Applied Natural Language Processing

Dr Jeff Mitchell, University of Sussex

Autumn 2025

Document classification

Previously

- Document classification
  - Feature extraction
  - Word list classifiers
- Evaluation
  - accuracy and error rate
  - the confusion matrix
  - precision, recall and F1-score

This time

- A machine learning approach
  - Some probability theory
  - Naïve Bayes classification

Probability theory

Probability problem for you

- At University Y, there are 250 CS students.
- 100 of the CS students live on campus.
- 35 of the CS students are regularly late for class.
- 14 of the CS students live on campus and are regularly late for class.
- Given this evidence, is there an association between ‘living on campus’ and ‘being regularly late for class’; or are these events independent?

Elementary probability theory

- A **random variable** X ranges over a set of predefined values
  - If C is the random variable "a student lives on campus" then the possible set of values are {true, false}
- The probability that **X** has some value **v** is written:
  - $P(X = v)$ or $P(v)$ if the identity of X can be assumed
  - For example: $P(C = \text{true}) = \frac{100}{250} = 0.4$
- The sum of all possible values of any random variable X is 1

uppercase for variables

lowercase for values

$$\sum_v P(X = v) = 1$$

More probability theory

- Two events, e.g., a='living on campus' (*C*=true) and b= 'being regularly late for class' (*L*=true), are independent IFF

conditional probability

$$P(b|a) = P(b)$$

Number of students who live on campus AND who are regularly late

Number of students who are live on campus

- $P(b) = P(L = \text{true}) = \frac{35}{250} = 0.14$
- $P(b|a) = P(L = true \,|C = true) = \frac{14}{100} = 0.14$
- Therefore the two events are **independent**
- There appears to be no association between students living on campus and being regularly late for class

Conditional probability

The **conditional** probability of two events is equal to the ratio of the **joint** probability to the **marginal** probability.

joint probability

conditional probability

$$P(a|b) = \frac{P(a,b)}{P(b)}$$

marginal probability

$$P(b|a) = \frac{P(a,b)}{P(a)}$$

eliminating the joint probability $P(a,b)$

$$P(a,b) = P(a|b).P(b) = P(b|a).P(a)$$

BAYES' LAW

and rearranging

$$P(a|b) = \frac{P(b|a) \times P(a)}{P(b)}$$

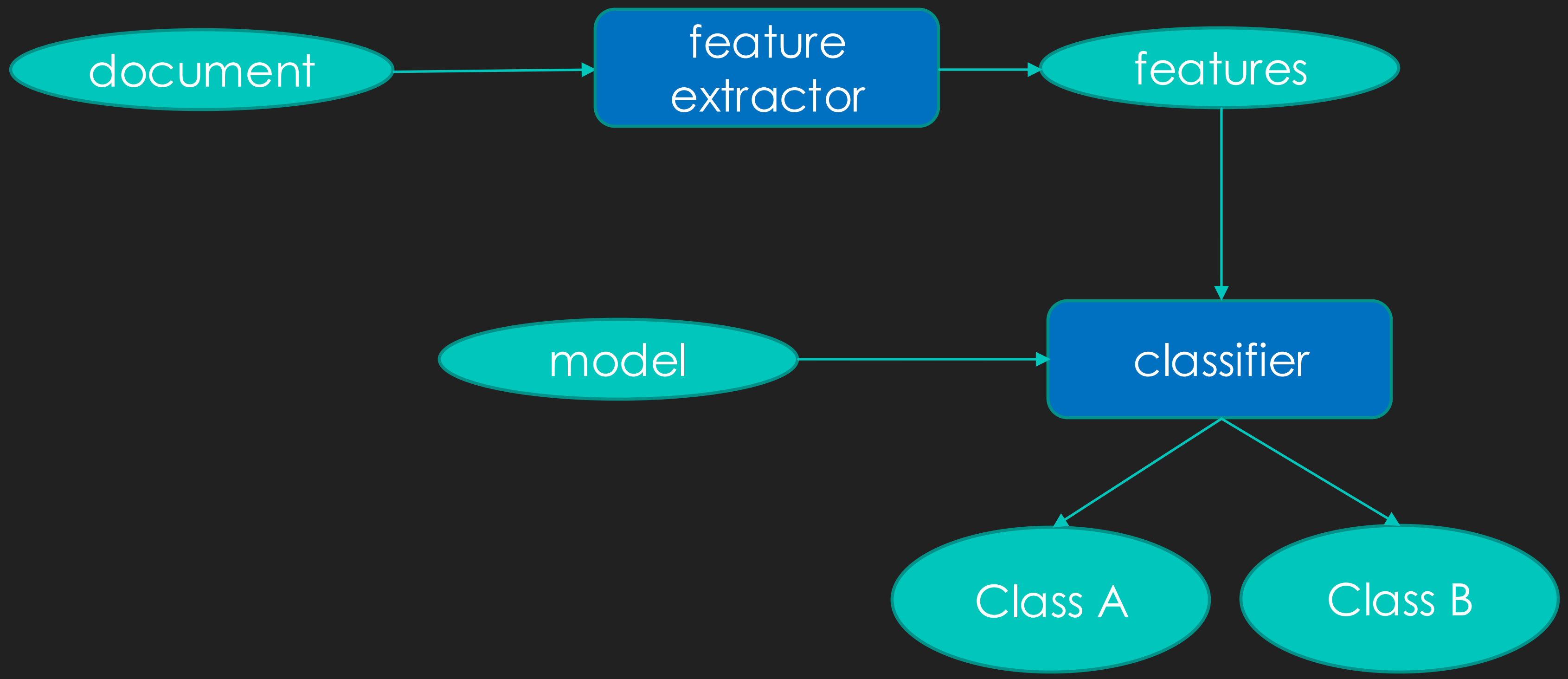
Joint probability of independent events

- We know $P(a, b) = P(a|b).P(b)$
- But $P(a|b) = P(a)$ if a and b are independent
- So, for **independent events** a and b:

