### Introduction to Unsupervised Learning and K Means

### Learning Objectives

* Explain the kinds of problems suitable for Unsupervised Learning approaches
* Describe the clustering process of the k-means algorithm
* Become familiar with k-means clustering syntax in scikit learn

Types of machine learning:

1. Supervised
   1. Data points have known outcome
2. Unsupervised
   1. Data points have unknown outcome

Types of unsupervised learning:

* Clustering - Id unknown structure in data
  + k-means
  + Hierarchical agglomerative clustering
  + DBSCAN
  + Mean shift
* Dimensionality reduction - Use structural characteristics to simplify data
  + Principal components analysis
  + Non-negative matrix factorization

Curse of dimensionality

* In theory, increasing features should improve performance
* In practice, too many features lead to worse performance.
* Number of training examples required increases exponentially with dimensionality

Common Clustering Use Cases:

* Classification
* Anomaly Detection
* Customer Segmentation
* Improve Supervised Learning

Introduction to clustering

Users of a web application:

* One feature (visits)
* Two clusters

k-means

- cluster

- Centroids

Which model is the right one

Smarter initialization of k-means clusters

* Pick one point at random as initial poin
* Pick next point with probability of an equation
* k-means++

Sometimes the question has a k:

* Clustering similar jobs on 4 cpu cores
* A clothing design in 10 diff. Sizes to cover most people (k=10)
* A navigation interace for browsing scientific papers with 20 disciplines (k=20)

Often, the number of clusters (k) is unclear, and we need an approach to select it.

Evaluating clustering performance

* Inertia: sum of squared distance from each point to its cluster
* Smaller value corresponds to tighter clusters
* Value sensitive to number of points in cluseter
* Distortion: avg. Of squared distance from each point to its cluster
* Smaller value corresponds to tighter clusters
* Doesn't generally increase as more points are added (relative to inertia)

Which model is the right one

Initiate mult. Times, and take the model with the best score.

Choosing the right number of clusters

* Inertia measures distance of point to cluster
* Value decreaeses with increasing K as long as cluster density increases
* Elbow

K means the syntax

From sklearn.cluster import kmeans

Kmeans= kmeans(n\_clusters=3, init=’k-means++)

Kmeans = kmeans.fit(X1)

Y\_predict = kmeans.predict(x2)

Elbow method syntax

Inertia = []

List\_clusters = list(range(10))

For k in list\_clusters:

K means = kmeans(n-clusters=k)

Kmeans.fit(x)

Inertia.append(km.

Kmean notebook pt.1

### **Unsupervised Learning Algorithms**

Unsupervised algorithms are relevant when we don’t have an outcome or labeled variable we are trying to predict.

They are helpful to find structures within our data set and when we want to partition our data set into smaller pieces.

Types of Unsupervised Learning:

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Unsupervised Learning | Data | Example | Algorithms |
| Clustering | Use unlabeled data, Identify unknown structure in data | Segmenting costumers into different groups | K-means, Hierarchical Agglomerative Clustering, DBSCAN, Mean shift |
| Dimensionality Reduction | Use structural characteristics to simplify data | Reducing size without losing too much information from our original data set | Principal Components Analysis, Non-negative Matrix, Factorization |

Dimensionality reduction is important in the context of large amounts of data.

**The Curse of Dimensionality**

In theory, a large number of features should improve performance. In theory, as models have more data to learn from, they should be more successful. But in practice, too many features lead to worse performance. There are several reasons why too many features end up leading to worse performance. If you have too many features, several things can be wrong, for example:

* Some features can be spurious correlations, which means they correlate into the data set but not outside your data set as long as new data comes in.
* Too many features create more noise than signal.
* Algorithms find hard to sort through non meaningful features if you have too many features.
* The number of training examples required increases exponentially with dimensionality.
* Higher dimensions slows performance.
* Larger data sets are computationally more expensive.
* Higher incidence of outliers.

To fix these problems in real life, it's best to reduce the dimension of the data set.

