ML_Chap4. Notes

Week1 - Intro to unsupervised ML and k-means (2022.1.30)

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.

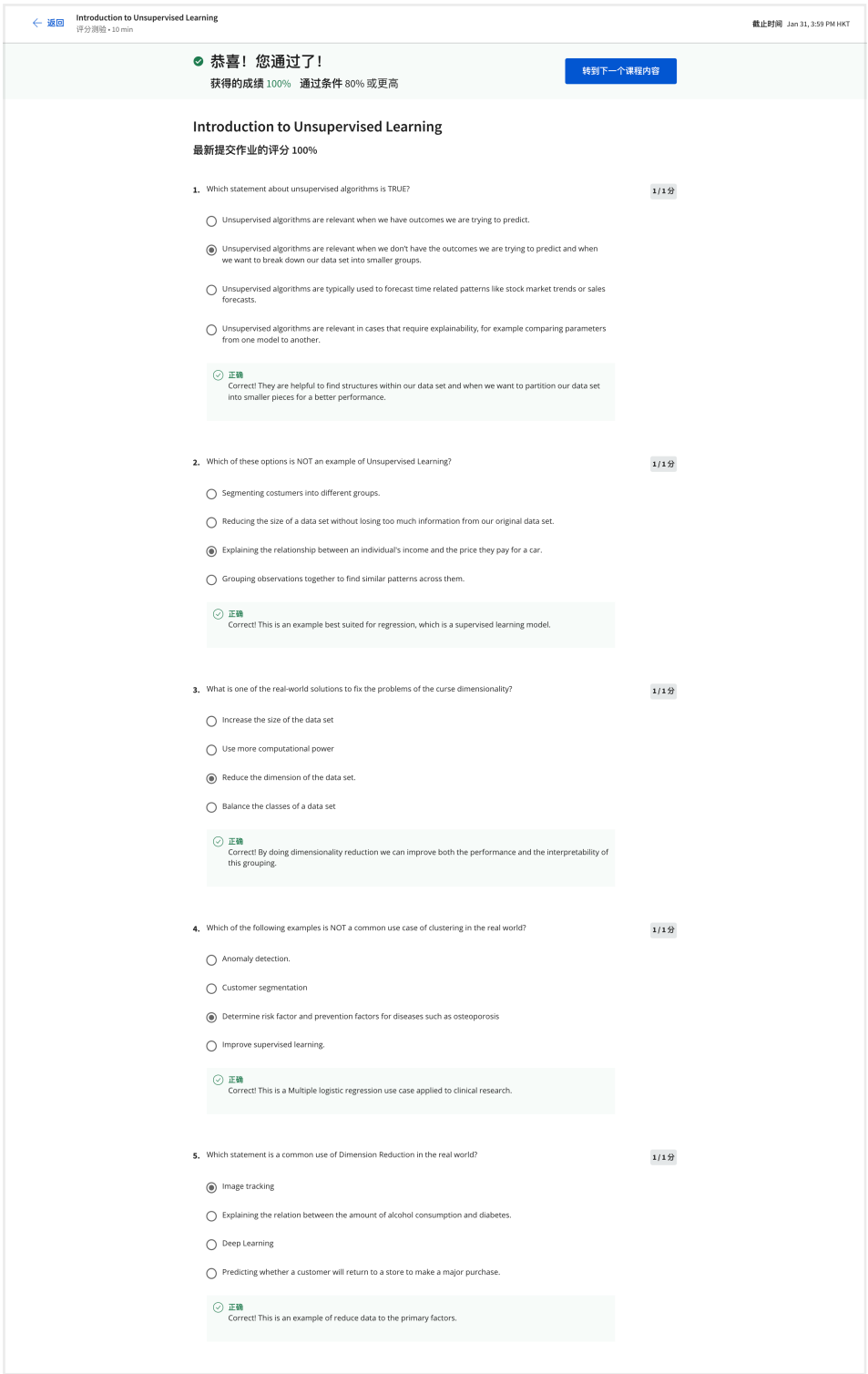
A common metric is ***Inertia***, defined as the sum of squares distance from each point to its cluster centroid.
Smaller values of Inertia correspond to tighter clusters, this means that we are penalizing spread out clusters and rewarding clusters that are tighter to their centroids.
The draw back of 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 actually 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 entropy 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.



