Title: **Examples of the Use of Data Mining in Financial Applications**

Summary: This article considers building mathematical models with financial data by using data mining techniques. In general, data mining methods such as neural networks and decision trees can be a useful addition to the techniques available to the financial analyst. However, the data mining techniques tend to require more historical data than the standard models and, in the case of neural networks, can be difficult to interpret.

This article considers building mathematical models with financial data by using data mining techniques. In general, data mining methods such as neural networks and decision trees can be a useful addition to the techniques available to the financial analyst. However, the data mining techniques tend to require more historical data than the standard models and, in the case of neural networks, can be difficult to interpret.

Stock market returns and foreign currency exchange rates data can be thought to fall into one of four categories as follows.

1. Five time series: index value at open, index value at close, highest index value, lowest index value and trading volume.
2. Fundamental factors: e.g., the price of gold, retail sales index, industrial production indices, foreign currency exchange rates.
3. Lagged returns from the time series of interest.
4. Technical factors: variables that are functions of one more time series, e.g., moving averages.

The standard approach to modeling stock market returns or exchange rates is to model the univariate time series with autoregressive (AR) and moving average (MA) models. A trader can determine an appropriate number of lags for AR and ARMA models based on experience and by analyzing the time series data. Similarly, an appropriate number of regimes for SETAR (self-exciting transition AR) and STAR (smooth transition AR) models can be determined. These models are deterministic in the sense that they attempt to use mathematical equations to describe the process that generates the time series. The advantage of these models lies in their interpretability.

Another approach, drawn from data mining, is to adopt a model that is flexible in the sense that it can approximate a wide class of functions with high accuracy. Such models are non-parametric in the sense that there need not be a direct relationship between the parameter values of a fitted model and the data. The advantages of using such a model include:

1. The ability to model highly complex functions.
2. The ability to use a high number of variables in the model and, therefore, to include other data (i.e. fundamental and technical factors) in addition to lagged time series data.

The disadvantage of non-parametric models is that they are not easy to interpret.

In the case of data mining time series data, the model of choice is a neural network. By adjusting the number of free parameters associated with a model, a trader controls its flexibility. Often, cross-validation, or hold-out data, is used to determine a suitable value for the number of free parameters contained in a neural network structure. The neural network most commonly used in financial applications is a multi-layer perceptron (MLP) with a single hidden layer of nodes.

The problem of predicting stock market returns or exchange rates at time *t+1* can be cast as either a regression or classification problem. Whereas the regression problem for exchange rate data involves modeling the actual exchange rate, the classification problem involves predicting whether the exchange rate has increased or decreased.

Applications that involve modeling returns from the stock market include portfolio management and trading futures (see below).

MLP regression example: portfolio management

The regression case involves predicting the (raw) return values. Such predictions can be used to manage a portfolio of n stocks as follows.

Suppose that historical data for N (N > n) stocks are used to fit N multi-layer perceptrons. At the end of each week the MLPs are re-fitted to include the latest historical data. For example, suppose company Z's pension fund has been managing a portfolio of $100 million since December 1993 using multi-layer perceptrons. The fund monitors a pool of 1,000 U.S. stocks on a weekly basis. For each of these stocks there is a MLP which models the future performance of the stock as a function of the stock's exposure to 40 fundamental and technical factors, and gives an estimate of its weekly price change. The company then selects a portfolio of the top n stocks and allocates the fund proportionately to predicted returns.

MLP classification example: trading futures

As an example of a data mining classifier, consider the problem of trading a future of stock A at price B on date C by using a neural network.

Firstly, the historical data is prepared. At each time step, data are classified into one of two categories according to whether it was profitable to buy or sell stock A at price B on date C:

1. Long: buy the stock on date C.
2. Short: sell the stock on date C.

Having fitted a model with this historical data, the model can be used to predict a profitable position at time t+1 (e.g., the next day or week). At the end of each time step the model is updated to include the new historical data.

By the time date C arrives, the trader should be in a profitable position (either long or short) given the current market value of stock A.

Trading rules

Trading rules can be determined from data with a categorical outcome, e.g., buy or sell, rise or fall. Such rules take the form of a set of conditional statements and an action, e.g.,

IF CONDITION1 AND CONDITION 2 THEN ACTION

and can be found by viewing a fitted decision tree model. Given an appropriate historical data set, this approach could be used to either validate a rule thought to exist, or generate new rules and ideas.

Suppose that a decision is fitted using historical data. Each internal node in the tree is a test on one of the variables used to predict the outcome in the historical data. If the variable, say X1, takes continuous values, this test is either:

(X1 >= VALUE) or (X1 < VALUE),

