Survey Paper
Pattern Matching and its application in Stock Market
Lam Nguyen

ABSTRACT

In this survey paper, we overview the methods currently published for graph similarity scoring and matching. We also examine different existing methods of predicting stock prices in various environments and their effectiveness. Some ideas on the application areas of graph matching are given.

Keywords
Pattern Matching, Artificial Intelligence, Support Vector Machine, Neural Networks.

1. Introduction

There are many different methods to approach the pattern matching problem, many of them proved to be very effective.

1.1 Outline of the Survey

We begin by providing basic definitions in Section 2. In section 3, we proceed to a discussion of graph similarity scoring and matching using coupled node-edge scoring as well as how BLAST 2 Sequences can be used to determine plots similarity score. Section 4 discusses some of the algorithms used for stock price prediction. Section 5 is about Rete, a fast algorithm for pattern matching problem, and in Section 6 we conclude the survey.

2. Terminology and definitions

2.1. Graph in Computer Science

In computer science, graph is an abstract data type that is used to implement the undirected and directed graph concepts. A graph data type consists for a finite set of nodes and a set of unordered pairs of these vertices called edges.

In some cases, a graph also has *weight* values assigned to edges, thus called weighted graph. Two main kinds of graph are undirected and directed graph. Figure 4 represents a directed weighted graph while Figure 5 represents an undirected graph. With a good graph similarity scoring, we can find and match multiple similar graphs, thus finding closely related sectors from those graphs.

2.2. Machine Learning and Statistical Classification

In the field of computer science, machine learning is the study of algorithms that can learn and make predictions from data, not from strictly static program instructions. In other words, machine learning gives the computer the ability to learn without being explicitly programmed.

In the field of machine learning, statistical classification is the problem of assigning a new observation into a category based on a training set of data where the category is known. Classification is an example of pattern recognition. An algorithm that implements classification is known as a classifier, which map input data to a category.

Supervised learning is the machine learning task of inferring a function from labeled training data, meaning that the training data consist of a set of training examples. The training data set normally include pair of input and desired output. The algorithm analyzes the training data set that was given and then can be used to map new examples.

2.2. Regression analysis

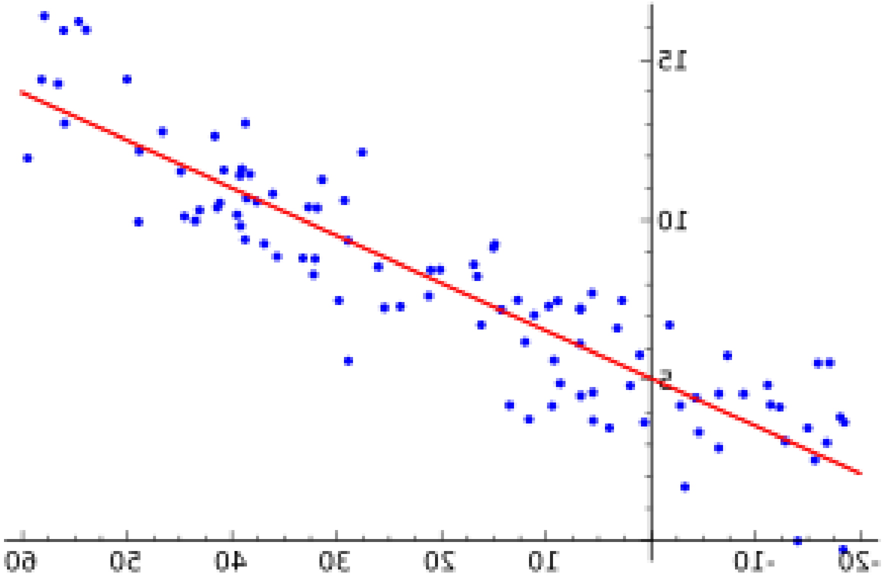


Figure 1. A linear regression

In statistical modeling, regression analysis is a process for estimating the relationship among variables. In other words, regression analysis help us to study how the value of a dependent value changes when we vary one independent variable while keeping the other independent variable constant.

2.3. Support Vector Machines

Support vector machines are supervised learning models that analyze data used for classification and regression analysis. SVM (support vector machine) is a binary linear classifier, meaning it’s given a training data set consists of examples that are marked to belong to one group or the other of two categories. The SVM algorithm then build a model that assign new values to one of the two groups.

