Discussion 10

2022.05.19

林维嘉 linwj@shanghaitech.edu.cn

Outline

Unsupervised Learning

- Clustering
- Dimension Reduction

Types of Learning

- Supervised Learning
  - Classification
  - Regression
- Semi-supervised Learning
- Active Learning
- Unsupervised Learning
  - Clustering
  - Dimension Reduction
- Reinforcement Learning
- …

Clustering Analysis

- Top-down
- Bottom-up
- Key questions:
  - How to measure proximity ?
  - How to choose the number of clusters ?
  - Initialization

K-means

Input: A set of n data points $\{x_1, x_2, \ldots, x_n\}$ in R^d and target # clusters k

Output: k representatives $c_1, c_2, \ldots, c_k$ in R^d

Objective: choose $c_1, c_2, \ldots, c_k$ to minimize

$$\min_c \sum_{i=1}^{n} \min_{j \in \{1,2,\ldots,k\}} ||x_i - c_j||^2$$

Lloyd's method

Input: A set of n data points $\{x_1, x_2, \ldots, x_n\}$ in R^d

Initialize: centers $c_1, c_2, \ldots, c_k$ in R^d and clusters $C_1, C_2, \ldots, C_k$ in any way.

Repeat until there is no further change in the cost.

- For each j: $C_j \leftarrow \{x \in S$ whose closest center is $c_j\}$
- For each j: $c_j \leftarrow$mean of C_j

Spectral Clustering

Basic Algorithm:

- Calculate the Laplacian L
- Calculate the first k eigenvectors (the eigenvectors corresponding to the k smallest eigenvalues of L)
- Consider the matrix formed by the first k eigenvectors; the $l\text{-}th$ row defines the features of graph node l
- Cluster the graph nodes based on these features (e.g. using k-means clustering)

Spectral Clustering

Construct a graph for all the data points: $G = (V, E, W)$

- V: vertices, in this case each data point is a vertex
- E: edges between two vertices
- W: weighted adjacency matrix, w_{ij} denotes the weight of the vertex between v_i and v_j
  - Non-negative: $w_{ij} \geq 0$
  - Symmetric: $w_{ij} = w_{ji}$
- Take the clustering problem as a graph cut problem

Spectral Clustering

First, we can construct a similarity matrix S of all the data points.

e.g. Euclidean Distance, $s_{ij} = ||x_i - x_j||_2^2$

Then, based on S, we can construct weighted adjacency matrix W

- ε-neighborhood graph

$$w_{ij} = \begin{cases} 0, s_{ij} > \varepsilon \\ \varepsilon, s_{ij} \ \leq \varepsilon \end{cases}$$

- KNN graph

$$w_{ij} = \begin{cases} 0, v_i \notin knn(v_j) \ or \ v_j \notin knn(v_i) \\ \dfrac{1}{s_{ij}}, v_i \in knn(v_j) \ and \ v_j \in knn(v_i) \end{cases}$$

- Fully connected graph

$$w_{ij} = e^{-\frac{||x_i - x_j||_2^2}{2\sigma^2}}$$

Spectral Clustering

Degree matrix D: diagonal

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

Unnormalized Graph Laplacian matrix: $L = D - W$

Properties of L:

(1) $\forall f \in R^n, f^T L f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$

(2) L is symmetric and positive semi-definite

(3) L's smallest eigenvalue is 0 and its corresponding eigenvector is the all one vector $\mathbf{1}$

(4) L has n non-negative, real-valued eigenvalues $\mathbf{0} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Spectral Clustering

Proof Property (1) by the definition of d_i

Plug $L = D - W$

$$
\begin{aligned}
f^T L f &= f^T D f - f^T W f \\
&= \sum_{i=1}^{n} D_{ii} f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j \\
&= \frac{1}{2}\left[\sum_{i=1}^{n} D_{ii} f_i^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} D_{jj} f_j^2\right] \\
&= \frac{1}{2}\left[\sum_{i=1}^{n}\left(\sum_{j=1}^{n} w_{ij}\right) f_i^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n}\left(\sum_{i=1}^{n} w_{ji}\right) f_j^2\right] \\
&= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2
\end{aligned}
$$

Spectral Clustering

Property (2) is obvious by Property (1)

