Machine Learning - Lecture 3: k-means clustering

Chris Thornton

October 14, 2011

Introduction

Nearest-neighbour methods are a simple way of using clump structure for prediction.

They avoid the need to construct a model.

Unfortunately, they have a serious weakness.

As the number of data increases, the task of finding nearest neighbours gets harder.

Eventually, the time (or storage costs) involved may be unacceptable.

At this point, we have no choice but to revert to an approach based on explicit modeling.

Clustering

The model we obtain in a particular case all depends on the patterns we go looking for, and the way in which we then represent those patterns.

Methods which aim to find and represent basic clumps in the data are known as **clustering** methods.

An enormous number of these have been devised, many in the field of statistics.

We will focus on just two of the best known methods.

Agglomerative clustering

This method builds a tree structure representing the way in which datapoints clump together in a hierarchical way.

There are four main steps:

Agglomerative clustering

This method builds a tree structure representing the way in which datapoints clump together in a hierarchical way.

There are four main steps:

(1) Initialise clusters so each datapoint has its own cluster.

Agglomerative clustering

This method builds a tree structure representing the way in which datapoints clump together in a hierarchical way.

There are four main steps:

(1) Initialise clusters so each datapoint has its own cluster.

(2) Calculate the similarity of every pair of clusters by averaging similarities of their datapoints.

Agglomerative clustering

This method builds a tree structure representing the way in which datapoints clump together in a hierarchical way.

There are four main steps:

(1) Initialise clusters so each datapoint has its own cluster.

(2) Calculate the similarity of every pair of clusters by averaging similarities of their datapoints.

(3) Form a new cluster by combining the two most similar clusters.

Agglomerative clustering

This method builds a tree structure representing the way in which datapoints clump together in a hierarchical way.

There are four main steps:

(1) Initialise clusters so each datapoint has its own cluster.

(2) Calculate the similarity of every pair of clusters by averaging similarities of their datapoints.

(3) Form a new cluster by combining the two most similar clusters.

(4) Exit if just one cluster left. Otherwise repeat from step 2.

Agglomerative clustering

This method builds a tree structure representing the way in which datapoints clump together in a hierarchical way.

There are four main steps:

(1) Initialise clusters so each datapoint has its own cluster.

(2) Calculate the similarity of every pair of clusters by averaging similarities of their datapoints.

(3) Form a new cluster by combining the two most similar clusters.

(4) Exit if just one cluster left. Otherwise repeat from step 2.

Example

Let's say we have a dataset based on two variables: VL and TD.

VL	TD
17	1
18	6
29	0
16	3
17	4
22	7
15	16
16	0
19	2
14	3

Visualisation

With low-dimensional data we have the great advantage of being able to look at the data as a distribution of points.

```
29  x
28
27
26
25
24
23
22                                     x
21
20
19            x
18                        x       x
17       x              x
16  x              x
15                                                                                  x
14                 x
    0    1    2    3    4    5    6    7    8    9  10   11   12   13   14   15   16
```

Hierarchy produced by agglomerative clustering

```
|-------------------------- <15 16>
|--
    |---------------------- <29 0>
    |--
        |------------------ <22 7>
        |--
            |-------------- <18 6>
            |--
                |--
                |   |------ <14 3>
                |   |--
                |       |-- <16 3>
                |       |-- <17 4>
                |--
                    |------ <19 2>
                    |--
                        |-- <17 1>
                        |-- <16 0>
```

Non-hierarchical clustering

While cluster hierarchies provide interesting information about groupings at different levels of description, what we really want for prediction is a method that divides the data into a single set of groups.

This effect can be obtained using **k-means clustering**.

This is the most widely used, non-hierarchical clustering method.

k-means clustering algorithm

Let's say the data distribute into k clumps, and we want to know where these clumps are.

For this method, we use imaginary datapoints called **means** or **centroids** to represent clump centres.

Having initialized exactly k centroids to random positions, we then apply the following two steps:

k-means clustering algorithm

Let's say the data distribute into k clumps, and we want to know where these clumps are.

For this method, we use imaginary datapoints called **means** or **centroids** to represent clump centres.

Having initialized exactly k centroids to random positions, we then apply the following two steps:

(1) Each datapoint is assigned to its closest centroid.

k-means clustering algorithm

Let's say the data distribute into k clumps, and we want to know where these clumps are.

For this method, we use imaginary datapoints called **means** or **centroids** to represent clump centres.

Having initialized exactly k centroids to random positions, we then apply the following two steps:

(1) Each datapoint is assigned to its closest centroid.

(2) The position of each centroid is set to be the average of all the datapoints assigned to it.

k-means clustering algorithm

Let’s say the data distribute into k clumps, and we want to know where these clumps are.

For this method, we use imaginary datapoints called **means** or **centroids** to represent clump centres.