返回 Introduction to Unsupervised Learning
评分测验 • 10 min
截止时间 Jan 31, 3:59 PM HKT

恭喜！您通过了！
获得的成绩 100% 通过条件 80% 或更高
转到下一个课程内容

Introduction to Unsupervised Learning
最新提交作业的评分 100%

1. Which statement about unsupervised algorithms is TRUE? 1 / 1 分

- Unsupervised algorithms are relevant when we have outcomes we are trying to predict.
- (selected) Unsupervised algorithms are relevant when we don't have the outcomes we are trying to predict and when we want to break down our data set into smaller groups.
- Unsupervised algorithms are typically used to forecast time related patterns like stock market trends or sales forecasts.
- Unsupervised algorithms are relevant in cases that require explainability, for example comparing parameters from one model to another.

正确
Correct! They are helpful to find structures within our data set and when we want to partition our data set into smaller pieces for a better performance.

2. Which of these options is NOT an example of Unsupervised Learning? 1 / 1 分

- Segmenting costumers into different groups.
- Reducing the size of a data set without losing too much information from our original data set.
- (selected) Explaining the relationship between an individual's income and the price they pay for a car.
- Grouping observations together to find similar patterns across them.

正确
Correct! This is an example best suited for regression, which is a supervised learning model.

3. What is one of the real-world solutions to fix the problems of the curse dimensionality? 1 / 1 分

- Increase the size of the data set
- Use more computational power
- (selected) Reduce the dimension of the data set.
- Balance the classes of a data set

正确
Correct! By doing dimensionality reduction we can improve both the performance and the interpretability of this grouping.

4. Which of the following examples is NOT a common use case of clustering in the real world? 1 / 1 分

- Anomaly detection.
- Customer segmentation
- (selected) Determine risk factor and prevention factors for diseases such as osteoporosis
- Improve supervised learning.

正确
Correct! This is a Multiple logistic regression use case applied to clinical research.

5. Which statement is a common use of Dimension Reduction in the real world? 1 / 1 分

- (selected) Image tracking
- Explaining the relation between the amount of alcohol consumption and diabetes.
- Deep Learning
- Predicting whether a customer will return to a store to make a major purchase.

正确
Correct! This is an example of reduce data to the primary factors.

恭喜！您通过了！

获得的成绩 80% 通过条件 80% 或更高

转到下一个课程内容

K Means Clustering

最新提交作业的评分 80%

1. (True/False) Is the following statement True or False? 1 / 1 分

"We initialize our K-means algorithm by taking 2 random points and these points are going to act as the centroids".

- (●) True
- ( ) False

正确
Correct! We initialize the algorithm by taking 2 random points and these are going to act as the centroids, with our centroids initiated we determinate to which cluster belongs to, by computing the distance to the nearest centroid and seeing which one is closer. You can find more information in the lesson *K-means part 1.*

2. Which of the following statements best describes the iterative part of the K-means algorithm? 1 / 1 分

- ( ) The k-means algorithm assigns a number of clusters at random.
- (●) The k-means algorithm adjusts the centroids to the new mean of each cluster, and then it keeps repeating this process until no example is assigned to another cluster.
- ( ) The k-means algorithm iteratively deletes outliers.
- ( ) The k-means algorithm iteratively calculates the distance from each point to the centroid of each cluster.

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

3. (True/False) Is the following statement True or False? 0 / 1 分

"The problem with K-means algorithm is that is sensitive to the choice of the initial points, so different initial configurations may yield different results".

