Machine Learning for Investor Behaviours on Cryptocurrencies

Chengkai Lu

Submitted for the Degree of Master of Science in

Data Science and Analytics with a Year in Industry

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

June 3, 2018

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name:

Date of Submission:

Signature:

Abstract

Cryptocurrencies like Bitcoin and Ripple are becoming popular in these years. They can be obtained through mining or transactions. Every individual cryptocurrency can be transferred directly using a public key in digital wallets. Their price is mostly tied to supply/demand and hard to be interfered by governments. In addition, cryptocurrencies have some dependencies because some of them like Dogecoin can only be bought by some major cryptocurrencies such as Bitcoin and Ethereum in cryptocurrency exchanges.

K-means clustering is a popular unsupervised learning algorithm for grouping unlabelled data. It aims on finding the natural way of separating observations into diffident clusters and is quite popular on the area of marketing for discovering potential customer behaviour. In other words, it might have the ability to discover an implicit pattern on the investors' behaviour.

The concept of deep learning becomes very popular in recent years because of better computation on machines and larger scale of data, and it can predict data more precisely even though the difficulty of explanation. Recurrent Neural Network is a multi-layer neural network with time-series in Deep Learning, and Long Short-Term Memory is one of the most popular one in it because of its capability of eliminating vanishing/exploding gradient problems. It performs well on time-series dataset especially on text and voice recognition. Accordingly, experiment on application of LSTM to the price of different cryptocurrencies may produce some unexpected results.

An implementation of modern algorithms combining nearly pure supply/demand problem is highly possible to capture investors investing behaviours. If patterns of investing behaviours are found, the most/least profitable products can be predicted accordingly and be used for investment.

Contents

List of Figures

List of Tables

Acknowledgement

1 Introduction

1.1 Motivation

1.2 Aims And Objectives

1.3 Project Structure

1.3.1 Technologies

1.3.2 Programme Structure

1.3.3 Report Structure

2 Background Research

2.1 Correlation Analysis

In statistics, covariance(2.1) defines the variability between two individual attributes, which means the level of influence from one feature to another. The numbers correspond to similarity/dissimilarity of the two variables. Positive numbers represent a similar behaviour between them, and vice versa[1].

$$\mathrm{cov}(X, Y) = \mathrm{E}[(X - \mu_X)(Y - \mu_Y)] \tag{2.1}$$

where:

- E is the expectation
- X, Y are vectors of all the samples
- cov is the covariance
- μ_X is the mean of X
- μ_Y is the mean of Y

However, if we want to measure the strength of the linear relationship in between, covariance is not enough. We also need to consider the variance in each feature to tell whether the linear relationship is strong. In this case, the correlation was introduced. Correlation is a normalised form of covariance. It restricts the numbers to a certain range which shows how strong the relationship is. The most commonly used correlation coefficient is Pearson correlation coefficient(2.2). it is calculated by considering the standard deviation of both groups. This can ensure that dispersion of either attribute does not interfere our identification on the strength of mutual

linear relationships[5].

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{2.2}$$

where:

- ρ is the Pearson correlation coefficient
- X, Y are vectors of all the samples
- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

The value of Pearson correlation coefficient is always between -1 and 1. A positive number means a positive linear correlation, and a negative number means a negative linear correlation. The closer the number towards the extremes, the stronger the relationship is. If the number is 0, it means there is no linear correlation among the pair(2.3).

$$\text{relationship} = \begin{cases} \text{total positive linear correlation} & \text{if } \rho = 1 \\ \text{positive linear correlation} & \text{if } \rho > 0 \\ \text{no linear correlation} & \text{if } \rho = 0 \\ \text{negative linear correlation} & \text{if } \rho < 0 \\ \text{total negative linear correlation} & \text{if } \rho = -1 \end{cases} \tag{2.3}$$

2.1.1 Correlation Matrix

Given a set of data with multiple attributes, we may want to tell people how these attributes interact with each other. In addition, the result of analysis, especially in a simple regression, may not be reasonable when those features are highly dependent.

To achieve this, we can create a matrix which contains all the correlation coefficient calculated from the expanded equation(2.4) with a set of given samples.

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.4}$$

where:

- ρ is the Pearson correlation coefficient
- n is the sample size
- x_i, y_i are the single samples indexed with i

- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean)
- $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ (the sample mean)

A correlation matrix is a symmetirc matrix to its main diagonal. The values on the main diagonal always equal to 1 because the attributes are fully dependent on themselves. Table 1 gives an example of how a correlation matrix looks like.

