



程序代写
作业CS编程辅导

COMP4121 Advanced Algorithms

WeChat: [estudante](#)

Assignment Project Exam Help

Aleks Ignjatović
Email: tutorcs@163.com

School of Computer Science and Engineering
University of New South Wales

<https://tutorcs.com>
Clustering algorithms

What is clustering?

程序代写代做 CS编程辅导



- Fundamentally important to data science

- Making sense of data **WeChat: cstutorcs**

- Data preprocessing for **Assignment Project Exam Help**

- It is a type of *unsupervised learning* **Email: tutorcs@163.com**

QQ: 749389476

<https://tutorcs.com>

What is clustering?

程序代写代做 CS编程辅导



- Fundamentally important to data science

- Making sense of data on its own

- Data preprocessing for Assignment Project Exam Help

- It is a type of *unsupervised learning*

QQ: 749389476

<https://tutorcs.com>

What is clustering?

程序代写代做 CS编程辅导



- Fundamentally important to data science
- Making sense of data on its own
- Data preprocessing for other algorithms
- It is a type of *unsupervised learning*

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

What is clustering?

程序代写代做 CS编程辅导



- Fundamentally important science
- Making sense of data on its own
- Data preprocessing for other algorithms
- It is a type of *unsupervised learning*

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

How many clusters are there?

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

How many clusters are there?

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

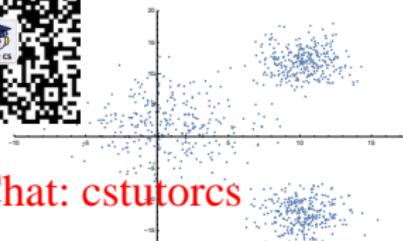
<https://tutorcs.com>

What are clusters?

程序代写代做 CS编程辅导

- Two kinds of clusters:

- center - based clustering



WeChat: cstutorcs

Assignment Project Exam Help

- high density clusters

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



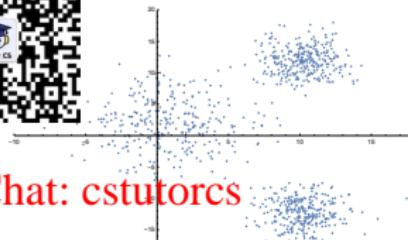
- A good clustering algorithm should be able to handle both kinds.

What are clusters?

程序代写代做 CS编程辅导

- Two kinds of clusters:

- ① center - based c



WeChat: cstutorcs

Assignment Project Exam Help

- ② high density clusters

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



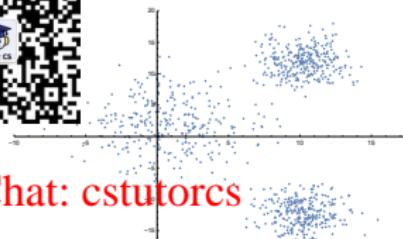
- A good clustering algorithm should be able to handle both kinds.

What are clusters?

程序代写代做 CS编程辅导

- Two kinds of clusters:

- ① center - based c



WeChat: cstutorcs

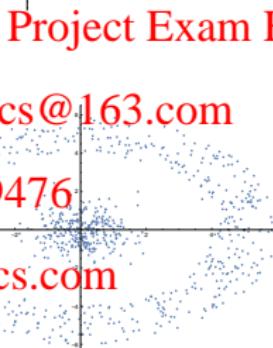
Assignment Project Exam Help

- ② high density clusters

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



- A good clustering algorithm should be able to handle both kinds.

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:
 - ① as vectors in \mathbb{R}^d
 - This is suitable if you have several numerical measurements of each object, like red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
 - Another example might be the relative frequencies of the key words in each document from a collection.
 - Note that d can be extremely large, corresponding to thousands or more of possible keywords.
 - This can be a problem when dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)
 - ② as a weighted graph
 - Data points are represented as vertices of the graph;
 - The weights of the edges reflect the degree of similarity (or dissimilarity) between data points.
 - Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.



WeChat: cstutorcs

Assignment Project Exam Help

Ring and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object. For example red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.



- Another example might be the relative frequencies of the key words in each document from a collection.

- Note that d can be extremely large, corresponding to thousands or more of possible keywords.

- This can be a problem when dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)

WeChat: cstutorcs

Assignment Project Exam Help
Email: tutorcs@163.com

② as a weighted graph

- Data points are represented as vertices of the graph;

- The weights of the edges reflect the degree of similarity (or dissimilarity) between data points

QQ: 749389476
<https://tutorcs.com>

- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object. For example red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.



- Another example might be the relative frequencies of the key words in each document from a collection.
- Note that d can be extremely large, corresponding to thousands or more of possible keywords.

- This can be a problem when dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

② as a weighted graph

- Data points are represented as vertices of the graph;

- The weights of the edges reflect the degree of similarity (or dissimilarity) between data points

QQ: 749389476

https://tutorcs.com

- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:
 - ① as vectors in \mathbb{R}^d
 - This is suitable if you have several numerical measurements of each object. For example, red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
 - Another example might be the relative frequencies of the key words in each document from a collection.
 - Note that d can be extremely large, corresponding to thousands or more of possible keywords.
 - This can be a problem when dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)
 - ② as a weighted graph
 - Data points are represented as vertices of the graph;
 - The weights of the edges reflect the degree of similarity (or dissimilarity);
 - Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.



WeChat: cstutorcs

Assignment Project Exam Help

Ring and handling
such high dimensional data. (this is where Johnson - Lindenstrauss
Theorem comes into play)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:
 - ① as vectors in \mathbb{R}^d
 - This is suitable if you have several numerical measurements of each object, e.g. red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
 - Another example might be the relative frequencies of the key words in each document from a collection.
 - Note that d can be extremely large, corresponding to thousands or more of possible keywords.
 - This can be a problem when dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)
 - ② as a weighted graph
 - Data points are represented as vertices of the graph;
 - The weights of the edges reflect the degree of similarity (or dissimilarity);
 - Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

程序代写代做 CS 编程辅导

①



WeChat: cstutorcs

Assignment Project Exam Help

Ring and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)

Email: tutorcs@163.com

②

as a weighted graph

QQ: 749389476

https://tutorcs.com

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:
 - ① as vectors in \mathbb{R}^d
 - This is suitable if you have several numerical measurements of each object, e.g. red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
 - Another example might be the relative frequencies of the key words in each document from a collection.
 - Note that d can be extremely large, corresponding to thousands or more of possible keywords.
 - This can be a problem when dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)
 - ② as a weighted graph
 - Data points are represented as vertices of the graph;
 - The weights of the edges reflect the degree of similarity (or dissimilarity);
 - Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

程序代写代做 CS 编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Handling and dealing with such high dimensional data. (this is where Johnson - Lindenstrauss Theorem comes into play)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object. For example, red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
- Another example might be the relative frequencies of the key words in each document from a collection.
- Note that d can be extremely large, corresponding to thousands or more of possible keywords.
- This can be a problem due to complexities of storing and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem come to play)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

② as a weighted graph

- Data points are represented as vertices of the graph;
- The weights of the edges reflect the degree of similarity (or dissimilarity) between data points.
- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object, e.g. red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
- Another example might be the relative frequencies of the key words in each document from a collection.
- Note that d can be extremely large, corresponding to thousands or more of possible keywords.
- This can be a problem due to complexities of storing and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem come to play)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

② as a weighted graph

- Data points are represented as vertices of the graph;
- The weights of the edges reflect the degree of similarity (or dissimilarity);
- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object. For example, red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
- Another example might be the relative frequencies of the key words in each document from a collection.
- Note that d can be extremely large, corresponding to thousands or more of possible keywords.
- This can be a problem due to complexities of storing and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem come to play)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

② as a weighted graph

- Data points are represented as vertices of the graph;
- The weights of the edges reflect the degree of similarity (or dissimilarity) between data points
- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object. For example, red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.
- Another example might be the relative frequencies of the key words in each document from a collection.
- Note that d can be extremely large, corresponding to thousands or more of possible keywords.
- This can be a problem due to complexities of storing and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem come to play)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

② as a weighted graph

- Data points are represented as vertices of the graph;
- The weights of the edges reflect the degree of similarity (or dissimilarity) of the data points.
- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

QQ: 749389476

<https://tutorcs.com>

Data representation

- We have to make sure that the data is adequately represented.
- In general, there are two most common representations:

① as vectors in \mathbb{R}^d

- This is suitable if you have several numerical measurements of each object, e.g. red blood cell count, white blood cell count, haemoglobin etc for each patient in a group of patients.



- Another example might be the relative frequencies of the key words in each document from a collection.
- Note that d can be extremely large, corresponding to thousands or more of possible keywords.

- This can be a problem due to complexities of storing and handling such high dimensional data. (this is where Johnson - Lindenstrauss Theorem come to play)

② as a weighted graph

- Data points are represented as vertices of the graph;
- The weights of the edges reflect the degree of similarity (or dissimilarity) of the data points.
- Data represented as vectors in \mathbb{R}^d can be represented as a weighted graph where the weights of edges reflect dissimilarity of the end points as measured by some form of distance between the end points.

Data representation

- The distance between two data points $x, y \in \mathbb{R}^d$ can be defined as either

$$d(x, y) = \sum_{i=1}^d \text{QR code} \quad \text{or as} \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- Such a distance $d(x, y)$ is taken as a measure of dissimilarity of x and y .
- If the scales of the i^{th} components x_i and x_j differ significantly, or if they are not of equal importance, we might consider instead

Assignment Project Exam Help

$d(x, y)^2 = \sum_{i=1}^d w_i (x_i - y_i)^2$
Email: tutorcs@163.com

- The weights w_i are chosen to reflect different variances of x_i and x_j or to encode their relative significances.
- Graph representation of data is often much a more compact representation than as vectors in \mathbb{R}^d , which does not suffer from problems of high dimensionality.
- Note that in the graph representation of data the geometry of the data points is lost, so the clustering is based only on mutual distances of pairs of points, which is good when clustering is not center-based.

Data representation

- The distance between two data points $x, y \in \mathbb{R}^d$ can be defined as either

$$d(x, y) = \sum_{i=1}^d \text{QR code} \quad \text{or as} \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- Such a distance $d(x, y)$ taken as a measure of dissimilarity of x and y .
- If the scales of the i^{th} WeChat: cstutorcs x_i and x_j differ significantly, or if they are not of equal importance, we might consider instead

Assignment Project Exam Help

$d(x, y)^2 = \sum_{i=1}^d w_i (x_i - y_i)^2$
Email: tutorcs@163.com

- The weights w_i are chosen to encode different variances of x_i and x_j or to encode their relative significances.
- Graph representation of data is often much a more compact representation than as vectors in \mathbb{R}^d , which does not suffer from problems of high dimensionality.
- Note that in the graph representation of data the geometry of the data points is lost, so the clustering is based only on mutual distances of pairs of points, which is good when clustering is not center-based.

Data representation

- The distance between two data points $x, y \in \mathbb{R}^d$ can be defined as either

$$d(x, y) = \sum_{i=1}^d \text{QR code} \quad \text{or as} \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- Such a distance $d(x, y)$ taken as a measure of dissimilarity of x and y .
- If the scales of the i^{th} and j^{th} coordinates x_i and x_j differ significantly, or if they are not of equal importance, we might consider instead

Assignment Project Exam Help

$$d(x, y)^2 = \sum_{i=1}^d w_i (x_i - y_i)^2$$

Email: tutorcs@163.com

- The weights w_i are chosen to reflect different variances of x_i and x_j or to encode their relative significances.
- Graph representation of data is often much a more compact representation than as vectors in \mathbb{R}^d , which does not suffer from problems of high dimensionality.
- Note that in the graph representation of data the geometry of the data points is lost, so the clustering is based only on mutual distances of pairs of points, which is good when clustering is not center-based.

Data representation

- The distance between two data points $x, y \in \mathbb{R}^d$ can be defined as either

$$d(x, y) = \sum_{i=1}^d \text{QR code} \quad \text{or as} \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- Such a distance $d(x, y)$ taken as a measure of dissimilarity of x and y .
- If the scales of the i^{th} and j^{th} coordinates x_i and x_j differ significantly, or if they are not of equal importance, we might consider instead

Assignment Project Exam Help

$$d(x, y)^2 = \sum_{i=1}^d w_i (x_i - y_i)^2$$

Email: tutorcs@163.com

- The weights w_i are chosen to normalize different variances of x_i and x_j or to encode their relative significances.
- Graph representation of data is often much a more compact representation than as vectors in \mathbb{R}^d , which does not suffer from problems of high dimensionality.
- Note that in the graph representation of data the geometry of the data points is lost, so the clustering is based only on mutual distances of pairs of points, which is good when clustering is not center-based.

Data representation

- The distance between two data points $x, y \in \mathbb{R}^d$ can be defined as either

$$d(x, y) = \sum_{i=1}^d \text{QR code} \quad \text{or as} \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- Such a distance $d(x, y)$ taken as a measure of dissimilarity of x and y .
- If the scales of the i^{th} and j^{th} coordinates x_i and x_j differ significantly, or if they are not of equal importance, we might consider instead

Assignment Project Exam Help

$$d(x, y)^2 = \sum_{i=1}^d w_i (x_i - y_i)^2$$

Email: tutorcs@163.com

- The weights w_i are chosen to normalize different variances of x_i and x_j or to encode their relative significances.
- Graph representation of data is often much a more compact representation than as vectors in \mathbb{R}^d , which does not suffer from problems of high dimensionality.
- Note that in the graph representation of data the geometry of the data points is lost, so the clustering is based only on mutual distances of pairs of points, which is good when clustering is not center-based.

Data representation

- The distance between two data points $x, y \in \mathbb{R}^d$ can be defined as either

$$d(x, y) = \sum_{i=1}^d \text{QR code} \quad \text{or as} \quad d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

- Such a distance $d(x, y)$ taken as a measure of dissimilarity of x and y .
- If the scales of the i^{th} and j^{th} coordinates x_i and x_j differ significantly, or if they are not of equal importance, we might consider instead

Assignment Project Exam Help

$$d(x, y)^2 = \sum_{i=1}^d w_i (x_i - y_i)^2$$

Email: tutorcs@163.com

- The weights w_i are chosen to normalize different variances of x_i and x_j or to encode their relative significances.
- Graph representation of data is often much a more compact representation than as vectors in \mathbb{R}^d , which does not suffer from problems of high dimensionality.
- Note that in the graph representation of data the geometry of the data points is lost, so the clustering is based only on mutual distances of pairs of points, which is good when clustering is not center based.

Center-based clustering algorithms

We assume data points are represented as vectors in \mathbb{R}^d

程序代写代做 CS 编程辅导

- ① **k -center clustering**: partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{a_1, \dots, a_n\}$ into k clusters, with the corresponding centers c_1, \dots, c_k , which minimizes the maximum distance between any data point and the center of its cluster.

- That is, we want to minimize

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

- This is the “fire-station location problem” since one can think of it as the problem of placing fire stations in a city so as to minimise the maximum distance a fire-truck needs to travel to put out a fire.
- The *radius* of a cluster \mathcal{C}_j is the largest distance of a point from A to its associated cluster centre.
- Thus, center-based clustering algorithms try to minimise the radius of the resulting clustering.