$$P(a, b) = P(a).P(b)$$

Naïve bayes classification

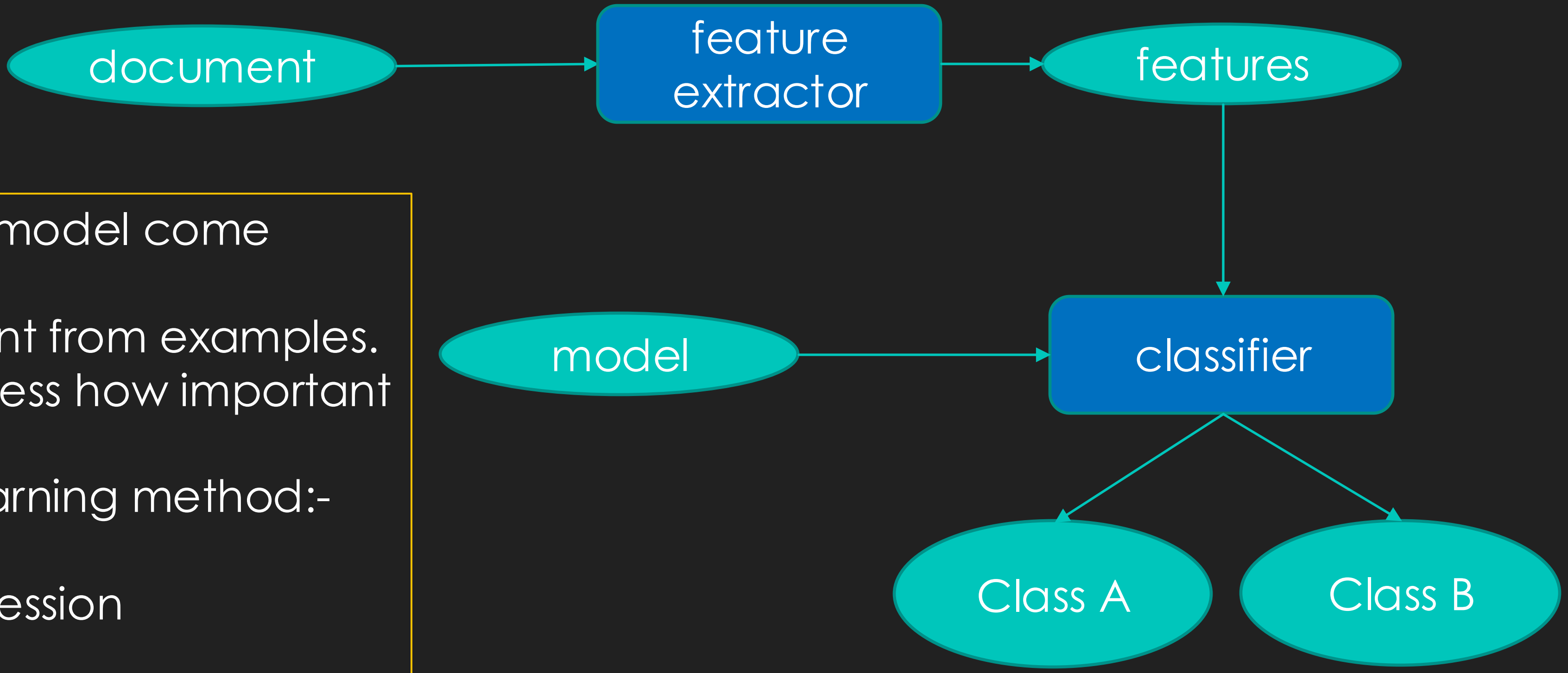
General Architecture for Document Classification (Recap)