Besides performing linear classification, SV can also perform non-linear classification using a kernel method, implicitly mapping their inputs into high-dimensional feature spaces.

2.4. Modular neural network

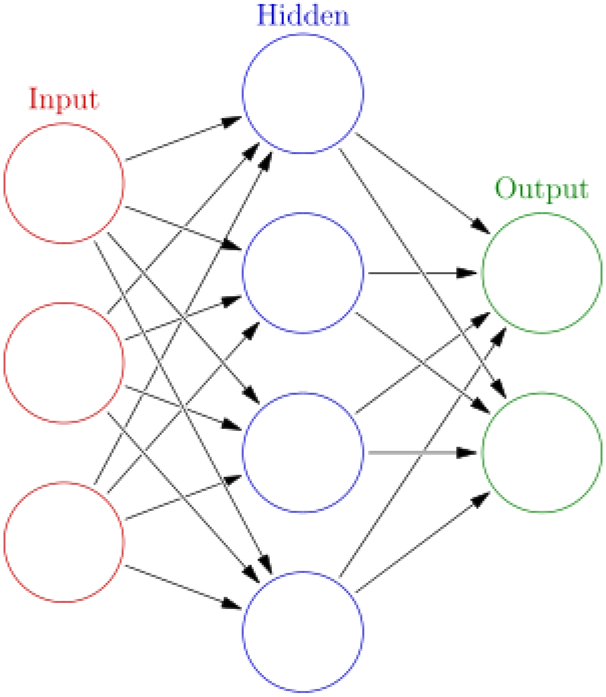


Figure 2. An artificial neural network

A modular neural network is an artificial neural network characterized by a series of independent neural networks moderated by some intermediary. An artificial neural network is normally inspired by the biological modularization found in the brain. An advantage that a modular neural network has over a normal artificial neural network is the reduced number of components, since the nodes are modularized and more manageable. In other words, modular neural network is a collection of different neural networks that work together to produce the result. These neural networks sometimes even interconnected and use each other’s output as input, thus increasing the complexity and performance of the algorithm.

2.5. Genetic algorithm

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions and normally used to generate high-quality solutions to optimization and search problems, using operators such as mutation, crossover, and selection.

2.6. Fuzzy neural network

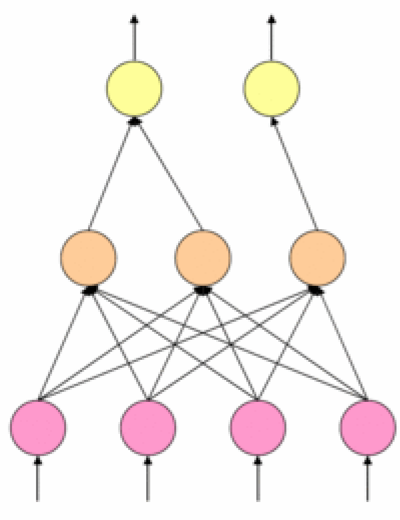


Figure 3. The architecture of a neuro-fuzzy system

A fuzzy neural network is a learning machine that finds the parameters of a fuzzy system by exploiting approximation techniques from neural networks. In other words, it is the combination of fuzzy system and neural networks. When combined, the algorithm can overcome the disadvantages of both fuzzy system and a neural network. A neural network can only be useful if the problem is expressed by sufficient observed examples, which means there is no easy way to extract comprehensible rules from its structure. On the other hand, a fuzzy system requires linguistic rules instead of examples as prior knowledge.

A fuzzy system or fuzzy set is a mathematical model of vague qualitative or quantitative data, frequently generated using the natural language.