Center-based clustering algorithms

We assume data points are represented as vectors in \mathbb{R}^d

程序代写代做 CS 编程辅导

- ① **k -center clustering** partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimizes the maximum distance between any data point and the center of its cluster.

- That is, we want to minimize

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

- This is the “fire-station location problem” since one can think of it as the problem of placing fire stations in a city so as to minimise the maximum distance a fire-truck needs to travel to put out a fire.
- The *radius* of a cluster \mathcal{C}_j is the largest distance of a point from A to its associated cluster centre.
- Thus, center-based clustering algorithms try to minimise the radius of the resulting clustering.

Center-based clustering algorithms

We assume data points are represented as vectors in \mathbb{R}^d

程序代写代做 CS 编程辅导

- ① **k -center clustering** partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimizes the maximum distance between any data point and the center of its cluster.

- That is, we want to minimize

WeChat: cstutorcs

$$\Phi(\mathcal{C}) = \max_{j=1}^k \max_{a \in \mathcal{C}_j} d(a, \mathbf{c}_j)$$

Email: tutorcs@163.com

- This is the “fire-station location problem” since one can think of it as the problem of finding fire-stations in a city so as to minimise the maximum distance a fire-truck needs to travel to put out a fire.
- The *radius* of a cluster \mathcal{C}_m is the largest distance of a point from A to its associated cluster centre.
- Thus, center-based clustering algorithms try to minimise the radius of the resulting clustering.

Center-based clustering algorithms

We assume data points are represented as vectors in \mathbb{R}^d

程序代写代做 CS 编程辅导

- ① **k -center clustering** partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimizes the maximum distance between any data point and the center of its cluster.

- That is, we want to minimize

WeChat: cstutorcs

$$\Phi(\mathcal{C}) = \max_{j=1}^k \max_{\mathbf{a} \in \mathcal{C}_j} d(\mathbf{a}, \mathbf{c}_j)$$

Email: tutorcs@163.com

- This is the “fire-station location problem” since one can think of it as the problem of building ~~8~~ fire stations in a city so as to minimise the maximum distance a fire-truck needs to travel to put out a fire.
- The *radius* of a clustering $\bigcup_{m=1}^k A_m$ is the largest distance of a point from A to its associated cluster centre.
- Thus, center-based clustering algorithms try to minimise the radius of the resulting clustering.

Center-based clustering algorithms

We assume data points are represented as vectors in \mathbb{R}^d

程序代写代做 CS 编程辅导

- ① **k -center clustering** partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimizes the maximum distance between any data point and the center of its cluster.

- That is, we want to minimize

WeChat: cstutorcs

$$\Phi(\mathcal{C}) = \max_{j=1}^k \max_{a \in \mathcal{C}_j} d(a, c_j)$$

Email: tutorcs@163.com

- This is the “fire-station location problem” since one can think of it as the problem of building ~~8~~ fire stations in a city so as to minimise the maximum distance a fire-truck needs to travel to put out a fire.
- The *radius* of a clustering $A = \bigcup_{m=1}^k A_m$ is the largest distance of a point from A to its associated cluster centre.
- Thus, center-based clustering algorithms try to minimise the radius of the resulting clustering.

Center-based clustering algorithms

We assume data points are represented as vectors in \mathbb{R}^d

程序代写代做 CS 编程辅导

- ① **k -center clustering** partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimizes the maximum distance between any data point and the center of its cluster.

- That is, we want to minimize

WeChat: cstutorcs

$$\Phi(\mathcal{C}) = \max_{j=1}^k \max_{a \in \mathcal{C}_j} d(a, c_j)$$

Email: tutorcs@163.com

- This is the “fire-station location problem” since one can think of it as the problem of building ~~8~~ fire stations in a city so as to minimise the maximum distance a fire-truck needs to travel to put out a fire.
- The *radius* of a clustering $\mathcal{C} = \bigcup_{m=1}^k A_m$ is the largest distance of a point from A to its associated cluster centre.
- Thus, center-based clustering algorithms try to minimise the radius of the resulting clustering.

Center-based clustering

程序代写代做 CS 编程辅导

- ➊ **k -median clustering:** Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of distances between data points and their corresponding centres.
 - That is, we want to minimize

WeChat: cstutorcs

$$\Phi(\mathcal{C}) = \sum_{j=1}^k \sum_{a \in \mathcal{C}_j} d(\mathbf{a}, \mathbf{c}_j)$$

Assignment Project Exam Help

- Note that $d(\mathbf{a}, \mathbf{c}_j)$ can be any distance metric, such as ℓ^1 metric
 $d_1(\mathbf{a}, \mathbf{c}_j) = \sum_{k=1}^d |(a)_k - (c_j)_k|$ or ℓ^2 metric
 $d_2(\mathbf{a}, \mathbf{c}_j) = \sqrt{\sum_{k=1}^d ((a)_k - (c_j)_k)^2}$.
- If the distance is the ℓ_1 distance, one can show that the coordinates of the optimal centers are the coordinate-wise medians of points in each cluster.

Center-based clustering

程序代写代做 CS 编程辅导

- ① **k -median clustering:** Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimise the sum of distances between data points and their corresponding centres.
 - That is, we want to minimise

WeChat: cstutorcs

$$\Phi(\mathcal{C}) = \sum_{j=1}^k \sum_{a \in \mathcal{C}_j} d(\mathbf{a}, \mathbf{c}_j)$$

Assignment Project Exam Help

- Note that $d(\mathbf{a}, \mathbf{c}_j)$ can be any distance metric, such as ℓ^1 metric
 $d_1(\mathbf{a}, \mathbf{c}_j) = \sum_{k=1}^d |(a)_k - (c_j)_k|$ or ℓ^2 metric
 $d_2(\mathbf{a}, \mathbf{c}_j) = \sqrt{\sum_{k=1}^d ((a)_k - (c_j)_k)^2}$.
- If the distance is the ℓ_1 distance, one can show that the coordinates of the optimal centers are the coordinate-wise medians of points in each cluster.

Center-based clustering

程序代写代做 CS编程辅导

- ① **k -median clustering:** Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of distances between data points and their corresponding centres.
 - That is, we want to minimize

WeChat: cstutorcs
$$\Phi(\mathcal{C}) = \sum_{j=1}^k \sum_{a \in \mathcal{C}_j} d(\mathbf{a}, \mathbf{c}_j)$$

Assignment Project Exam Help

- Note that $d(\mathbf{a}, \mathbf{c}_j)$ can be any distance metric, such as ℓ^1 metric
 $d_1(\mathbf{a}, \mathbf{c}_j) = \sum_{k=1}^d |(\mathbf{a})_k - (\mathbf{c}_j)_k|$ or ℓ^2 metric
 $d_2(\mathbf{a}, \mathbf{c}_j) = \sqrt{\sum_{k=1}^d ((\mathbf{a})_k - (\mathbf{c}_j)_k)^2}$.
Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>
- If the distance is the ℓ_1 distance, one can show that the coordinates of the optimal centers are the coordinate-wise medians of points in each cluster.

Center-based clustering algorithms

程序代写代做 CS 编程辅导

The most frequently used center-based clustering algorithm is the k -means algorithm.

① k -means clustering

Description: Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{a_1, \dots, a_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of the squares of distances between data points and their corresponding cluster centers.



- That is, we want

WeChat: cstutorcs

Assignment Project Exam Help

$$\Phi(\mathcal{C}) = \sum_{i=1}^n \sum_{j=1}^k d(a_i, \mathbf{c}_j)^2$$

Email: tutorcs@163.com

- k -means clustering penalises more for larger distances than the k -median clustering.
- k -means clustering has other nice properties; for example, if $d(\mathbf{a}, \mathbf{c}_j)^2 = \sum_{i=1}^d (a_i - c_{ji})^2$ then \mathbf{c}_j must be the centroids of the points in their cluster.

QQ: 749389476

<https://tutorcs.com>

Center-based clustering algorithms

程序代写代做 CS 编程辅导

The most frequently used center-based clustering algorithm is the k -means algorithm.

① k -means clustering



Definition: Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{a_1, \dots, a_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of the squares of distances between data points and their corresponding cluster centers.

- That is, we want

WeChat: cstutorcs

Assignment Project Exam Help

$$\Phi(\mathcal{C}) = \sum_{a \in A} \sum_{j=1}^k d(a, \mathbf{c}_j)^2$$

Email: tutorcs@163.com

- k -means clustering penalises more for larger distances than the k -median clustering.
- k -means clustering has other nice properties; for example, if $d(\mathbf{a}, \mathbf{c}_j)^2 = \sum_{i=1}^d (a_i - c_{ji})^2$ then \mathbf{c}_j must be the centroids of the points in their cluster.

QQ: 749389476
<https://tutorcs.com>

Center-based clustering algorithms

程序代写代做 CS 编程辅导

The most frequently used center-based clustering algorithm is the k -means algorithm.

① k -means clustering



Definition: Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of the squares of distances between data points and their corresponding cluster centers.

- That is, we want to minimize

WeChat: **tutorcs**

Assignment Project Exam Help

$$\Phi(\mathcal{C}) = \sum_{i=1}^n \sum_{j=1}^k d(\mathbf{a}_i, \mathbf{c}_j)^2$$

Email: tutorcs@163.com

- k -means clustering penalizes more for larger distances than the k -median clustering.
- k -means clustering has other nice properties; for example, if $d(\mathbf{a}, \mathbf{c}_j)^2 = \sum_{i=1}^d (a_i - c_{ji})^2$ then \mathbf{c}_j must be the centroids of the points in their cluster.

QQ: 749389476

<https://tutorcs.com>

Center-based clustering algorithms

程序代写代做 CS 编程辅导

The most frequently used center-based clustering algorithm is the k -means algorithm.

① k -means clustering



Definition: Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of the squares of distances between data points and their corresponding cluster centers.

- That is, we want to minimize

WeChat: **tutorcs**

$$\Phi(\mathcal{C}) = \sum_{i=1}^n \sum_{j=1}^k d(\mathbf{a}_i, \mathbf{c}_j)^2$$

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

- k -means clustering penalises more for larger distances than the k -median clustering.
- k -means clustering has other nice properties; for example, if $d(\mathbf{a}, \mathbf{c}_j)^2 = \sum_{i=1}^d (a_i - c_{ji})^2$ then \mathbf{c}_j must be the centroids of the points in their cluster.

Center-based clustering algorithms

程序代写代做 CS 编程辅导

The most frequently used center-based clustering algorithm is the k -means algorithm.

① k -means clustering



Definition: Find a partition $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ of a set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ into k clusters, with the corresponding centers $\mathbf{c}_1, \dots, \mathbf{c}_k$, which minimize the sum of the squares of distances between data points and their corresponding cluster centers.

- That is, we want to minimize

WeChat: **tutorcs**

$$\Phi(\mathcal{C}) = \sum_{i=1}^n \sum_{j=1}^k d(\mathbf{a}_i, \mathbf{c}_j)^2$$

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

- k -means clustering penalises more for larger distances than the k -median clustering.
- k -means clustering has other nice properties; for example, if $d(\mathbf{a}, \mathbf{c}_j)^2 = \sum_{i=1}^d (a_i - c_{ji})^2$ then \mathbf{c}_j must be the centroids of the points in their cluster.

Center-based clustering algorithms

程序代写代做 CS编程辅导

- What is the centroid of points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$?
- Keep in mind that each point is a vector in \mathbb{R}^d , so



$$\mathbf{a}_i = (a_{i1}, \dots, a_{id}).$$

WeChat: cstutorcs

- Let $\mathbf{c} = (c_1, \dots, c_d)$ with for all $1 \leq k \leq d$

Assignment Project Exam Help

Email: tutorcs@163.com

i.e., c_k is the arithmetic mean of the k^{th} coordinates of all of the points $\mathbf{a}_1, \dots, \mathbf{a}_n$.

QQ: 749389476

- Then \mathbf{c} is called *the centroid of the set of points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$* .

Center-based clustering algorithms

程序代写代做 CS编程辅导

- What is the centroid points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$?
- Keep in mind that each point is a vector in \mathbb{R}^d , so



$$\mathbf{a}_i = (a_{i1}, \dots, a_{id}).$$

WeChat: cstutorcs

- Let $\mathbf{c} = (c_1, \dots, c_d)$ with $c_l = (a_{1l} + \dots + a_{nl})/n$ for all $1 \leq l \leq d$

Assignment Project Exam Help

Email: tutorcs@163.com

i.e., c_k is the arithmetic mean of the k^{th} coordinates of all of the points $\mathbf{a}_1, \dots, \mathbf{a}_n$.

QQ: 749389476

- Then \mathbf{c} is called *the centroid of the set of points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$* .

Center-based clustering algorithms

程序代写代做 CS编程辅导

- What is the centroid points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$?
- Keep in mind that each point is a vector in \mathbb{R}^d , so



$$\mathbf{a}_i = (a_{i1}, \dots, a_{id}).$$

WeChat: cstutorcs

- Let $\mathbf{c} = (c_1, \dots, c_d)$ with for all $1 \leq k \leq d$

Assignment Project Exam Help

$$c_k = (a_{1k} + \dots + a_{nk})/n$$

i.e., c_k is the arithmetic mean of the k^{th} coordinates of all of the points $\mathbf{a}_1, \dots, \mathbf{a}_n$.

QQ: 749389476

- Then \mathbf{c} is called *the centroid of the set of points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$* .

<https://tutorcs.com>

Center-based clustering algorithms

程序代写代做 CS编程辅导

- What is the centroid points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$?
- Keep in mind that each point is a vector in \mathbb{R}^d , so



$$\mathbf{a}_i = (a_{i1}, \dots, a_{id}).$$

WeChat: cstutorcs

- Let $\mathbf{c} = (c_1, \dots, c_d)$ with for all $1 \leq k \leq d$

Assignment Project Exam Help

$$c_k = (a_{1k} + \dots + a_{nk})/n$$

i.e., c_k is the arithmetic mean of the k^{th} coordinates of all of the points $\mathbf{a}_1, \dots, \mathbf{a}_n$.

