CIVIL-557

Decision-aid methodologies in transportation

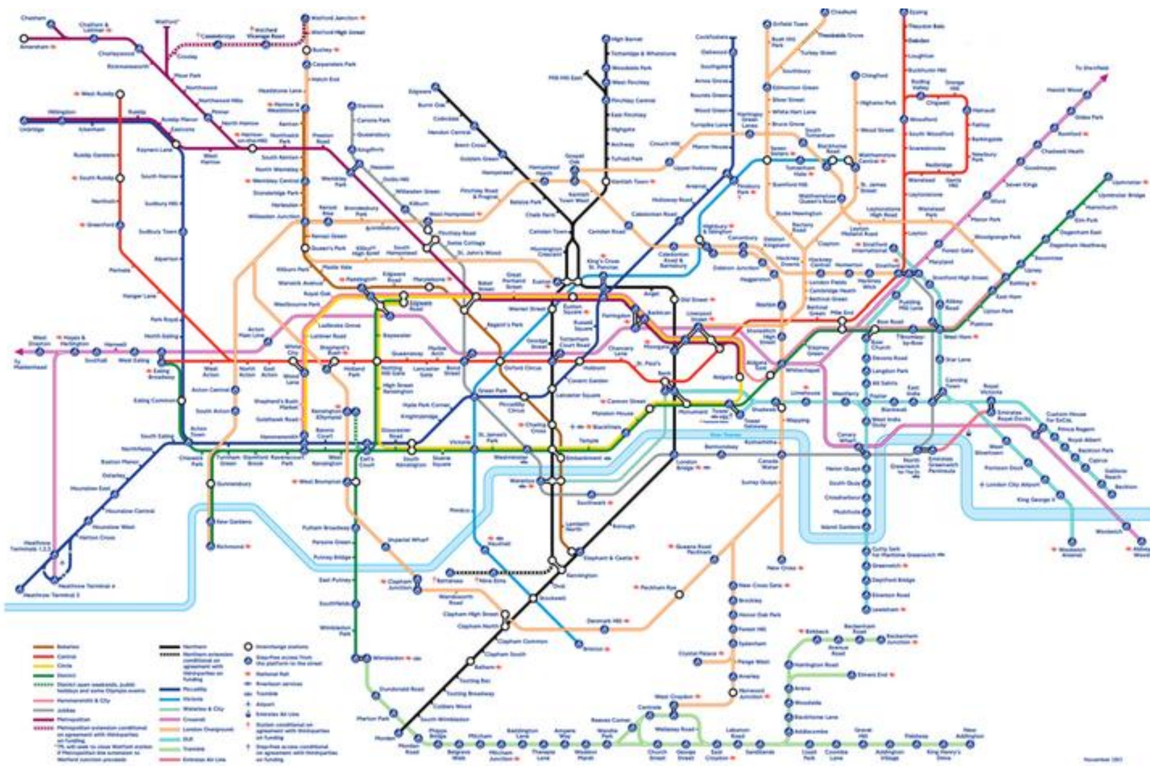
Lecture 3: Logistic regression and probabilistic metrics