Similar to feature selection, you can use Unsupervised Machine Learning models such as Principal Components Analysis.

**Common uses of clustering cases in the real world**

1. Anomaly detection

Example: Fraudulent transactions.

Suspicious fraud patterns such as small clusters of credit card transactions with high volume of attempts, small amounts, at new merchants. This creates a new cluster and this is presented as an anomaly so perhaps there’s fraudulent transactions happening.

2. Customer segmentation

You could segment the customers by recency, frequency, average amount of visits in the last 3 months. Or another common type of segmentation is by demographics and the level of engagement, for example: single costumers, new parents, empty nesters, etc. And the combinations of each with the preferred marketing channel, so you can use these insights for future marketing campaigns.

3. Improve supervised learning

Logistic regressions per cluster, this means training one model for each segment of your data to try to improve classification.

**Common uses of Dimension Reduction in the real world**

1. Turn high resolution images into compressed images

This means to come to a reduced, more compact version of those images so they can still contain most of the data that can tell us what the image is about.

2. Image tracking

Reduce the noise to the primary factors that are relevant in a video capture. The benefits of reducing the data set can greatly speed up the computational efficiency of the detection algorithms.

### **K-means Clustering**

K-means clustering is an iterative process in which similar observations are grouped together. To do that, this algorithm starts by taking 2 random points known as centroids, and starts calculating the distance of each observation to the centroid, and assigning each cluster to the nearest centroid. After the first iteration every point belongs to a cluster.

Next, the number of centroids increases by one, and the centroid for each cluster are recalculated as the points with the average distance to all points in a given cluster. Then we keep repeating this process until no example is assigned to another cluster.

And this process is repeated k-times, hence the name k-means. This algorithm converges when clusters do not move anymore.

We can also create multiple clusters, and we can have multiple solutions, by multiple solutions we mean that the clusters are not going to move anymore (they converged) but we can converge in different places where we no longer move those centroids.

**Advantages and Disadvantages of K-Means**

The main advantage of k-means algorithm is that it is easy to compute. One disadvantage is that this algorithm is sensitive to the choice of the initial points, so different initial configurations may yield different results.

To overcome this, there is a smarter initialization of K-mean clusters called K-means ++, which helps to avoid getting stuck at local optima. This is the default implementation of the K-means.

**Model Selection, choosing *K* number of clusters**

Sometimes you want to split your data into a predetermined number of groups or segments. Often, the number of clusters (K) is unclear, and you need an approach to select it.

A common metric is ***Inertia***, defined as the sum of squares distance from each point to its cluster.

Smaller values of Inertia correspond to tighter clusters, this means that we are penalizing spread out clusters and rewarding tighter clusters to their centroids.

The draw back of this metric is that its value sensitive to number of points in clusters. The more points you add, the more you will continue penalizing the inertia of a cluster, even if those points are relatively closer to the centroids than the existing points.

Another metric is ***Distortion*** defined as the average of squared distance from each point to its cluster.

Smaller values of distortion corresponds to tighter clusters.

An advantage, is that distortion doesn’t generally increase as more points are added (relative to inertia). This means that It doesn’t increase distortion, as closer points will aid an actual decreasing the average distance.

**Inertia Vs. Distortion**

Both are measures of entropy per cluster.

Inertia will always increase as more members are added to each cluster, while this will not be the case with distortion.

When the similarity of the points in the cluster are very relevant, you should use distortion and if you are more concerned that clusters should have a similar number of points, then you should use inertia.

**Finding the right cluster**

To find the cluster with a low entropy metric, you run a few k-means clustering models with different initial configurations, compare the results, and determine which one of the different initializations of configurations lead to the lowest inertia or distortion.

|  |  |
| --- | --- |
| **Started on** | Friday, July 22, 2022, 2:50 PM |
| **State** | Finished |
| **Completed on** | Friday, July 22, 2022, 3:18 PM |
| **Time taken** | 27 mins 40 secs |
| **Grade** | **10.00** out of 10.00 (**100**%) |
| **Feedback** | Congratulations, you earned a passing score. |

### Question 1

Correct

1.00 points out of 1.00

Flag question

#### Question text

(True/False) K-means clustering algorithm relies on finding clusters centers to group data points based on minimizing the sum of square errors between each data point and its cluster centroid.