- (●) False
- ( ) True

错误
Feedback: Incorrect. This is a true statement. Please review the lesson *K-means part 1.*

4. Which statement describes better "The Smarter initialization of K-mean clusters? 1 / 1 分

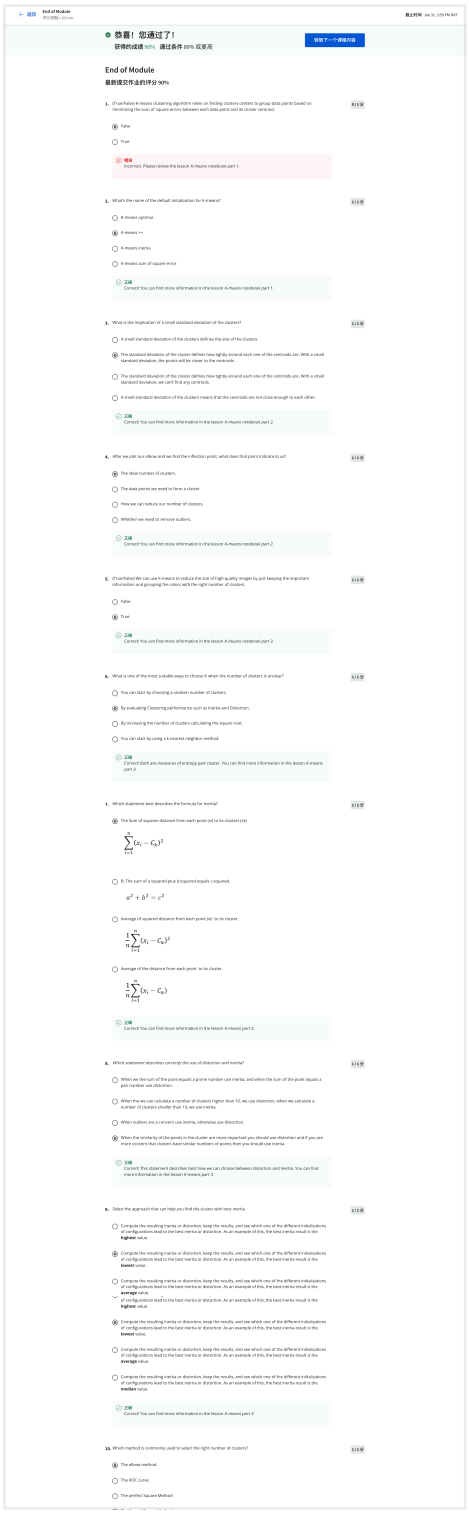
- ( ) "Draw a line between the data points to create 2 big clusters."
- ( ) "After we find our centroids, we calculate the distance between all our data points."
- (●) "Pick one point random as initial point and for the second pick instead of doing it randomly we prioritize by assigning the probability of the distance."
- ( ) "We start by having two centroids as far as possible between each other."

正确
Correct! This one defines it and remember: The smarter initialization of K-mean clusters is called, K-means ++, and it helps to avoid getting stuck at these local optima. This is the default implementation of the K-means. You can find more information in the lesson *K-means part 2.*

5. What happen with our second cluster centroid when we use the probability formula? 1 / 1 分

- ( ) When we use the probability formula, we put less weight on the points that are far away. So, our second cluster centroid is likely going to be closer.
- (●) When we use the probability formula, we put more weight on the points that are far away. So, our second cluster centroid is likely going to be more distant.
- ( ) When we use the probability formula, we put more weight on the lighter centroids, because it will take more computational power to draw our clusters. So, the second cluster centroid is likely going to be less distant.
- ( ) When we use the probability formula, we put less weight on the points that are far away. So, our second cluster centroid is likely going to be more distant.

正确
Correct! This happens because it will take a larger proportion of the total distance square of all our points. You can find more information in the lesson *K-means part 2.*



Week2 - Clustering (2022.2.7)

Distance Metrics

Clustering methods rely very heavily on our definition of distance. Our choice of Distance Metric will be extremely important when discussing our clustering algorithms and to clustering success.

Each metric has strengths and most appropriate use cases, but sometimes choosing a distance metric is also based on empirical evaluation to determine

which metric works best to achieve our goals.

These are the most common distance metrics:

Euclidean Distance

This one is the most intuitive distance metric, and that we use in K-means, another name for this is the L2 distance. You probably remember from your trigonometry classes.

