Learning and predicting price changes in financial markets:

An oil market study with Japanese Candle Sticks

Hideo Oda
B.A.(Mod.) Computer Science
Final Year Project April 2012
Supervisor: Khurshid Ahmad

School of Computer Science and Statistics
O’Reilly Institute, Trinity College, Dublin 2, Ireland
DECLARATION

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university

____________________  __________________
Name                      Date
Acknowledgements

I would like to thank my supervisor Khurshid Ahmad for all his help and constant guidance during this project, it has been a great help.
I would also like to thank my parents for all their supports and one of my friends, Hugo Llewellyn, for proof reading my report throughout this project.
Abstract

This project demonstrates purely statistics-based learning and prediction analysis of changes of prices in financial markets, using Japanese Candlestick charts. A dataset of WTI (West Texas Intermediate) historical price of oil over the past 20 years was used to run the program which was developed in this project.

As the Candlestick chart is visually friendly, one can see regularities in price changes in the chart intuitively, and some patterns are said to occur more often than others.

A program that can look at price changes during many trading sessions and identify the candlestick patterns was designed and implemented. When given an historic time series, the program can create a library of these patterns and note their frequency. The frequency can be used to assign a probability of occurrence. The program can also compute and output a list of patterns that occur more often than others, meaning when these patterns occur the program could predict how the price is most likely to change, using co-occurrence of two patterns within a specified period of time. In effect, it learns the behavior patterns, and outputs a frequency of occurrence that has been learnt.

The experimental results by the program proved that there are combinations of patterns that can be used to predict the price in the near future, in the markets.
Contents

Introduction.............................................................................................................................................. 6
Motivation and Literature review ............................................................................................................... 11
Method ...................................................................................................................................................... 13
Experiments and Evaluations..................................................................................................................... 20
Conclusions............................................................................................................................................. 36
Appendices.............................................................................................................................................. 38
Bibliography............................................................................................................................................ 52

Electronic Sources and Resources available on accompanying CD................................................. 51
Figure 1: Image of a candlestick chart generated by the program.

Introduction

There are a number of regularities that can be observed when prices of many equities and commodities are organized in a discrete set of data points in time – a time series of prices. One can see wave-like patterns, including sharp falls and rises; rising prices follow closing prices, but rising prices can be followed by rising prices (boom), and falling by falling (bust). An analysis of trading session shows that traders start with a price (open), the prices may rise to a peak (high) or drop to their lowest point (low) during the session, and lastly, there is the final price (close).

It was noted by many, including 18th-century Japanese rice farmers that in addition to wave like patterns over a whole period of trading, one can visualize the period into
many sessions. During each session, closing prices may be greater than opening prices, the interim highs/lows may be greater/lower than closing prices or opening prices. In order to visualize the sessional variation in prices, the prices were displayed in a candlestick pattern.

![Candlestick diagram](image)

Figure 2: Variables of candlesticks. Image from [1]

The area between open and close in the candlestick is called the real body, the wick above and below the real body are called shadows. The wick illustrates the highest and lowest prices of a commodity during the period of time that is represented. The main body illustrates the opening and closing prices [2].

Candlestick charts are thought to have been developed in the 16th century by a Japanese rice trader named Sokyu Honma. However, as several contradictions can be found in books that were thought to been written by him, it is uncertain if he was the real inventor of Candlestick charts [3]. They were introduced to the Western world by Steve Nison.
In the Edo period (1603-1868 CE), Japan pioneered the future trade, specifically in rice. The amount of money Sokyu earnt is said to be more than 10 billion euro in today's prices. He designed the de facto standard of technical analysis techniques for candlestick charts (called *Sakatagoho*) that is still heavily used in today’s market, two hundred years after his death [3].

![Figure 3: Example image of a candlestick chart, using MetaTrader 4. Image from [4](#)](image1)

![Figure 4: Example image of an ordinary chart of the same data, using MetaTrader 4. Image from [4](#)](image2)

As it can be seen in the two images above, although both charts look reasonably similar,
the top chart, which is a candlestick chart, has two different colors inside the chart, which are black and white depending on the direction the price is moving to. Each candle bar shows both the highest and lowest price that were reached in its time interval. Therefore, it holds more information.

*This software uses white candles for price fall and black candles for price rise, which is opposite to what is described throughout this report. In general, using white candles for price rise and black candles for price fall is the standard.

Candlestick charts perform well in analyzing prices that change rapidly, unlike the stock price of stable national banks. A candlestick chart can show traders’ and markets’ emotion, as it can visualize price changes within a given period of time [3]. As the market is largely affected by people’s thoughts, it would be reasonable to make an estimate that there might be some simple patterns that occur clearly more often than others. For example, one could think that after seeing 5 consecutive price rises, the market must have a higher chance of having the price fall in the next period of time. This may sound too simple, however, it is said that patterns which are more likely to occur than others do exist. For instance, the *heads and shoulders pattern* below occurs during price reversals and shows declining trends [5].

![Figure 5: Heads and shoulders pattern. Image from [6]](image-url)
Another pattern as an example would be the *island reversal* pattern shown below, where there is a gap in the pattern after which a trend starts. This is one of the trends which is said to be hard to discern [5].

![Island Reversal Pattern](image)

**Figure 6: Island reversal pattern. Image from [7]**

There are a few types of analysis techniques for investments when analyzing price changes of stocks, currencies, or futures contracts, which could be briefly described as either fundamental analysis or technical analysis [8]. As fundamental analysis considers all unpredictable factors such as the overall state of the economy that it adds extra complex uncertainty to its analysis process, it was decided that this project would focus only on technical analysis, which is to forecast the direction of price movements by only looking over the historical prices of the market.
Motivation and Literature review

As briefly described in the introduction, it is people’s minds that make decisions in the world’s politics and economics, and therefore we influence the strength of currencies, value of stocks, price of oil, etc. As those commodities and equities are traded by a large number of investors, they often affect the prices in the markets.