Having initialized exactly k centroids to random positions, we then apply the following two steps:

(1) Each datapoint is assigned to its closest centroid.

(2) The position of each centroid is set to be the average of all the datapoints assigned to it.

These two steps are then repeated as long as we see any change in the assignments.

The general effect is that the centroids ‘move’ rapidly to the k most densely populated parts of the dataspace.

Using centroids for prediction

Models based on centroids can be used for prediction by applying the nearest-neighbour approach.

For each centroid, we work out which classification is most common among its captured datapoints.

That classification is then associated with the entroid.

Then we just apply the nearest-neighbour rule but using centroids rather than datapoints.

Any new case is predicted to have the classification associated with its nearest *centroid*.

Demo

Demo

- Creating data: 1s, 0s, Xs

Demo

- Creating data: 1s, 0s, Xs
- Listing data

Demo

- Creating data: 1s, 0s, Xs
- Listing data
- What happens with a single centroid

Demo

- Creating data: 1s, 0s, Xs
- Listing data
- What happens with a single centroid
- What happens with more than one centroid

Demo

- Creating data: 1s, 0s, Xs
- Listing data
- What happens with a single centroid
- What happens with more than one centroid
- How do we find the right number of centroids?

Demo

- Creating data: 1s, 0s, Xs
- Listing data
- What happens with a single centroid
- What happens with more than one centroid
- How do we find the right number of centroids?
- Color coding for predictions

Demo

- Creating data: 1s, 0s, Xs
- Listing data
- What happens with a single centroid
- What happens with more than one centroid
- How do we find the right number of centroids?
- Color coding for predictions
- Use of unseen datapoints

Demo

- Creating data: 1s, 0s, Xs
- Listing data
- What happens with a single centroid
- What happens with more than one centroid
- How do we find the right number of centroids?
- Color coding for predictions
- Use of unseen datapoints

Summary

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.
- Eventually, the only alternative is to produce some form of explicit model.

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.
- Eventually, the only alternative is to produce some form of explicit model.
- Clustering methods aim to model the way data ‘clump’ together.

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.
- Eventually, the only alternative is to produce some form of explicit model.
- Clustering methods aim to model the way data ‘clump’ together.
- Agglomerative clustering produces a hierarchical model.

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.
- Eventually, the only alternative is to produce some form of explicit model.
- Clustering methods aim to model the way data ‘clump’ together.
- Agglomerative clustering produces a hierarchical model.
- k-means-clustering is a simple and efficient way of deriving a non-hierarchical model.

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.
- Eventually, the only alternative is to produce some form of explicit model.
- Clustering methods aim to model the way data ‘clump’ together.
- Agglomerative clustering produces a hierarchical model.
- k-means-clustering is a simple and efficient way of deriving a non-hierarchical model.
- Applying nearest-neighbour rules to cluster-centroids can be a way of generating predictions and classifications.

Summary

- As dataset size increases, nearest-neighbour methods get increasingly costly.
- Eventually, the only alternative is to produce some form of explicit model.
- Clustering methods aim to model the way data ‘clump’ together.
- Agglomerative clustering produces a hierarchical model.
- k-means-clustering is a simple and efficient way of deriving a non-hierarchical model.
- Applying nearest-neighbour rules to cluster-centroids can be a way of generating predictions and classifications.

Questions

Questions

- In k-means-clustering, why do centroids appear to ‘repel’ each other.

Questions

- In k-means-clustering, why do centroids appear to ‘repel’ each other.
- Estimate the amount of memory required in executing the k-means clustering algorithm.

Questions

- In k-means-clustering, why do centroids appear to ‘repel’ each other.
- Estimate the amount of memory required in executing the k-means clustering algorithm.
- On what assumptions can we say that that k-means-clustering performs induction?

Questions

- In k-means-clustering, why do centroids appear to ‘repel’ each other.
- Estimate the amount of memory required in executing the k-means clustering algorithm.
- On what assumptions can we say that that k-means-clustering performs induction?
- In a random distribution of positive and negative data-points, estimate the number of centroids needed to accurately model the data.

Questions

- In k-means-clustering, why do centroids appear to ‘repel’ each other.
- Estimate the amount of memory required in executing the k-means clustering algorithm.
- On what assumptions can we say that that k-means-clustering performs induction?
- In a random distribution of positive and negative data-points, estimate the number of centroids needed to accurately model the data.
- What sort of clump are we expecting when we use measurements of city-block distance for clustering?

Questions

- In k-means-clustering, why do centroids appear to ‘repel’ each other.
- Estimate the amount of memory required in executing the k-means clustering algorithm.
- On what assumptions can we say that that k-means-clustering performs induction?
- In a random distribution of positive and negative data-points, estimate the number of centroids needed to accurately model the data.
- What sort of clump are we expecting when we use measurements of city-block distance for clustering?