Tim Hillel

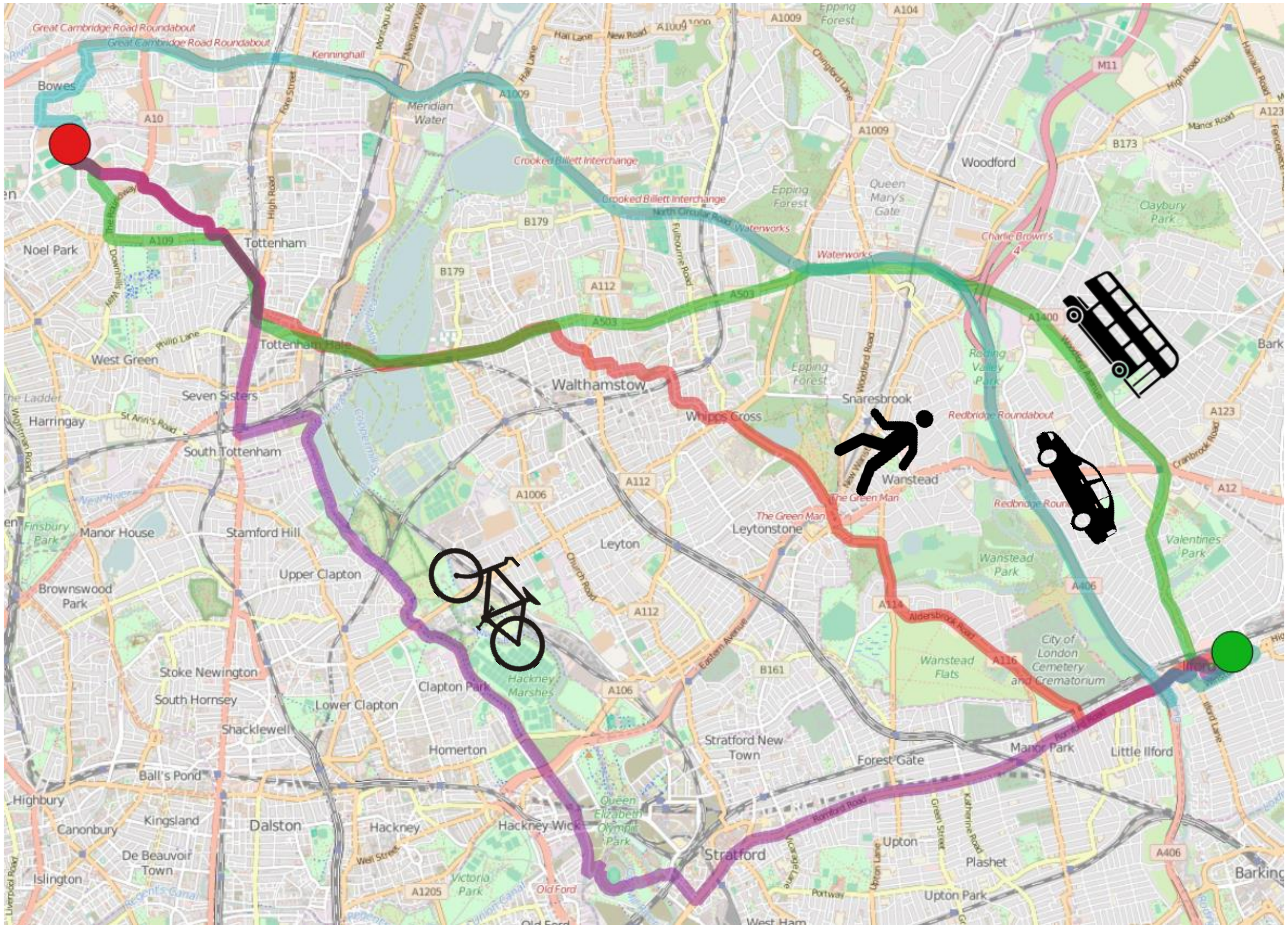
Transport and Mobility Laboratory TRANSP-OR
École Polytechnique Fédérale de Lausanne EPFL

Case study

Transport for London



Mode choice



Bowes
Noel Park
Tottenham
Meridian Water
Woodford
West Green
Harringay
Seven Sisters
South Tottenham
Tottenham Hale
Walthamstow
Whipps Cross
Snaresbrook
Wanstead
Leytonstone
Leyton
Manor House
Stamford Hill
Upper Clapton
Brownswood Park
Stoke Newington
South Hornsey
Clapton Park
Lower Clapton
Shacklewell
Homerton
Ball's Pond
Highbury
Canonbury
Kingsland
Dalston
Hackney
Hackney Wick
De Beauvoir Town
Islington
Stratford
Stratford New Town
Forest Gate
Manor Park
Little Ilford
Upton
Plashet
Upton Park
Wanstead Flats
Wanstead Park
Valentines Park
Claybury Park
Barking

Last week

- Data science process
- Dataset
- Deterministic methods
  - K-Nearest Neighbours (KNN)
  - Decision Tree (DT)
- Discrete metrics

Today

- Theory of probabilistic classification
- Probabilistic metrics
- Probabilistic classifiers
  - Logistic regression

But first…

…a bit more **feature processing**

TRANSP-OR

EPFL

Feature processing

- Last week - scaling
  - Crucial when algorithms consider **distance**
  - *Standard scaling* – zero-mean unit-variance
- What about missing values?
- And categorical data?