Proof Property (3)

$$
\begin{aligned}
L\mathbf{1} &= (D - W)\mathbf{1} \\
&= D\mathbf{1} - W\mathbf{1} \\
&= \begin{bmatrix} D_{11} \\ \vdots \\ D_{nn} \end{bmatrix} - \begin{bmatrix} \sum_{j=1}^{n} w_{1j} \\ \vdots \\ \sum_{j=1}^{n} w_{nj} \end{bmatrix} \\
&= \mathbf{0} = 0 \times \mathbf{1}
\end{aligned}
$$

Property (4) is obvious by Property (1)-(3)

Spectral Clustering

Normalized Graph Laplacian matrix:

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

Properties of L_{sym} and L_{rw}:

(1) $\forall\, f \in R^n, f^T L_{sym} f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij} (\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}})^2$

(2) λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{\frac{1}{2}} u$

(3) λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigenproblem $Lu = \lambda D u$

(4) 0 is an eigenvalue of L_{rw} with the constant one vector $\mathbf{1}$ as eigenvector; 0 is an eigenvalue of L_{sym} with eigenvector $D^{\frac{1}{2}}\mathbf{1}$

(5)L_{sym} and L_{rw} are positive semi-definite and have n non-negative, real-valued eigenvalues $\mathbf{0} = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$

Spectral Clustering

$G = (V, E)$: an undirected graph with non-negative weights

Given a subset $A \subset V$, we denote its complement $V \setminus A$ by $\bar{A}$

For two subsets $A, B \subset V$, we define:

$$W(A,B) = \sum_{v_i \in A, v_j \in B} w_{ij}$$

The non-empty sets $A_1, A_2, \ldots, A_k$ form a partition of the graph $G = (V, E)$ if $A_i \cap A_j = \emptyset$ and $A_1 \cup \cdots \cup A_k = V$

For a partition $A_1, A_2, \ldots, A_k$, we define:

$$cut(A_1, A_2, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k} W(A_i, \overline{A_i})$$

Spectral Clustering

How to measure the "size" of a subset $A \subset V$?

- $|A| \leftarrow$ the number of vertices in A
- $vol(A) \leftarrow \sum_{i \in A} D_{ii}$

RatioCut:

$$RatioCut(A_1, A_2, \dots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

Ncut:

$$NCut(A_1, A_2, \dots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{vol(A_i)} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{vol(A_i)}$$

Spectral Clustering

RatioCut Case:

We define k indicator vectors $h_j = \left[h_{1j}, h_{2j}, \dots, h_{nj}\right]^T$

$$h_{ij} = \begin{cases} \dfrac{1}{\sqrt{|A_j|}}, if\ v_i \in A_j \\ \quad 0 \quad\ , otherwise \end{cases}$$

where $i = 1,2, \dots, n$ and $j = 1,2, \dots, k$

Spectral Clustering

Let us consider: $h_p{}^T L h_p = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(h_{ip} - h_{jp})^2$

$$= \frac{1}{2}\sum_{v_i \in A_p, v_j \in \overline{A_p}} w_{ij}\left(\frac{1}{\sqrt{|A_p|}} - 0\right)^2 + \frac{1}{2}\sum_{v_i \in \overline{A_p}, v_j \in A_p} w_{ij}\left(\frac{1}{\sqrt{|A_p|}} - 0\right)^2$$

$$= \frac{1}{2}\frac{1}{|A_p|}\left[W\left(A_p, \overline{A_P}\right) + w(\overline{A_P}, A_p)\right]$$

$$= \frac{cut\left(A_p, \overline{A_P}\right)}{|A_p|}$$

Spectral Clustering

Let $H = [h_1, h_2, \dots, h_k] \in R^{n\times k}$, which contains those k indicator vectors as columns.

Note that the columns in H are orthonormal to each other, that is $H^T H = I$

$$h_p{}^T L h_p = (H^T L H)_{pp}$$

$$RatioCut(A_1, A_2, \dots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

$$= \sum_{i=1}^{k} h_i{}^T L h_i$$

$$= \sum_{i=1}^{k} (H^T L H)_{ii}$$

$$= Tr(H^T L H)$$

Spectral Clustering

The problem of minimizing $RatioCut(A_1, A_2, \ldots, A_k)$ can be rewritten as :

$$\min_{A_1,\ldots,A_k} Tr(H^T L H)$$
$$s.t.\ H^T H = I$$

Still NP-hard !!!