We calculate (d) by taking the square root of the square of each of this changes (values). We can move this to higher dimensions for example 3 dimensions, 4 dimensions etc. In general, for an n-dimensional space, the distance is:

$$d(p,q) = \sqrt{(P_1 - q_I)^2 + (P_2 - q_2)^2 + \cdots (p_i + q_i)^2 + \cdots (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (P_1 - q_i)^2}$$

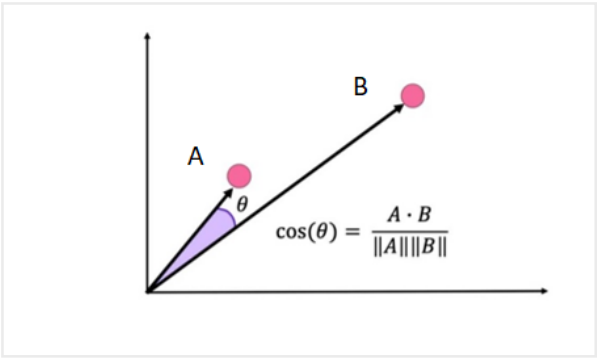
Manhattan Distance (L1 or City Block)

Another distance metric is the L1 distance or the Manhattan distance, and instead of squaring each term we are adding up the absolute value of each term. It will always be larger than the L2 distance, unless they lie on the same axis. We use this in business cases where there is very high dimensionality.

As high dimensionality often leads to difficulty in distinguishing distances between one point and the other, the L1 score does better than the L2 score in distinguishing these different distances once we move into a higher dimensional space.

Cosine Distance

This is a bit less intuitive distance metric. What we really care about the Cosine Distance is the angle between 2 points, for example, for two given points A and B:



This metric gives us the cosine of the angle between the two vectors defined from the origin to two given points in a two-dimensional space. To translate this definition into higher dimensions, we take the dot product of the vectors and divide it by the norm of each point.

The key to the Cosine distance is that it will remain insensitive to the scaling with respect to the origin, which means that we can move some of the points along the same line and the distance will remain the same. So, any two points on that same array, passing through the origin will have a distance of zero from

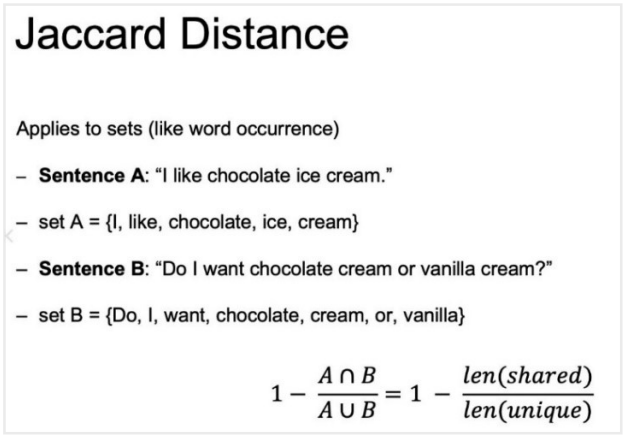
one another.