This project was based on the idea that simple patterns must exist in the markets, for example, it is understandable to feel fear if the price keeps falling for a long period of time, investors might start wanting to minimize their losses by relinquishing what they are holding, thereby abandoning the possibility of the price rising again.

Before launching this project, various research papers which were aimed at predicting the future prices in financial markets using Japanese Candlestick charts were examined. However, most of the researchers used very impressive-looking mathematical models and formulae that were full of complexities, and no research papers with the idea on which this project is based could be found. This idea is to implement a simple pattern recognition system that is used to define patterns as combinations of only low price, high price, open price, close price, and whether the price has risen or fallen in a certain period of time, which determines the color of a candle: white if the price has risen and black if the price has fallen. Consecutive white candles must give momentum to the market but it is certain that the trend will reverse, and consecutive black candles must accelerate sell orders in the market, worrying investors. But then again it is almost certain that at some point the trend will reverse when investors start feeling that price is too low and that it is worth buying whatever the commodity is.

This project was strongly influenced by the work that was done by Kouki Imoto, who is known as a winner of the first algorithmic trading contest that was held in Japan and works as a director of the Institute Of Probability Theory, which was exposed to the masses in one of the books he published [9]. As mentioned in the introduction, Sokyu Honma, who is believed to be a pioneer of candlestick charts, noted all his strategies and techniques for buying commodities or equities (in his case it was future trade of rice) and he also made a list of simple patterns that appear in candlestick charts, giving a name to each one of them. In the western world nowadays, investors who are aware of candlestick charts know those names but in the English version, that were
mostly translated in his books by Steve Nison. These patterns are mainly composed of a small number of candles, around 2 ~ 6.

![Patterns](image)

**Figure 7:** Various patterns which can be used to predict the future trend of a market.

Image from [10]

Kouki Imoto tested all those patterns to see if they actually appeared, and found that some patterns were reliable, so he wrote a program to trade stocks whenever the system sees these patterns and won the competition. As he proved that one could find patterns in commodity prices, by making good predictions of prices in the stock market by using statistics, this project had a further goal of not only testing and evaluating the existing patterns and using ones that had good results, but to find more patterns that had not yet been discovered.
Method

As a dataset, the price of oil, WTI (West Texas Intermediate) price over the past 20 years on a 5 minute chart was used. The price of WTI is often referenced in news reports on oil prices. Other important oil markers include the Dubai Crude and the OPEC Reference Basket [11]. A CSV file was provided by the supervisor of this project, which is composed of date, time, open price, high price, low price, and close price in respective order in each row, which looked like the following:

01/27/1992, 0950, 56.53, 56.57, 56.50, 56.54, 41, 0
01/27/1992, 0955, 56.57, 56.57, 56.52, 56.52, 48, 0
01/27/1992, 1000, 56.51, 56.52, 56.47, 56.52, 34, 0
01/27/1992, 1005, 56.51, 56.53, 56.48, 56.52, 17, 0
01/27/1992, 1010, 56.51, 56.57, 56.51, 56.56, 34, 0
01/27/1992, 1015, 56.57, 56.57, 56.52, 56.52, 13, 0
01/27/1992, 1020, 56.53, 56.58, 56.53, 56.56, 18, 0
01/27/1992, 1025, 56.57, 56.65, 56.57, 56.65, 37, 0

Figure 8: Sample CSV of WTI daily oil price which was used.

Python programming language was used to implement the system. The system is also able to create a visualization of the candlestick charts, when given a dataset of prices. The code for outputting chart images is shown in Appendix B.

The example code from the constructor of the system is as follows:
```python
def __init__(self, typeOfChart, dire, csvName):
    self.bars = []
    self.typeOfChart = typeOfChart
    self.dire = dire
    print 'Initializing System... Loading a File: "' + csvName + '"'
    self.loadBars(self.dire + csvName)
    print str(len(self.bars)) + " Bars loaded."
    self.myDrawer = Drawer.Drawer()
```

To get potential patterns out of the oil market, the following three approaches were taken:

1. The first approach is to combine 4 variables that appear on each candle on candlestick charts, which is open, high, low, and close and to define patterns using these 4 factors.

Patterns were represented using binary digits. The pattern length can be obtained by this formula:

\[
\text{len(PickedPatternInBinary)} < 1 + 5 \times (\text{NumOfBarsToConsider}-1)
\]

For example, one randomly picked pattern can look like this:

\[111101\]
This pattern will mean:

- This pattern is composed of 2 candles.
- The first candle is a white candle (close price higher than open price) because the first digit is 1.
- The second candle is another white candle because the second digit is 1.
- The second candle has a higher open price than the previous candle, because the third digit is 1.
- The second candle has higher high price than the previous candle, because the fourth digit is 1.
- The second candle has lower low price than the previous candle, because the fifth digit is 0.
- The second candle has higher close price than the previous candle, because the last digit is 1.
- Visually it will look something like this:

![Figure 9: Example image of a pattern “111101” generated by the program.](image)
Just to make it clearer, yet another example of patterns could be:

1111110000010101

- From the third candle onwards, it will always refer to the candle just before, in exactly the same manner as between the first and the second candle.
- Therefore if a completely random sequence of binary digits are selected, there will be a chance that the binary sequence will not make sense technically and does not exist (say pattern “10111”; it cannot exist, as it is not possible to have both the open price and close price higher than the previous white candle, if this candle is a black candle).
- This pattern should look like this:

![Diagram of a candlestick chart]

Figure 10: Example image of a pattern “1111110000010101” generated by the program.
Using this pattern definition, the program goes through all the price data in the CSV file that was provided, counting the number of times each pattern occurs. By using this approach, the program will be able to create a list of all possible patterns that appeared in the past chart and also to see which patterns are more likely to occur than others.

