Does past stock performance matter in investing?(Final Report)

KK and KT

Abstract. In our project we investigate the impact of past stock data on future price movement. More precisely, we consider two different approaches for predicting the future stock behavior: supervised learning based on feature vectors built from technical analysis indicators, and sequence mining for discovering frequent patterns of stock price movements for a given stock. *Optional: We also investigate the relationship between price movements for different stocks and use this to prediction of stock prices.*

Data

We gathered publicly available data from Yahoo! Finance for different stocks in two sectors: Basic materials and Technology. The Basic materials sector consists of companies involved with the discovery, development and processing of raw materials such as metals, chemical producers and forestry products. The Technology sector contains stock data for companies in areas like software development, electronics manufacturing, and other businesses related to information technology.

The Basic materials sector contains data for stocks positively correlated with a strong economy, e.g. the stocks of companies producing steal are dependent on the automotive industry, as well as stocks negatively correlated with the economic growth, e.g. the demands for gold rise during a bad economy cycle.

On the other hand, the technology sector is ... ?

Data preprocessing

In our case data preprocessing is a straightforward task since for each stock we are provided with a csv file consisting of the following features for the stock for each trading day: opening price, closing price, highest and lowest price and the volume of traded stocks for the day. We directly use the numerical values in order to compute technical analysis indicators and to label days, respectively longer time periods, in a way describing in an informative manner the behavior of the stock for the considered time interval.

Note that since our data contains no missing or inconsistent values no data cleaning is necessary.

Supervised learning

We incorporate several technical indicators about past stock behavior into feature vectors. We label the data as follows...

Frequent pattern mining

General setting

We consider the price movement of a given stock over a certain time period. We divide the period in small subintervals of at least one day which we consider data points in a sequence. Similarly to [1] we label the data points such that the label indicates the stock price movement. We use three distinct labels F, H and R indicating whether a given price will decrease its value over a given period (F), will retain approximately the same value (H), or will increase its price (R). They are obtained depending on the opening and closing price for the period.

A modified Apriori approach

Let us formally define our problem.

We are given a sequence $\mathcal{S}$ of symbols s over some alphabet $\mathcal{A}$. Let the support of a subsequence S be $\text{sup}(S) = \frac{\#S}{|\mathcal{S}|-|S|}$ where $\#S$ is the number of occurrences of the subsequence S in $\mathcal{S}$ and $|S|$ is the length of S. That is, the support of a given sequence S is the number of its occurrences divided by the number of all subsequences in $\mathcal{S}$ of length $|S|$. For a given support threshold σ and a confidence threshold τ we want to find all subsequences $S \subset \mathcal{S}$ such that $\text{sup}(S) \geq \sigma$ and $\text{conf}(S) \geq \tau$. The confidence $\text{conf}(S)$ of a rule derived from S is defined as $\frac{\text{sup}(S)}{\text{sup}(S \backslash \ell)}$ where ℓ is the last symbol in S.

Obviously, the above definitions are similar to the itemset and transaction setting in the classic frequent pattern mining context. However, an important difference is that one can build only linearly many subsequences of a given sequence, thus the following algorithm avoids the combinatorial explosion caused by Apriori for lower support thresholds and is thus much more efficient. (Note that our algorithm is different from the one proposed in [1].)

Algorithm:

1. Find frequent symbols, set $k = 2$.
2. Join frequent subsequences of length $k-1$ if the last $k-2$ symbols of the first subsequence are identical to the first $k-2$ symbols of the second subsequence
3. build a set of candidates of length k and filter out infrequent subsequences.
4. repeat steps 2–3 until no new frequent subsequences can be obtained.

Lemma 1 *The algorithm correctly identifies all frequent subsequences of a given sequence.*

Proof. For a subsequence S to be frequent the Apriori property that all proper subsequences S' of S are frequent must hold. Thus, assuming we have identified all frequent subsequences of length $k-1$, steps 2 and 3 return a superset of the frequent subsequences of length k. Correctness follows by a simple inductive argument.

Example: Let us consider a simple concrete example: the sequence $\mathcal{S} = abcbcaababcaabccbaabbabcbcacba$ contains 30 symbols. We are interested in all k-subsequences with frequency at least 0.15, i.e. occurring in at least 10% of all subsequences of length k. The 1-subsequences are the symbols themselves and we find that all of them are frequent since they appear more than 4.5 times (4.5 = 0.15*30). Then we build candidate subsequences as $aa, ab, ac, ba, bb, bc, ca, cb, cc$ since they all satisfy the joining condition: the last 0 symbols of the first subsequence agree with the first 0 symbols of the second one. From these symbols we find that the subsequence aa appears only once, ab – five times, ac – one time, ba – 4 times, bb – one time, bc– 6 times, ca – three times, cb– three times and cc – one time. The total number of 2-subsequences is 29, thus threshold is now $0.15 * 29 = 4.35$. Thus, the frequent 2-subsequences are ab, ba and bc. From these we build the following candidate 3-subsequences: aba, from ab and ba, bab, from ba and ab, and abc, from ab and bc. Note that ba and bc can not be combined since they don't satisfy the joining condition. Now aba appears only once, bab twice, and abc – four times. Thus, no 3-subsequence is frequent, i.e. occur more than $4.2 = 0.15 * 28$ times, and the algorithm terminates.
In the very same way as in Apriori we now compute the confidence of each rule.

Results

After labeling the data for a given stock over a certain time period, say 3 days, we perform frequent sequence mining over the sequence. We use the first 2/3 of the sequence for frequent pattern mining, and the last 1/3 for correctness testing of our predictions. We obtain the following results for chosen stocks: ...

Correlation pattern mining

It is often the case that the price movement of a certain stock is correlated with the prices of other stocks. For example higher prices for metals will affect negatively the automotive industry but the effect occurs after some time. Another example is...
Inspired by the above we implemented a sequence mining algorithm for detecting correlations between stocks within a given time span. For example is there a correlation between the price movement of stock A in the interval $[p, q]$ and the price of stock B in day $q + t$ for some appropriately chosen t. We experimented with different time intervals in order to mine dependencies between the stocks.

References

1. Jo Ting, Tak-Chung Fu, Fu-Lai Chung: Mining of Stock Data: Intra- and Inter-Stock Pattern Associative Classification. *DMIN 2006:* 30–36