1. Feature extractor turns document into a set or **vector** of features

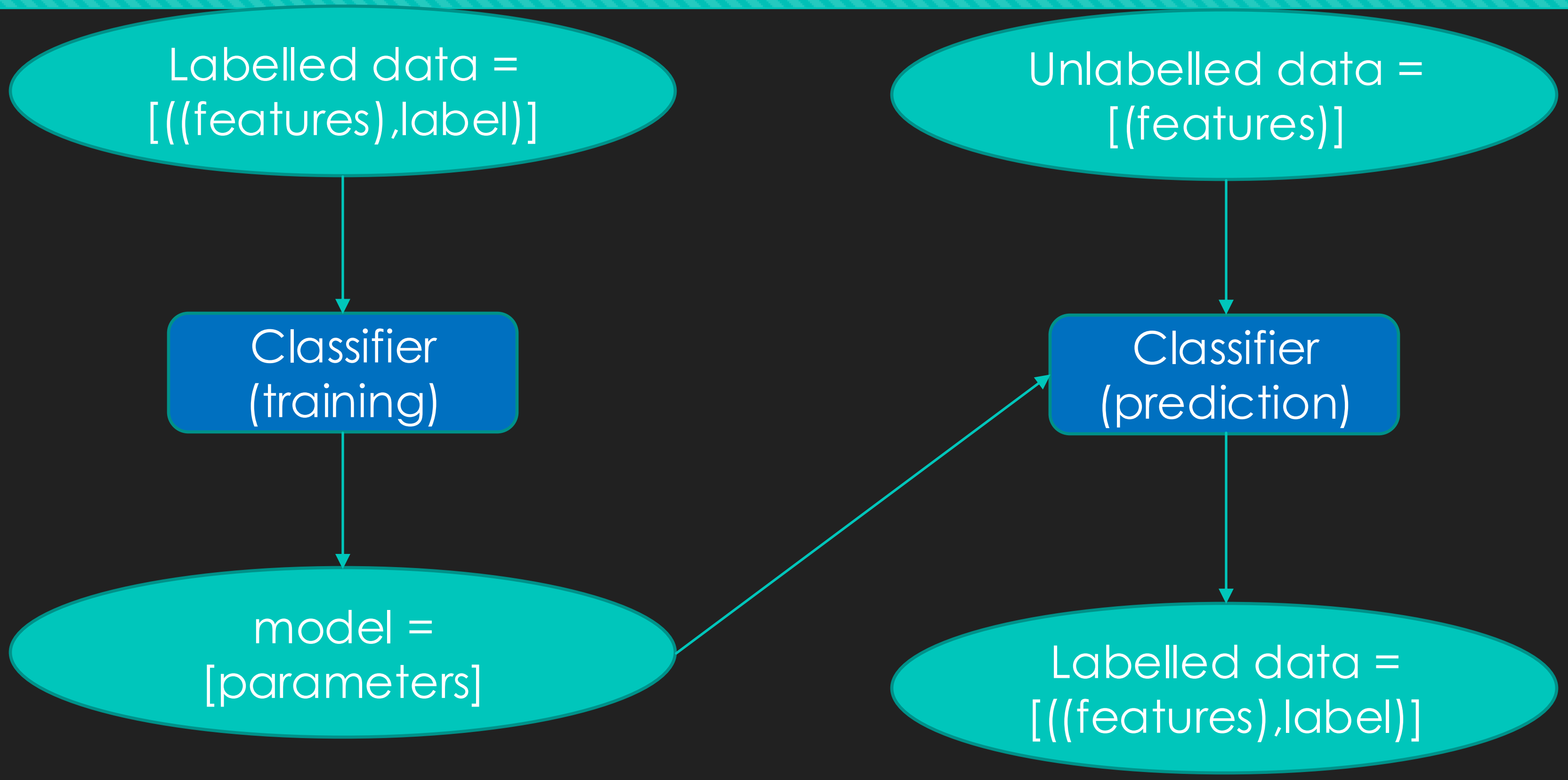
2. Classifier consults a model of what features to expect in different classes and decides the **most likely** class accordingly

Machine learning approach



- Where does the model come from?
- The model is learnt from examples.
- Don't second guess how important each feature is
- Just choose a learning method:-
  - Naïve Bayes
  - Logistic Regression
  - SVM
  - Neural network

Learning Classifiers



Problem instantiation

- A data item to classify is viewed as a tuple of features $(f_1, f_2, \ldots, f_n)$
- As a shorthand for $(f_1, f_2, \ldots, f_n)$ we write f_1^n
- The set of possible classes is C
- A particular class is denoted by c where $c \in C$
- The goal is to assign a class based on f_1^n
- Probability of some class given features f_1^n is $P(c|f_1^n)$
- We want the most probable class: $\underset{c}{\operatorname{argmax}} P(c|f_1^n)$

Naïve bayes classification

Applying Bayes rule

$$\underset{c}{\operatorname{argmax}} P(c|f_1^n) = \underset{c}{\operatorname{argmax}} \frac{P(f_1^n|c).P(c)}{P(f_1^n)}$$

Ignoring the denominator because it does not affect which class maximises the probability

$$\underset{c}{\operatorname{argmax}} P(c|f_1^n) = \underset{c}{\operatorname{argmax}} P(f_1^n|c).P(c)$$

Making the naïve assumption that features are independent, we can find their joint probability by multiplying the probabilities of individual features

$$\underset{c}{\operatorname{argmax}} P(c|f_1^n) \approx \underset{c}{\operatorname{argmax}} \left(\prod_{i}^{n} P(f_i|c)\right).P(c)$$

Naïve bayes parameters (Training)

The parameters of a Naïve Bayes model are:

- the **prior** probabilities:

$$P(c)$$

- the **class conditional** probabilities of each feature given each class:

$$P(f|c)$$

We estimate these probabilities from the labelled training data using maximum likelihood estimation (MLE)

Estimating the priors

- We want to estimate the probability of each class
- Suppose we have *k* documents in the training sample: $\{d_1, d_2, \dots, d_k\}$
- For each document in the sample, d_i, we know the correct label $label(d_i)$
- The MLE for $P(c)$ is the proportion of the labels of $\{d_1, d_2, \dots, d_k\}$ that are equal to c:

$$P(c) = \frac{|\{i | label(d_i) = c\}|}{k}$$

Question for you

- In a training set of 100 documents, 40 are labelled as positive and 60 are labelled as negative. What is the prior probability of a document being labelled positive?