3. Graph similarity scoring and matching

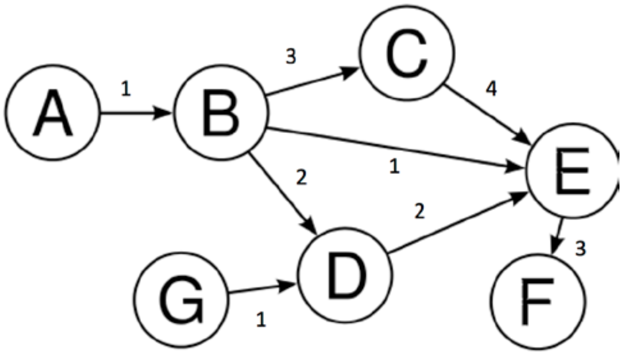


Figure 4. A weighted graph

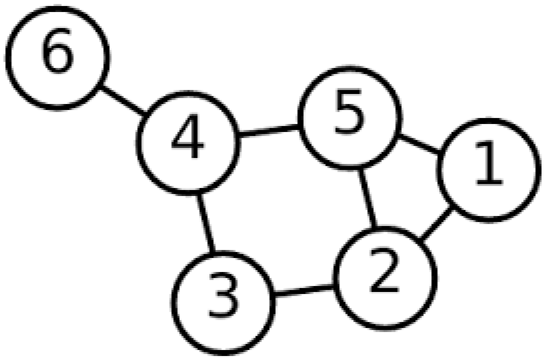


Figure 5. An undirected graph

3.1. Coupled node-edge scoring

Zager et al. (2008) proposed using the similarity of local neighborhoods to derive pairwise similarity score for the nodes of two different graphs. This method is used to generate both node and edge similarity scores, which then can be used for graph matching. To score the "similarity" between two graphs, the authors choose to use coupled node-edge scoring, which assume that two graph elements are similar if their neighborhoods are similar. In other words, an edge in graph A is like an edge in graph B if their respective source and terminal nodes are similar.

Not only can this algorithm help to score and match different graphs, it can also be applied to pairs of isomorphic graphs to generate self-similarity matrices. There is also the possibility of identifying two or even more similar sectors of one graph, thus providing a hint on the pattern of that graph.

3.2. Maximum Common Edge Subgraphs

Raymond et al. (2002) suggest a procedure for comparing labeled graphs using maximum common edge subgraph (MCES) detection algorithm to compute the exact degree and composition of similarity. They propose to use RASCAL (Rapid Similarity CALculation) which consists of two components, screening and rigorous graph matching (using MCES). The screening phase is used to rapidly determine if the graphs being compared exceed some specified minimum similarity threshold, so that we can avoid unnecessary computations.

For screening phase, the authors use 2 tier cost-vector algorithms. The 1st tier cost-vector algorithm utilize local connectivity and vertex labels to help eliminate unnecessary and costly MCES comparison. It considers different sets of labeled vertices, which are partitioned into l partitions based on their label type, and then sorted in non-increasing order by degree. The second-tier screening algorithm takes into account edge labeling, which is costlier than the 1st tier screening.

MCES, in comparison to screening algorithms, is very costly but necessary to determine the similarity of two graphs. In order to detect clique, the authors use a branch and bound algorithm that incorporates improvements in both lower and upper bounding of node selection. While RASCAL has been designed for use in chemical information management, the algorithm is conceptually general and applicable to any graph-based similarity application.

3.3. BLAST 2 Sequences

In their paper, Tatusova et al. (1999) discusses about BLAST's algorithm and its application. BLAST is a rapid sequence comparison tool that uses a heuristic approach to construct alignments by optimizing a measure of local similarity.

If we can convert a graph of financial data into simple classifications, we would be able to use BLAST to compare the similarity of two different graphs. One naïve approach would be to divide the range of data into smaller equal ranges (named A, B, C, etc.) and compare two different graphs using those ranges.

4. Stock prediction

4.1. Using support vector machine

Huang et al. (2005) and Yang et al. (2002) both discusses the possibility of using Support Vector Machine to predict stock market movement direction.

Per Huang (2005), Japanese's economy growth has a close relationship with Japanese export, which in turn makes the United States' (Japanese's main export target) economic condition determines Japan economy. The USA's economy is represented by the S&P 500 Index, while Japan economy is represented by the NIKKEI 225 Index. In this experiment, S&P 500 Index is served as input for the Support Vector Machine model. To evaluate the forecasting ability of SVM, the authors use the random walk model (RW) as a benchmark for comparison. They also included linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and elman backpropagation neural networks (EBNN). A combined model is also developed, using different weights for different classification method. The result is fascinating, with SVM's hit ratio was around 73%, the highest compare to another method. The combined method has even higher hit ratio with 75%.