Missing values

	start_time_linear	age	female	driving_license	car_ownership	distance
20679	10.000000	46	0	1.0	0	5410
7276	7.500000	68	1	0.0	0	7725
11816	10.250000	37	0	NaN	1	2939
66175	8.750000	18	0	1.0	2	20205
73216	10.750000	44	1	NaN	1	12646
33259	15.333333	19	1	0.0	0	11436
27837	17.666667	62	1	1.0	1	1425
56018	21.500000	48	0	1.0	1	8820

Missing values

Possible solutions?

- Remove **rows** (instances)
- Remove **columns** (features)
- Assume **default value**
  - Zero
  - Mean
  - Random value
  - Other?

Missing values

- Removing rows – can introduce **sampling bias**!
- Removing columns – reduces available features
- Default value – needs to make sense

Categorical variables

	travel_mode	purpose	fueltype	faretype	bus_scale	survey_year
20679	pt	B	Average_Car	full	1.0	1
7276	pt	B	Average_Car	free	0.0	1
11816	drive	HBO	Petrol_Car	full	1.0	1
66175	drive	NHBO	Petrol_Car	dis	0.0	3
73216	pt	HBE	Diesel_Car	full	1.0	3
33259	pt	HBO	Average_Car	full	0.0	2
27837	drive	HBO	Diesel_Car	free	0.0	2
56018	drive	HBO	Petrol_Car	full	1.0	3

Categorical data

All data must be numerical – possible solutions?

- Numerical encoding

- Binary encoding

- Remove feature?

Numerical encoding

Blue	Silver	Red
1	2	3

Implies **order:**

Blue < Silver < Red

and **distance:**

Red – Silver = Silver - Blue

TRANSP-OR

EPFL

Binary encoding

Blue

Silver

Red

Car	Colour:Blue	Colour:Silver	Colour:Red
1	0	1	0
2	1	0	0
3	0	0	1
4	1	0	0
…	…	…	…

A lot more features!

Validation schemes

Train-test split:

Test set must be **unseen data.**

- How to sample test set?
- How to test model hyperparameters?

Sampling test set

Previously used **random sampling**

- Assumes data is **independent**
- Not always a good assumption
  - Sampling bias/recording errors
  - Hierarchical data

Instead, use **external validation**

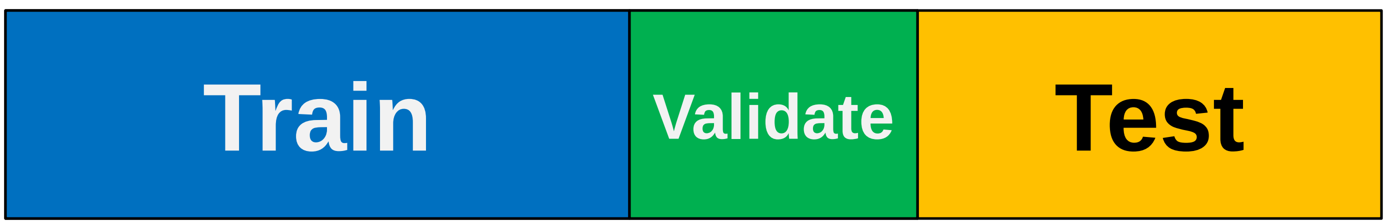
- Validate the model on data **sampled separately** from the training data (e.g. separate year)