Estimating the conditional probabilities

- For each feature, we want to know its probability of occurrence for each class
- To estimate these probabilities we know the label of each document $label(d_i)$ and the features of each document $feats(d_i)$
- We distinguish three different event models
  - (multi-variate) Bernoulli model
  - multinomial model
  - multinomial model truncated to 1

Bernoulli Naïve Bayes model

- Only considers whether a feature is in a document or not
- Document is represented as a vector of Booleans, one for each feature
- Maximum likelihood estimate for conditional probability *P(f_j | c)* is the **proportion of documents labelled with class c that have feature f_j**

$$P(f_j|c) = \frac{|\{i|label(d_i) = c \text{ and } f_j \text{ in } feats(d_i)\}|}{|\{i|label(d_i) = c\}|}$$

Multinomial Naïve Bayes Model

- Considers all occurrences of a feature in a document
- Document is represented as a vector of counts, one for each feature
- Maximum likelihood estimate for conditional probability $P(f_j \mid c)$ is the **proportion of features in documents labelled with class c that are feature f_j**

$$P(f_j|c) = \frac{count(c, f_j)}{\sum_{i=0}^{|V|} count(c, f_i)}$$

Multinomial model truncated to 1

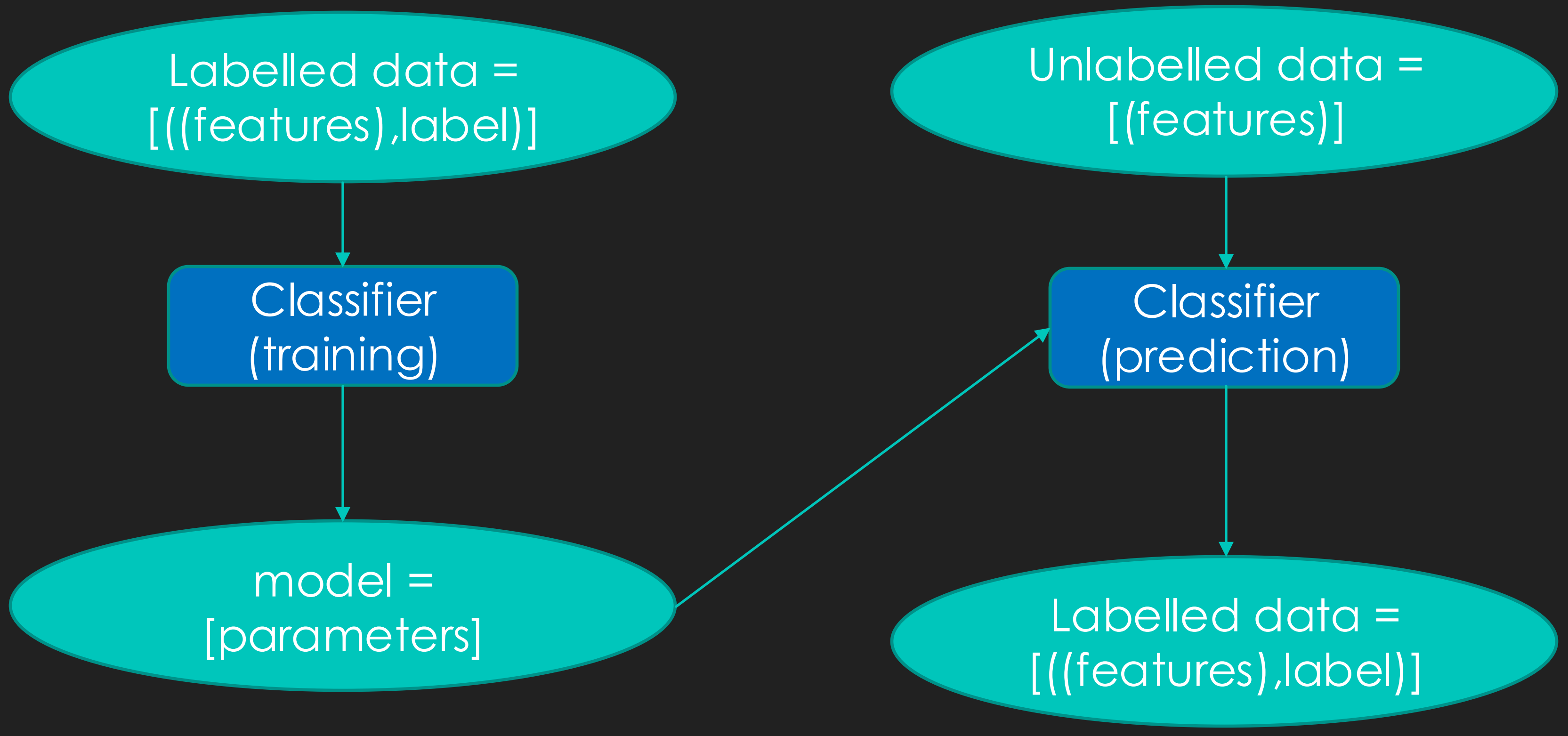
- Only considers the first occurrence of each feature in a document
  - so similar to the Bernoulli model where the feature representation is binary
- Maximum likelihood estimate for conditional probability $P(f_j \mid c)$ is the **proportion of features in documents labelled with class c that are feature f_j**
  - making it a multinomial model

