

7: Catchup Session & very short intro to other classifiers

Machine Learning and Real-world Data (MLRD)

Paula Buttery

Lent 2018

What happens in a catchup session?

- Lecture and practical session as normal.
- New material is non-examinable.
- Time for you to catch-up or attempt some starred ticks.
- Demonstrators help as per usual.

Naive Bayes is a probabilistic classifier

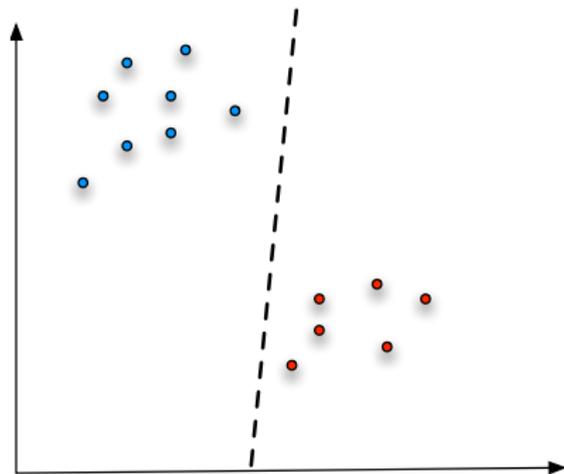
- Given a set of input features a probabilistic classifier provide a distribution over classes.
- That is, for a set of observed features O and classes $c_1 \dots c_n \in C$ gives $P(c_i|O)$ for all $c_i \in C$
- For us O was the set all the words in a review $\{w_1, w_2, \dots, w_n\}$ where w_i is the i th word in a review, $C = \{\text{POS}, \text{NEG}\}$
- We decided on a single class by choosing the one with the highest probability given the features:

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c|O)$$

An SVM is a popular non-probabilistic classifier

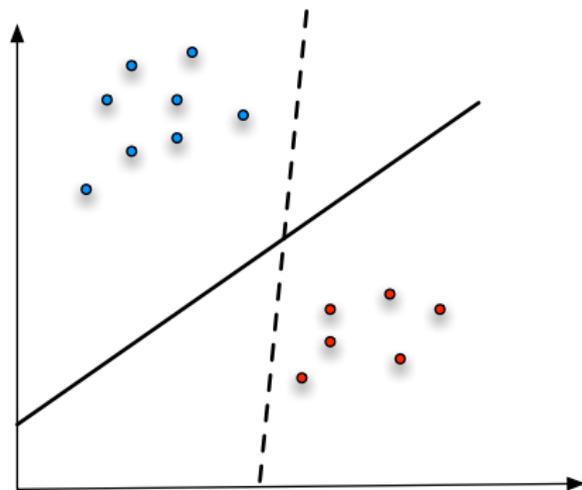
- A Support Vector Machine (SVM) is a non-probabilistic binary linear classifier
- SVMs assign new examples to one category or the other
- SVMs can reduce the amount of labeled data required to gain good accuracy
- A linear-SVM can be considered to be a base-line for non-probabilistic approaches
- SVMs can be efficiently adapted to perform a non-linear classification

SVMs find hyper-planes that separate classes



- Our classes exist in a multidimensional feature space
- A linear classifier will separate the points with a hyper-plane

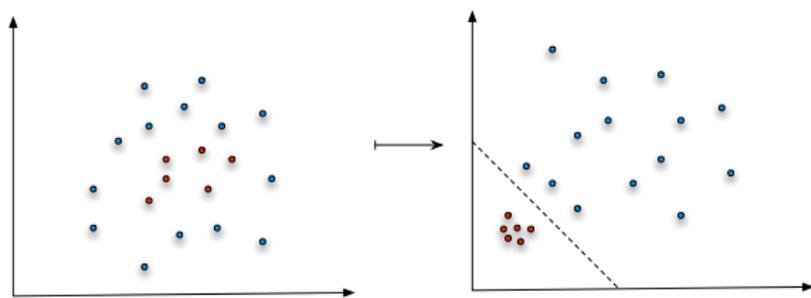
SVMs find a maximum-margin hyperplane in noisy data