Euclidean VS Cosine distances

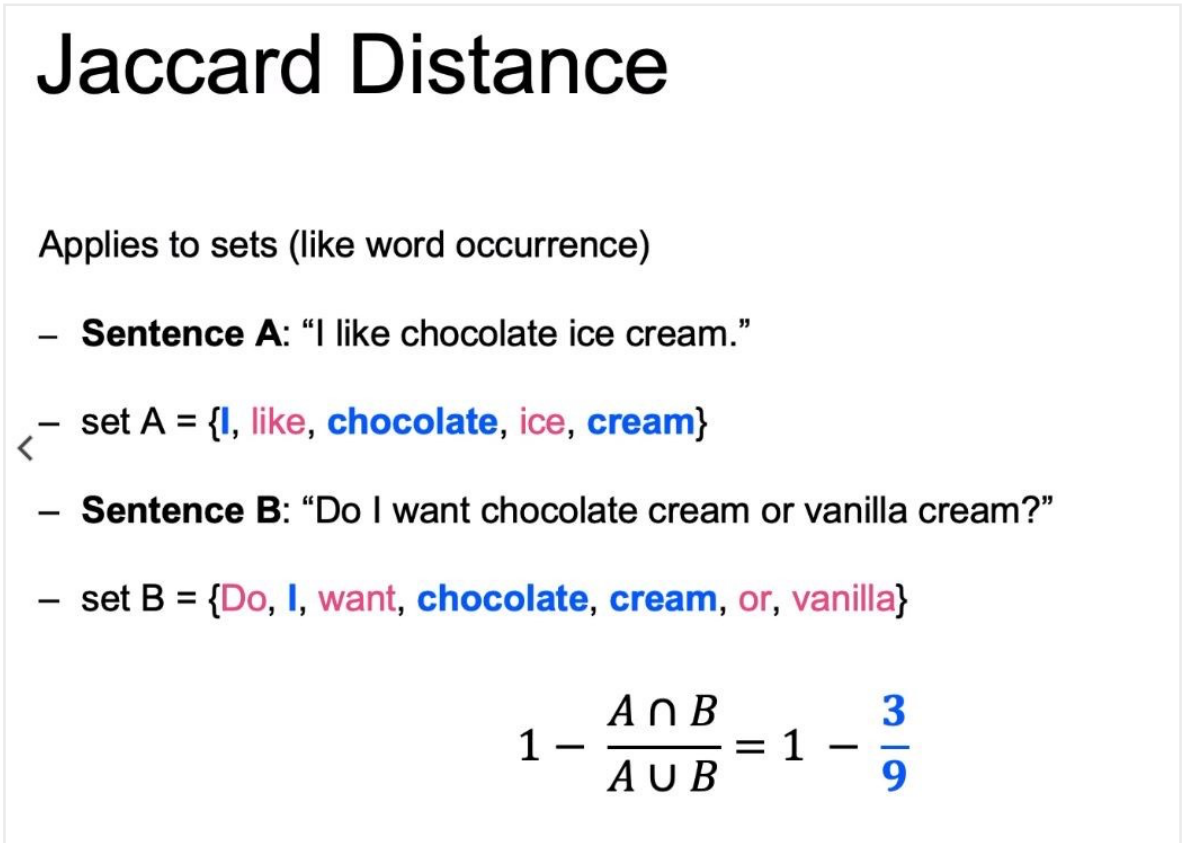
- Euclidean distance is useful for coordinate based measurements.
- Euclidean distance is more sensitive to curse of dimensionality
- Cosine is better for data such as text where location of occurrence is less important.

Jaccard Distance

This distance is useful for texts and is often used to word occurrence. Consider the following example:



In this case, the Jaccard Distance is going to be one minus the amount of value shared. So, the intersection over that union. This intersection means, the shared values of the two sentences over the length of the total unique values between sentecnes A and B.



It can be useful in cases you have text documents and you want to group

similar topics together.

Hierarchical Clustering
This clustering algorithm, will try to continuously split out and merge new clusters successively until it reaches a level of convergence.
This algorithm identifies first the pair of points which has the minimal distance and it turns it into the first cluster, then the second pair of points with the second minimal distance will form the second cluster, and so on. As the algorithm continues doing this with all the pairs of closest points, we can turn our points into just one cluster, which is why HAC also needs a stopping criterion.