QQ: 749389476

- Then \mathbf{c} is called the *centroid* of the set of points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

<https://tutorcs.com>

Center-based clustering algorithms

程序代写代做 CS编程辅导

- We denote by $\mathbf{x} \cdot \mathbf{y}$ the scalar product of vectors \mathbf{x} and \mathbf{y} and by $\|\mathbf{x}\|$ the norm of a vector \mathbf{x} , i



$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^d x_i y_i \quad \text{and} \quad \|\mathbf{x}\| = \sqrt{\sum_{i=1}^d x_i^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

WeChat: cstutorcs

- Note that $\|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance of points \mathbf{x} and \mathbf{y} :

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

- Note also that

<https://tutorcs.com>

$$\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2 \mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$$

Center-based clustering algorithms

程序代写代做 CS编程辅导

- We denote by $\mathbf{x} \cdot \mathbf{y}$ the scalar product of vectors \mathbf{x} and \mathbf{y} and by $\|\mathbf{x}\|$ the norm of a vector \mathbf{x} , i



$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^d x_i y_i \quad \text{and} \quad \|\mathbf{x}\| = \sqrt{\sum_{i=1}^d x_i^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

WeChat: cstutorcs

- Note that $\|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance of points \mathbf{x} and \mathbf{y} :

Assignment Project Exam Help

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

QQ: 749389476

- Note also that

<https://tutorcs.com>

$$\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2 \mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$$

Center-based clustering algorithms

程序代写代做 CS编程辅导

- We denote by $\mathbf{x} \cdot \mathbf{y}$ the scalar product of vectors \mathbf{x} and \mathbf{y} and by $\|\mathbf{x}\|$ the norm of a vector \mathbf{x} , i



$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^d x_i y_i \quad \text{and} \quad \|\mathbf{x}\| = \sqrt{\sum_{i=1}^d x_i^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

WeChat: cstutorcs

- Note that $\|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance of points \mathbf{x} and \mathbf{y} :

Assignment Project Exam Help

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

QQ: 749389476

- Note also that <https://tutorcs.com>

$$\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2 \mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$$

Center-based clustering algorithms

- **Theorem:** Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points and \mathbf{x} another point, all in \mathbb{R}^d . Let also \mathbf{c} be the centroid of A . Then

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

- Proof: Using $\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2\mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$

WeChat: cstutorcs

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|(\mathbf{a}_i - \mathbf{c}) + (\mathbf{c} - \mathbf{x})\|^2$$

Assignment Project Exam Help

$$= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + 2(\mathbf{c} - \mathbf{x}) \cdot \sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) + n\|\mathbf{c} - \mathbf{x}\|^2$$

Email: tutorcs@163.com

QQ: 749389476

Since \mathbf{c} is the centroid, $\mathbf{c} = (\sum_{i=1}^n \mathbf{a}_i) / n$, we have

<https://tutorcs.com>

$$\sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) = \sum_{i=1}^n \mathbf{a}_i - n\mathbf{c} = 0$$

and we have proved the claim.

Center-based clustering algorithms

- **Theorem:** Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points and \mathbf{x} another point, all in \mathbb{R}^d . Let also \mathbf{c} be the centroid of A . Then

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

- **Proof:** Using $\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2\mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$

WeChat: cstutorcs

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|(\mathbf{a}_i - \mathbf{c}) + (\mathbf{c} - \mathbf{x})\|^2$$

Assignment Project Exam Help

$$= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + 2(\mathbf{c} - \mathbf{x}) \cdot \sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) + n\|\mathbf{c} - \mathbf{x}\|^2$$

Email: tutorcs@163.com

QQ: 749389476

Since \mathbf{c} is the centroid, $\mathbf{c} = (\sum_{i=1}^n \mathbf{a}_i) / n$, we have

<https://tutorcs.com>

$$\sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) = \sum_{i=1}^n \mathbf{a}_i - n\mathbf{c} = 0$$

and we have proved the claim.

Center-based clustering algorithms

- **Theorem:** Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points and \mathbf{x} another point, all in \mathbb{R}^d . Let also \mathbf{c} be the centroid of A . Then

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

- **Proof:** Using $\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2\mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$

WeChat: cstutorcs

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|(\mathbf{a}_i - \mathbf{c}) + (\mathbf{c} - \mathbf{x})\|^2$$

Assignment Project Exam Help

$$= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + 2(\mathbf{c} - \mathbf{x}) \cdot \sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) + n\|\mathbf{c} - \mathbf{x}\|^2$$

Email: tutorcs@163.com

QQ: 749389476

Since \mathbf{c} is the centroid, $\mathbf{c} = (\sum_{i=1}^n \mathbf{a}_i) / n$, we have

<https://tutorcs.com>

$$\sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) = \sum_{i=1}^n \mathbf{a}_i - n\mathbf{c} = 0$$

and we have proved the claim.

Center-based clustering algorithms

- **Theorem:** Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points and \mathbf{x} another point, all in \mathbb{R}^d . Let also \mathbf{c} be the centroid of A . Then

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

- **Proof:** Using $\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2\mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$

$$\begin{aligned} \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 &= \sum_{i=1}^n \|(\mathbf{a}_i - \mathbf{c}) + (\mathbf{c} - \mathbf{x})\|^2 \\ &= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + 2(\mathbf{c} - \mathbf{x}) \cdot \sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) + n\|\mathbf{c} - \mathbf{x}\|^2 \end{aligned}$$

Since \mathbf{c} is the centroid, $\mathbf{c} = (\sum_{i=1}^n \mathbf{a}_i)/n$, we have

<https://tutorcs.com>

$$\sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) = \sum_{i=1}^n \mathbf{a}_i - n\mathbf{c} = 0$$

and we have proved the claim.

Center-based clustering algorithms

- **Theorem:** Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points and \mathbf{x} another point, all in \mathbb{R}^d . Let also \mathbf{c} be the centroid of A . Then

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

- **Proof:** Using $\|\mathbf{x} + \mathbf{y}\|^2 = (\mathbf{x} + \mathbf{y}) \cdot (\mathbf{x} + \mathbf{y}) = \|\mathbf{x}\|^2 + 2\mathbf{x} \cdot \mathbf{y} + \|\mathbf{y}\|^2$

$$\begin{aligned} \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 &= \sum_{i=1}^n \|(\mathbf{a}_i - \mathbf{c}) + (\mathbf{c} - \mathbf{x})\|^2 \\ &= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + 2(\mathbf{c} - \mathbf{x}) \cdot \sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) + n\|\mathbf{c} - \mathbf{x}\|^2 \end{aligned}$$

Since \mathbf{c} is the centroid, $\mathbf{c} = (\sum_{i=1}^n \mathbf{a}_i) / n$, we have

<https://tutorcs.com>

$$\sum_{i=1}^n (\mathbf{a}_i - \mathbf{c}) = \sum_{i=1}^n \mathbf{a}_i - n\mathbf{c} = 0$$

and we have proved the claim.

Center-based clustering algorithms

程序代写代做 CS编程辅导

- **Corollary:** Let $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$ be a set of points in \mathbb{R}^d . Then

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

WeChat: cstutorcs
is minimised when \mathbf{x} is the centroid $\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$.

- **Proof:** By the previous slide

Email: tutorcs@163.com
$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

QQ: 749389476

The first summand does not depend on \mathbf{x} and the second is zero when $\mathbf{x} = \mathbf{c}$. Thus $\mathbf{x} = \mathbf{c}$ minimises $D(\mathbf{x})$.

Center-based clustering algorithms

程序代写代做 CS编程辅导

- **Corollary:** Let $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$ be a set of points in \mathbb{R}^d . Then


$$= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2$$

WeChat: cstutorcs

is minimised when \mathbf{x} is the centroid $\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$.

- **Proof:** By the previous theorem.

Assignment Project Exam Help

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

Email: tutorcs@163.com
QQ: 749389476

The first summand does not depend on \mathbf{x} and the second is zero when $\mathbf{x} = \mathbf{c}$. Thus $\mathbf{x} = \mathbf{c}$ minimises $D(\mathbf{x})$.

Center-based clustering algorithms

程序代写代做 CS编程辅导

- **Corollary:** Let $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$ be a set of points in \mathbb{R}^d . Then



$$= \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2$$

WeChat: cstutorcs

is minimised when \mathbf{x} is the centroid $\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$.

- **Proof:** By the previous theorem.

Assignment Project Exam Help

$$\sum_{i=1}^n \|\mathbf{a}_i - \mathbf{x}\|^2 = \sum_{i=1}^n \|\mathbf{a}_i - \mathbf{c}\|^2 + n\|\mathbf{c} - \mathbf{x}\|^2$$

Email: tutorcs@163.com

QQ: 749389476

The first summand does not depend on \mathbf{x} and the second is zero when $\mathbf{x} = \mathbf{c}$. Thus $\mathbf{x} = \mathbf{c}$ minimises $D(\mathbf{x})$.

<https://tutorcs.com>

Center-based clustering algorithms

- Thus, if we are given a set of points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ in \mathbb{R}^d and the problem is to find a partition of A into k disjoint components $A = \bigcup_{i=1}^k A_i$ and k points $\mathbf{x}_1, \dots, \mathbf{x}_k$ such that the sum



$$\sum_{j=1}^k \sum_{\mathbf{a}_i \in A_j} \|\mathbf{a}_i - \mathbf{x}_j\|^2$$

WeChat: cstutorcs

is as small as possible, then, whatever such an optimal partition $\{A_j : 1 \leq j \leq k\}$ might be, the points \mathbf{x}_j must be the centroids \mathbf{c}_j of sets A_j .

- Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points in \mathbb{R}^d and let \mathbf{c} be the centroid of A . Then

Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>

- This is easy to see just by replacing \mathbf{c} by its definition plus doing some obvious algebra.

Center-based clustering algorithms

- Thus, if we are given a set of points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ in \mathbb{R}^d and the problem is to find a partition of A into k disjoint components $A = \bigcup_{i=1}^k A_i$ and k points $\mathbf{x}_1, \dots, \mathbf{x}_k$ such that the sum



$$\sum_{j=1}^k \sum_{\mathbf{a}_i \in A_j} \|\mathbf{a}_i - \mathbf{x}_j\|^2$$

WeChat: cstutorcs

is as small as possible, then, whatever such an optimal partition $\{A_j : 1 \leq j \leq k\}$ might be, the points \mathbf{x}_j must be the centroids \mathbf{c}_j of sets A_j .

- Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points in \mathbb{R}^d and let \mathbf{c} be the centroid of A . Then

$$\frac{1}{2n} \sum_{i,j=1}^n \|\mathbf{a}_i - \mathbf{a}_j\|^2 = \sum_{m=1}^n \|\mathbf{a}_m - \mathbf{c}\|^2$$

QQ: 749389476
<https://tutorcs.com>

- This is easy to see just by replacing \mathbf{c} by its definition plus doing some obvious algebra.

Center-based clustering algorithms

- 程序代写代做CS编程辅导
- Thus, if we are given a set of points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ in \mathbb{R}^d and the problem is to find a partition of A into k disjoint components $A = \bigcup_{i=1}^k A_i$ and k points $\mathbf{x}_1, \dots, \mathbf{x}_k$ such that the sum



$$\sum_{j=1}^k \sum_{\mathbf{a}_i \in A_j} \|\mathbf{a}_i - \mathbf{x}_j\|^2$$

WeChat: cstutorcs

is as small as possible, then, whatever such an optimal partition $\{A_j : 1 \leq j \leq k\}$ might be, the points \mathbf{x}_j must be the centroids \mathbf{c}_j of sets A_j .

- Assignment Project Exam Help
- Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a set of points in \mathbb{R}^d and let \mathbf{c} be the centroid of A . Then

Email: tutorcs@163.com

QQ: 749389476

$$\frac{1}{2n} \sum_{i,j=1}^n \|\mathbf{a}_i - \mathbf{a}_j\|^2 = \sum_{m=1}^n \|\mathbf{a}_m - \mathbf{c}\|^2$$

<https://tutorcs.com>

- This is easy to see just by replacing \mathbf{c} by its definition plus doing some obvious algebra.

Center-based clustering algorithms

程序代写代做 CS编程辅导

- Thus, finding disjoint clusters $A = \bigcup_{m=1}^k A_m$ which minimises



$$\sum_{m=1}^k \sum_{\mathbf{a}_j \in A_m} \|\mathbf{a}_j - \mathbf{c}_m\|^2$$

WeChat: csutorcs

is equivalent to minimising

Assignment Project Exam Help

Email: tutorcs@163.com

$$\sum_{m=1}^k \frac{1}{2|A_m|} \sum_{\mathbf{a}_i, \mathbf{a}_j \in A_m} \|\mathbf{a}_i - \mathbf{a}_j\|^2$$

<https://tutorcs.com>

程序代写代做 CS编程辅导

- Finding the optimal clustering is an NP hard problem which cannot be solved in polynomial time, so we have to look at approximate solutions.
- The best known approach to k-means clustering algorithm is Lloyd's algorithm.
- Lloyd's Clustering Algorithm.



WeChat: cstutorcs

① Start with an initial set of cluster centres $\{c_m^{(0)} | 1 \leq m \leq k\}$ (we will later explain how to obtain such an initial set).

② Cluster all points by associating each $a \in A$ with the nearest cluster centre.

③ Replace cluster centres with the centroids of thus obtained clusters.

④ Repeat 2 and 3 until cluster centres (and thus also clusters) stop changing.

Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>

程序代写代做 CS编程辅导

- Finding the optimal clustering is an NP hard problem which cannot be solved in polynomial time, so we have to look at approximate solutions.
- The best known approach to k-means clustering algorithm is Lloyd's algorithm.
- Lloyd's Clustering Algorithm.

WeChat: cstutorcs

① Start with an initial set of cluster centres $\{c_m^{(0)} : 1 \leq m \leq k\}$ (we will later explain how to obtain such an initial set).

② Cluster all points by associating each $a \in A$ with the nearest cluster centre.

③ Replace cluster centres with the centroids of thus obtained clusters.

④ Repeat 2 and 3 until cluster centres (and thus also clusters) stop changing.

Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>



程序代写代做 CS编程辅导

- Finding the optimal clustering is an NP hard problem which cannot be solved in polynomial time, so we have to look at approximate solutions.
- The best known approach to k-means clustering algorithm is Lloyd's algorithm.
- Lloyd's Clustering Algorithm:



WeChat: cstutorcs

- ① Start with an initial set of cluster centres $\{\mathbf{c}_m^{(0)} : 1 \leq m \leq k\}$ (we will later explain how to obtain such an initial set).
- ② Cluster all points $a \in A$ by associating each $a \in A$ with the nearest cluster centre.
- ③ Replace cluster centres with the centroids of thus obtained clusters.
- ④ Repeat 2 and 3 until cluster centres (and thus also clusters) stop changing.

Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导

- Finding the optimal clustering is an NP hard problem which cannot be solved in polynomial time, so we have to look at approximate solutions.
- The best known approach to k-means clustering algorithm is Lloyd's algorithm.
- Lloyd's Clustering Algorithm:



WeChat: cstutorcs

- ① Start with an initial set of cluster centres $\{\mathbf{c}_m^{(0)} : 1 \leq m \leq k\}$ (we will later explain how to obtain such an initial set).
- ② Cluster all points $\mathbf{a} \in A$ into clusters A_m by associating each $\mathbf{a} \in A$ with the nearest cluster centre.
- ③ Replace cluster centres with the centroids of thus obtained clusters.
- ④ Repeat 2 and 3 until cluster centres (and thus also clusters) stop changing.

Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>

程序代写代做 CS编程辅导

- Finding the optimal clustering is an NP hard problem which cannot be solved in polynomial time, so we have to look at approximate solutions.
- The best known approach to k-means clustering algorithm is Lloyd's algorithm.
- Lloyd's Clustering Algorithm:



WeChat: cstutorcs

- ① Start with an initial set of cluster centres $\{\mathbf{c}_m^{(0)} : 1 \leq m \leq k\}$ (we will later explain how to obtain such an initial set).
- ② Cluster all points $\mathbf{a} \in A$ into clusters A_m by associating each $\mathbf{a} \in A$ with the nearest cluster centre.
- ③ Replace cluster centres with the centroids of thus obtained clusters.
- ④ Repeat 2 and 3 until cluster centres (and thus also clusters) stop changing.