Features	f1	f2	f3	f4	f5
f1	1	0.74	-0.38	0.12	0.43
f2	0.74	1	0.26	0.88	-0.57
f3	-0.38	0.26	1	0.61	0.59
f4	0.12	0.88	0.61	1	-0.22
f5	0.43	-0.57	0.59	-0.22	1

Table 1: Correlation Matrix

2.2 Principal Component Analysis

It is always challenging to analyse a dataset with high-dimensional data points. Due to the curse of dimensionality, which was discovered by Richard Ernest Bellman in 1961, higher-dimensional space increases the difficulties of analysing and organising data exponentially[2]. Especially in machine learning, given a certain number of samples, the accuracy of predictions on these samples will increase followed by the rising dimensions to a peak but then gradually drop. This is known as Hughes phenomenon[7]. Figure 1 shows how the dimensionality influence the accuracy of predictions.

In order to reduce the dimensionality, there are two approaches can be implemented:

- Feature Selection: To select a subset that is more informative or relevant among all the attributes[8].
- Feature Extraction: To generate new features from the initial attributes of existing data[3].

The reasons and benefits of executing dimensionality reduction can be summarised as follow:

1. Computational efficiency: Fewer features mean less computation on dissimilarity between pairs of data points and lower arithmetic complexity. It also implies less storage usage as the variables in each sample decrease.

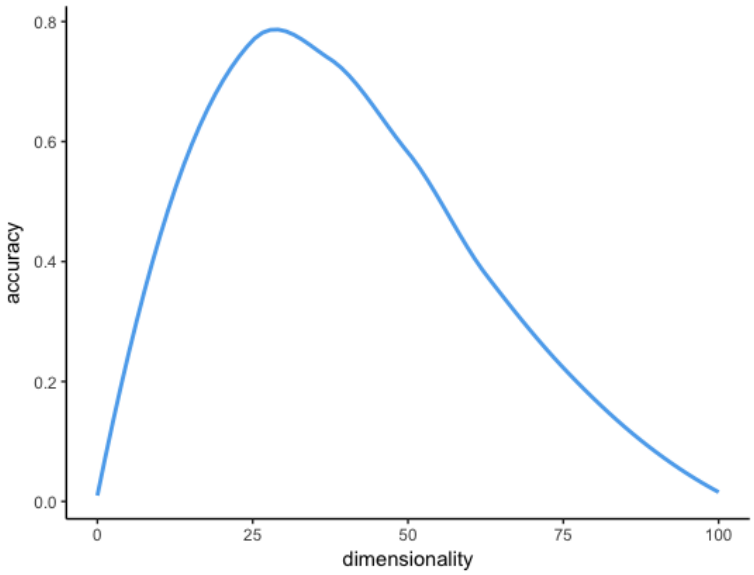


Figure 1: Hughes phenomenon

2. Statistical generalisation: By removing noise or irrelevant information from the inputs for building models, the prediction rules can be more general among the datasets.

3. Better explanation: Visualising a lower-dimensional space is much easier. We can effortlessly illustrate the structure of data when the dimension is lower than 3. Higher-dimensional space will be more challenging to visualise, explain and comprehend.

Principal component analysis(PCA) is a method for feature extraction. It projects features onto a lower-dimensional space. A traditional PCA is a kind of single representation approach as opposed to classification on revealing underlying information in a lower-dimensional space with a linear function.

In practice, an optimal mapping function is usually non-linear. In order to fit the data in a non-linear way, we can apply a kernel method on top of the traditional PCA, and this is called kernel PCA. It performs a linear PCA mapping in a higher dimensional kernel Hilbert space to provide a better classification. The kernel can be a polynomial function, a radial function or other functions[4]. However, in this project, we will assume that the relationship between the dimensions(cryptocurrencies) are linear and will

only use a standard linear PCA to perform the dimensionality reduction.

2.2.1 Principal Components

The new features derived are called principal components(PCs). They represent new orthogonal axes in an order based on the amount of information it contains.

2.3 Clustering

Clustering, cluster analysis or data segmentation is a non-parametric algorithm in the subtree of unsupervised learning. It is used to separate data into different groups using their dissimilarities(similarities) or possible distributions. Unlike supervised learning, this type of learning algorithms does not have any indicator for assessing the quality of results, and this means that it does not have any meaning or objective itself. Instead, it discovers the distribution of data and uses the definition given by people who have the specific domain knowledge. By giving the rules for partitioning data self-defined meanings, useful information can be obtained and utilised in different domains[8].