Select one:

True

False

#### Feedback

Correct! You can find more information in the lesson K-means notebook part 1.

The correct answer is 'True'.

### Question 2

Correct

1.00 points out of 1.00

Flag question

#### Question text

What’s the name of the default initialization for K-means?

Select one:

A.

K-means sum of square error

B.

K-means optimal.

C.

K-means inertia

D.

K-means ++

#### Feedback

Correct! You can find more information in the lesson K-means notebook part 1.

The correct answer is: K-means ++

### Question 3

Correct

1.00 points out of 1.00

Flag question

#### Question text

What is the implication of a small standard deviation of the clusters?

Select one:

A.

The standard deviation of the cluster defines how tightly around each one of the centroids are. With a small standard deviation, we can’t find any centroids.

B.

The standard deviation of the cluster defines how tightly around each one of the centroids are. With a small standard deviation, the points will be closer to the centroids.

C.

A small standard deviation of the clusters means that the centroids are not close enough to each other.

D.

A small standard deviation of the clusters defines the size of the clusters.

#### Feedback

Correct! You can find more information in the lesson K-means notebook part 2.

The correct answer is: The standard deviation of the cluster defines how tightly around each one of the centroids are. With a small standard deviation, the points will be closer to the centroids.

### Question 4

Correct

1.00 points out of 1.00

Flag question

#### Question text

After we plot our elbow and we find the inflection point, what does that point indicate to us?

Select one:

A.

The ideal number of clusters.

B.

The data points we need to form a cluster

C.

Whether we need to remove outliers.

D.

How we can reduce our number of clusters.

#### Feedback

Correct! You can find more information in the lesson K-means notebook part 2.

The correct answer is: The ideal number of clusters.

### Question 5

Correct

1.00 points out of 1.00

Flag question

#### Question text

(True/False) We can use K-means to reduce the size of high-quality images by just keeping the important information and grouping the colors with the right number of clusters.

Select one:

True

False

#### Feedback

Correct! You can find more information in the lesson K-means notebook part 3.

The correct answer is 'True'.

### Question 6

Correct

1.00 points out of 1.00

Flag question

#### Question text

What is one of the most suitable ways to choose K when the number of clusters is unclear?

Select one:

A.

By evaluating Clustering performance such as Inertia and Distortion.

B.

You can start by choosing a random number of clusters.

C.

You can start by using a k nearest neighbor method.

D.

By increasing the number of clusters calculating the square root.

#### Feedback

Correct! Both are measures of entropy part cluster. You can find more information in the lesson K-means part 3.

The correct answer is: By evaluating Clustering performance such as Inertia and Distortion.

### Question 7

Correct

1.00 points out of 1.00

Flag question

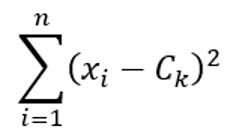
#### Question text

Which statement best describes the formula for Inertia?

Select one:

A.

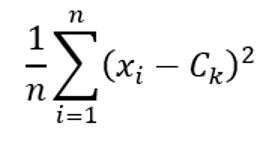
The Sum of squares distance from each point (xi) to its clusters (ck)



Correct! You can find more information in the lesson *K-means part 3.*

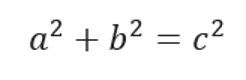
B.

Average of squared distance from each point (xi) to its cluster.



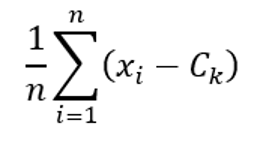
C.

The sum of *a* squared plus *b* squared equals *c* squared.