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导

- Finding the optimal clustering is an NP hard problem which cannot be solved in polynomial time, so we have to look at approximate solutions.
- The best known approach to k-means clustering algorithm is Lloyd's algorithm.
- Lloyd's Clustering Algorithm:



WeChat: cstutorcs

- ① Start with an initial set of cluster centres $\{\mathbf{c}_m^{(0)} : 1 \leq m \leq k\}$ (we will later explain how to obtain such an initial set).
- ② Cluster all points $\mathbf{a} \in A$ to clusters A_m by associating each $\mathbf{a} \in A$ with the nearest cluster centre.
- ③ Replace cluster centres with the centroids of thus obtained clusters.
- ④ Repeat 2 and 3 until cluster centres (and thus also clusters) stop changing.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导

- **Claim:** At every round of the loop, Lloyd's algorithm reduces the size of



$$\sum_{\substack{m=1 \\ \mathbf{a}_j \in A_m^{(p)}}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

where $A_m^{(p)}$ are the “temporary” clusters and $\mathbf{c}_m^{(p)}$ is the “temporary” centre of cluster $A_m^{(p)}$ at round p of the loop.

- This is obvious, because both steps of the loop have this property: replacing the cluster **Email: tutorcs@163.com** by the cluster recuses every summand $\sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$ and so does associating every point with the nearest cluster centre.
- However, the algorithm might stop at a local minimum, not the globally optimal minimum.

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导

- **Claim:** At every round of the loop, Lloyd's algorithm reduces the size of



$$\sum_{\substack{m=1 \\ \mathbf{a}_j \in A_m^{(p)}}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

where $A_m^{(p)}$ are the “temporary” clusters and $\mathbf{c}_m^{(p)}$ is the “temporary” centre of cluster $A_m^{(p)}$ at round p of the loop.

- This is obvious, because both steps of the loop have this property: replacing the cluster centres with the centroid of the cluster reduces every summand $\sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$ and so does associating every point with the nearest cluster centre.
- However, the algorithm might stop at a local minimum, not the globally optimal minimum.

Email: tutorcs@163.com

QQ: 749389476

Lloyd's Algorithm

程序代写代做 CS编程辅导

- **Claim:** At every round of the loop, Lloyd's algorithm reduces the size of



$$\sum_{\substack{m=1 \\ \mathbf{a}_j \in A_m^{(p)}}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

where $A_m^{(p)}$ are the “temporary” clusters and $\mathbf{c}_m^{(p)}$ is the “temporary” centre of cluster $A_m^{(p)}$ at round p of the loop.

- This is obvious, because both steps of the loop have this property: replacing the cluster centres with the centroid of the cluster reduces every summand $\sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$ and so does associating every point with the nearest cluster centre.
- However, the algorithm might stop at a local minimum, not the globally optimal minimum.

Email: tutors@163.com

QQ: 749389476

Lloyd's Algorithm

程序代写代做 CS 编程辅导

- In lots of applications this local minimum provides a good clustering.
- However, sometimes it's better to run the algorithm several times with different initial set of cluster centres and picking the solution from the run with the smallest sum



$$\sum_{a_j \in A_m} \sum_{m=1}^k \|a_j - c_m^{(p)}\|^2$$

WeChat: cstutorcs

is the smallest.

Assignment Project Exam Help

- How can we obtain good starting centres of clusters for the algorithm?
- One good option is to pick a random point a_q from A as the first centre
 $c_1^{(0)} = a_q$.
- For the second centre pick another point from A which is the farthest away from a_q .
- Continue in this manner to get all k cluster centres, always picking as the next centre a point from A which has the largest minimal distance to all previously picked centres.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS 编程辅导

- In lots of applications this local minimum provides a good clustering.
- However, sometimes the results are obtained by running the algorithm several times with different initial set of cluster centres and picking the solution from the run with the sum



$$\sum_{m=1}^M \sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

is the smallest.

Assignment Project Exam Help

- How can we obtain good starting centres of clusters for the algorithm?
- One good option is to pick a random point \mathbf{a}_q from A as the first centre $\mathbf{c}_1^{(0)} = \mathbf{a}_q$.
- For the second centre pick another point from A which is the farthest away from \mathbf{a}_q .
- Continue in this manner to get all k cluster centres, always picking as the next centre a point from A which has the largest minimal distance to all previously picked centres.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS 编程辅导

- In lots of applications this local minimum provides a good clustering.
- However, sometimes the results are obtained by running the algorithm several times with different initial set of cluster centres and picking the solution from the run with the sum



$$\sum_{m=1}^M \sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

is the smallest.

Assignment Project Exam Help

- How can we obtain good starting centres of clusters for the algorithm?
- One good option is to pick a random point \mathbf{a}_q from A as the first centre $\mathbf{c}_1^{(0)} = \mathbf{a}_q$.
- For the second centre pick another point from A which is the farthest away from \mathbf{a}_q .
- Continue in this manner to get all k cluster centres, always picking as the next centre a point from A which has the largest minimal distance to all previously picked centres.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS 编程辅导

- In lots of applications this local minimum provides a good clustering.
- However, sometimes the results are obtained by running the algorithm several times with different initial set of cluster centres and picking the solution from the run with the sum



$$\sum_{m=1}^M \sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

is the smallest.

Assignment Project Exam Help

- How can we obtain good starting centres of clusters for the algorithm?
- One good option is to pick a random point \mathbf{a}_q from A as the first centre
 $\mathbf{c}_1^{(0)} = \mathbf{a}_q$.
- For the second centre pick another point from A which is the farthest away from \mathbf{a}_q .
- Continue in this manner to get all k cluster centres, always picking as the next centre a point from A which has the largest minimal distance to all previously picked centres.

Email: tutorcs@163.com
QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS 编程辅导

- In lots of applications this local minimum provides a good clustering.
- However, sometimes the results are obtained by running the algorithm several times with different initial set of cluster centres and picking the solution from the run with the sum



$$\sum_{m=1}^M \sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

is the smallest.

Assignment Project Exam Help

- How can we obtain good starting centres of clusters for the algorithm?
- One good option is to pick a random point \mathbf{a}_q from A as the first centre $\mathbf{c}_1^{(0)} = \mathbf{a}_q$.
- For the second centre pick another point from A which is the farthest away from \mathbf{a}_q .
- Continue in this manner to get all k cluster centres, always picking as the next centre a point from A which has the largest minimal distance to all previously picked centres.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS 编程辅导

- In lots of applications this local minimum provides a good clustering.
- However, sometimes the results are obtained by running the algorithm several times with different initial set of cluster centres and picking the solution from the run with the sum



$$\sum_{m=1}^M \sum_{\mathbf{a}_j \in A_m^{(p)}} \|\mathbf{a}_j - \mathbf{c}_m^{(p)}\|^2$$

WeChat: cstutorcs

is the smallest.

Assignment Project Exam Help

- How can we obtain good starting centres of clusters for the algorithm?
- One good option is to pick a random point \mathbf{a}_q from A as the first centre
 $\mathbf{c}_1^{(0)} = \mathbf{a}_q$.
- For the second centre pick another point from A which is the farthest away from \mathbf{a}_q .
- Continue in this manner to get all k cluster centres, always picking as the next centre a point from A which has the largest minimal distance to all previously picked centres.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导



- In fact, such a procedure for finding the initial centres $\mathbf{c}_m^{(0)}$ of initial clusters is used as a step in a clustering algorithm in itself. You first choose cluster centers as described and then simply clustering points from A according to which is the closest centre $\mathbf{c}_m^{(0)}$ to that point.
- This algorithm is usually called The Farthest Traversal k -clustering algorithm.
- The Farthest Traversal k -clustering algorithm, despite its simplicity, provides a reasonably good approximation clustering in the following sense.
- (Remember that the *radius* of a clustering $A = \bigcup_{m=1}^k A_m$ is the largest distance of a point from its associated cluster centre.)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导



- In fact, such a procedure for finding the initial centres $\mathbf{c}_m^{(0)}$ of initial clusters is used as a step in a clustering algorithm in itself. You first choose cluster centers as described and then simply clustering points from A according to which is the closest centre $\mathbf{c}_m^{(0)}$ to that point.
- This algorithm is usually called The Farthest Traversal k -clustering algorithm.
- The Farthest Traversal k -clustering algorithm, despite its simplicity, provides a reasonably good approximation clustering in the following sense.
- (Remember that the *radius* of a clustering $A = \bigcup_{m=1}^k A_m$ is the largest distance of a point from its associated cluster centre.)

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



- In fact, such a procedure for finding the initial centres $\mathbf{c}_m^{(0)}$ of initial clusters is used as a step in a clustering algorithm in itself. You first choose cluster centers as described and then simply clustering points from A according to which is the closest centre $\mathbf{c}_m^{(0)}$ to that point.
- This algorithm is usually called The Farthest Traversal k -clustering algorithm.
- The Farthest Traversal k -clustering algorithm, despite its simplicity, provides a reasonably good approximate clustering in the following sense.
- (Remember that the *radius* of a clustering $A = \bigcup_{m=1}^k A_m$ is the largest distance of a point from its associated cluster centre.)

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



- In fact, such a procedure for finding the initial centres $\mathbf{c}_m^{(0)}$ of initial clusters is used as a step in a clustering algorithm in itself. You first choose cluster centers as described and then simply clustering points from A according to which is the closest centre $\mathbf{c}_m^{(0)}$ to that point.
- This algorithm is usually called The Farthest Traversal k -clustering algorithm.
- The Farthest Traversal k -clustering algorithm, despite its simplicity, provides a reasonably good approximate clustering in the following sense.
- (Remember that the *radius* of a clustering $A = \bigcup_{m=1}^k A_m$ is the largest distance of a point from A to its associated cluster centre.)

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $a \in A$ at a distance to its cluster larger than $2r$. This means that the distance of a to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise a would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (a plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Lloyd's Algorithm

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $\mathbf{a} \in A$ at a distance to its cluster larger than $2r$. This would mean that the distance of \mathbf{a} to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise \mathbf{a} would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (\mathbf{a} plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $\mathbf{a} \in A$ at a distance to its cluster larger than $2r$. This would mean that the distance of \mathbf{a} to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise \mathbf{a} would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (\mathbf{a} plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $\mathbf{a} \in A$ at a distance to its cluster larger than $2r$. This would mean that the distance of \mathbf{a} to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise \mathbf{a} would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (\mathbf{a} plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $\mathbf{a} \in A$ at a distance to its cluster larger than $2r$. This would mean that the distance of \mathbf{a} to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise \mathbf{a} would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (\mathbf{a} plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $\mathbf{a} \in A$ at a distance to its cluster larger than $2r$. This would mean that the distance of \mathbf{a} to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise \mathbf{a} would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (\mathbf{a} plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导

- **Theorem:** If A has a k -clustering of radius r , then the Farthest Traversal k -clustering produces a clustering of radius at most $2r$.
- **Proof:** Suppose otherwise there is $\mathbf{a} \in A$ at a distance to its cluster larger than $2r$. This would mean that the distance of \mathbf{a} to all cluster centres is larger than $2r$.
- But this implies that also the distances between all pairs of cluster centres must also be larger than $2r$ because otherwise \mathbf{a} would have been chosen as one of the cluster centres.
- Thus, we would have at least $k + 1$ points (\mathbf{a} plus the k cluster centres) which are all pairwise distances larger than $2r$.
- Since we have $k + 1$ points in k clusters, two such points must be in the same cluster.
- But no such two points can be in the same cluster of radius r , because their distance would be at most the diameter of the circle which is $2r$.
- Thus there cannot be a k clustering of radius r , which is a contradiction.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Ward's Algorithm

程序代写代做 CS 编程辅导

- Loyd's algorithm has been recently randomized by instead of picking always the furthest point, by picking a point with probability proportional to the distance to one of already picked points.
- Let A_m be a cluster. Let us set its centroid \mathbf{c}_m as its centre; let us set

$$\text{cost}(A_m) = \sum_{\mathbf{a}_i \in A_m} \|\mathbf{a}_i - \mathbf{c}_m\|^2$$

WeChat: cstutorcs

Assignment Project Exam Help

- The k -means clustering algorithms are trying to minimise the sum $\sum_{m=1}^k \text{cost}(A_m)$. Email: tutorcs@163.com
- Note that if we have two clusters A_m and A_l and take their union $B = A_m \cup A_l$ and the centroid of B as the cluster centre of B , then QQ: 749389476

<https://tutorcs.com> + cost(A_l),

because two centres can get closer to the points in $B = A_m \cup A_l$ than just one point.

Ward's Algorithm

程序代写代做 CS 编程辅导

- Loyd's algorithm has been recently randomized by instead of picking always the furthest point, by picking a point with probability proportional to the distance to one of already picked points.
- Let A_m be a cluster. Let us set its centroid \mathbf{c}_m as its centre; let us set

$$\text{cost}(A_m) = \sum_{\mathbf{a}_i \in A_m} \|\mathbf{a}_i - \mathbf{c}_m\|^2$$

WeChat: cstutorcs

Assignment Project Exam Help

- The k -means clustering algorithms are trying to minimise the sum $\sum_{m=1}^k \text{cost}(A_m)$. Email: tutorcs@163.com
- Note that if we have two clusters A_m and A_l and take their union $B = A_m \cup A_l$ and the centroid of B as the cluster centre of B , then

$$\text{cost}(B) < \text{cost}(A_m) + \text{cost}(A_l),$$

because two centres can get closer to the points in $B = A_m \cup A_l$ than just one point.

Ward's Algorithm

程序代写代做 CS编程辅导

- Lloyd's algorithm has been recently randomized by instead of picking always the furthest point, by picking a point with probability proportional to the distance to one of already picked points.
 - Let A_m be a cluster. Let us choose centroid \mathbf{c}_m as its centre; let us set



$$\text{cost}(A_m) = \sum_{\mathbf{a}_i \in A_m} \|\mathbf{a}_i - \mathbf{c}_m\|^2$$

Assignment Project Exam Help

- The k -means clustering algorithms are trying to minimise the sum $\sum_{m=1}^k \text{cost}(A_m)$. Email: tutorcs@163.com
 - Note that if we have two clusters A_m and A_l and take their union $B = A_m \cup A_l$ and the centroid of B as the cluster centre of B , then QQ: 749389476

$$\text{cost}(B) \geq \text{cost}(A_m) + \text{cost}(A_l),$$

because two centres can get closer to the points in $B = A_m \cup A_l$ than just one point.

Ward's Algorithm

- Ward's algorithm is a greedy k -means algorithm:

- ① Start with every point p_i in its own cluster.
- ② While the number of clusters is larger than k repeat:
find two clusters C, C' such that



$$\text{cost}(C \cup C') - \text{cost}(C) - \text{cost}(C')$$

is as small as possible. Merge them with a single merged cluster $C \cup C'$ with its centroid as its centre.

Assignment Project Exam Help

- How do we cluster data when clusters are not centre based??

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



- This done using the *spectral clustering*.