2. The second approach is to define patterns similar to the first approach and to count the number of times they occur, but in a much simpler way. In the first approach, all the possible variables of each candle were taken into account, but this second approach considers only one of the 4 variables, for example checking only if the high price was higher than the previous candle’s high price. The most obvious example would be if a candle is a white candle or a black candle, meaning the system considers only if the close price of candles was higher than its open price. For instance, a random pattern “1001” will simply mean, white candle, black candle, black candle, and white candle appearing consecutively in an order which will look something like this:

![Figure 11: Example image of a pattern “1001” generated by the program.](image-url)
The following is a part of the code that well describes the main idea of how to generate patterns using this approach:

```python
while True:
    if (self.bars[iStart+pattern_index].closesHigher()):
        result_pattern += "1"
    else:
        result_pattern += "0"

    pattern_index += 1
    if (iStart+pattern_index > iEnd):
        break
```

3. The third approach is to define patterns in exactly the same way as the way that was used in the second approach, and to count the number of time they occur. This time however, not only to find out which patterns are more likely to occur than others, but also to calculate the co-occurrence of two patterns within a specified period of time. Through this project, simple models to calculate the co-occurrence of two patterns were used. More precisely, to calculate the co-occurrence of all patterns that appeared in the past market, with both white and black candles. This is equivalent to saying “After this pattern occurs, will the price be more likely to rise or fall in the next time interval?” This approach was based on the idea that it will not help you predict the price of the near future when you are trading, even if you have a list of patterns that are more likely to occur than others, if that’s not the pattern or price of the market you are seeing on the screen right now.

The following is another example of code for building a library of patterns with their occurrence:
patternsInfo = {}

while (NumOfBarsToConsider <= MaxNumOfBarsToConsider):
    pattern = self.getPattern(index, index + NumOfBarsToConsider-1, my_mode)
    if (patternsInfo.has_key(pattern)):
        patternsInfo[pattern] += 1
    else:
        patternsInfo[pattern] = 1
    NumOfBarsToConsider += 1
    index += 1
Experiments and Evaluations

1. For the first approach described in the method section, below is a part of the automatically generated analysis report when running the program on WTI 5min price over the past 20 years. The program was set up to go through the first 80% of the entire data to do analysis (pattern finding), learn and compare with the analysis of the rest.

The following is part of an analysis report generated by the program:

```
SortMax10Cdl5s01.27.1992-01.21.2010.txt   Top 20 of patterns of 8 candles:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Number of occurrences</th>
</tr>
</thead>
</table>
| 111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
Example of an automatically generated analysis report of given sets of price data. Patterns are shown on the left hand side, and the number of times each pattern occurred are listed on the right hand side, and sorted.

Predictably, by far the most frequently occurring pattern was a pattern composed only of 1s which will look like this:

```
11000011111111111000011111111111111111
110000111111111111111111111111111111
1111111111100001111111111000011111
111111100001111111111111111111111000
```

Figure 12: Example image of a pattern “111111111111111111111111111111111111” generated by the program.
Intuitively, this was not a surprising result. For example, when a random pattern of candles “1111” occurs, it will produce 3 of the patterns that are composed of three 1s. After seeing the pattern “1111”, the system will count:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Therefore it was thought that the system will learn a relatively high number of patterns that are all composed of candles of the same color (1, in this case). However, when the system sees a pattern of candles, for example, “10011”, it will produce:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>10011</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
<tr>
<td>0011</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
</tbody>
</table>

The above example table was thought to be able to describe what just happened in the outcome of the system. None of the shorter patterns inside the original long pattern will match each other; therefore those patterns will have less chance of having a high frequency.
Although this explanation was first thought to be the reason for this phenomenon, which is that all the simple patterns composed of candles of the same color (same kind, 1, in the previous example) dominated in the total statistics.

However, mathematically it was proven that this consideration was not correct. In order to ascertain the veracity of this consideration, a simple script of Python code was written. The script can generate a sequence of random binary digits if given a length, and counts the number of times each subsequences of the sequence of given length occurs.

The short script is as follows:

```python
import random

def generate_sequence(length):
    result_seq = ""

    counter = 0
    while(counter < length):
        result_seq += str(random.randint(0,1))
        counter += 1

    return result_seq
```
```python
def count_occurrences(sequence):
    index = 0
    len_of_pattern = 3
dic_of_patterns = {}

    while (index + len_of_pattern <= len(sequence)):
        pattern = sequence[index] + sequence[index+1] + sequence[index+2]
        if (dic_of_patterns.has_key(pattern)):
            dic_of_patterns[pattern] += 1
        else:
            dic_of_patterns[pattern] = 1
        index += 1

    for pattern in dic_of_patterns:
        print pattern + " " + str(dic_of_patterns[pattern])

count_occurrences(generate_sequence(10000))
```

For example, if the script is run and given a length of 10, it will produce a sequence similar to:

```
1001101000
```

And then if given a random length of 3 for its subsequences, it will return:

```
1001101000
```
In a sense this script is simulating a financial chart by generating a sequence of combinations of 1s and 0s randomly, and also it counts the number of times the subsequences appear, which are equivalent to patterns occurring in the chart. Patterns which are composed of 3 candles were considered and counted, and the result was outputted.

With this simple script, a length of 10000 (as shown in the code) for the whole sequence, and again the subsequence of length 3 were used, and it returned:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern</th>
<th>occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>1274</td>
</tr>
<tr>
<td>011</td>
<td>1232</td>
</tr>
<tr>
<td>001</td>
<td>1222</td>
</tr>
<tr>
<td>000</td>
<td>1230</td>
</tr>
<tr>
<td>111</td>
<td>1230</td>
</tr>
<tr>
<td>110</td>
<td>1274</td>
</tr>
<tr>
<td>100</td>
<td>1233</td>
</tr>
<tr>
<td>101</td>
<td>1287</td>
</tr>
</tbody>
</table>
The table above shows that the consideration made on the result the system outputted was not correct, as all the patterns should have nearly the same probability of occurring. Therefore, all those patterns that occurred more often than others the system found were not found because of the way that the system counts the number of times each pattern occurs; but these patterns actually do occur much more often than others.

In addition to that fact, from the result analysis, it can be seen that both 1 and 0 always appear in a group of 4. This implies not having special combinations of high price and close price (for example, if the second candle in the pattern has a higher high price and a lower close price than a previous candle, followed by a third candle with a lower high price than the second candle, etc.), instead, simple and typical trends all composed of open price, high price, low price, and close price moving in the same direction consecutively, dominated the total statistics. This phenomenon can be considered a trend or momentum of the market, and it was just proved by the system that those continuations of price rise or fall (trend) occur and last, resulting in domination of the whole statistics.

The reason for having more 1s (white candles) than 0s (black candles) is that, in the past 20 years the price of oil rose in general, therefore there were more periods of time where the price rose than fell.

The following is an automatically generated comparison report of 2 different analysis reports on common patterns which are the first 80% of the whole WTI oil price data used and the rest. For both datasets, the program looked for the 20 most common patterns according to the number of their occurrences. The system outputted that both datasets share 14 common patterns in their top 20. This may sound like those patterns that are found have much higher frequency than other patterns, however this could be explained by simple mathematics. As the model this approach is using sees each period as either a price rise or fall, for patterns of 5 candles there are only $2^5 = 32$ possibilities, and even among those top 20 patterns, most of them are patterns which are composed of mainly consecutive 1s or 0s, therefore it is reasonable that the system outputted 14 common patterns out of 32 possibilities.
The report after the following is the result of more candles, which is an analysis of 8 candles. As it can be seen, there are only 2 matches and one of them is composed entirely of 1s. It can be concluded that the pattern definition was too detailed for the analysis of the large number of candles. Therefore approach 2, which is basically a simpler version of approach 1, was designed and implemented so that although it will lose some information that each pattern holds, there will be more matches between patterns.

Analysis of patterns of 5 candles: 14 matches in Top 20

000000000001001111111 :  
426 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1078 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

01011111111111111111 :  
698 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1574 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

00000000000000000000 :  
612 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1383 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

010001111111111111111 :  
507 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1239 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

Example of an automatically generated analysis comparison report of given lists of analysis.
Patterns followed by their occurrences in each analysis report.
Full list available in Appendix C.1.
Analysis of patterns of 8 candles: 2 matches in Top 20

010011111111111111111111111111111111 : 
25 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
168 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

111111111111111111111111111111111111 : 
62 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1000 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

Example of an automatically generated analyses comparison report of given lists of analysis.
Patterns followed by their occurrences in each analysis report.

2. For the second approach described in the method section, below is a part of the automatically generated analysis report when running the program on WTI 5min price over the past 20 years. The program was set up to go through the first 80% of the entire data to do analysis (pattern finding), learn, and compare with the analysis of the rest.

The report shows that even for 9 candles there are still 11 common patterns between 2 different sets of data, however, once again the most popular pattern was composed of all 1s and the rest of patterns were composed of almost all 1s but with a single or two lone 0s. No interesting or surprising patterns were outputted.

The problem with these approaches was that there were always patterns which were composed of consecutive 1s or 0s dominating the whole ranking, and the system could not output patterns which are as interesting as famous existing patterns which have names (Island reversal pattern), because even though there are such things, they still do not occur often enough that they beat patterns which are composed of consecutive 1s or 0s.
Analysis of patterns of 9 candles: 11 matches in Top 20

111011111 :
5260 occurrences in UDMax10Cdls01.27.1992-01.21.2010.txt
393 occurrences in UDMax10Cdls01.21.2010-01.25.2012.txt

110111111 :
5431 occurrences in UDMax10Cdls01.27.1992-01.21.2010.txt
395 occurrences in UDMax10Cdls01.21.2010-01.25.2012.txt

111100111 :
2526 occurrences in UDMax10Cdls01.27.1992-01.21.2010.txt
400 occurrences in UDMax10Cdls01.21.2010-01.25.2012.txt

110011111 :
2468 occurrences in UDMax10Cdls01.27.1992-01.21.2010.txt
384 occurrences in UDMax10Cdls01.21.2010-01.25.2012.txt

...
pattern and then outputs the list of those patterns at the end.

For example, one random candle can be picked from the past market data. It can be anything, such as even “1” (white candle, meaning in this time interval the price rose). The simplest estimation the system could do would be to go back through all the data, count how many “10”s (one white candle followed by one black candle) patterns there were and compare them with that of the alternative pattern, which is “11”. This approach may sound too simple, but this is merely an example considering only two candles. The consideration can be done from even a single candle to any number of candles. For example, if the decision is made to consider seven candles and the latest candle displayed on the screen is (or it could be anywhere in the dataset to simulate the prediction feature) the last candle of a pattern “100101”, the system can count the number of times patterns “1001011” and “1001010” occur, to find out which pattern is more common in general so that this information can be used to help predict the next move of the price.

The following example is a prediction of the next price movement after the latest candle price in the CSV file, outputted by the program.