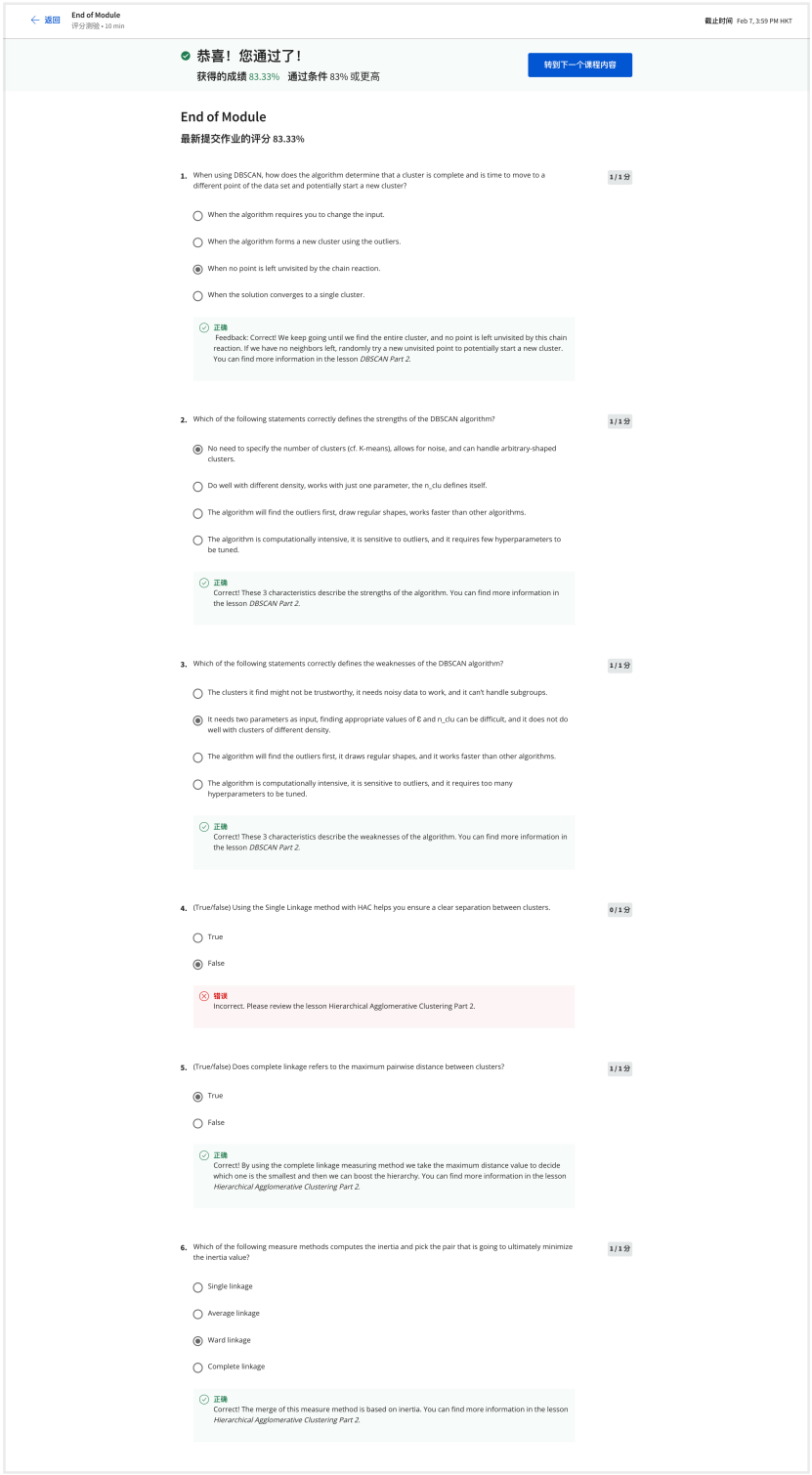
There are a few linkage types or methods to measure the distance between clusters. these are the most common:
Single linkage: minimum pairwise distance between clusters.
It takes the distance between specific points and declare that as the distance between 2 clusters and then it tries to find for all these pairwise linkages which one is the minimum and then we will combine those together as we move up to a higher hierarchy.
Pros: It helps ensuring a clear separation between clusters.
Cons: It won't be able to separate out cleanly if there is some noise between 2 different clusters.
Complete linkage: maximum pairwise distance between clusters.
Instead of taking the minimum distance given the points within each cluster, it will take the maximum value. Then from those maximum distances it decides which one is the smallest and then we can move up that hierarchy.
Pro: It would do a much better job of separating out the clusters if there's a bit of noise or overlapping points of two different clusters.
Cons: Tends to break apart a larger existing cluster depending on where that maximum distance of those different points may end up lying
Average linkage: Average pairwise distance between clusters.
Takes the average of all the points for a given cluster and use those averages or clusters centroids to determine the distance between the different clusters.
Pros: The same as the single and complete linkage.
Cons: It also tends to break apart a larger existing cluster.
Ward linkage: Cluster merge is based on inertia.
Computes the inertia for all pairs of points and picks the pair that will ultimately minimizes the value of inertia.

The pros and cons are the same as the average linkage.

Syntax for Agglomerative Clusters
First, import AgglomarativeClustering
From sklearn.cluster import AgglomerativeClustering
then create an instance of class,
agg = AgglomerativeClustering (n_clusters=3, affinity='euclidean', linkage='ward')

and finally, fit the instance on the data and then predict clusters for new data

```
agg=agg.fit(X1)
y_predict=agg.predict(X2)
```



Week3 Dimensionality reduction (2022.2.11)

Non Negative Matrix Decomposition

Non Negative Matrix Decomposition is another way of reducing the number of

dimensions.
Similar to PCA, it is also a matrix decomposition method in the form V=WxH.

