OxML: Clustering and Classification

Richard Willis

May 2, 2024

Outline

1. Intro
2. K-means
3. Naive Bayes Classifier

Intro

We will go through two Google Colab worksheets:

1. K-means, an example of unsupervised clustering
2. Naive Bayes, an example of supervised classification

Please check that you can assess and run them!

K-means

- Unsupervised learning [1] means that we have a dataset, which is unlabelled, and we wish to discover patterns and insights with it.
- *Clustering* algorithms are one example of unsupervised learning algorithms, which find groups of points that are more similar to each other than those in other groups, or clusters.
- The K-means algorithm is the most simple clustering algorithm, and it works by partitioning the feature space into n clusters. Amazing visualisation blog

[1] `https://cloud.google.com/discover/what-is-unsupervised-learning`

Algorithm

Steps

Fitting

- Randomly create *n* cluster centres
- Loop:
  1. Compute the distance of each point to each cluster centre
  2. Assign each point to its closest cluster centre
  3. Update the cluster centres to be the mean of the points assigned to them

Inference

1. Compute the distances to each cluster centre.
2. Assign the data point to the closest cluster centre.

Details

We can either continue until the cluster centres do not update (this will happen when no point has changed its assigned cluster), or terminate early, as long as no cluster centre moved by more than a tolerance value.

Naive Bayes Classifier

Used with categorical data for supervised learning

Bayes theorem

$$P(\text{Hypothesis} \mid \text{Data}) = \frac{P(\text{Data} \mid \text{Hypothesis}) \times P(\text{Hypothesis})}{P(\text{Data})} \tag{1}$$

- $P(\text{Hypothesis})$ is the prior probability of the hypothesis before observing the data.
- $P(\text{Data} \mid \text{Hypothesis})$ is the likelihood of the data given that the hypothesis is true.
- $P(\text{Hypothesis} \mid \text{Data})$ is the posterior probability of the hypothesis given the data
- $P(\text{Data})$ is the probability of observing the data under all possible hypotheses.

Example (1)

Weather	Temperature	Wind	Tennis
Sun	Hot	Strong	Yes
Rain	Hot	Mild	No
Cloudy	Cold	Mild	Yes
Sun	Cold	Mild	Yes

- Prior : $P(tennis = yes) = \frac{3}{4}$
- Conditional Probability : $P(weather = sun|tennis = yes) = \frac{2}{3}$
- *Naive* because we assume that the features are *conditionally independent*, so that

$$P(weather, temperature, wind|tennis) = P(weather|tennis)P(temperature|tennis)P(wind|tennis)$$

Without the assumption of conditional independence, we would need a vast amount of data. To work out $P(sun, hot, mild|yes)$, we would need to have observed this *exact* outcome multiple times. We therefore tend to use Naive Bayes even when this assumption is violated.

Conditional independence

The features are not *independent*: the weather is causally related to the temperature.
Are the features *conditionally independent*? Suppose I never play tennis when it is both raining and windy. Then $P(rain|yes)$ and $P(strong|yes)$ are not independent: if one is true then the other must be false.

What might conditional independence look like?

Weight	Likes baths	Animal
Light	No	Cat
Heavy	Yes	Dog
Heavy	No	Cat

Cats typically hate baths. But fat (heavy) cats are just as likely to hate baths as normal cats. Therefore, conditional on the animal, knowing the weight does not influence whether they enjoy bathing.

Example (2)

Table: Do I play tennis?

Weather	Temperature	Wind	Tennis
Sun	Hot	Strong	Yes
Rain	Hot	Mild	No
Cloud	Cold	Mild	Yes
Sun	Cold	Mild	Yes
Rain	Cold	Strong	???

$$P(tennis = yes|data) \propto P(yes)P(sun|yes)P(cold|yes)P(mild|yes)$$
$$= \frac{3}{4} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} = \frac{2}{9}$$

$$P(tennis = no|data) \propto P(no)P(sun|no)P(cold|no)P(mild|no)$$
$$= \frac{1}{4} \times \frac{0}{1} \times \frac{0}{1} \times \frac{1}{1} = 0$$

Therefore, $P(yes|data) = \frac{P(yes|data)}{P(yes|data)+P(no|data)} = \frac{\frac{2}{9}}{\frac{2}{9}+0} = 1$

Algorithm

Steps

Fitting

1. Compute the prior
2. Compute the feature likelihoods

Inference

1. For each class, multiply the prior probability by the likelihood, which is the product of the conditional feature probabilities
2. Select the class with the highest probability

Details

We use log probabilities, to avoid floating point precision errors.