```
total number of "1": 408329.0  0.598042403365
total number of "0": 274447.0  0.401957596635

pattern, occurrence: ['11', 249210]  probability: 0.610316680912
pattern, occurrence: ['10', 159119]  probability: 0.389683319088

pattern, occurrence: ['111', 155798]  probability: 0.625167529393
pattern, occurrence: ['110', 93412]   probability: 0.374832470607

pattern, occurrence: ['0111', 55188]  probability: 0.590802038282
pattern, occurrence: ['0110', 38224]  probability: 0.409197961718

pattern, occurrence: ['00010011',388] probability: 0.552706552707
```
Example of an estimation output. Full list available in Appendix D.1.

The above table shows the total number of times pattern “1” and “0” occurred (if a candle in one specific period of time was white, meaning rise in price, or black, meaning fall in price). Each pair of rows starts with patterns, the number of times they were counted and the ratio of the number of times they occurred to the number of times the other pattern was counted, in order. The further the percentage of the ratio gets, from either 60% or 40% of the original ratio of the number of times “1” occurred to the number of times “0” occurred, the more likely it is that one of the two patterns occurs more often the other.

It considers from two to nine candles. For each length of patterns, it counts their occurrence in the past data. The longer the patterns get, the smaller the number of times they occur, in general.

The ratio of total number of “1”s to the total number of “0”s was almost 60% to 40%. Therefore it will be natural to have more “1”s in general. The result showed that, surprisingly, all the patterns of different length showed nearly the same ratio as the original.

For instance, the short pattern “11” occurred 249210 times and “10” occurred 159119 times, therefore the ratio is 61% to 39%. As for the longest patterns, the pattern “0000100111” occurred 388 times, and “0000100110” occurred 314 times, therefore the ratio is 55% to 45%. It can be said that this ratio is relatively close, for one pair of patterns occurs nearly 640 times more often than the other pair. However, there is a definitely a difference between this and the whole ratio of 60% to 40% which is 5%. This means that under the condition both patterns “1” and “0” occurring 50% of the time, pattern “0000100110” has a higher chance of occurring than pattern “0000100111”, meaning in the long term this system can be used to
make a profit, again under the condition of pattern “1” and “0” occurring with the same frequency. Given a current pattern that’s composing the market price, the system can output whether one should buy or sell, as it was just proved that it is possible that one direction of price change can have higher probability of occurring than the other direction, in this example.

To get more information and for accurate observations, another random candle pattern closer to the end in the dataset was picked, to see another sample of the estimation output.

```
=================================================================
total number of "1": 379666.0  0.603766683788
total number of "0": 249163.0  0.396233316212

pattern, occurrence: ['11', 234297] probability: 0.617113462886
pattern, occurrence: ['10', 145369] probability: 0.382886537114

pattern, occurrence: ['1011', 51099] probability: 0.600345410969
pattern, occurrence: ['1010', 34017] probability: 0.399654589031

pattern, occurrence: ['101011', 11571] probability: 0.590628349752
pattern, occurrence: ['101010', 8020] probability: 0.409371650248

pattern, occurrence: ['0011101011', 667] probability: 0.590788308237
pattern, occurrence: ['0011101010', 462] probability: 0.409211691763
...
=================================================================
```

Another example of an Estimation output. Full list available in Appendix D.2.
Again, all different lengths of patterns showed relatively close ratios to their original ratio of the total number of 1s to 0s. However, one thing that still needs to be discovered is whether or not this phenomenon of patterns not affecting the price change in the next period of time, applies to all cases. Therefore an extra experiment was run to make those estimations at every single point of candles over the past chart data.

A model of taking the first 80% of the entire data as learning sets was used. The rest of the data was used for the experiment where the program records patterns whenever the ratio of number of occurrence of patterns followed by 0 to that of 1, differs by more than 8% from the ratio of the total number of 0s to that of 1s. The following is a part of the output from the program.

```
0:1 = 0.401955270442:0.598044729558  Pattern: 11111111
count of occurrence of this pattern followed by '0': 5982
ratio of this pattern followed by '0' to '1': 0.239107842354

0:1 = 0.401889696828:0.598110303172  Pattern: 101111111
count of occurrence of this pattern followed by '0': 1677
ratio of this pattern followed by '0' to '1': 0.290439903014

0:1 = 0.401908932238:0.598091067762  Pattern: 111101111
count of occurrence of this pattern followed by '0': 1763
ratio of this pattern followed by '0' to '1': 0.306875543951