This can't be more than the number of documents

$$P(f_j|c) = \frac{count(c, f_j)}{\sum_{i=0}^{|V|} count(c, f_i)}$$

This depends on the number of documents and the size of the vocabulary

Learning Classifiers



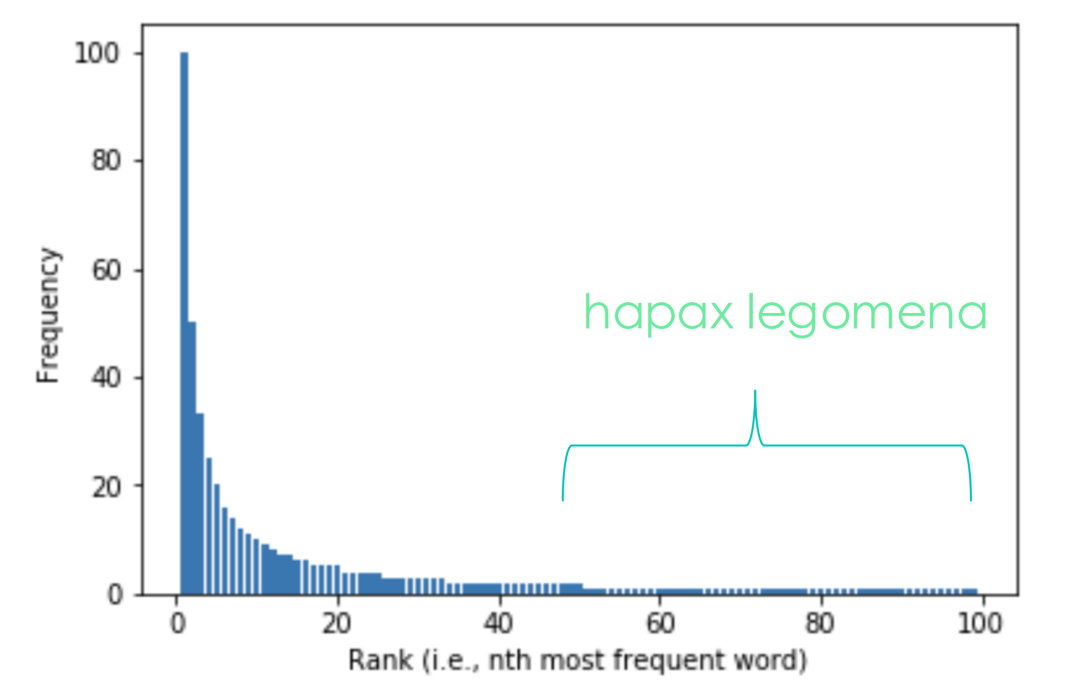
Labelling unseen data (Prediction)

- Suppose we have an unseen, unlabeled document *d*
- We want to use our model to predict the correct label for *d*, *label(d)*
- Let the features of d be $(f_1, \dots, f_n)$
- We predict the label we get from

$$label(d) = \underset{c}{\operatorname{argmax}} \left(\prod_{i}^{n} P(f_i|c) \right) . P(c)$$

where the probabilities are those estimates found from the labelled data

Problem: Data sparseness



- Remember Zipf's Law from last week?
  - Zipf's Law states that "**the product of a word's frequency and its rank frequency is approximately constant.**"
    - So, if the most frequent word occurs 100 times,
    - the 2nd most frequent word will occur 50 times,
    - the 3rd most frequent word will occur 33 times,
    - And so on.

- Many events are rare
- Many events in the test data will not have occurred in the training data

Problem: Data sparseness

$$label(d) = \underset{c}{\operatorname{argmax}} \left(\prod_{i}^{n} P(f_i|c) \right) . P(c)$$

- We are multiplying together lots of probabilities
- What happens if a feature in the test document was never seen in a document of the given class in the training sample?

Smoothing

- Nothing is impossible!
- We need to smooth the estimated probability distributions
- Simplest form of smoothing is **add-one** smoothing
- Simply add one to all of the counts when estimating $P(f \mid c)$:
  - So if a feature has not been seen with class c, we give it a count of 1
  - If it has been seen once, we give it a count of 2
  - And so on
- We can also smooth the prior distributions (but not usually necessary).

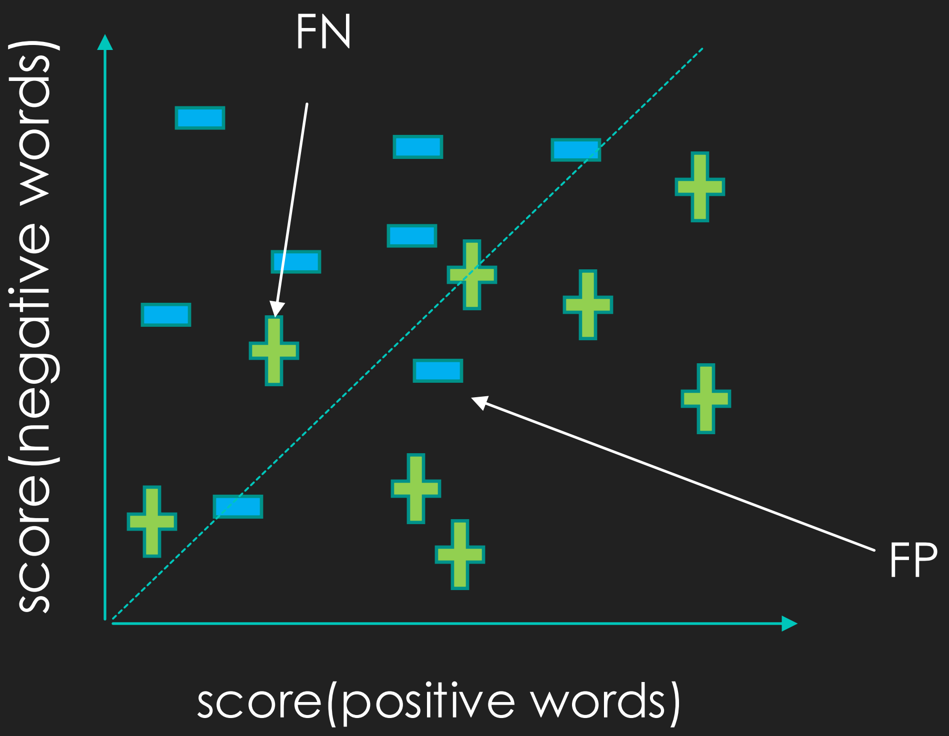
Precision and Recall (again)