Testing model hyperparameters

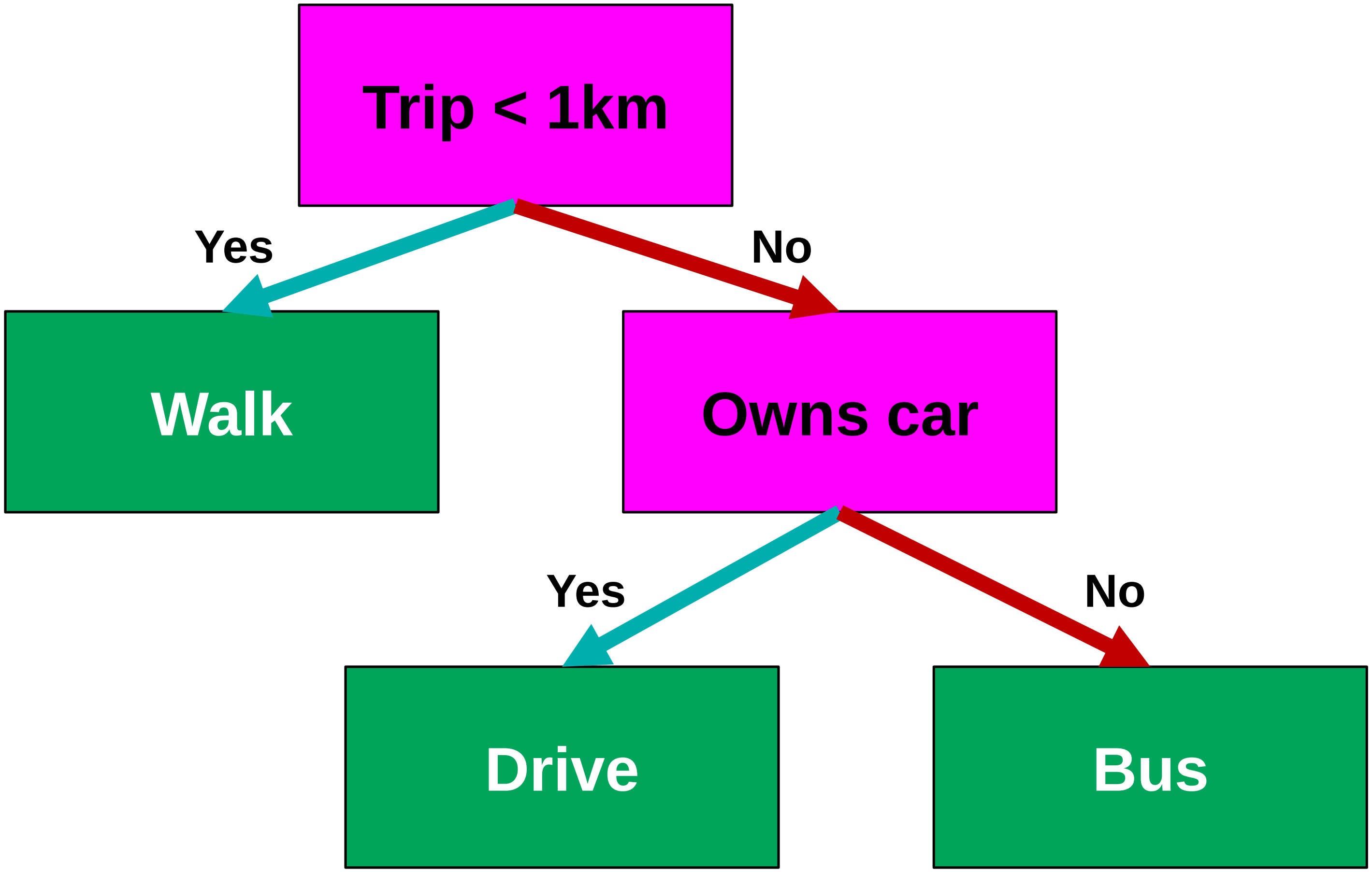
Can only be performed on the training data

Possible solution – split again:

Train-validate-test



Decision trees



DT Metrics

GINI Impurity:

$$G(p) = \sum_{i=1}^{J} p_i(1 - p_i) = 1 - \sum_{i=1}^{J} {p_i}^2$$

Entropy:

$$H(p) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

where p_i is the **proportion** of class i

Decision trees

Remember: Decision trees only see ranking of feature values….