- There are many possible hyperplanes
- SVMs find the best hyperplane such that the distance from it to the nearest data point from each class is maximised
- i.e. the hyperplane that passes through the widest possible gap (hopefully helps to avoid over-fitting)

SVMs can be very efficient and effective

- Efficient when learning from a large number of features (good for text)
- Effective even with relatively small amounts of labelled data (we only need points close to the plane to calculate it)
- We can choose how many points to involve (size of margin) when calculating the plane (tuning vs. over-fitting)
- Can separate non-linear boundaries by increasing the feature space (using a kernel function)



Choice of classifier will depend on the task

Comparison of a SVM and Naive Bayes on the same task:

- 2000 imdb movie reviews, 400 kept for testing
- preprocess with improved tokeniser (lowercased, removed uninformative words, dealt with punctuation, lemmatised words)

	SVM	Naive Bayes
Accuracy on train	0.98	0.96
Accuracy on test	0.84	0.80

- But from Naive Bayes I know that *character*, *good*, *story*, *great*, ... are informative features
- SVMs are more difficult to interpret

Decision tree can be used to visually represent classifications



- Simple to interpret
- Can mix numerical and categorical data
- You specify the parameters of the tree (maximum depth, number of items at leaf nodes—both change accuracy)
- But finding the optimal decision tree can be np-complete

Information gain can be used to decide how to split

- Information gain is defined in terms of entropy H

Entropy of tree node:

$$H(n) = - \sum_p p_i \log_2 p_i$$

where p 's are the fraction of each class at node n

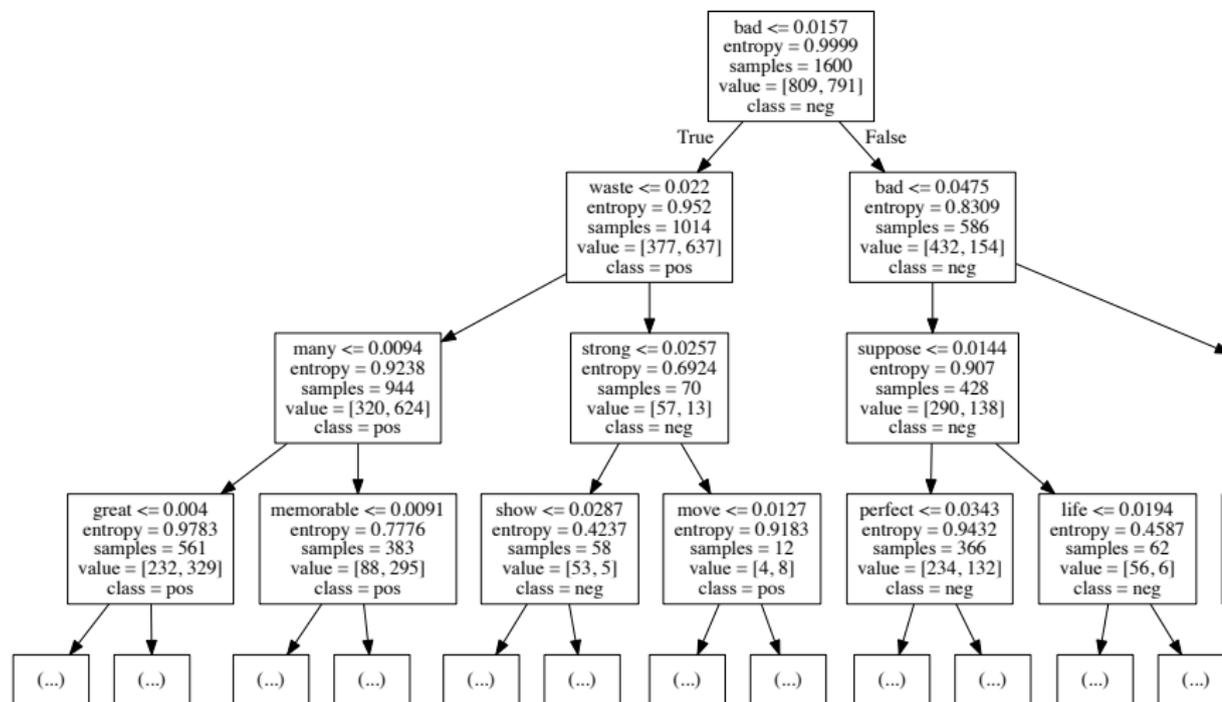
- Information gain I is used to decide which feature to split on at each step in building the tree