0:1 = 0.396605121437:0.603394878563  Pattern: 110000000
count of occurrence of this pattern followed by '0': 215
ratio of this pattern followed by '0' to '1': 0.476718403548
```

Example of an estimation output for every candle. Each row is composed of the ratio of the
number of times pattern “1” and “0” have occurred at the point this pattern was found, the pattern, the number of times the pattern of the current pattern followed by “0” occurred, and its ratio to that of the current pattern + “1”. Full list available in Appendix E.

In this output, it can be seen which patterns occur more often than others. In other experiments, it was always the case that patterns which were composed of consecutive rises or falls in price, dominated other patterns, however in this method, just by changing the deviation percentage (8% in the example above), all patterns that can be used to predict the price change in the next time interval could be found.

In the first row of this output, the ratio of 0 to 1 following the pattern “11111111” is 0.24:0.76, where the ratio of total number of 0 to 1 is 0.4:0.6. This means that there is a much higher chance of price rise happening in the next time interval. As for both predicting price rise (seeing pattern “1” next) and predicting price fall (seeing pattern “0” next) in the next time interval, the system sets its 50:50 position fairly by changing it according to the past and current counts of the patterns (patterns of either “1” or “0” in this example) of the whole dataset (learning), and therefore it is able to estimate in which direction the price is going to move, under the condition of having probability of price rise and fall happening by 50% chance in general.

However, this obviously is not the case, so in this example, although some patterns that were composed of many “0”s, had a high chance of being followed by another “0” pattern, often (or always) it was still the case that the probability of that pattern followed by “1” was higher. That implies that whatever the current pattern or price is on the screen, one should always buy, not sell, but again this is because the price of the market which was used for this project, rose in the past overall. One cannot know if it will be the same in the future. However this is always the case when it comes to learning and analyzing from the past: you are supposed to assume that the things that happened in the past will happen again. Therefore if this system is ever going to be used for a real prediction of the future price of oil for trading, it makes sense to only buy oil, when the latest pattern shown on the screen is in the list above (or more, depending on the deviation percentage it is set to), as the most recent trend of the market was an uptrend. Again, as the result showed many patterns which are composed of consecutive rises or falls in price, it could be described as a
“Momentum” or “Trend”.

Also, the fact that there are no short patterns which are composed of consecutive rises or falls in this output supports the theory that long patterns which are composed of price movements of the same kind, tend to continue as more and more people join thinking there is a trend occurring. But for those short patterns which are composed of price movements of the same kind, these are followed by a new price movement (either pattern “1” or “0” in the example above), according to the ratio of the total number of pattern “0” to “1”. Most patterns in the list above which are quite different from typical patterns (composed of price movements of the same kind), have a reasonably low number of occurrences (i.e. pattern “100100000” occurred only 182 times), so there is quite a high chance that these appeared on the list due to a bias in the dataset that was used.

Therefore, if predictions were to be made using this system, it would always be safer and more accurate to trust the patterns which have a reasonably high number of occurrences depending on the length of the pattern.
Conclusions

There are two main types of investment analysis when analyzing price changes of stocks, currencies, or futures contracts, which are fundamental analysis and technical analysis. In this project the technical analysis approach was used, where the system which was developed for this project can parse the past dataset of the price of oil, create a library of these patterns and note their frequency using candlestick charts. The frequency was used to assign a probability of occurrence.

Three different approaches were taken to find patterns in price movements in the market. The first approach considered all four major variables that each bar in the candlestick chart holds, which are open, high, low, and close. The result showed that, although the system found patterns which occur more frequently than others, there were not many, as patterns were defined in so much detail that they did not match easily with other patterns.

The second approach was a simpler version of the first approach, where it considers only high and close, meaning whether the price rises or falls in a specific period of time. Using this approach, more patterns were found. However they were somewhat similar to the results of the first approach, in the sense that many patterns which were found, were composed of consecutive rises or falls in price movements.

The third approach took the idea of computing the probability of the co-occurrence of two patterns within a specified period of time, learning the behavior patterns, and outputting a frequency of occurrence that has been learnt. The experimental results showed that by using co-occurrence of two patterns, we can find out which patterns have a higher probability of occurring than other patterns. In this project, only patterns of price rise or fall in the next time interval were considered to get co-occurrence with every pattern that was predefined in the second approach.

Although this project was very successful with obtaining a list of patterns which occur more frequently than others and also a list of patterns with their probability of price rise or price fall in the next time interval, according to the point of time in the dataset the system is looking at, the result was heavily affected by the market trend in the long term, therefore it would not be guaranteed that this system will help investors make profits if
it is ever going to be used, as no one can be sure of a long term trend in the future.

The most common kinds of patterns which were found throughout this project, were composed mostly of consecutive rises or falls in price. This fact proved that so-called “Momentum” or “Trend” exists in the market.

Future enhancements could take the following approaches:

- Improved design of defining patterns.
- Algorithm for computing occurrence according to its current trend in the market.
Appendices

Appendix A   Class which represents each bar in the candlestick chart

```python
class Bar:
    VALUE = 1.009

    def __init__(self, date, time, openPrice, highPrice, lowPrice, closePrice):
        self.date = date
        self.time = time
        self.openPrice = float(openPrice)
        self.highPrice = float(highPrice)
        self.lowPrice = float(lowPrice)
        self.closePrice = float(closePrice)

    def isWhite(self):
        return (self.closePrice >= self.openPrice * Bar.VALUE)

    def isBlack(self):
        return (self.closePrice * Bar.VALUE < self.openPrice)

    def closesHigher(self):
        return (self.closePrice >= self.openPrice)

    def closesLower(self):
        return (self.closePrice < self.openPrice)
```

Bar class that represents a single Candle Stick, written in Python.
Appendix B  Drawer class to visualize the candlestick chart given a dataset of price

```python
from Tkinter import *
import Bar

class Drawer:
    WIDTH = 800.0
    HEIGHT = 400.0
    CANDLE_WIDTH = 25.0
    CANDLE_HEIGHT = 200.0
    RATIO = 30  # positive value only

    def __init__(self):
        self.canvas = Canvas(width=Drawer.WIDTH, height=Drawer.HEIGHT, bg='white')
        self.canvas.pack(expand=YES, fill=BOTH)
        self.top = 0.0
        self.center = 0.0
        self.bottom = 0.0
        self.CnvRtio = 0.0
        self.PrevOpen = 0.0
        self.PrevHigh = 0.0
        self.PrevLow = 0.0
        self.PrevClose = 0.0