…therefore **scaling** does not need to be applied!

Notebook 1: Categorical features

Probabilistic classification

- Previously considered **deterministic classifiers**
  - Classifier predicts discrete class $\hat{y}$

- Now consider **probabilistic classification**
  - Classifier predicts continuous probability for each class

Notation

- Dataset D contains N elements
- J classes
- Each element n associated with:
  - Feature vector x_n of k real features
    - $x_n \in \mathbb{R}^k$
  - Set of class indicators $y_n \in 0,1^J$:
    - $\sum_{j=1}^{J} y_{in} = 1$
- Selected class (ground-truth)
  - $i_n = \sum_{i=1}^{J} i y_{in}$

TRANSP-OR

EPFL

Notation continued

- Classifier P maps feature vector x_n to probability distribution across J classes
  - $P: \mathbb{R}^k \rightarrow [0,1]^J$
- Probability for each class i:
  - $P(i|x_n) > 0$
  - $\sum_{i=1}^{J} P(i|x_n) = 1$
- Therefore, probability for **selected** (ground-truth) class:
  - $P(i_n|x_n)$

Example

- Four modes (*walk, cycle, pt, drive*)
  - $J = 4$
- For *pt* trip:
  - $y_n = (0, 0, 1, 0)$
  - $i_n = 3$
- Predicted probabilities for arbitrary classifier P
  - $P(x_n) = (0.1, 0.2, 0.4, 0.3)$
  - $P(i_n | x_n) = 0.4$

Likelihood

n	i_n	$P(1\|x_n)$	$P(2\|x_n)$	$P(3\|x_n)$	$P(4\|x_n)$	$P(i_n\|x_n)$
1	2	0.11	0.84	0.03	0.02	**0.84**
2	1	0.82	0.04	0.06	0.08	**0.82**
3	4	0.11	0.18	0.05	0.66	**0.66**
4	1	0.57	0.22	0.12	0.09	**0.57**
5	3	0.10	0.03	0.75	0.12	**0.75**

- Likelihood:
  - $= \prod_{n=1}^{N} P(i_n|x_n)$
  - $= 0.84 \times 0.82 \times 0.66 \times 0.57 \times 0.75$
  - $= 0.2036$

Likelihood

n	i_n	$P(1\|x_n)$	$P(2\|x_n)$	$P(3\|x_n)$	$P(4\|x_n)$	$P(i_n\|x_n)$
1	4	0.11	0.84	0.03	0.02	**0.02**
2	1	0.82	0.04	0.06	0.08	**0.82**
3	4	0.11	0.18	0.05	0.66	**0.66**
4	1	0.57	0.22	0.12	0.09	**0.57**
5	3	0.10	0.03	0.75	0.12	**0.75**

- Likelihood:
  - $= \prod_{n=1}^{N} P(i_n|x_n)$
  - $= 0.02 \times 0.82 \times 0.66 \times 0.57 \times 0.75$
  - $= 0.0046$

Log-likelihood

- Likelihood bound between 0 and 1
- Tends to 0 as N increases
  - Computational issues with small numbers

- Use *log-likelihood:*
  - $= \ln(\prod_{n=1}^{N} P(i_n|x_n))$
  - $= \sum_{n=1}^{N} \ln P\,(i_n|x_n)$

Log-likelihood

n	i_n	$P(1\|x_n)$	$P(2\|x_n)$	$P(3\|x_n)$	$P(4\|x_n)$	$P(i_n\|x_n)$
1	2	0.11	0.84	0.03	0.02	**0.84**
2	1	0.82	0.04	0.06	0.08	**0.82**
3	4	0.11	0.18	0.05	0.66	**0.66**
4	1	0.57	0.22	0.12	0.09	**0.57**
5	3	0.10	0.03	0.75	0.12	**0.75**

- Log-likelihood:
  - $= \sum_{n=1}^{N} \ln P(i_n|x_n)$
  - $= \ln(0.84 \times 0.82 \times 0.66 \times 0.57 \times 0.75)$
  - $= -1.638$