Ward's Algorithm

- Ward's algorithm is a greedy k -means algorithm:
 - ① Start with every point in its own cluster.
 - ② While the number of clusters is larger than k repeat:
find two clusters such that
$$\text{cost}(C \cup C') - \text{cost}(C) - \text{cost}(C')$$



is as small as possible. Merge them with a single merged cluster $C \cup C'$ with its centroid as its centre.

**WeChat: cstutorcs
Assignment Project Exam Help**

- How do we cluster data when clusters are not centre based??

Email: tutorcs@163.com



- This done using the *spectral clustering*.

Ward's Algorithm

- Ward's algorithm is a greedy k -means algorithm:
 - ① Start with every point x_i in its own cluster.
 - ② While the number of clusters is larger than k repeat:
find two clusters C and C' such that
$$\text{cost}(C \cup C') - \text{cost}(C) - \text{cost}(C')$$

is as small as possible and replace them with a single merged cluster $C \cup C'$ with its centroid as its centre.

WeChat: **tutorcs**
Assignment Project Exam Help

- How do we cluster data when clusters are not centre based??
Email: tutorcs@163.com



- This done using the *spectral clustering*.

Ward's Algorithm

- Ward's algorithm is a greedy k -means algorithm:
 - ① Start with every point x_i in its own cluster.
 - ② While the number of clusters is larger than k repeat:
find two clusters C and C' such that
$$\text{cost}(C \cup C') - \text{cost}(C) - \text{cost}(C')$$

is as small as possible and replace them with a single merged cluster $C \cup C'$ with its centroid as its centre.

WeChat: **tutorcs**
Assignment Project Exam Help

- How do we cluster data when clusters are not centre based??

Email: **tutorcs@163.com**



- This done using the *spectral clustering*.

Ward's Algorithm

- Ward's algorithm is a greedy k -means algorithm:
 - ① Start with every point in its own cluster.
 - ② While the number of clusters is larger than k repeat:
 - find two clusters such that
$$\text{cost}(C \cup C') - \text{cost}(C) - \text{cost}(C')$$

is as small as possible and replace them with a single merged cluster $C \cup C'$ with its centroid as its centre.

WeChat: **tutorcs**
Assignment Project Exam Help

- How do we cluster data when clusters are not centre based??

Email: tutorcs@163.com



- This done using the *spectral clustering*.

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_n\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of an edge $e = (v_i, v_j)$ is equal to some form of similarity measure of the data points a_i, a_j which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means that the vertices v_i and v_j correspond to completely dissimilar data points a_i, a_j , and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|a_i - a_j\|$ between the corresponding data points a_i and a_j , such as $e^{-\frac{\|a_i - a_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points a_i, a_j which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|a_i - a_j\|$ between the corresponding data points a_i and a_j , such as $e^{-\frac{\|a_i - a_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points a_i, a_j which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|a_i - a_j\|$ between the corresponding data points a_i and a_j , such as $e^{-\frac{\|a_i - a_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points $\mathbf{a}_i, \mathbf{a}_j$ which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|\mathbf{a}_i - \mathbf{a}_j\|$ between the corresponding data points \mathbf{a}_i and \mathbf{a}_j , such as $e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points $\mathbf{a}_i, \mathbf{a}_j$ which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|\mathbf{a}_i - \mathbf{a}_j\|$ between the corresponding data points \mathbf{a}_i and \mathbf{a}_j , such as $e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ji} = w_{ij}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points $\mathbf{a}_i, \mathbf{a}_j$ which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|\mathbf{a}_i - \mathbf{a}_j\|$ between the corresponding data points \mathbf{a}_i and \mathbf{a}_j , such as $e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points $\mathbf{a}_i, \mathbf{a}_j$ which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|\mathbf{a}_i - \mathbf{a}_j\|$ between the corresponding data points \mathbf{a}_i and \mathbf{a}_j , such as $e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- We represent a set of data points $A = \{a_1, \dots, a_m\}$ as the set of vertices $\{v_1, \dots, v_n\}$ of an undirected weighted graph $G = (V, E)$.
- The weight $w_{ij} \geq 0$ of edge $e = (v_i, v_j)$ is equal to some form of similarity measure of data points $\mathbf{a}_i, \mathbf{a}_j$ which correspond to vertices v_i, v_j .
- If $w_{ij} = 0$ this means vertices v_i and v_j correspond to completely dissimilar data points and in this case the graph does not include an edge of the form $e = (v_i, v_j)$.
- The similarity of vertices v_i and v_j depend, for example, on a decreasing function of the Euclidean distance $\|\mathbf{a}_i - \mathbf{a}_j\|$ between the corresponding data points \mathbf{a}_i and \mathbf{a}_j , such as $e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2}}$.
- Since the graph is undirected, $w_{ij} = w_{ji}$.
- We put all the weights w_{ij} into a symmetric adjacency matrix $W = (w_{ij})_{i,j=1}^n$.
- Recall that $w_{ij} = 0$ means that there is no edge between vertices v_i and v_j .
- Since the graph is weighted, the degree d_i of a vertex v_i is defined as

$$d_i = \sum_{j=1}^n w_{ij}$$

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a complete summary of the set of data points $A = \{a_1, \dots, a_n\}$.

- The geometry of A is completely lost, and only pairwise similarities between the data points are preserved.

- This is actually good for us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

**WeChat: cstutorcs
Assignment Project Exam Help**

- Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

Email: tutorcs@163.com

- Given a subset $S \subset V$ of vertices, we denote by $\mathbb{1}_S$ the indicator vector $\mathbb{1}_S = (f_1, \dots, f_n) \in \mathbb{R}^n$

$$\mathbb{1}_S = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a summary of the set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.
- The geometry of A is lost, and only pairwise similarities between the data points are preserved.
- This is actually good for us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

QQ: 749389476

$$\frac{1}{\|f\|_1} f_i = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a complete summary of the set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

- The geometry of A is completely lost, and only pairwise similarities between the data points are preserved.

- This is actually good for us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

WeChat: cstutorcs
Assignment Project Exam Help

- Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

Email: tutorcs@163.com

- Given a subset $S \subset V$ of vertices, we denote by $\mathbb{1}_S$ the indicator vector $\mathbb{1}_S = (f_1, \dots, f_n) \in \mathbb{R}^n$

$$\mathbb{1}_S = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a summary of the set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

- The geometry of A is lost, and only pairwise similarities between the data points are preserved.

- This is actually good because it will allow us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

WeChat: cstutorcs

Assignment Project Exam Help

- Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

Email: tutorcs@163.com

- Given a subset $S \subset V$ of vertices, we denote by $\mathbb{1}_S$ the indicator vector $\mathbb{1}_S = (f_1, \dots, f_n) \in \mathbb{R}^n$

$$\mathbb{1}_S = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a summary of the set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

- The geometry of A is lost, and only pairwise similarities between the data points are preserved.

- This is actually good because it will allow us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

WeChat: cstutorcs

Assignment Project Exam Help

- Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

Email: tutorcs@163.com

- Given a subset $S \subset V$ of vertices, we denote by $\mathbb{1}_S$ the indicator vector $\mathbb{1}_S = (f_1, \dots, f_n) \in \mathbb{R}^n$

QQ: 749389476

$$\mathbb{1}_S = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a complementary of the set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

- The geometry of A is completely lost, and only pairwise similarities between the data points are preserved.

- This is actually good because it will allow us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

WeChat: cstutorcs

Assignment Project Exam Help

- Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

Email: tutorcs@163.com

- Given a subset $S \subset V$ of vertices, we denote by $\mathbb{1}_S$ the indicator vector $\mathbb{1}_S = (f_1, \dots, f_n) \in \mathbb{R}^n$ where

$$f_i = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

- The *degree matrix* D is defined as a diagonal matrix with degree d_i of vertex v_i on the i^{th} entry of the diagonal of D and zeros everywhere off diagonal.



- So graph G is a summary of the set of data points $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.

- The geometry of A is lost, and only pairwise similarities between the data points are preserved.

- This is actually good because it will allow us to handle clustering of points which is not centre based, but is based on “local similarity” of data points, as we will see later.

WeChat: cstutorcs

Assignment Project Exam Help

- Given a subset $S \subset V$ of vertices, we denote by \bar{S} the complement $V \setminus S$ of S in V .

Email: tutorcs@163.com

- Given a subset $S \subset V$ of vertices, we denote by $\mathbb{1}_S$ the indicator vector $\mathbb{1}_S = (f_1, \dots, f_n) \in \mathbb{R}^n$ where

$$f_i = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

- For simplicity, we abbreviate $v_i \in S$ as $i \in S$.

Similarity Graphs

程序代写代做 CS编程辅导

- For any two subsets



we define

$$B) = \sum_{i \in S, j \in B} w_{ij}$$

WeChat: cstutorcs

and for any set S we define two types of measurements of the “size” of S :

Assignment Project Exam Help

① $|S|$ is the number of elements in S ,

② $\text{vol}(S) = \sum_{i \in S} d_i$

Email: tutorcs@163.com

- Recall that $d_i = \sum_{j=1}^n w_{ij}$ is the degree of vertex v_i .

QQ: 749389476

- A natural partition of vertices of a graph $G = (V, E)$ is into its connected components.

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导

- For any two subsets



we define

$$\text{vol}(B) = \sum_{i \in S, j \in B} w_{ij}$$

WeChat: cstutorcs

and for any set S we define two types of measurements of the “size” of S :

① $|S|$ is the number of elements in S ;

② $\text{vol}(S) = \sum_{i \in S} \text{vol}(i)$

Email: tutorcs@163.com

- Recall that $d_i = \sum_{j=1}^n w_{ij}$ is the degree of vertex v_i .

QQ: 749389476

- A natural partition of vertices of a graph $G = (V, E)$ is into its connected components.

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导

- For any two subsets



we define

$$w(S, B) = \sum_{i \in S, j \in B} w_{ij}$$

WeChat: cstutorcs

and for any set S we define two types of measurements of the “size” of S :

Assignment Project Exam Help

① $|S|$ is the number of elements in S ;

② $\text{vol}(S) = \sum_{i \in S} d_i$

Email: tutorcs@163.com

- Recall that $d_i = \sum_{j=1}^n w_{ij}$ is the degree of vertex v_i .

QQ: 749389476

- A natural partition of vertices of a graph $G = (V, E)$ is into its connected components.

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导

- For any two subsets



we define

$$\text{sim}(S, B) = \sum_{i \in S, j \in B} w_{ij}$$

WeChat: cstutorcs

and for any set S we define two types of measurements of the “size” of S :

Assignment Project Exam Help

① $|S|$ is the number of elements in S ,

② $\text{vol}(S) = \sum_{i \in S} d_i$

Email: tutorcs@163.com

- Recall that $d_i = \sum_{j=1}^n w_{ij}$ is the degree of vertex v_i .

QQ: 749389476

- A natural partition of vertices of a graph $G = (V, E)$ is into its connected components.

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导

- For any two subsets



we define

$$w(S, B) = \sum_{i \in S, j \in B} w_{ij}$$

WeChat: cstutorcs

and for any set S we define two types of measurements of the “size” of S :

Assignment Project Exam Help

① $|S|$ is the number of elements in S ,

② $\text{vol}(S) = \sum_{i \in S} d_i$

Email: tutorcs@163.com

- Recall that $d_i = \sum_{j=1}^n w_{ij}$ is the degree of vertex v_i .

QQ: 749389476

- A natural partition of vertices of a graph $G = (V, E)$ is into its connected components.

<https://tutorcs.com>

程序代写代做 CS编程辅导



- Given a set of data points $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ we associate vertices $\{v_1, \dots, v_n\}$ of a similarity graph G , but there are many ways how we can associate weights w_{ij} which measure the similarity of data points \mathbf{a}_i and \mathbf{a}_j that correspond to vertices v_i and v_j .

(1) The ε -neighbourhood graph:

- We connect all pairs of vertices v_i, v_j such that the distances between the associated data points $\mathbf{a}_i, \mathbf{a}_j$ are smaller than ε .
- The distance is usually the Euclidean distance

$$\|\mathbf{a}_i - \mathbf{a}_j\| = \sqrt{\sum_{p=1}^d (a_{ip} - a_{jp})^2},$$

where $\mathbf{a}_i = (a_{i1}, \dots, a_{id}) \in \mathbb{R}^d$.

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导



(2) The k -nearest neighbour graphs:

- There are two types of k -nearest neighbour graphs:
 - ① *Unidirectional k -nearest neighbour graph.* We connect v_i with v_j if either v_j is among k nearest neighbours of v_i or vice versa, v_i is among k nearest neighbours of v_j .
 - ② *Mutual k -nearest neighbour graph.* We connect v_i with v_j if both v_j is among k nearest neighbours of v_i and v_i is also among k closest neighbours of v_j .
- In both cases the edge is then weighted with the degree of similarity of the vertices v_i and v_j .

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导



(2) The k -nearest neighbour graphs:

- There are two forms of k -nearest neighbour graphs:
 - ① *Unidirectional k -nearest neighbour graph.* We connect v_i with v_j if either v_j is among k nearest neighbours of v_i or vice versa, v_i is among k nearest neighbours of v_j .
 - ② *Mutual k -nearest neighbour graph.* We connect v_i with v_j if both v_j is among k nearest neighbours of v_i and v_i is also among k closest neighbours of v_j .
- In both cases the edge is then weighted with the degree of similarity of the vertices v_i and v_j .

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导



(2) The k -nearest neighbour graphs:

- There are two forms of k -nearest neighbour graphs:
 - ① *Unidirectional k -nearest neighbour graph.* We connect v_i with v_j if either v_j is among k nearest neighbours of v_i or vice versa, v_i is among k nearest neighbours of v_j .
 - ② *Mutual k -nearest neighbour graph.* We connect v_i with v_j if both v_j is among k nearest neighbours of v_i and v_i is also among k closest neighbours of v_j .
- In both cases the edge is then weighted with the degree of similarity of the vertices v_i and v_j .

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导



(2) The k -nearest neighbour graphs:

- There are two forms of k -nearest neighbour graphs:
 - ① *Unidirectional k -nearest neighbour graph.* We connect v_i with v_j if either v_j is among k nearest neighbours of v_i or vice versa, v_i is among k nearest neighbours of v_j .
 - ② *Mutual k -nearest neighbour graph.* We connect v_i with v_j if both v_j is among k nearest neighbours of v_i and v_i is also among k closest neighbours of v_j .
- In both cases the edge is then weighted with the degree of similarity of the vertices v_i and v_j .

<https://tutorcs.com>

Similarity Graphs

程序代写代做 CS编程辅导

(3) The fully connected graphs:

- We simply connect all pairs of vertices v_i and v_j for which the corresponding vectors \mathbf{a}_i and \mathbf{a}_j have a strictly positive similarity, or similar to zero, higher than some prescribed threshold ε .
- To ensure that such a graph represents local neighbourhood relationships, the similarity measure must be chosen to respect such localisation constraints.
- Often we take weights which reflect such local similarities by the following formula.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

- Here σ is a parameter which determines “the size” of the neighbourhood, namely how fast the similarity decreases as distance increases.
- Unfortunately, there is not a simple way how to choose a similarity graph.
► The best is just to try several and pick the one which eventually produces the most informative clustering.