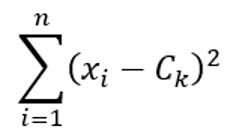


D.

Average of the distance from each point to its cluster.



#### Feedback

The correct answer is: The Sum of squares distance from each point (xi) to its clusters (ck)  


### Question 8

Correct

1.00 points out of 1.00

Flag question

#### Question text

Which statement describes correctly the use of distortion and inertia?

Select one:

A.

When the we can calculate a number of clusters higher than 10, we use distortion, when we calculate a number of clusters smaller than 10, we use inertia.

B.

When outliers are a concern use inertia, otherwise use distortion.

C.

When we the sum of the point equals a prime number use inertia, and when the sum of the point equals a pair number use distortion.

D.

When the similarity of the points in the cluster are more important you should use distortion and if you are more concern that clusters have similar numbers of points then you should use inertia.

#### Feedback

Correct! This statement describes best how we can choose between distortion and inertia. You can find more information in the lesson K-means part 3.

The correct answer is: When the similarity of the points in the cluster are more important you should use distortion and if you are more concern that clusters have similar numbers of points then you should use inertia.

### Question 9

Correct

1.00 points out of 1.00

Flag question

#### Question text

Select the approach that can help you find the cluster with best inertia

Select one:

A.

Compute the resulting inertia or distortion, keep the results, and see which one of the different initializations of configurations lead to the best inertia or distortion. As an example of this, the best inertia result is the **highest** value.

B.

Compute the resulting inertia or distortion, keep the results, and see which one of the different initializations of configurations lead to the best inertia or distortion. As an example of this, the best inertia result is the **lowest** value.

C.

Compute the resulting inertia or distortion, keep the results, and see which one of the different initializations of configurations lead to the best inertia or distortion. As an example of this, the best inertia result is the **median** value.

D.

Compute the resulting inertia or distortion, keep the results, and see which one of the different initializations of configurations lead to the best inertia or distortion. As an example of this, the best inertia result is the **average** value.

#### Feedback

Correct! You can find more information in the lesson K-means part 3.

The correct answer is: Compute the resulting inertia or distortion, keep the results, and see which one of the different initializations of configurations lead to the best inertia or distortion. As an example of this, the best inertia result is the **lowest** value.

### Question 10

Correct

1.00 points out of 1.00

Flag question

#### Question text

Which method is commonly used to select the right number of clusters?

Select one:

A.

The perfect Square Method

B.

The ROC curve.

C.

The elbow method.

Correct! The method consists of plotting the interpreted variation as a function of the number of clusters, and selecting the elbow of the curve as the number of clusters to use. You can find more information in the lesson K-means part 4.

D.

The Sum of Square Method

#### Feedback

The correct answer is: The elbow method.

## **Unsupervised Learning Algorithms**

Unsupervised algorithms are relevant when we don’t have an outcome or labeled variable we are trying to predict.

They are helpful to find structures within our data set and when we want to partition our data set into smaller pieces.

Types of Unsupervised Learning:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Unsupervised Learning** | **Data** | **Example** | **Algorithms** |
| Clustering | Use unlabeled data, Identify unknown structure in data | Segmenting costumers into different groups | K-means, Hierarchical Agglomerative Clustering, DBSCAN, Mean shift |
| Dimensionality Reduction | Use structural characteristics to simplify data | Reducing size without losing too much information from our original data set | Principal Components Analysis, Non-negative Matrix, Factorization |

Dimensionality reduction is important in the context of large amounts of data.

**The Curse of Dimensionality**

In theory, a large number of features should improve performance. In theory, as models have more data to learn from, they should be more successful. But in practice, too many features lead to worse performance. There are several reasons why too many features end up leading to worse performance. If you have too many features, several things can be wrong, for example:

- Some features can be spurious correlations, which means they correlate into the data set but not outside your data set as long as new data comes in.

- Too many features create more noise than signal.

- Algorithms find it hard to sort through non meaningful features if you have too many features.

- The number of training examples required increases exponentially with dimensionality.