Stop and think

- In a collection of 100 documents, 20 of them are actually relevant to NLE. If a classifier predicts 30 of them as being relevant and its recall is 50%, what is it's precision?

Trading Off Precision and Recall

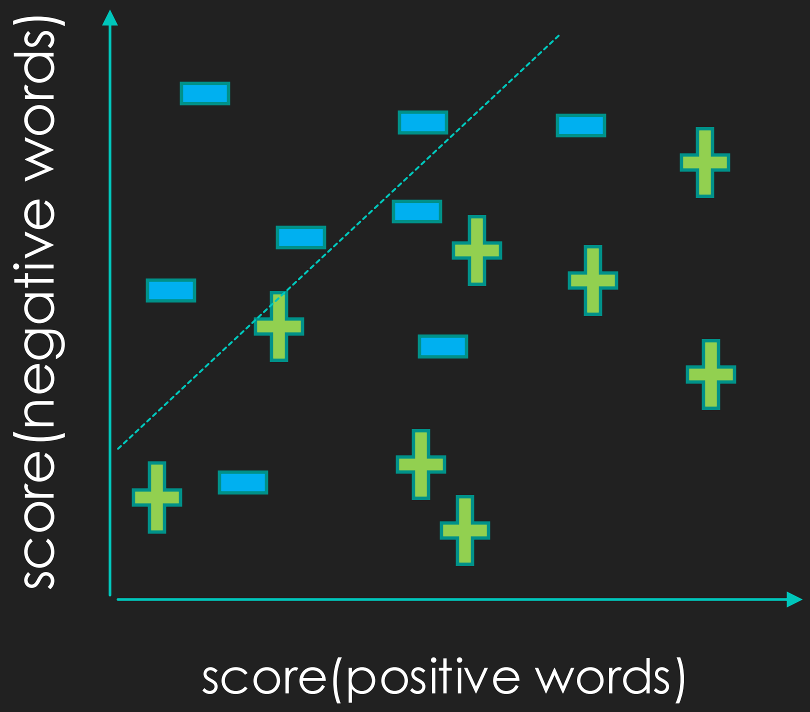
For any given classifier, we can change the **decision boundary**, making it more or less likely to classify an item as positive.



- The standard decision rule for a wordlist classifier is **score(positive words) > score(negative words)**
- Everything below the boundary of the graph is classified as positive, everything above is classified as negative
- If the true labels are as shown, we get some FN and some FP
- Affecting precision and recall

Trading Off Precision and Recall

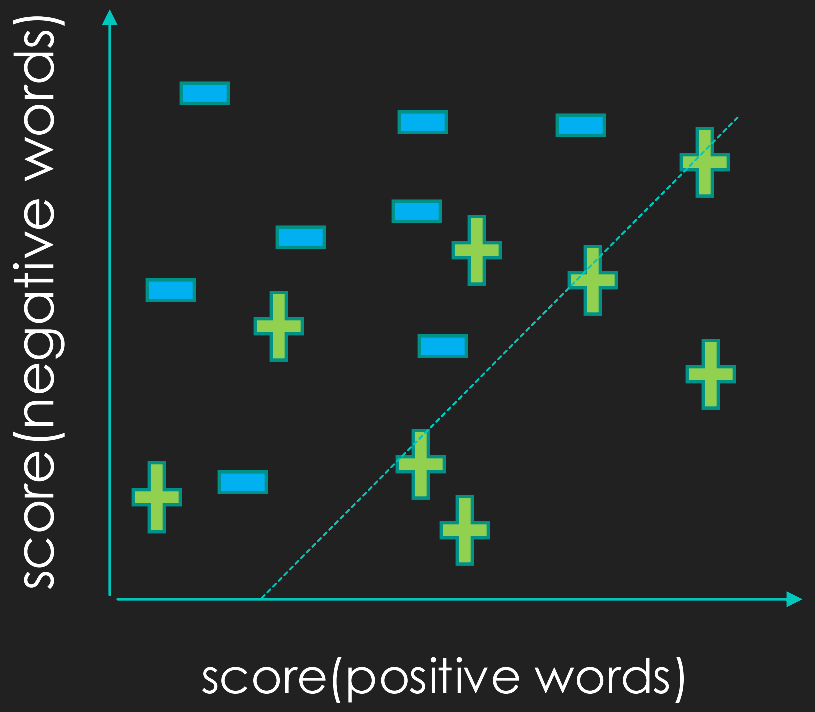
- For any given classifier, we can change the **decision boundary**, making it more or less likely to classify an item as positive.