Similarity Graphs

程序代写代做 CS编程辅导

(3) The fully connected graphs:

- We simply connect all pairs of vertices v_i and v_j for which the corresponding distance $\| \mathbf{a}_i - \mathbf{a}_j \|$ between \mathbf{a}_i and \mathbf{a}_j have a strictly positive similarity, or similarly higher than some prescribed threshold ε .
- To ensure that such a graph represents local neighbourhood relationships, the similarity measure must be chosen to respect such localisation conditions.
- Often we take weights which reflect such local similarities by the following formula.

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

- Here σ is a parameter which determines “the size” of the neighbourhood, namely how fast the similarity decreases as distance increases.
- Unfortunately, there is not a simple way how to choose a similarity graph.
► The best is just to try several and pick the one which eventually produces the most informative clustering.

Similarity Graphs

程序代写代做 CS编程辅导

(3) The fully connected graphs:

- We simply connect all pairs of vertices v_i and v_j for which the corresponding vectors \mathbf{a}_i and \mathbf{a}_j have a strictly positive similarity, or similarly higher than some prescribed threshold ε .
- To ensure that the graph represents local neighbourhood relationships, the similarity measure must be chosen to respect such localisation conditions.
- Often we take weights which reflect such local similarities by the following formula:

$$w_{ij} = e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2\sigma^2}}$$

Email: tutorcs@163.com

QQ: 749389476

- Here σ is a parameter which determines “the size” of the neighbourhood, namely how fast the similarity decreases as distance increases.
- Unfortunately, there is not a simple way how to choose a similarity graph.
► The best is just to try several and pick the one which eventually produces the most informative clustering.

Similarity Graphs

程序代写代做 CS编程辅导

(3) The fully connected graphs:

- We simply connect all pairs of vertices v_i and v_j for which the corresponding vectors \mathbf{a}_i and \mathbf{a}_j have a strictly positive similarity, or similarly higher than some prescribed threshold ε .
- To ensure that the graph represents local neighbourhood relationships, the similarity measure must be chosen to respect such localisation conditions.
- Often we take weights which reflect such local similarities by the following formula:

$$w_{ij} = e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2\sigma^2}}$$

Email: tutorcs@163.com

QQ: 749389476

- Here σ is a parameter which determines “the size” of the neighbourhood, namely how fast the similarity decreases as distance increases.
- Unfortunately, there is not a simple way how to choose a similarity graph.
- The best is just to try several and pick the one which eventually produces the most informative clustering.

Similarity Graphs

程序代写代做 CS编程辅导

(3) The fully connected graphs:

- We simply connect all pairs of vertices v_i and v_j for which the corresponding data vectors \mathbf{a}_i and \mathbf{a}_j have a strictly positive similarity, or similarly higher than some prescribed threshold ε .
- To ensure that the graph represents local neighbourhood relationships, the similarity measure must be chosen to respect such localisation conditions.
- Often we take weights which reflect such local similarities by the following formula:

$$w_{ij} = e^{-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2\sigma^2}}$$

Email: tutorcs@163.com

QQ: 749389476

- Here σ is a parameter which determines “the size” of the neighbourhood, namely how fast the similarity decreases as distance increases.
- Unfortunately, there is not a simple way how to choose a similarity graph.
- The best is just to try several and pick the one which eventually produces the most informative clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

WeChat: cstutorcs

- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
- Graph Laplacians are Assignment Project Exam Help
- A matrix M of size $n \times n$ is *positive semi-definite* if for all vectors $f \in \mathbb{R}^n$ we have

Email: tutorcs@163.com

$$f^T M f \geq 0$$

QQ: 749389476

- From linear algebra <https://tutorcs.com> metric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

- WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
 - Graph Laplacians are positive semi-definite if for all vectors $f \in \mathbb{R}^n$ we have

$$f^T M f \geq 0$$

- From linear algebra a symmetric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

WeChat: cstutorcs

- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
- Graph Laplacians are Assignment Project Exam Help
- A matrix M of size $n \times n$ is *positive semi-definite* if for all vectors $f \in \mathbb{R}^n$ we have Email: tutorcs@163.com

$$f^T M f \geq 0$$

QQ: 749389476

- From linear algebra <https://tutorcs.com> metric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

- WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
 - Graph Laplacians are crucial for spectral clustering.
 - A matrix M of size $n \times n$ is *positive semi-definite* if for all vectors $f \in \mathbb{R}^n$ we have

$$f^T M f \geq 0$$

- From linear algebra <https://tutorcs.com> metric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

WeChat: cstutorcs

- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
- Graph Laplacians are crucial for spectral clustering.
- A matrix M of size $n \times n$ is *positive semi-definite* if for all vectors $f \in \mathbb{R}^n$ we have

Email: tutorcs@163.com

$$f^T M f \geq 0$$

QQ: 749389476

- From linear algebra a symmetric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

- WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
 - Graph Laplacians are crucial for spectral clustering.
 - A matrix M of size $n \times n$ is *positive semi-definite* if for all vectors $f \in \mathbb{R}^n$ we have

$$f^T M f \geq 0$$

- From linear algebra we know that a symmetric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

程序代写代做 CS编程辅导

- Recall that the $n \times n$ diagonal matrix D has the degrees d_i of vertices v_i on its diagonal, where $d_i = \sum_{j=1}^n w_{ij}$.
- The (unnormalised) Laplacian matrix L is defined as



$$L = D - W$$

where $W = (w_{ij})_{i,j=1}^n$

- WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
- Clearly, L is symmetric and it does not depend on w_{ii} , $1 \leq i \leq n$.
 - Graph Laplacians are crucial for spectral clustering.
 - A matrix M of size $n \times n$ is *positive semi-definite* if for all vectors $f \in \mathbb{R}^n$ we have

$$f^T M f \geq 0$$

- From linear algebra we know that a symmetric matrix is positive semi-definite iff all of its eigenvalues are real and non-negative.
- The next theorem summarises their main properties important for spectral clustering.

Spectral Graph Theory

- **Theorem:** The matrix $L = D - W$ has the following properties:

- (1) For every vector $f \in \mathbb{R}^n$



$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

- (2) L is a symmetric positive semi-definite matrix.
 - (3) The smallest eigenvalue of L is 0 and its corresponding eigenvector is $\mathbf{1} = (1, 1, \dots, 1)$.
- Proof:

Assignment Project Exam Help

$$(1) f^\top L f = f^\top D f - f^\top W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j$$

$$= \frac{1}{2} \left(\sum_{i=1}^n \left(\sum_{j=1}^n w_{ij} \right) f_i^2 - 2 \sum_{i,j=1}^n w_{ij} f_i f_j + \sum_{j=1}^n \left(\sum_{i=1}^n w_{ij} \right) f_j^2 \right)$$

$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2.$$

Spectral Graph Theory

- **Theorem:** The matrix $L = D - W$ has the following properties:

- (1) For every vector $f \in \mathbb{R}^n$.



$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

- (2) L is a symmetric positive semi-definite matrix.
 - (3) The smallest eigenvalue of L is 0 and its corresponding eigenvector is $\mathbf{1} = (1, 1, \dots, 1)$.
- Proof:

Assignment Project Exam Help

$$(1) f^\top L f = f^\top D f - f^\top W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j$$

$$= \frac{1}{2} \left(\sum_{i=1}^n \left(\sum_{j=1}^n w_{ij} \right) f_i^2 - 2 \sum_{i,j=1}^n w_{ij} f_i f_j + \sum_{j=1}^n \left(\sum_{i=1}^n w_{ij} \right) f_j^2 \right)$$

$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2.$$

Spectral Graph Theory

- **Theorem:** The matrix $L = D - W$ has the following properties:

- (1) For every vector $f \in \mathbb{R}^n$.



$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

- (2) L is a symmetric positive semi-definite matrix.
 - (3) The smallest eigenvalue of L is 0 and its corresponding eigenvector is $\mathbf{1} = (1, 1, \dots, 1)$.
- Proof:

Assignment Project Exam Help

$$(1) f^\top L f = f^\top D f - f^\top W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j$$

$$= \frac{1}{2} \left(\sum_{i=1}^n \left(\sum_{j=1}^n w_{ij} \right) f_i^2 - 2 \sum_{i,j=1}^n w_{ij} f_i f_j + \sum_{j=1}^n \left(\sum_{i=1}^n w_{ij} \right) f_j^2 \right)$$

$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2.$$

Spectral Graph Theory

- **Theorem:** The matrix $L = D - W$ has the following properties:

- (1) For every vector $f \in \mathbb{R}^n$.



$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

- (2) L is a symmetric positive semi-definite matrix.
 - (3) The smallest eigenvalue of L is 0 and its corresponding eigenvector is $\mathbf{1} = (1, 1, \dots, 1)$.
- Proof:

Assignment Project Exam Help

$$(1) f^\top L f = f^\top D f - f^\top W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j$$

$$= \frac{1}{2} \left(\sum_{i=1}^n \left(\sum_{j=1}^n w_{ij} \right) f_i^2 - 2 \sum_{i,j=1}^n w_{ij} f_i f_j + \sum_{j=1}^n \left(\sum_{i=1}^n w_{ij} \right) f_j^2 \right)$$

$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2.$$

Spectral Graph Theory

- **Theorem:** The matrix $L = D - W$ has the following properties:

(1) For every vector $f \in \mathbb{R}^n$.



$$= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

- (2) L is a symmetric positive semi-definite matrix.
- (3) The smallest eigenvalue of L is 0 and its corresponding eigenvector is $\mathbf{1} = (1, 1, \dots, 1)$.
- **Proof:**

Assignment Project Exam Help

$$\begin{aligned} (1) \quad f^\top L f &= f^\top D f - f^\top W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j \\ &= \frac{1}{2} \left(\sum_{i=1}^n \left(\sum_{j=1}^n w_{ij} \right) f_i^2 - 2 \sum_{i,j=1}^n w_{ij} f_i f_j + \sum_{j=1}^n \left(\sum_{i=1}^n w_{ij} \right) f_j^2 \right) \\ &= \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2. \end{aligned}$$

Spectral Graph Theory

程序代写代做 CS编程辅导

- (2) Since we have shown that $f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$ and since $w_{ij} \geq 0$, L satisfies $f^T L f \geq 0$ for all vectors f and is thus positive semi-definite.



- (3) Note that $L\mathbf{1} = D\mathbf{1}$. However, it is easy to see that both $D\mathbf{1}$ and $W\mathbf{1}$ produce the same vector with the i^{th} coordinate equal to the degree d_i of vertex v_i . Thus, $L\mathbf{1} = D\mathbf{1} = W\mathbf{1} = \mathbf{0} = 0 \cdot \mathbf{1}$. So, since $L\mathbf{1} = 0 \cdot \mathbf{1}$, 0 is the smallest eigenvalue of L (because all eigenvalues are non-negative) and 1

WeChat: cstutorcs
Assignment Project Exam Help

- Let M be an $n \times n$ matrix. Then an eigenvalue λ of this matrix has:
 - An algebraic multiplicity k if the characteristic polynomial $P_n(x)$ has a form $P_n(x) = (x - \lambda)^k Q(x)$ where $Q(x)$ is a polynomial of degree $n - k$. <https://tutorcs.com>
 - A geometric multiplicity k if λ has k linearly independent eigenvectors.

Spectral Graph Theory

程序代写代做 CS编程辅导

- (2) Since we have shown that $f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$ and since $w_{ij} \geq 0$, L satisfies $f^T L f \geq 0$ for all vectors f and is thus positive semi-definite.



- (3) Note that $L\mathbf{1} = D\mathbf{1} - W\mathbf{1}$. However, it is easy to see that both $D\mathbf{1}$ and $W\mathbf{1}$ produce the same vector with the i^{th} coordinate equal to the degree d_i of vertex v_i . Thus, $L\mathbf{1} = D\mathbf{1} - W\mathbf{1} = \mathbf{0} = 0 \cdot \mathbf{1}$. So, since $L\mathbf{1} = 0 \cdot \mathbf{1}$, 0 is the smallest eigenvalue of L (because all eigenvalues are non-negative) and 1

WeChat: cstutorcs
Assignment Project Exam Help

- Let M be an $n \times n$ matrix. Then an eigenvalue λ of this matrix has:
 - An algebraic multiplicity k if the characteristic polynomial $P_n(x)$ has a form $P_n(x) = (x - \lambda)^k Q(x)$ where $Q(x)$ is a polynomial of degree $n - k$. <https://tutorcs.com>
 - A geometric multiplicity k if λ has k linearly independent eigenvectors.

Spectral Graph Theory

程序代写代做 CS编程辅导

- (2) Since we have shown that $f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$ and since $w_{ij} \geq 0$, L satisfies $f^T L f \geq 0$ for all vectors f and is thus positive semi-definite.



- (3) Note that $L\mathbf{1} = D\mathbf{1} - W\mathbf{1}$. However, it is easy to see that both $D\mathbf{1}$ and $W\mathbf{1}$ produce the same vector with the i^{th} coordinate equal to the degree d_i of vertex v_i . Thus, $L\mathbf{1} = D\mathbf{1} - W\mathbf{1} = \mathbf{0} = 0 \cdot \mathbf{1}$. So, since $L\mathbf{1} = 0 \cdot \mathbf{1}$, 0 is the smallest eigenvalue of L (because all eigenvalues are non-negative) and $\mathbf{1}$ is the corresponding eigenvector.

- Let M be an $n \times n$ matrix. Then an eigenvalue λ of this matrix has:
 - An algebraic multiplicity k if the characteristic polynomial $P_n(x)$ has a form $P_n(x) = (x - \lambda)^k Q(x)$ where $Q(x)$ is a polynomial of degree $n - k$. <https://tutorcs.com>
 - A geometric multiplicity k if λ has k linearly independent eigenvectors.

Spectral Graph Theory

程序代写代做 CS编程辅导

- (2) Since we have shown that $f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$ and since $w_{ij} \geq 0$, L satisfies $f^T L f \geq 0$ for all vectors f and is thus positive semi-definite.



- (3) Note that $L\mathbf{1} = D\mathbf{1} - W\mathbf{1}$. However, it is easy to see that both $D\mathbf{1}$ and $W\mathbf{1}$ produce the same vector with the i^{th} coordinate equal to the degree d_i of vertex v_i . Thus, $L\mathbf{1} = D\mathbf{1} - W\mathbf{1} = \mathbf{0} = 0 \cdot \mathbf{1}$. So, since $L\mathbf{1} = 0 \cdot \mathbf{1}$, 0 is the smallest eigenvalue of L (because all eigenvalues are non-negative) and $\mathbf{1}$ is the corresponding eigenvector.

- Let M be an $n \times n$ matrix. Then an eigenvalue λ of this matrix has:
 - An algebraic multiplicity k if the characteristic polynomial $P_n(x)$ has a form $P_n(x) = (x - \lambda)^k Q(x)$ where $Q(x)$ is a polynomial of degree $n - k$.
 - A geometric multiplicity k if λ has k linearly independent eigenvectors.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS编程辅导

- (2) Since we have shown that $f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$ and since $w_{ij} \geq 0$, L satisfies $f^T L f \geq 0$ for all vectors f and is thus positive semi-definite.