Yang et al. (2002) try to apply Support Vector Regression (SVR) to financial prediction tasks. They propose an improved model based on a normal SCR model, which consider margins adaptation. When using SVM in regression tasks, the SVR need to use a cost function to calculate the risk to minimize the regression error. The margin used by those function is very important because when the margin is zero and very small, it is possible to overfit the data with poor generalization. On the other hand, if the margin is too high, one run into the risk of having higher testing error. For financial data, because of the embedded noise, one must use a suitable margin to obtain a good prediction. Two experiments were conducted to illustrate the effect of FASM (Fixed and Symmetrical Margin), FAAM (Fixed and Asymmetrical Margin), and NASM (Non-fixed and Symmetrical Margin). Through their finding, they concluded that in financial applications, setting a suitable margin is critical to the performance of the prediction tool used.

4.2. Using modular neural networks

Kimoto et al. (1990) discuss a buying and selling timing prediction system for stocks on the Tokyo Stock Exchange and analysis of internal representation, using a modular neural network algorithm. The input consists of several technical and economic indexes, including some neural networks learned the relationships between the past technical and economic indexes and the appropriate buy/sell time. A prediction system that is made up of modular neural networks was claimed to be correct, with the simulation of buying and selling stocks using the prediction system shows an excellent profit.

Below is the graph that the authors used to describe the overall architecture of the prediction system. Using different indexes such as turnover rat, foreign exchange rate, etc. the data is then passed to a preprocessor before passed on to neural networks to predict the right time to buy and sell stocks.

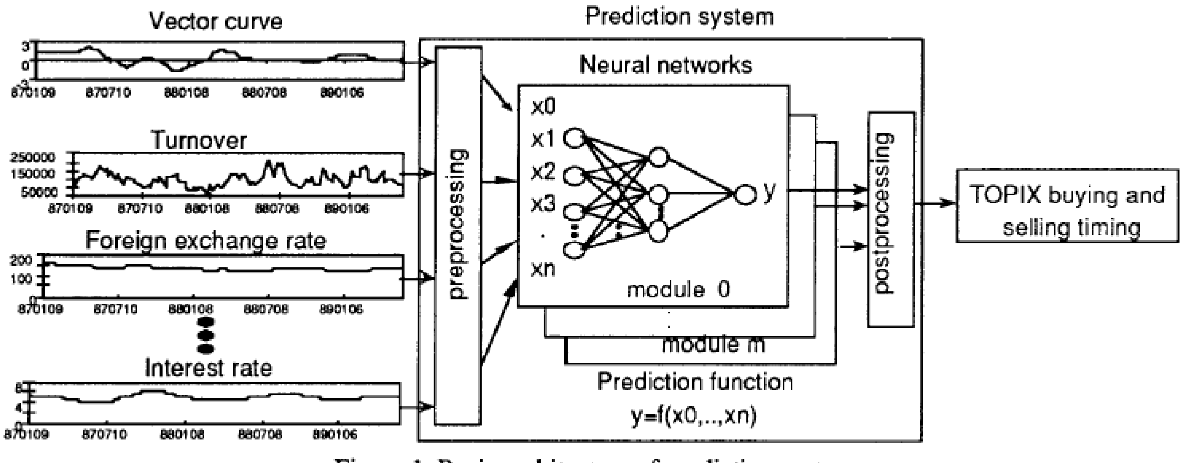
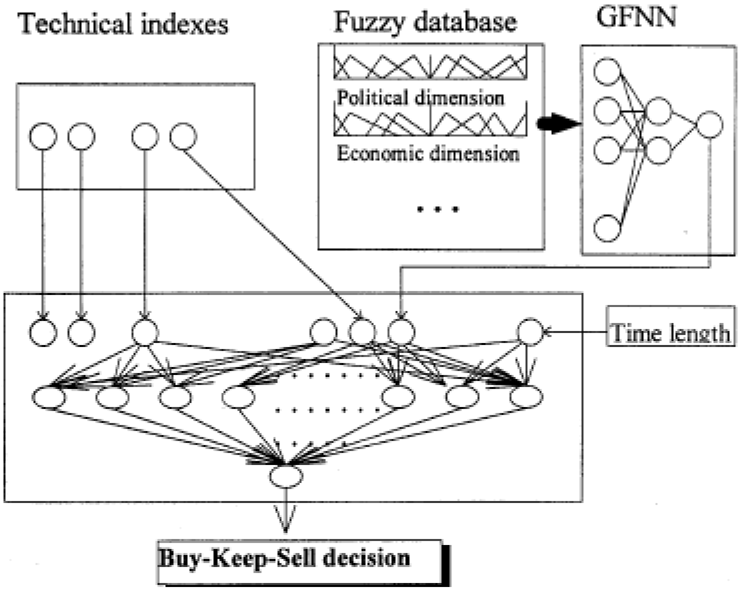