Log-likelihood

n	i_n	$P(1\|x_n)$	$P(2\|x_n)$	$P(3\|x_n)$	$P(4\|x_n)$	$P(i_n\|x_n)$
1	4	0.11	0.84	0.03	0.02	**0.02**
2	1	0.82	0.04	0.06	0.08	**0.82**
3	4	0.11	0.18	0.05	0.66	**0.66**
4	1	0.57	0.22	0.12	0.09	**0.57**
5	3	0.10	0.03	0.75	0.12	**0.75**

- Log-likelihood:
  - $= \sum_{n=1}^{N} \ln P(i_n|x_n)$
  - $= \ln(0.02 \times 0.82 \times 0.66 \times 0.57 \times 0.75)$
  - $= -5.376$

Cross-entropy loss

- Log-likelihood bound between $-\inf$ and 0
  – Tends to $-\inf$ as N increases

- Can't compare between different dataset sizes
  – Normalize (divide by N) to get cross-entropy loss (CEL) H
  – Typically take negative

$$L = \frac{-1}{n} \sum_{n=1}^{N} \ln P\left(i_n | x_n\right)$$

Cross-entropy loss

- Can also derive from **Shannon's cross entropy** $H(p,q)$ (hence name)

- Minimising cross-entropy loss equivalent to **maximising likelihood of data under the model**
  - Maximum likelihood estimation (MLE)

- Other metrics exist e.g. Brier score (MSE), but CEL is *golden standard*

Probabilistic classifiers

- Need a model which generates multiclass probability distribution from feature vector x_n

- Lot's of possibilities!

- Today, **logistic regression**

Linear regression

- Logistic regression is an extension of **linear regression**
  - Often called *linear models*

$$f(x) = \sum_{k=1}^{K} \beta_k x_k + \beta_o$$

- Find parameters (β) using **gradient descent**
  - Typically minimise MSE

Logistic regression

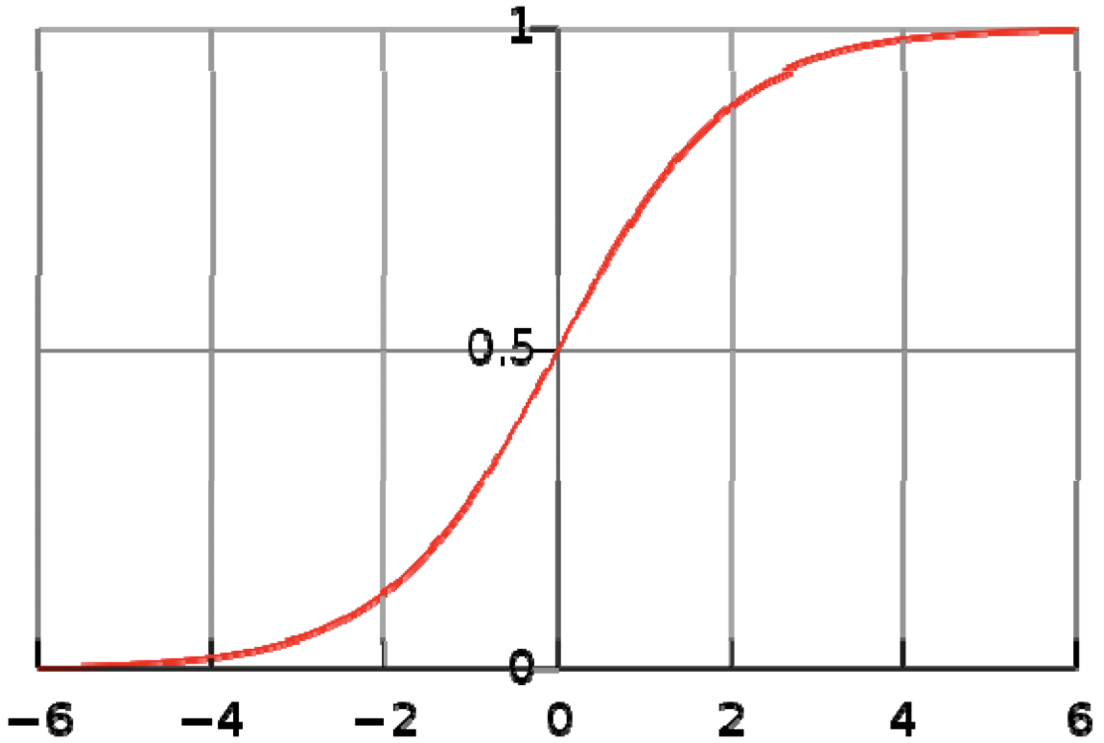
- Pass linear regression through function σ

