

BMEG3105

Recap of Lecture 7

Clustering

Hierarchical clustering

Mahalanobis distance

Classification

K-nearest neighbors classification (KNN)

KNN code example

Comparison

Recap of Lecture 7

Clustering analysis

- Find groups of objects
 - similar to one another
 - different from other groups

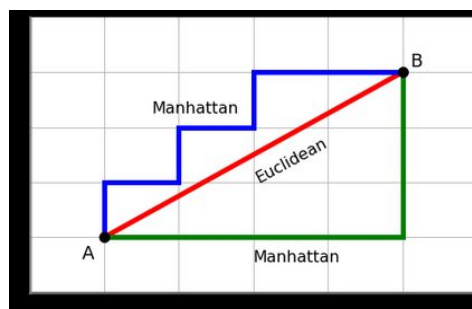
Minkowski distance

r=1: Manhattan distance, L_1 norm

- $|y_2 - y_1| + |x_2 - x_1|$

r=2: Euclidean distance, L_2 norm

- $\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$



$r \rightarrow \infty$: Supremum distance, L_{max} or L_∞ norm

$$\text{dist}(\mathbf{p}, \mathbf{q}) = \left(\sum_{k=1}^m |p_k - q_k|^r \right)^{\frac{1}{r}}$$

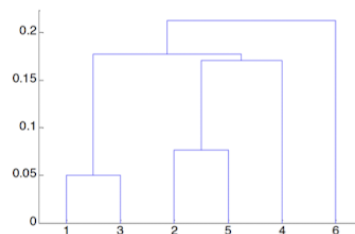
$$\begin{aligned} \lim_{r \rightarrow \infty} \text{dist}(\mathbf{p}, \mathbf{q}) &= \lim_{r \rightarrow \infty} \left(\sum_{k=1}^m |p_k - q_k|^r \right)^{\frac{1}{r}} \\ &= \max_{\mathbb{R}} [|p_k - q_k|] \end{aligned}$$

Clustering

- small intra-cluster distance, large inter-cluster distance
- distance matrix: Cosine similarity, Correlation, Euclidean distance, Manhattan distance, Mahalanobis distance

Hierarchical clustering

- nested cluster → dendrogram → taxonomies
 - dendrogram: tree-like diagram with merging clusters



- taxonomies eg: phylogenetic tree, gene clustering...

Steps:

1. Compute distance matrix
2. Let each data point be a cluster

3. Merge two closest clusters
4. Update distance matrix until only a single cluster remain

Updating distance matrix:

- min, max, group average distance, centroids distance

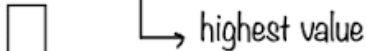
Explaining lecture examples:

distance matrix = correlation

range of Pearson's correlation: 0 - 1, higher = more correlated

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770					
At1g30720	0.9733				
At4g27450	-1	-0.9733			
At2g34930	0.9493	0.9909	-0.9493		
At2g05540	0.5774	0.562	-0.5774	0.4528	



the cell with highest value is related to gene2 and gene4

- gene2 and gene4: same cluster

construct another table, eliminating the cell with highest value

	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770					
At1g30720	0.9733				
At4g27450	-1	-0.9733 ->-0.9493			
At2g34930	0.9493 ->0.9733		-0.9493		
At2g05540	0.5774	0.562	-0.5774	0.4528 ->0.562	

check all the cells related to gene2 or gene4

cell (2,3): -0.9733 → -0.9493

- cell (2,3) = -0.9733 is related to gene2
- cell (3,4) = -0.9493 is related to gene4

-0.9733 < -0.9493

cell (2,3) < cell (3,4)

cell (2,3) → cell (3,4)

repeat the process until only one cluster remains

nodal tree can also be constructed

	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770					
At1g30720					
At4g27450	-0.5774	-0.5774			
At2g34930			-0.5774		
At2g05540			-0.5774		

Mahalanobis distance

- calculating distance considering data distribution
- how many standard deviation away

eg1:

set 1 contains A, B; set 2 contains C, D

- A-B = 1, std = 10
- C-D = 1, std = 1

eg2:

$\text{mahal}(\vec{p}, \vec{q}) = (\vec{p} - \vec{q})^T \Sigma^{-1} (\vec{p} - \vec{q})$

Given $\Sigma = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{pmatrix}$ $\vec{a} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ $\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $\vec{c} = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}$

$\det \Sigma = 0.3^2 - 0.2^2 = 0.05$

$\Sigma^{-1} = \frac{1}{0.05} \begin{pmatrix} 0.3 & -0.2 \\ -0.2 & 0.3 \end{pmatrix} = \begin{pmatrix} 6 & -4 \\ -4 & 6 \end{pmatrix}$

① $\text{Mahal}(\vec{a}, \vec{b})$

$\vec{a} - \vec{b} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$

$\text{Mahal}(\vec{a}, \vec{b}) = (\vec{a} - \vec{b})^T \Sigma^{-1} (\vec{a} - \vec{b})$

$= \begin{pmatrix} 0.5 & -0.5 \end{pmatrix} \begin{pmatrix} 6 & -4 \\ -4 & 6 \end{pmatrix} \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$

$= \begin{pmatrix} 5 & -5 \end{pmatrix} \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$

$= 5$

Inverse of 2x2 Matrix

If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then

$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

Inverse of A Determinant of A Adjoint of A

Note: A^{-1} exists only when $ad - bc \neq 0$

② $\text{Mahal}(\vec{a}, \vec{c})$

$\vec{a} - \vec{c} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$

$\text{Mahal}(\vec{a}, \vec{c}) = (\vec{a} - \vec{c})^T \Sigma^{-1} (\vec{a} - \vec{c})$

$= \begin{pmatrix} -1 & -1 \end{pmatrix} \begin{pmatrix} 6 & -4 \\ -4 & 6 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix}$

$= \begin{pmatrix} -2 & -2 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix}$

$= 4$

Mahalanobis distance: generalization of Euclidean distance

- when attributes are correlated, have different ranges of values (different variances)
- distribution is approximately Gaussian (normal distribution)
- gives less emphasis to the direction of largest variance than Euclidean
- If the attributes are relatively uncorrelated, but have different ranges, then standardizing the variables is sufficient.

Classification

class or category \subset attributes or features \subset records or training set
 (innermost) (outermost)

- assign class of unseen data, based on attributes & training set

Step:

1. Training data with calss
2. Trained by specified classification method
3. Input new data
4. Output of result

K-nearest neighbors classification (KNN)

- store all available instances
- classify new instance based on distance metric

Step:

1. Training-1: store all available instances
2. Training-2: normalization of data
3. Training-3: compute distance
4. Prediction-1: identify K most similar data
5. Prediction-2: mode class / return most frequent class label among K instance

Choose:

- value of K
 - range: 5-10 (for low-dimensional data set)
 - cross-validation
- weighing function (closer data point = higher weighting)

Note:

Updating the distance matrix → **×** need original data matrix

KNN code example

```
>>> X = [[0], [1], [2], [3]]
>>> y = [0, 0, 1, 1]
>>> from sklearn.neighbors import KNeighborsClassifier
>>> neigh = KNeighborsClassifier(n_neighbors=3)
>>> neigh.fit(X, y)
KNeighborsClassifier(...)
>>> print(neigh.predict([[1.1]]))
[0]
>>> print(neigh.predict_proba([[0.9]]))
[[0.666... 0.333...]]
```

scikit-learn library

Comparison

Clustering vs classification

	Clustering	Classification
Goal	Find similarity (clusters) in the data	Assign class to the new data
Data	Data without class	Training data with class and testing data without class
Classes	Unknown number of classes	Known number of classes
Output	The cluster index for each point	The class assignment of the testing data
Algorithm	One phase	Two phases (training and application)

Unsupervised vs Supervised learning:

	Unsupervised learning	Supervised learning
Function	analyse and cluster unlabelled data	Classify and predict outcomes, trained on labelled data
Example	clustering & dimension reduction	classification & regression