    def drawBars(self, bars):
        tmpMax = 0.0
        tmpMin = 300.0
        for bar in bars:
            if(bar.lowPrice < tmpMin):
                tmpMin = bar.lowPrice
            if(bar.highPrice > tmpMax):
                tmpMax = bar.highPrice
        tmpMin -= 0.01
        tmpMax += 0.01
```
self.top = tmpMax
self.bottom = tmpMin
self.center = (tmpMax+tmpMin)/2

colour = ""
if (bars[0].openPrice < bars[0].closePrice):
    colour = "white"
else:
    colour = "black"

self.canvas.create_rectangle(19.5+Drawer.CANDLE_WIDTH/2, self.HEIGHT-(self.HEIGHT/2+(bars[0].highPrice-self.center)*self.CnvRtio), 20.5+Drawer.CANDLE_WIDTH/2, self.HEIGHT-(self.HEIGHT/2+(bars[0].lowPrice-self.center)*self.CnvRtio), width=4, fill='black')

    self.canvas.create_rectangle(20, self.HEIGHT-(self.HEIGHT/2+(bars[0].openPrice-self.center)*self.CnvRtio), 20+Drawer.CANDLE_WIDTH, self.HEIGHT-(self.HEIGHT/2+(bars[0].closePrice-self.center)*self.CnvRtio), width=4, fill=colour)

i = 1
while (i < len(bars)):

    if (bars[i].openPrice < bars[i].closePrice):
        colour = "white"
    else:
        colour = "black"

    self.canvas.create_rectangle(19.5+Drawer.CANDLE_WIDTH/2+(Drawer.CANDLE_WIDTH+20)*i, self.HEIGHT-(self.HEIGHT/2+(bars[i].highPrice-self.center)*self.CnvRtio), 20.5+Drawer.CANDLE_WIDTH/2+(Drawer.CANDLE_WIDTH+20)*i, self.HEIGHT-(self.HEIGHT/2+(bars[i].lowPrice-self.center)*self.CnvRtio), width=4, fill='black')

40
self.canvas.create_rectangle(20+(Drawer.CANDLE_WIDTH+20)*i,
self.HEIGHT-(self.HEIGHT/2+(bars[i].openPrice-self.center)*self.CnvRti o),
20+(Drawer.CANDLE_WIDTH+20)*i+Drawer.CANDLE_WIDTH,
self.HEIGHT-(self.HEIGHT/2+(bars[i].closePrice-self.center)*self.CnvRt io), width=4, fill=colour)

i += 1
mainloop()

if __name__ == "__main__":
bars = []
bars.append(Bar.Bar(0, 0, 104.2, 106.12, 99.12, 103.26))
bars.append(Bar.Bar(0, 0, 104.76, 107.12, 100.12, 104.61))

myDrawer = Drawer()
myDrawer.drawBars(bars)
Appendix C.1 Analysis report for patterns composed of 5 candles, 1

=================================================================

Analysis of patterns of 5 candles: 14 matches in Top 20

000000000001001111111 : 
426 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1078 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

010111111111111111111 : 
698 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1574 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

000000000000000000000 : 
612 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1383 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

010001111111111111111 : 
507 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1239 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

000000101111111111111 : 
628 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1376 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

111111111111111111111 : 
1579 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
9995 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt

111111111111111110100 : 
663 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt
1484 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt
11111111111011000000 :  
562 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1374 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt  

00000010001111111111 :  
479 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1065 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt 

00000010011111111111 :  
488 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1424 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt 

111111111111101110 :  
599 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1683 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt 

111111111111110110 :  
428 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1124 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt 

010011111111111111 :  
489 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
2039 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt 

111111111111110100000000 :  
537 occurrences in SortMax10Cdls01.21.2010-01.25.2012.txt  
1108 occurrences in SortMax10Cdls01.27.1992-01.21.2010.txt
Appendix C.2 Analysis report for patterns composed of 5 candles, 2

=================================================================

Analysis of patterns of 9 candles: 11 matches in Top 20

111011111 :
5260 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
393 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

110111111 :
5431 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
395 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

111100111 :
2526 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
400 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

110011111 :
2468 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
384 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

111110111 :
5288 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
387 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

111110011 :
2506 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
373 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

111101111 :
5358 occurrences in UMax10Cdls01.27.1992-01.21.2010.txt
387 occurrences in UMax10Cdls01.21.2010-01.25.2012.txt

111001111 :
2492 occurrences in UDMx10Cdl01.27.1992-01.21.2010.txt
381 occurrences in UDMx10Cdl01.21.2010-01.25.2012.txt

111111111 :
18656 occurrences in UDMx10Cdl01.27.1992-01.21.2010.txt
380 occurrences in UDMx10Cdl01.21.2010-01.25.2012.txt

111110111 :
5233 occurrences in UDMx10Cdl01.27.1992-01.21.2010.txt
385 occurrences in UDMx10Cdl01.21.2010-01.25.2012.txt

111111101 :
5291 occurrences in UDMx10Cdl01.27.1992-01.21.2010.txt
397 occurrences in UDMx10Cdl01.21.2010-01.25.2012.txt

=================================================================
Appendix D.1 Estimation output, 1

=================================================================
total number of "1": 408329.0  0.598042403365
total number of "0": 274447.0  0.401957596635

pattern, occurrence: ['11', 249210]  probability: 0.610316680912
pattern, occurrence: ['10', 159119]  probability: 0.389683319088

pattern, occurrence: ['111', 155798]  probability: 0.625167529393
pattern, occurrence: ['110', 93412]  probability: 0.374832470607

pattern, occurrence: ['0111', 55188]  probability: 0.590802038282
pattern, occurrence: ['0110', 38224]  probability: 0.409197961718

pattern, occurrence: ['00111', 22207]  probability: 0.577961117039
pattern, occurrence: ['00110', 16216]  probability: 0.422038882961

pattern, occurrence: ['100111', 13075]  probability: 0.587059985632