Information gain:

$$I(n, D) = H(n) - H(n|D)$$

where $H(n|D)$ is the weighted entropy of the daughter nodes.

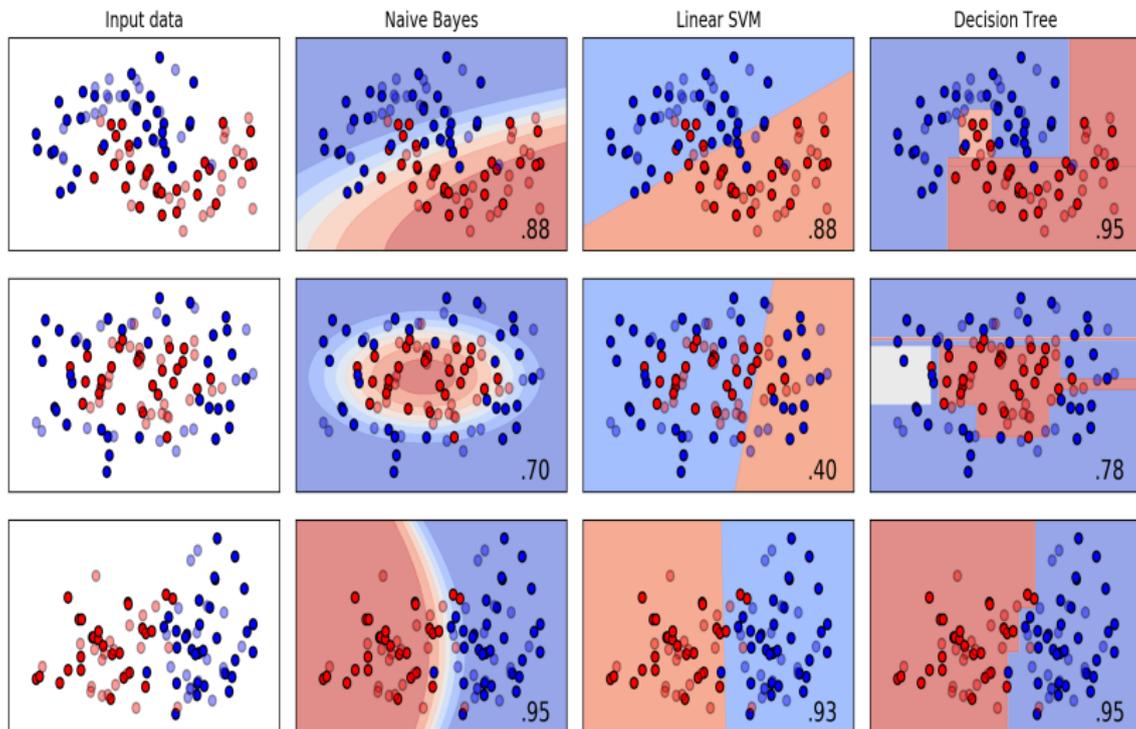
Information gain can be used to decide how to split



Results on the 2000 movie reviews:

	SVM	Naive Bayes	DTree (max depth 7)
Accuracy on train	0.98	0.96	0.80
Accuracy on test	0.84	0.8	0.69

Classifier comparison on sample data



Modified from SciKit Learn Classifier Comparison

Today

- Come to see lecturers if you are behind
- New topic starts on Monday—try to have ticks 1–6 by end of today