- (3) Note that $L\mathbf{1} = D\mathbf{1} - W\mathbf{1}$. However, it is easy to see that both $D\mathbf{1}$ and $W\mathbf{1}$ produce the same vector with the i^{th} coordinate equal to the degree d_i of vertex v_i . Thus, $L\mathbf{1} = D\mathbf{1} - W\mathbf{1} = \mathbf{0} = 0 \cdot \mathbf{1}$. So, since $L\mathbf{1} = 0 \cdot \mathbf{1}$, 0 is the smallest eigenvalue of L (because all eigenvalues are non-negative) and $\mathbf{1}$ is the corresponding eigenvector.

- Let M be an $n \times n$ matrix. Then an eigenvalue λ of this matrix has:
 - An algebraic multiplicity k if the characteristic polynomial $P_n(x)$ has a form $P_n(x) = (x - \lambda)^k Q(x)$ where $Q(x)$ is a polynomial of degree $n - k$.
 - A geometric multiplicity k if λ has k linearly independent eigenvectors.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Spectral Graph Theory

- 程序代写代做 CS 编程辅导
- **Theorem:** Let G be an undirected weighted graph with n vertices and non-negative weights which has exactly k connected components A_1, \dots, A_k . Then the eigenvalues of L are equal to k and the eigenspace of eigenvalue 0 is spanned by the indicators $\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}$ of those components.
 - **Proof:** Assume first that the graph is connected, i.e., that it has exactly one connected component and let $f \in \mathbb{R}^n$ be an eigenvector with eigenvalue 0. Then, **WeChat: cstutorcs** thus also $f^\top L f = 0$. Consequently, by the previous theorem we have

Assignment Project Exam Help

$0 = f^\top L f = \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$
Email: tutorcs@163.com

QQ: 749389476

Thus, if two vertices v_i and v_j are connected by an edge, then $w_{ij} > 0$ and consequently $f_i = f_j$. Going along any path we get that the coordinates of f must be constant at all vertices along that path. Since G is connected, obviously all coordinates of f must be constant. Thus, $\mathbf{1}_V = (1, 1, \dots, 1)$ is an eigenvector. (Note that we do not normalise eigenvectors).

Spectral Graph Theory

- **Theorem:** Let G be an undirected weighted graph with n vertices and non-negative weights which has exactly k connected components A_1, \dots, A_k . Then the eigenvalues of L are equal to k and the eigenspace of eigenvalue 0 is spanned by the indicators $\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}$ of those components.
- **Proof:** Assume first that the graph is connected, i.e., that it has exactly one connected component and let $f \in \mathbb{R}^n$ be an eigenvector with eigenvalue 0. Then, $\Delta f = f = 0$ and thus also $f^\top L f = 0$. Consequently, by the previous theorem we have

Assignment Project Exam Help

$$0 = f^\top L f = \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

QQ: 749389476

Thus, if two vertices v_i and v_j are connected by an edge, then $w_{ij} > 0$ and consequently $f_i = f_j$. Going along any path we get that the coordinates of f must be constant at all vertices along that path. Since G is connected, obviously all coordinates of f must be constant. Thus, $\mathbf{1}_V = (1, 1, \dots, 1)$ is an eigenvector. (Note that we do not normalise eigenvectors).

Spectral Graph Theory

- **Theorem:** Let G be an undirected weighted graph with n vertices and non-negative weights which has exactly k connected components A_1, \dots, A_k . Then the eigenvalues of L are equal to k and the eigenspace of eigenvalue 0 is spanned by the indicators $\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}$ of those components.
- **Proof:** Assume first that the graph is connected, i.e., that it has exactly one connected component and let $f \in \mathbb{R}^n$ be an eigenvector with eigenvalue 0. Then, $\Delta f = f = 0$ and thus also $f^\top L f = 0$. Consequently, by the previous theorem we have

Assignment Project Exam Help

$$0 = f^\top L f = \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$$

QQ: 749389476

Thus, if two vertices v_i and v_j are connected by an edge, then $w_{ij} > 0$ and consequently $f_i = f_j$. Going along any path we get that the coordinates of f must be constant at all vertices along that path. Since G is connected, obviously all coordinates of f must be constant. Thus, $\mathbf{1}_V = (1, 1, \dots, 1)$ is an eigenvector. (Note that we do not normalise eigenvectors).

Spectral Graph Theory

程序代写代做 CS 编程辅导

- **Proof (continued):** Assume now that G has k connected components. We can assume that the vertices of the connected components are listed in order of the components they belong to. Thus, matrix W is a block matrix of the form



$$W = \begin{pmatrix} V_1 & & & 0 \\ W_2 & & & \\ \vdots & & & \\ 0 & & & W_k \end{pmatrix}$$

WeChat: cstutorcs
Assignment Project Exam Help

where the block W_i corresponds to the connected component A_i and with 0's outside the blocks.

Email: W_tutorcs@163.com

- But then $L = D - W$ also has the same structure:

QQ: 749389476

$$L = \begin{pmatrix} L_1 & & & 0 \\ & \ddots & & \\ 0 & & \ddots & L_k \end{pmatrix}$$

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS 编程辅导

- **Proof (continued):** Assume now that G has k connected components. We can assume that the vertices of the connected components are listed in order of the components they belong to. Thus, matrix W is a block matrix of the form



$$W = \begin{pmatrix} V_1 & W_2 & & 0 \\ 0 & V_2 & W_3 & \\ & & \ddots & \\ & & & V_k \end{pmatrix}$$

WeChat: cstutorcs
Assignment Project Exam Help

where the block W_i corresponds to the connected component A_i and with 0's outside the blocks.

Email: tutorcs@163.com

- But then $L = D - W$ also has the same structure:

QQ: 749389476

$$L = \begin{pmatrix} L_1 & & & 0 \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}$$

<https://tutorcs.com>

程序代写代做 CS编程辅导



- **Proof (continued)**: The eigenvalues of block matrices are the union of the eigenvalues of each block as a separate matrix, and the eigenvectors of a block matrix are the eigenvectors of the blocks with zeros appended outside each block. **WeChat: cstutorcs**
- Thus, since each L_i corresponds to a single connected component, it has 0 as the smallest eigenvalue. **Assignment Project Exam Help Email: tutorcs@163.com**
- Consequently, the matrix L has as many eigenvalues 0 as there are connected components, and its corresponding eigenvectors are the indicator vectors of the connected components.

QQ: 749389476

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS编程辅导



- **Proof (continued)**: Eigenvalues of block matrices are the union of the eigenvalues of each block as a separate matrix, and the eigenvectors of a block matrix are the eigenvectors of the blocks with zeros appended outside each block. **WeChat: cstutorcs**
- Thus, since each L_i corresponds to a single connected component, it has 0 as the smallest eigenvalue with an eigenvector $\mathbf{1}_{A_i}$. **Assignment Project Exam Help**
- Consequently, the matrix L has as many eigenvalues 0 as there are connected components, and its corresponding eigenvectors are the indicator vectors of the connected components. **Email: tutorcs@163.com**

QQ: 749389476

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS编程辅导



- **Proof (continued)**: Eigenvalues of block matrices are the union of the eigenvalues of each block as a separate matrix, and the eigenvectors of a block matrix are the eigenvectors of the blocks with zeros appended outside each block. **WeChat: cstutorcs**
- Thus, since each L_i corresponds to a single connected component, it has 0 as the smallest eigenvalue with an eigenvector $\mathbf{1}_{A_i}$. **Assignment Project Exam Help**
- Consequently, the matrix L has as many eigenvalues 0 as there are connected components, and the corresponding eigenvectors are the indicator vectors of the connected components. **Email: tutorcs@163.com**

QQ: 749389476

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS编程辅导

- Assume now that a graph G has k connected components which we would like to find.
- After forming the Laplacian matrix L we would use a standard software to find its eigenvalues.
- Such a software would output that 0 is an eigenvalue of L with multiplicity k and it would output k eigenvectors corresponding to eigenvalue 0.
- However, generally, these vectors would not be the indicators $\mathbb{1}_{A_i}$ of the connected components but k (mutually orthogonal) linear combinations of these indicator vectors, because any k orthogonal vectors in the eigensubspace corresponding to the eigenvalue 0 are equally good candidates.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

- Thus, we will obtain k vectors from \mathbb{R}^n which look like this:

QQ: 749389476

$$\mathbf{e}_1 = \alpha_1^1 \mathbb{1}_{A_1} + \alpha_2^1 \mathbb{1}_{A_2} + \dots + \alpha_k^1 \mathbb{1}_{A_k}$$

.....

$$\mathbf{e}_k = \alpha_1^k \mathbb{1}_{A_1} + \alpha_2^k \mathbb{1}_{A_2} + \dots + \alpha_k^k \mathbb{1}_{A_k}$$

Spectral Graph Theory

程序代写代做 CS编程辅导

- Assume now that a graph G has k connected components which we would like to find.
- After forming the Laplacian matrix L we would use a standard software to find its eigenvalues.
- Such a software would output that 0 is an eigenvalue of L with multiplicity k and it would output k eigenvectors corresponding to eigenvalue 0.
- However, generally, these vectors would not be the indicators $\mathbb{1}_{A_i}$ of the connected components but k (mutually orthogonal) linear combinations of these indicator vectors, because any k orthogonal vectors in the eigensubspace corresponding to the eigenvalue 0 are equally good candidates.
- Thus, we will obtain k vectors from \mathbb{R}^n which look like this:

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

$$\mathbf{e}_1 = \alpha_1^1 \mathbb{1}_{A_1} + \alpha_2^1 \mathbb{1}_{A_2} + \dots + \alpha_k^1 \mathbb{1}_{A_k}$$

.....

$$\mathbf{e}_k = \alpha_1^k \mathbb{1}_{A_1} + \alpha_2^k \mathbb{1}_{A_2} + \dots + \alpha_k^k \mathbb{1}_{A_k}$$

Spectral Graph Theory

程序代写代做 CS编程辅导

- Assume now that a graph G has k connected components which we would like to find.
- After forming the Laplacian matrix L we would use a standard software to find its eigenvalues.
- Such a software would output that 0 is an eigenvalue of L with multiplicity k and it would output k eigenvectors corresponding to eigenvalue 0.

WeChat: cstutorcs

- However, generally, these vectors would not be the indicators $\mathbb{1}_{A_i}$ of the connected components but k (mutually orthogonal) linear combinations of these indicator vectors, because any k orthogonal vectors in the eigensubspace corresponding to the eigenvalue 0 are equally good candidates.

Email: tutorcs@163.com

- Thus, we will obtain k vectors from \mathbb{R}^n which look like this:

QQ: 749389476

$$\mathbf{e}_1 = \alpha_1^1 \mathbb{1}_{A_1} + \alpha_2^1 \mathbb{1}_{A_2} + \dots + \alpha_k^1 \mathbb{1}_{A_k}$$

.....

$$\mathbf{e}_k = \alpha_1^k \mathbb{1}_{A_1} + \alpha_2^k \mathbb{1}_{A_2} + \dots + \alpha_k^k \mathbb{1}_{A_k}$$

Spectral Graph Theory

程序代写代做 CS编程辅导

- Assume now that a graph G has k connected components which we would like to find.
- After forming the Laplacian matrix L we would use a standard software to find its eigenvalues.
- Such a software would output that 0 is an eigenvalue of L with multiplicity k and it would output k eigenvectors corresponding to eigenvalue 0.
- However, generally, these vectors would not be the indicators $\mathbb{1}_{A_i}$ of the connected components but k (mutually orthogonal) linear combinations of these indicator vectors, because any k orthogonal vectors in the eigensubspace corresponding to the eigenvalue 0 are equally good candidates.
- Thus, we will obtain k vectors from \mathbb{R}^n which look like this:

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

$$\mathbf{e}_1 = \alpha_1^1 \mathbb{1}_{A_1} + \alpha_2^1 \mathbb{1}_{A_2} + \dots + \alpha_k^1 \mathbb{1}_{A_k}$$

.....

$$\mathbf{e}_k = \alpha_1^k \mathbb{1}_{A_1} + \alpha_2^k \mathbb{1}_{A_2} + \dots + \alpha_k^k \mathbb{1}_{A_k}$$

Spectral Graph Theory

程序代写代做 CS编程辅导

- Assume now that a graph G has k connected components which we would like to find.
- After forming the Laplacian matrix L we would use a standard software to find its eigenvalues.
- Such a software would output that 0 is an eigenvalue of L with multiplicity k and it would output k eigenvectors corresponding to eigenvalue 0.
- However, generally, these vectors would not be the indicators $\mathbb{1}_{A_i}$ of the connected components but k (mutually orthogonal) linear combinations of these indicator vectors, because any k orthogonal vectors in the eigensubspace corresponding to the eigenvalue 0 are equally good candidates.
- Thus, we will obtain k vectors from \mathbb{R}^n which look like this:

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

$$\mathbf{e}_1 = \alpha_1^1 \mathbb{1}_{A_1} + \alpha_2^1 \mathbb{1}_{A_2} + \dots + \alpha_k^1 \mathbb{1}_{A_k}$$

...

$$\mathbf{e}_k = \alpha_1^k \mathbb{1}_{A_1} + \alpha_2^k \mathbb{1}_{A_2} + \dots + \alpha_k^k \mathbb{1}_{A_k}$$

程序代写代做 CS编程辅导



- We can now form a matrix $\mathbf{e} \in \mathbb{R}^{n \times k}$ of size $n \times k$ whose k columns are the eigenvectors $\mathbf{e}_i \in \mathbb{R}^n$. Consider n vectors $\mathbf{y}_j \in \mathbb{R}^k$ which are the rows of E .
- Note that for every two vertices v_i, v_j which belongs to the same connected component A_m the corresponding vectors \mathbf{y}_i and \mathbf{y}_j are identical and equal ($\alpha^1, \alpha^2, \dots, \alpha^k$) because only $\mathbf{1}_{A_m}$ has 1's at positions i and j ; all other $\mathbf{1}_{A_l}$ for $l \neq m$ have zeros at these positions, given that v_i, v_j belong to the connected component A_m and that the components are disjoint.
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
- This allows us to easily identify the connected components, by selecting points i with identical corresponding vectors v_i .

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS编程辅导



- We can now form a matrix $\mathbf{e} \in \mathbb{R}^{n \times k}$ of size $n \times k$ whose k columns are the eigenvectors $\mathbf{e}_i \in \mathbb{R}^n$. Consider n vectors $\mathbf{y}_j \in \mathbb{R}^k$ which are the rows of E .
- Note that for every two vertices v_i, v_j which belongs to the same connected component A_m the corresponding vectors \mathbf{y}_i and \mathbf{y}_j are identical and equal $(\alpha_m^1, \alpha_m^2, \dots, \alpha_m^k)$ because only $\mathbf{1}_{A_m}$ has 1's at positions i and j ; all other $\mathbf{1}_{A_l}$ for $l \neq m$ have zeros at these positions, given that v_i, v_j belong to the connected component A_m and that the components are disjoint.
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
- This allows us to easily identify the connected components, by selecting points i with identical corresponding vectors v_i .

