BMEG3105 Fall 2022

Classification

Lecturer: Yu LI (李煜) from CSE
Liyu95.com, liyu@cse.cuhk.edu.hk Wednesday, 28 September2022

Post-course survey results (from last lecture):

One question:

How do we know which type of distance measurement (e.g., correlation, cosine, minkowski distance, mahalanobis distance...) should we use? Similarly, how do we choose the type of cluster distance measurement (min, max, group average...) for the distance matrix after merging clusters?

Professor answers:

Correlation is a good way to measure gene similarity, because it can avoid some errors. For example, it can ignore the same difference in different gene expression comparison.

Four comments:

Fun.

Generally interesting.

It's great.

This lecture is tough. I don't really understand the material at all. But I will spend time to work on it after the lecture.

Recap from last lecture:

Clustering:

The reasons why we do the clustering, the definition of the clustering and the way to do the clustering.

Similarity and dissimilarity measurement:

Minkowski distance (City block distance, Euclidean distance, supremum distance)

Correlation: it measures the linear relationship between objects.

Mahalanobis distance: it calculates the distance considering the data distribution.

Hierarchical clustering:

Change a whole data to one cluster by iterations. In iterations, we calculate the distance, merge the 2 closest clustering, and update the data matrix in order.

Classification:

Why classification:

For classifying items, we can get better organization and know where to put the new items.

For classifying people, we can provide different treatment to different groups of patients and we can know if the customer is the targets of us.

For biology, we can give a new gene expression profile.

What is classification:

Given the collection of records (training set), find a method to assign the class of previous unseen records based on their other attributes and the training set as accurately as possible.

How to do classification:

First, get the training data with classes.

Then, get the classification method.

Then, input the data which needs to be classified.

Finally, give the results.

K-nearest neighbor classification:

Definition:

KNN is a simple algorithm that stores all available instances and classifies new instance based on a distance metric to the available ones.

Procedure:

Training process: Store the available training instances.

Predicting process: First, find the K training instances that are closest to the query instance. Then, return the most frequent class label among those K instances.

Importantly, during those two processes, the data should be normalized, and using a value of K somewhere between 5 and 10 gives good results for most low-dimensional data sets.

The problem of KNN:

Need to store all the data.

Need to calculate the distance matrix.

Predicting is slow.

A solution: we can use logistic function which is a formula considering the attributes and labels to help classify the new data. In this part, we use the training sets to get the weights of each attributes.

Clustering VS Classification:

	Clustering	Classification
Goal	Find similarity 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 testing data
Algorithm	One phase	Two phases (training and application)

Unsupervised learning and supervised learning:

Unsupervised learning:

Definition: Machine learning algorithm to analyze and cluster unlabeled data.

Example: clustering and dimension reduction

Supervised learning:

Definition: Machine learning algorithms to classify and predict outcomes, trained on labeled data.

Examples: classification and regression

Further Study:

Reinforcement learning in machine learning.

Decision tree/ SVM/ Bayesian
Model overfitting

Some useful links:

KNN in python:
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
Logistic regression in python:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html