Life Is a Random Walk

Abstract

In this project, we look into a game with 300 players, each having played the game on 8 different fields. We develop a model to determine different strategies for players in all datasets using Markov Chain theory. K-means clustering algorithm will be implemented to classify players into different groups. Simulations are given for further analysis and conclusions.

1. Introduction

There is a game with data files including 300 players and 8 different fields. The objective of the game is for a player to travel from coordinate (1,1) to coordinate(512,512). At each step, a player can choose either one of the seven directions or going back. The field consists of ares of four levels of difficulty from 0 (the easiest) to 3 (the hardest). Each player has uses the same strategy when playing on different fields. An algorithm is expected to be developed to classify 300 players into different groups.

2. Background

2.1 Markov Chain

We describe a Markov chain as follows: We have a set of states, S = {s1, s2, . . . , sr}. The process starts in one of these states and moves successively from one state to another. Each move is called a step. If the chain is currently in state si, then it moves to state sj at the next step with a probability denoted by pij, and this probability does not depend upon which states the chain was in before the current state.

The probabilities pij are called transition probabilities. The process can remain in the state it is in, and this occurs with probability pii. An initial probability distribution, defined on S, specifies the starting state. Usually this is done by specifying a particular state as the starting state.

a process satisfies the Markov property if one can make predictions for the future of the process based solely on its present state just as well as one

could knowing the process's full history. i.e., conditional on the present state of the system, its future and past are independent.

2.2 Transition Matrix

A square array for Markov Chain is called the matrix of transition probabilities, or the transition matrix. Let P be the transition matrix of a Markov chain, and let u be the probability vector which represents the starting distribution. Then the probability that the chain is in state si after n steps is the ith entry in the vector:

$$u(n) = uP(n)$$

2.3 K-means Algorithm

The K-means clustering technique is simple, and we begin with a description of the basic algorithm. We first choose K initial centroids, where K is a user- specified parameter, namely, the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update steps until no point changes clusters, or equivalently, until the centroids remain the same. The pseudocode is shown as follow:

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

3. Analysis & Method

The main objective of this project is to classify 300 players based on their strategies. This process can be regarded as unsupervised machine learning with a task of inferring a function to describe hidden structure from unlabelled data. Here we take clustering technique as our approach towards this problem.Cluster analysis or clustering is the task of grouping

a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups.

As for the preprocessing of the data files, we make use of the idea of Markov Chain theory. In total we have 8 data files recording how every single player takes their moves using their strategies in all 8 fields. For each player's strategy, there are in fact 9 options to choose from. A player in a certain position can move to 8 direction(right, left, top, bottom, top-right, top-left, bottom-right, bottom-left) around it or go back to former position before reaching the final point or running out of the move limitations. Based on the idea of Markov Chain theory, we assume that the decision for next move is only dependent to the present state. Therefore, in order to infer each strategy for each player, we extract the data of movement direction decisions to our training dataset. In my model, the mapping table is described as follow:

Coordinate	Direction	State
(1,0)	right	1
(-1,0)	left	2
(0-1)	top	3
(0,1)	bottom	4
(-1,-1)	top-right	5
(-1,1)	top-left	6
(1,1)	bottom-right	7
(-1,1)	bottom-left	8
/	back	9

After extracting and refining all the raw data files into training datasets, we use clustering technique to further achieve classification results on all player data. Among all clustering algorithms, we mainly use k-means clustering. In k-means clustering, the value of k (number of clusters) and the picking of the starting point are crucial to the result of clustering. Therefore, optimisations are needed to refine the basic k-means algorithm.

4. Implementation

According to Markov Chain theory and k-means algorithm mentioned above, we can implement the model for this project using the following functions:

4.1 Kmeans

The function Kmeans() mainly uses the refined k-means clustering model to complete data clustering and evaluations based on the preprocessing results. It returns a vector of clustering result with 3 main sets. Set C contains all centres for each cluster. Set I includes the final clustering result with group ID for each player. It indicates which group a player should be classified into. Parameter iter indicates the number of iteration for convergence. The algorithm for the selection of the starting points is implemented using the following technique: first it will choose the point as C1(probabilistic data vector) that is closest to the median of the random sample set; next it will choose the point with the longest distance from all the vector points in set C (so far there is only C1) from the random sample set and labelled as C2; this process will iterate up to k times.

