Bottom-system: Data Mining, Data Processing, Supporting the Decision-making

Alisa Dammer

December 3, 2014

1 Introduction

The modern trend in stock-movements prediction is taking social opinion into account. Prices of individual shares and different indicators are heavily influenced by people's expectation and general mood. This data is called "sentiment data". There are different tools existing that extract the general mood or particular information from social media. One of the programs working with sentiment data specially for stock exchange is called "StockPulse". According to the information given on the official web site, the whole process is broken into 7 Steps:

1. Data collection
2. Spam-filter (Data selection)
3. Mood detection
4. Topic - assigning title to message (Message coding?)
5. Relevance assignment (Message rating)
6. Aggregation and analyses
7. Trend signal

The main focus of the system is to support trader's decision by providing them with the forecast and general trade analysis. However the output of the program can be used mainly in short-term decision making.

In this work I would like to check one hypothesis based on two general Assumptions:

Assumption 1: Every index can be presented as a combination of several other indexes (Here "best-match" model with limited number of compositors will be considered).

Assumption 2: Some (From 1 to N) of the sub-indexes can have an in time postponed impact on the main index

Hypothesis 1: The analysis of each sub-index instead of directly connected data, can give an extra advantage for a trader in the mid-term decision making.

Hypothesis 2: The indirect data analysis can result in strategic advantage for an portfolio-investor.

In order to check the hypothesis certain system will be build that deals with every stage mentioned above. But first of all limitations for the system need to be set.

2 Limitations for the system and Data-mining

As was mentioned in introduction, this work will meet many restrictions and limitations in order to reduce the complexity, run-time and make it more transparent.
First of all the top limitation is to choose the primary index (stock ration, interest, price of an obligation). Also this index should be relatively easy to present as a composition. (here give examples of possible primary indexes and choose one to work with. The pyramid o the increasing parameter's number can be added to graphically demonstrate why limitations are important, at the same time give the O-notation, maybe o-notation as well)
After choosing the primary index, we will have to choose a composition of other indexes, that will present the main index as correct as possible. The limitations for this stage are:

1. The number of compositors should be pretty small. Let's say between 3 and 5 (Here is very important to remember that the processing of each parameter is working with N text-sources which leads to M*N complexity increase just on this stage. Here M - is amount of compositors, first-level parameters, N - the number of texts processed) establish some sort of cross-correlation matrix).

2. The compositors itself should be as simple as it gets. Their own influence should be significant and those indexes should not be closely correlated to each other. The reason for this limitation is again complexity and run-time reduction. (see figure with complexity-pyramid above). Do we include negative-impact-indexes? Only if they are obvious

3. The number of the text-files reviewed for each compositor should be restricted. Here let's set the number to 25-50 texts. All found texts will be processed and each text will get an unique id. The way how these IDs are composed defines the number of further limitations (transparency and complexity reduction)

   (a) The information from every text analyzed is extracted into several indexes: ID and desirable estimation (forecast for specific index here again not to forget. What to do with cross information.

What if on text contains 90% info about index(1) and 10% about index(2), 80/20 etc. How do indexes interact with each other? Weighted importance index - weighted cross-correlation?) and saved into independent arrays.

(b) The ID contains the information about the content (here not yet sure how this information will be presented: number of keywords with distance between them, the amount of "noise" etc.). This information is needed while mining, the system will do following thing: If the list of text's ids is empty, the text will be explored, and proper ID is assigned, ID is added to the special array. Next text is explored, ID is assigned but now the ID's array is not empty, so new ID will be compared with existing ones. If they match, than the counter for this ID is increased, but ID itself is not added to the array. (here array.count(x) is not equal to [id,(count, weight)], because we have ration for sources and the same text in Facebook won't have the same importance, as posted in business magazines about stock exchange). If the ID is unique, it will be added to the array.

Texts itself won't be saved anywhere, because of the rights policy and space saving. Instead (sub-array or multidimensional array for ID?) we will save the text's link (URL) the source as a part of unique ID (currently I see ID as a number, where first 3-4 digits present the link to the source, and the rest presents the content, the URL will be assigned an unique ID)
here graphically show how increasing number of texts influence spacing and run-time

These were main limitations for data mining. (types of sources will be better to describe in the "Data Mining" chapter, otherwise the information will be logically mixed, which is not appreciated here at all)

3 Data Mining

After setting limitations for the number of parameters, we start an important step - searching and choosing proper information sources. There are three main types of sources, only two will be used in this work.

1. Official business articles and government's reports - these sources are self-sufficient separated pieces of the information with the highest priority. The highest priority is given to the official reports, because it is not an interpretation in any form, but the fact - raw numbers. Actual and up-to-date reports are important for the fine tuning of the priority list and thus importance-weight matrix. (additional array for fine-tuning) For estimation I consider separate statements and announcements and official government's forecasts as a good and trusted source (but still not an absolute truth).

2. The second type of texts are articles in specialized magazines (it can be a bunch of articles, or debate-like article but with clear conclusion). This source is different from the first one, because it requires preprocessing to find one text and also, the level of experts there can be considered a bit lower in comparison to the previous source. At the same time the experts published in magazines are different, that mean, that not only institutes (publishing institute here) have ranks, but also individuals can have quite a big weight and their opinion may be considered more important as, for example, an article from a non-specialized magazine. (probably specialization here should influence the ratio)

3. The third type of the information sources publicity - social media, like Facebook, blogs, Live-Journals and tweeter. These sources are pretty sticky - you can't check the personality behind the text (u can't always believe, that the person behind the text should be ranked high or as a "spam"). Thus I decided not to use this source, to decrease run-time and increase transparency - there will be huge problems with rating such texts (also re-tweets are an enormous headache)

4 Data processing

The main idea is to break main index into several sub-indexes in order to get some additional ”hidden” information, that i not discussed now, but will have huge impact in the future.

Separate data processing as an output will have confidence interval for each of the sub-indexes. And than, the way the index is broken into several indexes and the way all results are integrated into one forecast are different separate results will be integrated into one forecast that will contain following information: forecast itself, probability and all th for several time steps. So the general output should look like an array ordered by time.

I haven’t consider the way to integrate separate results yet. I will need some guidance here as first step I will do it manually based on some logic and original model, that was used in composition.