- Higher dimensions slows performance.

- Larger data sets are computationally more expensive.

- Higher incidence of outliers.

To fix these problems in real life, it's best to reduce the dimension of the data set.

Similar to feature selection, you can use Unsupervised Machine Learning models such as Principal Components Analysis.

### **Common uses of clustering cases in the real world**

1. Anomaly detection

Example: Fraudulent transactions.

Suspicious fraud patterns such as small clusters of credit card transactions with high volume of attempts, small amounts, at new merchants. This creates a new cluster and this is presented as an anomaly so perhaps there’s fraudulent transactions happening.

2. Customer segmentation

You could segment the customers by recency, frequency, average amount of visits in the last 3 months. Another common type of segmentation is by demographics and the level of engagement, for example: single costumers, new parents, empty nesters, etc. And the combinations of each with the preferred marketing channel, so you can use these insights for future marketing campaigns.

3. Improve supervised learning

You can perform a Logistic regression for each cluster. This means training one model for each segment of your data to try to improve classification.

**Common uses of Dimension Reduction in the real world**

1. Turn high resolution images into compressed images

This means to come to a reduced, more compact version of those images so they can still contain most of the data that can tell us what the image is about.

2. Image tracking

Reduce the noise to the primary factors that are relevant in a video capture. The benefits of reducing the data set can greatly speed up the computational efficiency of the detection algorithms.

### **K-means Clustering**

K-means clustering is an iterative process in which similar observations are grouped together. To do that, this algorithm starts by taking 2 random points known as centroids, and starts calculating the distance of each observation to the centroid, and assigning each cluster to the nearest centroid. After the first iteration every point belongs to a cluster.

Next, the number of centroids increases by one, and the centroid for each cluster are recalculated as the points with the average distance to all points in a given cluster. Then we keep repeating this process until no example is assigned to another cluster.

And this process is repeated k-times, hence the name k-means. This algorithm converges when clusters do not move anymore.

We can also create multiple clusters, and we can have multiple solutions, by multiple solutions we mean that the clusters are not going to move anymore (they converged) but we can converge in different places where we no longer move those centroids.

**Advantages and Disadvantages of K-Means**

The main advantage of k-means algorithm is that it is easy to compute. One disadvantage is that this algorithm is sensitive to the choice of the initial points, so different initial configurations may yield different results.

To overcome this, there is a smarter initialization of K-mean clusters called K-means ++, which helps to avoid getting stuck at local optima. This is the default implementation of the K-means.

**Model Selection, choosing *K* number of clusters**

Sometimes you want to split your data into a predetermined number of groups or segments. Often, the number of clusters (K) is unclear, and you need an approach to select it.

A common metric is ***Inertia***, defined as the sum of squares distance from each point to its cluster centroid.

Smaller values of Inertia correspond to tighter clusters, this means that we are penalizing spread out clusters and rewarding clusters that are tighter to their centroids.

The draw back of this metric is that its value sensitive to number of points in clusters. The more points you add, the more you will continue penalizing the inertia of a cluster, even if those points are relatively closer to the centroids than the existing points.

Another metric is ***Distortion*** defined as the average of squared distance from each point to its cluster.

Smaller values of distortion corresponds to tighter clusters.

An advantage of distortion is that it doesn’t generally increase as more points are added (relative to inertia). This means that It doesn’t increase distortion, as closer points will actualyl decrease the average distance to the cluster centroid.

**Inertia Vs. Distortion**

Both Inertia and Distortion are measures of entropy per cluster.

Inertia will always increase as more members are added to each cluster, while this will not be the case with distortion.

When the similarity of the points in the cluster are very relevant, you should use distortion and if you are more concerned that clusters should have a similar number of points, then you should use inertia.

### **Finding the right cluster**

To find the cluster with a low entopy metric, you can run a few k-means clustering models with different initial configurations, compare the results, and determine which one of the different initializations of configurations lead to the lowest inertia or distortion.