Relaxation: allow the entries of the matrix H to take arbitrary real values.

$$\min_{H \in R^{n \times k}} Tr(H^T L H)$$
$$s.t.\ H^T H = I$$

Spectral Clustering

Recall: L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

According to Rayleigh-Ritz theorem, the solution is given by choosing H as the matrix which contains the first k eigenvectors of L as columns.

Spectral Clustering

NCut Case:

We define k indicator vectors $h_j = \left[h_{1j}, h_{2j}, \dots, h_{nj}\right]^T$

$$h_{ij} = \begin{cases} \dfrac{1}{\sqrt{vol(A_j)}}, if\ v_i \in A_j \\ 0 \quad , otherwise \end{cases}$$

where $i = 1,2,\dots,n$ and $j = 1,2,\dots,k$

$$h_p{}^T L h_p = \frac{cut\left(A_p, \overline{A_P}\right)}{vol(A_p)}$$

Spectral Clustering

Let $H = [h_1, h_2, \ldots, h_k] \in R^{n \times k}$, which contains those k indicator vectors as columns.

Note that $H^T H = I$

$$h_p{}^T L h_p = (H^T L H)_{pp}$$

$$h_p{}^T D h_p = 1 \Rightarrow H^T D H = I$$

The problem of minimizing $NCut(A_1, A_2, \ldots, A_k)$ can be rewritten as :

$$\min_{A_1, \ldots, A_k} Tr(H^T L H)$$

$$s.t.\ H^T D H = I$$

Spectral Clustering

Relaxation: allow the entries of the matrix H to take arbitrary real values.

$$\min_{H \in R^{n \times k}} Tr(H^T L H)$$
$$s.t.\ H^T D H = I$$

Let $B = D^{\frac{1}{2}} H \Rightarrow H = D^{-\frac{1}{2}} B$

$$\min_{H \in R^{n \times k}} Tr(B^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} B)$$
$$s.t.\ B^T B = I$$

According to Rayleigh-Ritz theorem, the solution is given by choosing B as the matrix which contains the first k eigenvectors of L_{sym} as columns.

Spectral Clustering

Basic Algorithm:

- Calculate the Laplacian L
- Calculate the first k eigenvectors (the eigenvectors corresponding to the k smallest eigenvalues of L)
- Consider the matrix formed by the first k eigenvectors; the $l\text{-}th$ row defines the features of graph node l
- Cluster the graph nodes based on these features (e.g. using k-means clustering)

PCA

A set of n data points $\{x_1, x_2, \dots, x_n\}$ in R^D , $X \in R^{D\times n}$

Principal components: let $v_1, v_2, \dots, v_d$ denote the d principal component

- Projections of data ($d \ll D$)
- Mutually Uncorrelated (orthogonal)

$v_i . v_j = 0,\ i \neq j$ and $v_i . v_j = 1,\ i = j$

- Ordered in variance

PCA

Centralization: $x_i - \bar{x}$

Covariance matrix: $\frac{1}{n} X X^T$

Objective function:

$$\max_{V} V^T X X^T V$$
$$s.t. V^T V = I$$

Apply Lagrange Multiplier:

$$(X X^T) V = V \Lambda$$

where Λ is a diagonal matrix

Kernel PCA

$\phi: R^D \to R^p \ (D < P) \ F: R^p$

Data matrix: $X = [x_1, x_2, \ldots, x_n] \in R^{D \times n}$; $\phi(X) = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)] \in R^{P \times n}$

Centralization

Covariance matrix: $C_F = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T = \frac{1}{n} \phi(X)\phi(X)^T$

Eigenvalue Decomposition: $C_F P = \lambda P$

Consider $\lambda \neq 0 \Rightarrow P = \frac{1}{n} \frac{1}{\lambda} \sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T P = \frac{1}{n} \sum_{i=1}^{n} \alpha_i \phi(x_i) = \frac{1}{n} \phi(X)\alpha$

Kernel PCA

$$\frac{1}{n}\phi(X)\phi(X)^T\frac{1}{n}\phi(X)\alpha = \lambda\frac{1}{n}\phi(X)\alpha$$

$$\Rightarrow \frac{1}{n}\phi(X)^T\phi(X)\phi(X)^T\phi(X)\alpha = \lambda\phi(X)^T\phi(X)\alpha$$

Let $K = \phi(X)^T\phi(X)$

$$\frac{1}{n}KK\alpha = \lambda K\alpha$$

where $K \in R^{n\times n}$ and $K_{ij} = \phi(x_i)^T\phi(x_j)$