pattern, occurrence: ['100110', 9197]  probability: 0.412940014368

pattern, occurrence: ['0100111', 5259]  probability: 0.569279064733
pattern, occurrence: ['0100110', 3979]  probability: 0.430720935267

pattern, occurrence: ['000100111', 941]  probability: 0.566526189043
pattern, occurrence: ['000100110', 720]  probability: 0.433473810957

pattern, occurrence: ['0000100111', 388]  probability: 0.552706552707
pattern, occurrence: ['0000100110', 314]  probability: 0.447293447293
=================================================================
Appendix D.2 Estimation output, 2

=================================================================
total number of "1": 379666.0 0.603766683788
total number of "0": 249163.0 0.396233316212

pattern, occurrence: ['11', 234297]  probability: 0.617113462886
pattern, occurrence: ['10', 145369]  probability: 0.382886537114

pattern, occurrence: ['011', 86162]  probability: 0.592716416268
pattern, occurrence: ['010', 59206]  probability: 0.407283583732

pattern, occurrence: ['1011', 51099]  probability: 0.600345410969
pattern, occurrence: ['1010', 34017]  probability: 0.399654589031

pattern, occurrence: ['01011', 19932]  probability: 0.586442273744
pattern, occurrence: ['01010', 14056]  probability: 0.413557726256

pattern, occurrence: ['101011', 11571]  probability: 0.590628349752
pattern, occurrence: ['101010', 8020]  probability: 0.409371650248

pattern, occurrence: ['1101011', 6941]  probability: 0.600536424987
pattern, occurrence: ['1101010', 4617]  probability: 0.399463575013

pattern, occurrence: ['11101011', 4231]  probability: 0.616404428904
pattern, occurrence: ['11101010', 2633]  probability: 0.383595571096

pattern, occurrence: ['011101011', 1652]  probability: 0.594030924128
pattern, occurrence: ['011101010', 1129]  probability: 0.405969075872

pattern, occurrence: ['0011101011', 667]  probability: 0.590788308237
pattern, occurrence: ['0011101010', 462]  probability: 0.409211691763

=================================================================
Appendix E  Full list of patterns with both their occurrence and co-occurrence with another pattern, generated by the system

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Occurrence of pattern followed by '0'</th>
<th>Ratio of pattern followed by '0' to '1'</th>
</tr>
</thead>
<tbody>
<tr>
<td>110111111</td>
<td>1817</td>
<td>0.311877789221</td>
</tr>
<tr>
<td>111111111</td>
<td>4246</td>
<td>0.223051061147</td>
</tr>
<tr>
<td>000011100</td>
<td>296</td>
<td>0.472089314195</td>
</tr>
<tr>
<td>11111</td>
<td>20533</td>
<td>0.304892716609</td>
</tr>
<tr>
<td>000010000</td>
<td>340</td>
<td>0.468965517241</td>
</tr>
</tbody>
</table>

48
occurrence of this pattern followed by '0': 182
ratio of this pattern followed by '0' to '1': 0.469072164948

0:1 = 0.388522753633:0.611477246367          Pattern: 01000000
occurrence of this pattern followed by '0': 188
ratio of this pattern followed by '0' to '1': 0.468827930175

0:1 = 0.3931411797:0.6068588203          Pattern: 00100000
occurrence of this pattern followed by '0': 224
ratio of this pattern followed by '0' to '1': 0.473572938689

0:1 = 0.388454209129:0.611545790871          Pattern: 00001000
occurrence of this pattern followed by '0': 146
ratio of this pattern followed by '0' to '1': 0.469453376206

0:1 = 0.401896034684:0.598103965316          Pattern: 11011111
occurrence of this pattern followed by '0': 2710
ratio of this pattern followed by '0' to '1': 0.31747891284

0:1 = 0.401957036588:0.598042963412          Pattern: 01111111
occurrence of this pattern followed by '0': 2661
ratio of this pattern followed by '0' to '1': 0.307879208608

0:1 = 0.394263259618:0.605736740382          Pattern: 00001010
occurrence of this pattern followed by '0': 201
ratio of this pattern followed by '0' to '1': 0.475177304965

0:1 = 0.401955270442:0.598044729558          Pattern: 11111111
occurrence of this pattern followed by '0': 8643
ratio of this pattern followed by '0' to '1': 0.256765990315

0:1 = 0.401909521227:0.598090478773          Pattern: 11111011
occurrence of this pattern followed by '0': 1645
ratio of this pattern followed by '0' to '1': 0.292808828765

0:1 = 0.401933179609:0.598066820391          Pattern: 11111101
occurrence of this pattern followed by '0': 1744
ratio of this pattern followed by '0' to '1': 0.306610407876

0:1 = 0.394260950734:0.605739049266  Pattern: 001010101
occurrence of this pattern followed by '0': 359
ratio of this pattern followed by '0' to '1': 0.474867724868

0:1 = 0.401896131955:0.598103868045  Pattern: 11101111
occurrence of this pattern followed by '0': 1668
ratio of this pattern followed by '0' to '1': 0.295064567486

0:1 = 0.401955270442:0.598044729558  Pattern: 111111
occurrence of this pattern followed by '0': 13151
ratio of this pattern followed by '0' to '1': 0.280932239597

0:1 = 0.397113151078:0.602886848922  Pattern: 001010000
occurrence of this pattern followed by '0': 240
ratio of this pattern followed by '0' to '1': 0.477137176938

0:1 = 0.401909521227:0.598090478773  Pattern: 11110111
occurrence of this pattern followed by '0': 2655
ratio of this pattern followed by '0' to '1': 0.316071428571

0:1 = 0.395593512457:0.604406487543  Pattern: 000000011
occurrence of this pattern followed by '0': 267
ratio of this pattern followed by '0' to '1': 0.481949458484

=================================================================
Appendix F  Electronic Sources and Resources

Please see the CD attached to this project for all the electronic sources and resources (code, more lists of patterns, etc.). The CD also contains a README.txt file which describes the following.

The folders are:

src     - The Python source files can be found here, where:
  • ChartA main class is the latest code that was used.
  • ChartThread is the same program which was written in parallel programming style. Not in use anymore, as parallel programming did not help the system performance.

docs    - Main result documents which were generated by the program, where:
  • All the files are formatted as described throughout the report.
  • demo_CL_5min_24hr_max.txt is a part of the main CSV file that was used. The original CSV file cannot be put on the CD for confidentiality reasons.

project - All the project files and folders are put here, therefore the codes are executable. More documents generated by the program in a format described throughout the report can also be found here, as well as all other codes which were written before the final implementation of the system.
Bibliography