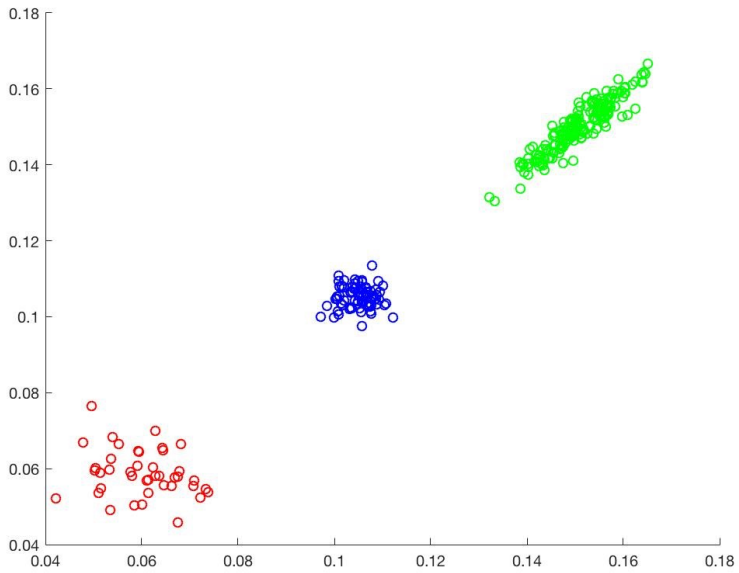
4.2 modelMain

The main file acts as the main model. It will classify all the player datasets given as the original resources. First of all, the clustering model will load the given dataset. There are some pre-processing operations for the raw data. Based on the information from the mapping table above, we transform all step records into direction records, which correspond to a set of state numbers and a set of direction vectors. By using the Kmeans functions, the clustering results will be generated in the modelMain function. The evaluation phase is to calculate the average distance between the centre point vector from set C and all the vectors of the players in the same group. The specific value of k corresponding to the minimum distance result will be the perfect selection for the number of clusters for the problem in this project.

5. Results

For performance evaluation, we test the trained models using 8 given datasets. First of all, we mainly focus on the clustering result using refine

k-means algorithm. In the Kmeans function, we set the rate of error tolerance TOL as 0.005, the number of iteration ITER as 40 and the number of clusters kappa as 3 according to our evaluation. The clustering result is demonstrated as follow:



During the clustering phase, we mainly pick three vectors out of ten in the probabilistic matrix that are more distinguishable to generate a graph of the experimental results and visualise the relationship between each cluster. As we can see from the figure above, 300 players can be classified into 3 clusters according to their strategy difference or transition matrix difference. Further clustering results are demonstrated in the attached result file. In addition, the central points of each cluster represent the average strategies for each group. In other words, different groups can be classified based on their average strategies. Details are shown below:

CENTER POINTS PROBABILISTIC DATA									
	state 1	**state 2**	**state 3**	**state 4**	**state 5**	**state 6**	**state 7**	**state 8**	**state 9**
Center Point 1	0.06023	0.04668	0.04554	0.05881	0.04552	0.06941	0.42264	0.04472	0.20641

CENTER POINTS PROBABILISTIC DATA									
Center Point 2	0.15089	0	0	0.15014	0	0	0.69896	0	0
Center Point 3	0.10505	0.08964	0.09003	0.10527	0.08949	0.08928	0.13762	0.08963	0.20394

6. Conclusion

Clustering algorithms have efficient performance on unsupervised machine learning. In this project we implemented k-means clustering algorithm on data files with 300 players. Using the idea of Markov Chain theory and transition matrix, we transform player strategies into probabilistic matrix and process clustering operations on our datasets. The experimental results are encouraging and the player data files have been properly classified according to their strategies differences.

7. Reference

[1] K. Alsabti, S. Ranka, and V. Singh. An efficient k-means clustering algorithm. In Proc. 1st Workshop on High Perfor- mance Data Mining, 1998.

[2] Arthu, D. and S. Vassilvitskii, 2007. K-means++: The advantages of careful seeding. Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis, Jan. 7-9, ACM Press, New Orleans, Louisiana, pp:1027-1035.

[3] Bremaud, P. (1999). Markov Chains. Gibbs Fields, Monte Carlo Simulation, and Queues. Springer, New York.