Figure 1 Basic architecture of prediction system

The authors use high-speed learning algorithm called supplementary learning, which is based on the error back propagation. The algorithm automatically schedules pattern presentation and changes learning constants when needed. In supplementary, the weights are changed according to the sum of error signals after presentation of all learning data.

4.3. Genetic algorithm based fuzzy neural network

Kuo et al. (2001) developed a genetic algorithm based fuzzy neural network (GFNN) to formulate the knowledge base of fuzzy inference rules which can measure the qualitative effect (e.g., political effect) on the stock market.



The methodology the authors used is described in the above diagram. This study develops an intelligent stock trading decision support system based on the viewpoint of system integration. There are three main parts in this model: factor identification, qualitative model (GFNN), and decision integration.

5. Rete algorithm

The Rete Match Algorithm is an efficient method for comparing a large collection of patterns to a large collection of objects. Since the pattern matching process can be very expensive, the algorithm can help to speed up the run time, especially in a time-critical application such as a financial tool. Forgy (1982) discusses in detail the design and implementation of the algorithm in his paper.

In pattern matching problems, it is very common to compare a collection of patterns to a collection of objects, and then determine all matches. The Rete algorithm uses some important techniques in order to reduce the runtime, such as avoiding iterating over working/production memory, using node types, etc. One important detail about the paper is that the methods described are developed for production system interpreters, but is still useful in many other scenarios.

6. Conclusion

There have been many attempts on using different artificial intelligence algorithm to predict the stock price as well as graphs' similarity scoring and matching. Many of them proved to be relatively efficient and accurate. However, there are more that can be done to improve the accuracy of the algorithms. A combination between different methods used is also likely to produce a more accurate and fast algorithm. One possibility is to look outside of the market and find the correlation between the stock price and another quantitative product.

References:

[1] Evertsz, C. J. G., & Berkner, K. (1995). Large deviation and self-similarity analysis of graphs: DAX stock prices. Chaos, Solitons & Fractals, 6, 121–130. https://doi.org/10.1016/0960-0779(95)80019-D

[2] Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence, 19(1), 17–37. https://doi.org/10.1016/0004-3702(82)90020-0

[3] Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32(10), 2513–2522. https://doi.org/10.1016/j.cor.2004.03.016

[4] Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In 1990 IJCNN International Joint Conference on Neural Networks (pp. 1–6 vol.1). https://doi.org/10.1109/IJCNN.1990.137535

[5] Kuo, R. J., Chen, C. H., & Hwang, Y. C. (2001). An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. Fuzzy Sets and Systems, 118(1), 21–45. https://doi.org/10.1016/S0165-0114(98)00399-6

[6] Raymond, J. W., Gardiner, E. J., & Willett, P. (2002). RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs. The Computer Journal, 45(6), 631–644. https://doi.org/10.1093/comjnl/45.6.631

[7] Tatusova, T. A., & Madden, T. L. (1999). BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences. FEMS Microbiology Letters, 174(2), 247–250.

[8] Yang, H., Chan, L., & King, I. (2002). Support Vector Machine Regression for Volatile Stock Market Prediction. In H. Yin, N. Allinson, R. Freeman, J. Keane, & S. Hubbard (Eds.), Intelligent Data Engineering and Automated Learning — IDEAL 2002 (pp. 391–396). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45675-9_58

[9] Zager, L. A., & Verghese, G. C. (2008). Graph similarity scoring and matching. Applied Mathematics Letters, 21(1), 86–94. https://doi.org/10.1016/j.aml.2007.01.006