The main difference is that it can only be applied to matrices that have positive values as inputs, for example:

- pixels in a matrix
- positive attributes that can be zero or higher

In the case of word and vocabulary recognition, each row in the matrix can be considered a document, while each column can be considered a topic.

NMF has proven to be powerful for:

- **word and vocabulary recognition**
- **image processing,**
- **text mining**
- **transcribing**
- **encoding and decoding**
- **decomposition of video, music, or images**

There are advantages and disadvantages of only dealing with non negative values.

- An advantage, is that NMF leads to features that tend to be more interpretable. For example, in facial recognition, the decomposed components match to something more interpretable like, for example, the nose, the eyebrows, or the mouth.

- A disadvantage is that NMF truncates negative values by default to impose the added constraint of only positive values. This truncation tends to lose more information than other decomposition methods.

Unlike PCA, it does not have to use orthogonal latent vectors, and can end up using vectors that point in the same direction.

NMF for NLP
In the case of Natural Language Processing, NMF works as below given these inputs, parameters to tune, and outputs:
Inputs
Given vectorized inputs, which are usually pre-processed using count vectorizer or vectorizers in the form Term Frequency - Inverse Document Frequency (TF-IDF).
Parameters to tune
The main two parameters are:

- Number of Topics
- Text Preprocessing (stop words, min/max document frequency, parts of speech, etc)

Output
The output of NMF will be two matrices:

1. W Matrix telling us how the terms relate to the different topics.
2. H Matrix telling us how to use those topics to reconstruct our original documents.

Syntax

The syntax consists of importing the class containing the clustering method:

from sklearn.decomposition import NMF

creating the instance of the class:

nmf=NMF(n_components=3, init='random')

and fit the instance and create a transformed version of the data:

x_nmf=NMF.fit(X)

准备好后再次尝试

获得的成绩 25%　通过条件 75% 或更高

再试

Non Negative Matrix Factorization

最新提交作业的评分 25%

1. (True/False) In some applications, NMF can make for more human interpretable latent features. 1 / 1 分

- (●) True
- ( ) False

正确
Correct! You can find more information in the Non Negative Matrix Factorization lesson.

2. Which of the following set of features is the least adapted to NMF? 0 / 1 分

- ( ) Word Count of the different words present in a text.
- (●) Pixel color values of a an Image.
- ( ) Spectral decomposition of an audio file.
- ( ) Monthly returns of a set of stock portfolios.

错误
Incorrect. Please review the Non Negative Matrix Factorization lesson.

3. (True/False) The NMF can produce different outputs depending on its initialization. 0 / 1 分

- (●) True
- ( ) False

错误
Incorrect. Please review the Non Negative Matrix Factorization lesson.

4. Which option is the dense representation of the matrix below? 0 / 1 分

[(1, 1, 2), (1, 2, 3), (3, 4, 1), (2, 4, 4), (4, 3, 1)]

- (●) [[2 0 0 0],
  [3 0 0 0],
  [0 0 0 1],
  [0 4 1 0]]
- ( ) [[0 0 0 1],
  [0 2 0 0],
  [0 0 0 3],
  [0 4 1 0]]
- ( ) [[1 0 0 0],
  [0 3 0 0],
  [0 2 0 0],
  [0 0 4 2]]
- ( ) [[0 0 0 2],
  [0 3 4 0],
  [0 0 0 0],
  [0 0 1 0]]

错误
Incorrect. Please review the Non Negative Matrix Factorization lesson.

Dimensionality Reduction: Approaches

Dimensionality reduction is common across a wide range of applications

Some rules of thumb for selecting an approach:

Method	Use case
Principal Components Analysis (PCA)	Identify small number of transformed variables with different effects, preserving variance
Kernel PCA	Useful for situations with nonlinear relationships, but requires more computation than PCA
Multidimensional Scaling	Like PCA, but new (transformed features) are determined based on preserving distance between points, rather than explaining variance
Non-negative Matrix Factorization	Useful when you want to consider only positive values (word matrices, images)