$$f(x) = \sigma\left(\sum_{k=1}^{K} \beta_k x_k + \beta_o\right)$$

- σ needs to take real value and return value in [0,1]
  - $\sigma(z) \in [0,1] \quad \forall\, x \in \mathbb{R}$

Binary case – sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



TRANSP-OR

EPFL

Multinomial case – softmax function

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{J} e^{z_i}}$$

TRANSP-OR

Logistic regression

- Separate set of parameters (β) for each class i
  – Normalised to zero for one class
- Separate value z for each class
  – *Utilities* in **Discrete Choice Models**

- Minimise cross-entropy loss (MLE)
- Solve using gradient descent for parameters
  – $\beta^* = \arg\min_{\beta} L_\beta$

Issues

- Overconfidence
  - Linearly separable data – parameters tend to infinity (unrealistic probability estimates)
- Outliers
  - Outliers can have disproportionate effect on parameters (high variance)
- Multi-colinearity
  - If features are highly correlated, can cause numerical instability

Solution: Regularisation

- Penalise model for large parameter values
  - Reduces variance (therefore *overfitting*)
- Regularisation
  - $\beta^* = \arg\min_{\beta}\{L_\beta + C \times f(\beta)\}$
- Two candidates for f
  - L1 normalisation (LASSO)
    - $\sum_i |\beta_i|$ (similar to Manhattan distance)
  - L2 normalisation (Ridge)
    - $\sum_i \beta^2$ (similar to Euclidean distance)

Regularisation

- L1 regularisation shrinks less important parameters to zero
  - Feature selection?

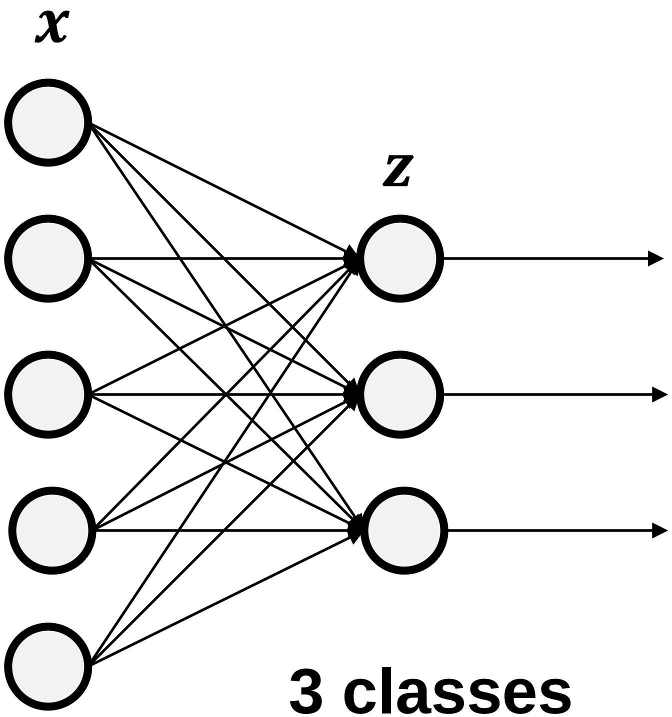
- L2 regularisation penalises larger parameters more

Regularisation

- C and choice of regularisation (L1/L2) are **hyperparameters**

- Larger C, more regularisation (higher bias, lower variance)

Logistic regression



Softmax

3 classes

5 features

Homework

Notebook 2: Logistic regression