Talked about the regression task in more detail, and how it's different from classification.
- We've described what linear models are, and how they can be used for both tasks.
- We've talked about the assumptions of linear least squares regression, and how we can interpret the output as the predicted probability of an observation.
- We've seen how to minimise the residual error using the pseudo-inverse.

Next lecture:

- Another approach for fitting regression models to data.
- How to manipulate data to match our assumptions.
- Other ways that regression can be used.

UNIVERSITY OF SUSSEX