In general, clustering can be defined into two types, parametric and non-parametric. A parametric clustering groups the clusters with a a assumed density function which is usually a Gaussian, while a non-parametric one does not have any assumed distribution, it only aims on finding natural groupings within the givien dataset. In this project, we will only focus on the K-means algorithm in non-parametric methods.

K-means clustering and hierarchical clustering are two of the most popular methods in the non-parametric cluster analysis. In K-means clustering, we specify the number of groups we want to classify into. In contrast, hierarchical clustering does not have an initial number of clusters that we want for the result. It instead shows all the possible clusters into a tree structure and allows us to choose the number of clusters we want at the end.

2.3.1 K-means Clustering

K-means clustering is a intuitive approach which allows us to separate data points into distinct groups. To implement this, we first need to specify an initial number of clusters - K, and then randomly assign a number(cluster) from 1 to K to each object(data point). In this case, the clusters will have two features[6]:

1. $C_1 \cup C_2 \cup ... \cup C_K = \{O_1, O_2, ..., O_n\}$
2. $C_k \cap C_{k'} = \emptyset, \quad \text{for} \wedge k \neq k'$

where:

- C_k is the kth cluster
- O_n is the nth object

These properties mean that each single object will be in exactly one cluster and the clusters does not overlap. After this initial setting, we want to optimise(2.5) the grouping because the previous assignment is just a random initialisation. We want to make sure that the objects are concentrated which means the data point fit the best in the assigned cluster.

$$\underset{\mathbf{C_1},...,\mathbf{C_K}}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j}) \right\} \tag{2.5}$$

where:

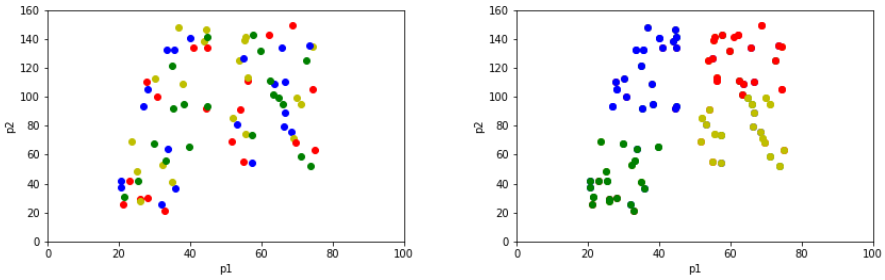
- C_k is the kth cluster
- x_{ij} is the jth attribute of the ith object

To fulfil the condition above which is (2.5), we can simplify the apporach as below to classify our data points into the multiple clusters[6]:

1. Randomly assign an initial number from 1 to K to each observation.
2. Iterate over the following steps until the assigned cluster of each observation stops changing:
   (a) for i in range(1, K):
   Compute the centroid of each cluster which is the mean of vectors with the same k(cluster) assigned.
   (b) Calculate the distance(Euclidean distance) between each object and each of the clusters.
   (c) Assign the nearest k(cluster) to each observation.

Figure 2 shows the difference after an optimisation of the cluster assignment. The colors indicate different clusters, and the groups are separated after the implementation of K-means clustering.

K = 4



(a) Before Iteration (b) After Iteration

Figure 2: K-means Clustering

2.4 Recurrent Neural Network

2.4.1 Long Short-Term Memory

3 K-means Clustering on Price Fluctuations

3.1 Data Prepocessing

3.1.1 Data Formats

3.1.2 Normalisation

3.2 Data Analysis

3.2.1 Shifting Along Timeline

3.2.2 Model Training

3.3 Result

References

[1] Wikipedia contributors. Covariance — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Covariance`. [Online; accessed 26-May-2018].

[2] Wikipedia contributors. Curse of dimensionality — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Curse_of_dimensionality. [Online; accessed 27-May-2018].

[3] Wikipedia contributors. Feature selection — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Feature_selection. [Online; accessed 27-May-2018].

[4] Wikipedia contributors. Kernel principal component analysis — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Kernel_principal_component_analysis. [Online; accessed 28-May-2018].

[5] Wikipedia contributors. Pearson correlation coefficient — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. [Online; accessed 26-May-2018].

[6] Trevor Hastie Gareth James, Daniela Witten and Robert Tibshirani. *An Introduction to Statistical Learning, with Applications in R*. Springer, New York, sixth edition, 2015.

[7] G. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, January 1968.

[8] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, second edition, 2009.