- Moving the boundary to the left reduces the number of FN
- Recall → 1

Trading Off Precision and Recall

- For any given classifier, we can change the **decision boundary**, making it more or less likely to classify an item as positive.



- Moving the boundary to the right reduces the number of FP
- Precision → 1

Naïve Bayes decision rule

- If $P(+ve|f_1^n) > P(-ve|f_1^n)$ then choose +ve class
- If $P(-ve|f_1^n) > P(+ve|f_1^n)$ then choose -ve class
- The standard decision boundary is therefore defined by:

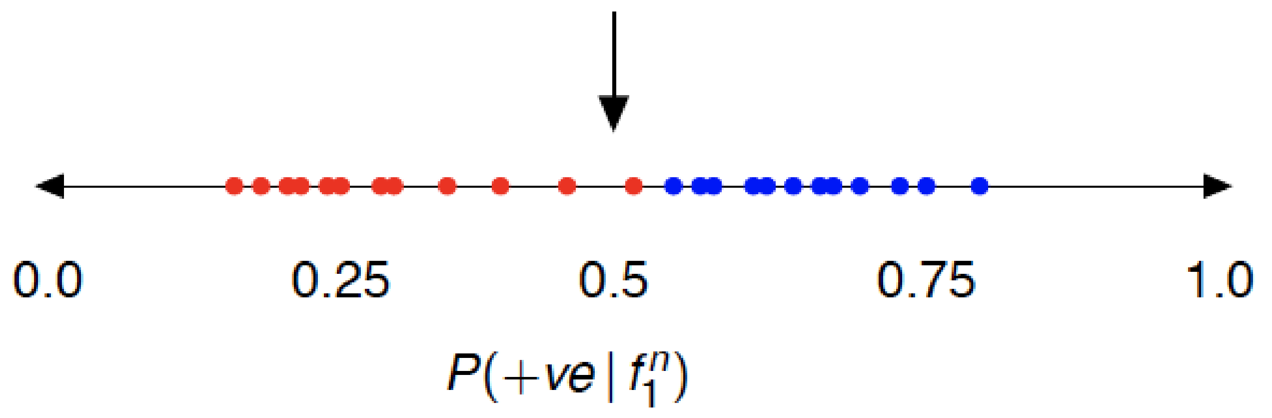
$$P(+ve|f_1^n) = 0.5$$

- We can generalize this (and trade-off precision and recall) to:

$$P(+ve|f_1^n) = p$$

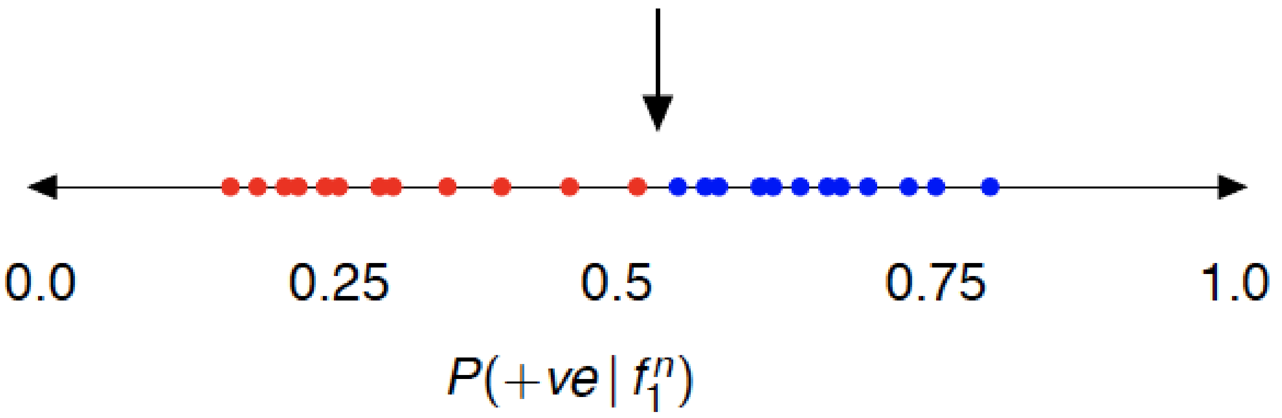
for some $0 \leq p \leq 1$

Easy decision boundary



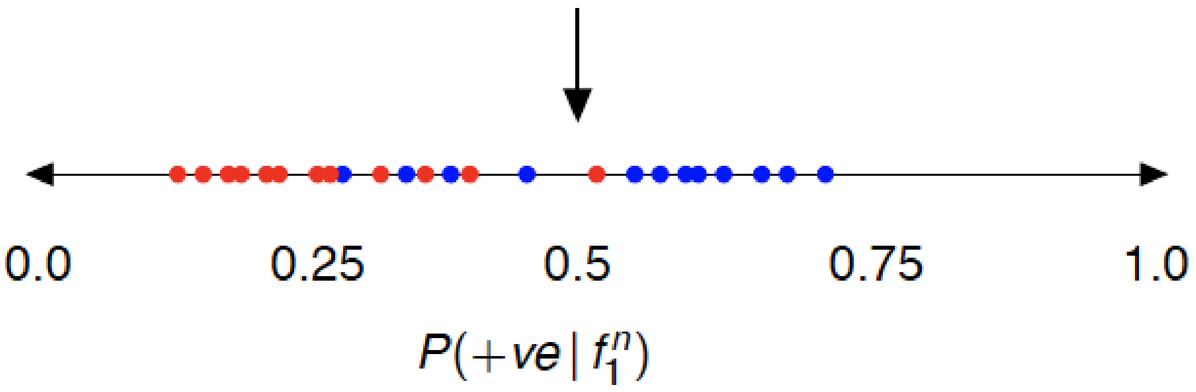
0 blue documents mis-classified
1 red document mis-classified