<https://tutorcs.com>

Spectral Graph Theory

程序代写代做 CS编程辅导



- We can now form a matrix $\mathbf{e} \in \mathbb{R}^{n \times k}$ of size $n \times k$ whose k columns are the eigenvectors $\mathbf{e}_i \in \mathbb{R}^n$. Consider n vectors $\mathbf{y}_j \in \mathbb{R}^k$ which are the rows of E .
- Note that for every two vertices v_i, v_j which belongs to the same connected component A_m the corresponding vectors \mathbf{y}_i and \mathbf{y}_j are identical and equal $(\alpha_m^1, \alpha_m^2, \dots, \alpha_m^k)$ because only $\mathbf{1}_{A_m}$ has 1's at positions i and j ; all other $\mathbf{1}_{A_l}$ for $l \neq m$ have zeros at these positions, given that v_i, v_j belong to the connected component A_m and that the components are disjoint.
- This allows us to easily identify the connected components, by selecting points i with identical corresponding vectors v_i .

<https://tutorcs.com>

Spectral Clustering

- When we cluster points into disjoint clusters we want any two vertices v_i, v_j with high similarity (i.e., with high weight w_{ij} of the corresponding edge (v_i, v_j)) to be in the same cluster, and any two vertices from different clusters either not be connected with an edge (i.e., $w_{ij} = 0$) or to be connected with a weight w_{ij} as small as possible.
- Thus, the clusters should, in a sense, “approximate connected components” of tightly connected vertices with weak edges between such “approximate components”.
- We can take k eigenvectors e_1, \dots, e_k corresponding to the k smallest eigenvalues (in place of eigenvalue 0 of multiplicity k) and form the corresponding matrix E with these eigenvectors as columns.
- We again consider the row vectors y_1, y_2, \dots, y_n . If the graph G had k connected components, then for every two points v_i and v_j from the same components we saw that the corresponding vectors y_i and y_j would be identical. **QQ: 749389476**
- For clustering this is no longer the case, but since within each optimal cluster the weights of edges between different clusters are low, we can hope that the points v_i and v_j from the same optimal cluster will have similar vectors y_i and y_j and points v_i and v_j from different optimal clusters will have substantially different vectors y_i and y_j .

程序代写代做 CS 编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Spectral Clustering

- When we cluster points into disjoint clusters we want any two vertices v_i, v_j with high similarity (i.e., with high weight w_{ij} of the corresponding edge (v_i, v_j)) to be in the same cluster, and any two vertices from different clusters either not be connected with an edge (i.e., $w_{ij} = 0$) or to be connected with a weight w_{ij} as small as possible.
- Thus, the clusters should in a sense, “approximate connected components” of tightly connected vertices with weak edges between such “approximate components”.
- We can take k eigenvectors e_1, \dots, e_k corresponding to the k smallest eigenvalues (in place of eigenvalue 0 of multiplicity k) and form the corresponding matrix E with these eigenvectors as columns.
- We again consider the row vectors y_1, y_2, \dots, y_n . If the graph G had k connected components, then for every two points v_i and v_j from the same components we saw that the corresponding vectors y_i and y_j would be identical. QQ: 749389476
- For clustering this is no longer the case, but since within each optimal cluster the weights of edges between different clusters are low, we can hope that the points v_i and v_j from the same optimal cluster will have similar vectors y_i and y_j and points v_i and v_j from different optimal clusters will have substantially different vectors y_i and y_j .

Spectral Clustering

- When we cluster points into disjoint clusters we want any two vertices v_i, v_j with high similarity (i.e., with high weight w_{ij} of the corresponding edge (v_i, v_j)) to be in the same cluster, and any two vertices from different clusters either not be connected with an edge (i.e., $w_{ij} = 0$) or to be connected with a weight w_{ij} as small as possible.
- Thus, the clusters should in a sense, “approximate connected components” of tightly connected vertices with weak edges between such “approximate components”.
- We can take k eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_k$ corresponding to the k smallest eigenvalues (in place of eigenvalue 0 of multiplicity k) and form the corresponding matrix E with these eigenvectors as columns.
- We again consider the row vectors $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$. If the graph G had k connected components, then for every two points v_i and v_j from the same components we saw that the corresponding vectors \mathbf{y}_i and \mathbf{y}_j would be identical. **QQ: 749389476**
- For clustering this is no longer the case, but since within each optimal cluster the weights of edges between different clusters are low, we can hope that the points v_i and v_j from the same optimal cluster will have similar vectors \mathbf{y}_i and \mathbf{y}_j and points v_i and v_j from different optimal clusters will have substantially different vectors \mathbf{y}_i and \mathbf{y}_j .

Spectral Clustering

- When we cluster points into disjoint clusters we want any two vertices v_i, v_j with high similarity (i.e., with high weight w_{ij} of the corresponding edge (v_i, v_j)) to be in the same cluster, and any two vertices from different clusters either not be connected with an edge (i.e., $w_{ij} = 0$) or to be connected with a weight w_{ij} as small as possible.
- Thus, the clusters should be, in a sense, “approximate connected components” of tightly connected vertices with weak edges between such “approximate components”.
- We can take k eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_k$ corresponding to the k smallest eigenvalues (in place of eigenvalue 0 of multiplicity k) and form the corresponding matrix E with these eigenvectors as columns.
- We again consider the row vectors $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$. If the graph G had k connected components, then for every two points v_i and v_j from the same components we saw that the corresponding vectors \mathbf{y}_i and \mathbf{y}_j would be identical.
- For clustering this is no longer the case, but since within each optimal cluster the weights of edges between different clusters are low, we can hope that the points v_i and v_j from the same optimal cluster will have similar vectors \mathbf{y}_i and \mathbf{y}_j and points v_i and v_j from different optimal clusters will have substantially different vectors \mathbf{y}_i and \mathbf{y}_j .

程序代写代做 CS 编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Spectral Clustering

- When we cluster points into disjoint clusters we want any two vertices v_i, v_j with high similarity (i.e., with high weight w_{ij} of the corresponding edge (v_i, v_j)) to be in the same cluster, and any two vertices from different clusters either not be connected with an edge (i.e., $w_{ij} = 0$) or to be connected with a weight w_{ij} as small as possible.
- Thus, the clusters should in a sense, “approximate connected components” of tightly connected vertices with weak edges between such “approximate components”.
- We can take k eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_k$ corresponding to the k smallest eigenvalues (in place of eigenvalue 0 of multiplicity k) and form the corresponding matrix E with these eigenvectors as columns.
- We again consider the row vectors $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$. If the graph G had k connected components, then for every two points v_i and v_j from the same components we saw that the corresponding vectors \mathbf{y}_i and \mathbf{y}_j would be identical.
- For clustering this is no longer the case, but since within each optimal cluster the weights of edges are high and the weights of edges between different clusters are low, we can hope that the points v_i and v_j from the same optimal cluster will have similar vectors \mathbf{y}_i and \mathbf{y}_j and points v_i and v_j from different optimal clusters will have substantially different vectors \mathbf{y}_i and \mathbf{y}_j .

程序代写代做 CS 编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

① Construct a similarity graph G by one of the ways described; let **WeChat: cstutorcs** $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.

② Compute the Laplacian $L = D - W$

③ Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues

Email: tutorcs@163.com

④ Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns

QQ:749389476

⑤ For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .

https://tutorcs.com

⑥ Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

① Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.

② Compute the Laplacian $L = D - W$

③ Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues

Email: tutorcs@163.com

④ Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns

QQ:749389476

⑤ For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .

https://tutorcs.com

⑥ Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- ① Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- ② Compute the Laplacian $L = D - W$
- ③ Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues
- ④ Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns
- ⑤ For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- ⑥ Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- Compute the Laplacian $L = D - W$.
- Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues.
- Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns.
- For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- Compute the Laplacian $L = D - W$.
- Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues.
- Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns.
- For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- Compute the Laplacian $L = D - W$.
- Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues.
- Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns.
- For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- Compute the Laplacian $L = D - W$.
- Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues.
- Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns.
- For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- Compute the Laplacian $L = D - W$.
- Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues.
- Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns.
- For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering

- The previous heuristic analysis suggests the following clustering algorithm which can produce clusters of similar points that are not centre based.
- Spectral Clustering**



Input: a set of data points $\{a_1, \dots, a_n\}$, number k of clusters to construct.

- Construct a similarity graph G by one of the ways described; let $W = \{w_{ij} : 1 \leq i, j \leq n\}$ be its weighted adjacency matrix.
- Compute the Laplacian $L = D - W$.
- Compute the k eigenvectors e_1, \dots, e_k of L which correspond to k smallest eigenvalues.
- Let E be the matrix of size $n \times k$ containing the eigenvectors e_1, \dots, e_k as columns.
- For $i = 1, \dots, n$, let y_i be the vector corresponding to the i^{th} row of E .
- Cluster points $\{y_1, \dots, y_n\}$ using the k-means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with A_i defined as $A_i = \{v_j : y_j \in C_i\}$.

Spectral Clustering as graph partitioning

程序代写代做 CS编程辅导

- For a given number k of subsets, the MinCut approach to graph partitioning simply consists of choosing a partition A_1, \dots, A_k which minimises



$$\text{cut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^n W(A_i, \overline{A_i}) \quad (1)$$

WeChat: cstutorcs

- (Note that if $k = 2$ this is just the standard MinCut problem like we had for Karger's algorithm)
Assignment Project Exam Help
- Recall that for any two sets A and B , we have $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$,
Email: tutorcs@163.com and that \overline{A} denotes the complement of set A .
- Note that the factor $\frac{1}{2}$ in (1) is present because every edge is counted twice.
- Unfortunately, minimising (1) is not a good idea.
- The reason is that minimising (2) often produces clusters some of which contain only a single vertex or just a few vertices.

Spectral Clustering as graph partitioning

- A better idea is to find a partition A_1, \dots, A_k which minimises

$$\text{Ratio}(\mathbf{A}_1, \dots, \mathbf{A}_k) = \frac{1}{2} \sum_{i=1}^n \frac{W(A_i, \overline{A_i})}{|A_i|} \quad (2)$$

- Having $|A_i|'$ s in the ratio encourages the algorithm to find clusters A_i which all have a reasonably large number of points, rather than just a few.
- For every partition A_1, \dots, A_k we can define the corresponding set of orthonormal indicator vectors $\mathbf{h}_j = (h_{1j}, h_{2j}, \dots, h_{nj})^\top$ by setting $\mathbf{h}_j = 1/|A_j| \cdot \mathbf{1}_{A_j}$, i.e., by letting

Email: tutorcs@163.com
$$h_{ij} = \begin{cases} \frac{1}{\sqrt{|A_j|}} & \text{if } v_i \in A_j \\ 0 & \text{if } v_i \notin A_j \end{cases}$$

- Note that

$$\|\mathbf{h}_j\|^2 = \sum_{v_i \in A_j} \left(\frac{1}{\sqrt{|A_j|}} \right)^2 = |A_j| \frac{1}{|A_j|} = 1$$

and, since A'_j 's are pairwise disjoint, $\mathbf{h}_m \cdot \mathbf{h}_l = 0$ if $m \neq l$.

Spectral Clustering as graph partitioning

- Also, note that we have shown that

$$\begin{aligned}\mathbf{h}_j^\top L \mathbf{h}_j &= \frac{1}{2} \sum_{m,l=1}^n w_m (\mathbf{h}_{mj} - \mathbf{h}_{lj})^2 \\&= \frac{1}{2} \left(\sum_{m \in A_j; l \notin A_j} w_{ml} (h_{mj} - h_{lj})^2 + \sum_{l \in A_j; m \notin A_j} w_{ml} (h_{mj} - h_{lj})^2 \right) \\&= \frac{1}{2} \left(\sum_{m \in A_j; l \notin A_j} \frac{1}{|A_j|} + \sum_{l \in A_j; m \notin A_j} \frac{1}{|A_j|} \right) \\&= \frac{\text{cut}(A_j, \overline{A_j})}{|A_j|} \quad \text{QQ: 749389476}\end{aligned}$$

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com

- Let H be the matrix of size $n \times k$ with vectors \mathbf{h}_j , $1 \leq j \leq k$ as columns.
- Then the above equality implies that the sum of the diagonal elements of $H^\top L H$, i.e. the trace of $H^\top L H$ satisfies

$$\text{Tr}(H^\top L H) = \sum_{j=1}^k \frac{\text{cut}(A_j, \overline{A_j})}{|A_j|}$$

Spectral Clustering as graph partitioning

- Since we have proved that

程序代写代做 CS 编程辅导

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{j=1}^k \frac{\text{cut}(A_j, \overline{A_j})}{|A_j|} = \text{Tr}(H^\top L H)$$


we can conclude that to minimise $\text{RatioCut}(A_1, \dots, A_k)$ we have to find disjoint sets A_1, \dots, A_k which minimise $\text{Tr}(H^\top L H)$ where the columns \mathbf{h}_j of H are of the form $\mathbf{h}_j = 1/|A_j| \cdot \mathbb{1}_{A_j}$.

- This is an NP hard problem, so we find only an approximate solution by solving the following relaxation of it:

WeChat: cs1tutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Find $H \in \mathbb{R}^{n \times k}$ which minimises $\text{Tr}(H^\top L H)$

QQ: 749389476

subject to the constraint $H^\top H = I$

<https://tutorcs.com>

- The Rayleigh-Ritz theorem from linear algebra tells us that such an $H \in \mathbb{R}^{n \times k}$ is obtained as the matrix with k eigenvectors corresponding to the k smallest eigenvalues of L as the k columns of H .

Spectral Clustering as graph partitioning

程序代写代做 CS编程辅导

- Just as before, these vectors might be approximations of linear combinations of the indicator functions $\mathbb{1}_{A_j}$, so we again cluster the n rows of H into k clusters C_1, \dots, C_k to obtain the partition A_1, \dots, A_k defined by



$v_j \in A_m$ if and only if the j^{th} row of H belongs to C_m

- But notice that this is precisely what our spectral clustering algorithm does.
- Thus, besides the original heuristics with the connected components, we now see that the spectral clustering algorithm finds an approximate solution to the problem of finding A_1, \dots, A_k which minimise the ratio cut

Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{j=1}^k \frac{\text{cut}(A_j, \overline{A_j})}{|A_j|}$$

Spectral Clustering as graph partitioning

- It is possible to normalise the Laplacian so that the solution approximately minimises the so called Ncut defined as



$$N(A_k) = \sum_{j=1}^k \frac{\text{cut}(A_j, \overline{A_j})}{\text{vol}(A_j)}$$

where $\text{vol}(A)$ is the sum of the degrees of all vertices in A :

WeChat: cstutorcs

$$\text{vol}(A) = \sum_{v_i \in A} d_i = \sum_{v_i \in A} \left(\sum_{j=1}^n w_{ij} \right)$$

Email: tutorcs@163.com

- This sometimes produces better clustering, which also has a nice interpretation via random walk on graphs (a random walk seldom switches between different clusters).
- You can find all the <https://tutorcs.com> really written tutorial by Ulrike von Luxburg from the Max Planck Institute for Biological Cybernetics, available at http://www.fml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf, which we have followed closely in a part of our presentation.