Easy decision boundary



0 blue documents mis-classified
0 red documents mis-classified

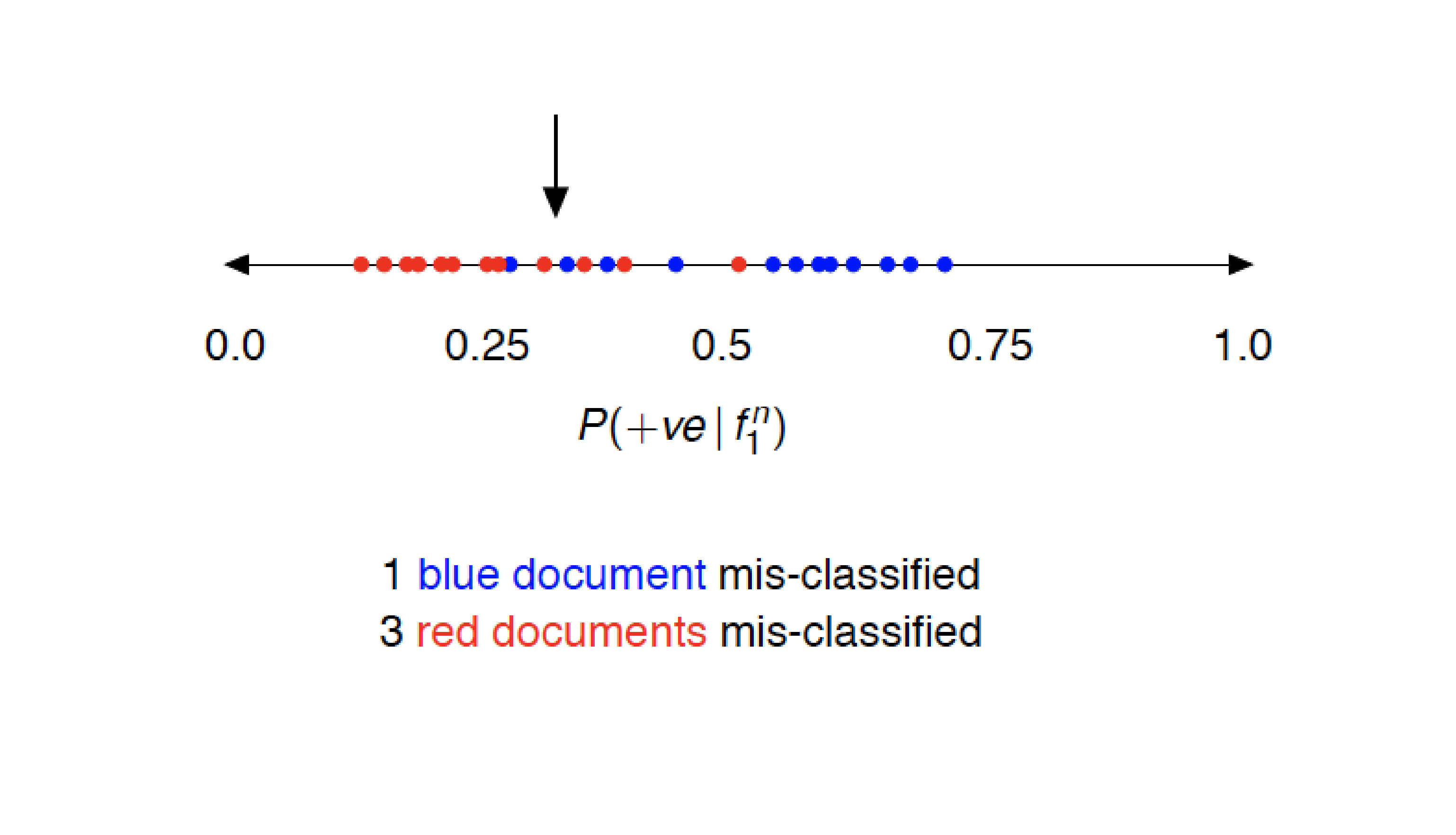
Harder decision boundary



4 blue documents mis-classified
1 red document mis-classified

- FN > FP
- High precision
- Low recall

Harder decision boundary



- FP > FN
- Low precision
- High recall

Making progress

- This week you should complete **all** of the exercises in both notebooks for week 4 on Further Document Classification:

  - ☐ Part 1: Lab_4_1.ipynb
  - ☐ Part 2: Lab_4_2.ipynb

Keywords Check

binary classification		Naïve Bayes classifier	
bag-of-words		prior probabilities	
supervised learning		class conditional probabilities	
over-fitting		smoothing	
hyperparameter		Bernoulli event model	
random variable		multinomial event model	
joint probability		maximum likelihood estimation	
conditional probability			
marginal probability			
Bayes Law			

More Python

Pandas

- = **Python Data Analysis library**
- use it to store and analyse tabular data
- lots of functionality
- here, we are mainly using it for visualisation of experimental results