RESEARCH PAPERS OF WROCŁAW UNIVERSITY OF ECONOMICS No. 147
Advanced Information Technologies for Management – AITM 2010 2010

Aleksander Fafuła
Wrocław University of Economics, Wrocław, Poland
e-mail: aleksander.fafula@ue.wroc.pl

A CONCEPT OF A PLATFORM FOR DATA-DRIVEN APPROACH TO DETECT COGNITIVE BIASES WITH FEEDBACK

Abstract: Cognitive biases often influence decision processes related to investment on stock markets. Mainly, these concern complex problems with perception and understanding of surrounding reality. This research was aimed to detect cognitive biases in the data-driven manner. A few basic cognitive biases are examined: gambler's fallacy, base rate fallacy, clustering illusion, bandwagon effect, overconfidence effect. This paper presents a new concept of a platform which can detect specific users behaviours. Finally users evaluate algorithms and rules found. Along with the standard functionalities of a stock market simulator a few methods of data mining will be applied: inductive decision trees, associations, clustering and more.

Keywords: cognitive biases, trading simulator, trading support software, patterns in behaviour.

1. Introduction

The main question in this project is the following: is it possible to detect cognitive biases automatically? Let us assume we can create decision rules, associations or clusters which can help us detect investors behaving in the way which can bring them to the bankruptcy. What if these rules are sufficient enough to even give live feedback and help users of the system displaying "You are doing it wrong" dialog? Finally we want our user to evaluate hints sent by the machine. But... how eager are we to admit that we are making mistakes?

The science of Investment Psychology points out various cognitive biases. They have been widely tested for many decades. The best known publications in this area are: *Judgment Under Uncertainty: Heuristics and Biases* by Kahneman, Slovic, Tversky (1982), *A Mind of Its Own: How Your Brain Distorts and Deceives* by Fine (2006), *Cognitive Biases in Market Forecasts* by Fisher and Statman. Polish best known work seems to be *Psychologiczne pułapki oceniania i podejmowania decyzji* (*Psychological traps in judgment and decision-making*) by T. Tyszka (2000). Due to

the nature of this matter some models of behaviour are known better than the others. In this study a few of them have been chosen as they appear to be more common, namely:

1. Gambler's fallacy – the tendency to think that future probabilities are altered by past events, when in reality they are unchanged. Results from an erroneous conceptualization of the law of large numbers. For example, if an abundance of heads comes up on a coin, observers may be heard to assert that a tail is due; that it is more likely to come up than another head [O'Neill, Puza 2004].

2. Base rate fallacy – this is people's tendency to ignore base rates in favour of, e.g., individuating information (when such is available), rather than integrate the two. This tendency has important implications for understanding judgment phenomena in many clinical, legal, and social-psychological settings [Bar-Hillel 1980].

3. Clustering illusion – the tendency to see patterns where actually none exist. For example T. Gilovich found that most people thought that the sequence "OXXXOXXXOXXOOOXOOXXOO" looked non-random, whereas, in fact, it has several characteristics maximally probable for a "random" stream, such as an equal number of each result and an equal number of adjacent results with the same outcome for both possible outcomes [Gilovich 1991].

4. Bandwagon effect – the tendency to do (or believe) things because many other people do (or believe) the same. Related to groupthink and herd behaviour [Leibenstein 1950].

5. Overconfidence effect – excessive confidence in one's own answers to questions. For example, for certain types of question, answers that people rate as "99% certain" turn out to be wrong 40% of the time [Pallier et al. 2002].[1]

Full list and extensive descriptions of experiments on cognitive biases detection can be found in the literature. However, cognitive science has not been introduced to data-mining at scale provided by computing power and network availability. It is unclear and thus needs further investigation if data-driven approach is able to give additional point of view for cognitive biases detection.

This paper presents a brand new concept of a system which can help improve traders by reducing cognitive biases ratio. Simulator presents the data-driven approach to the detection of cognitive biases. To make results more clear the study focuses on newbie investors. After patterns are detected users evaluate them. Finally, patterns with the highest scores create the rules for the next investors.

This paper describes seven stages to build and deploy the trading platform which can track users' behaviours and inform them about the cognitive biases. The technology is clearly defined, based on the author's experience. This early methodology should be treated as the first debut in this area of research. In the beginning the paper

[1] The numbers above correspond with the numbers of questions provided in the next chapter. Probably only one cognitive bias should be tested at once. This will be taken under consideration in the future work.

describes technical issues, along with the main algorithm. Finally some conclusions and affiliations with the virgin run of the prototype of the platform are made. The most important elements of the system are visualised by the diagram (Figure 1).

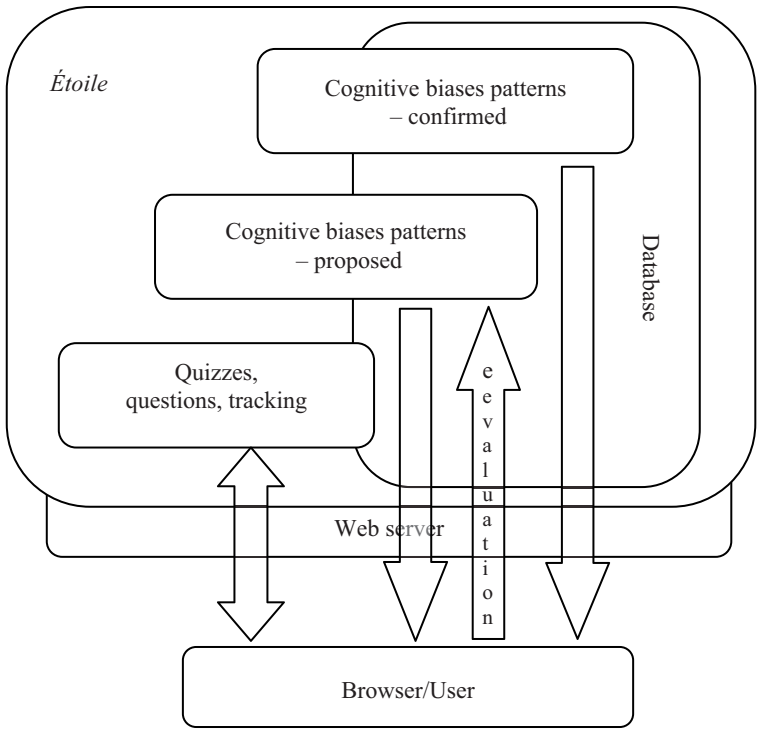

Figure 1. Elements of the platform with their interactions

The diagram above describes basic data and mechanisms' arrangement of the experimental platform. This does not reflect the real database structure, but provides reader with the broad view of the architecture. Elements included and used by the platform are: user interface (served by web server) – quizzes and stock market simulator's controls; layer of the data – cognitive patterns, users behavioural data, stock data (stored in database). The following section of this paper describes them in detail, proposing very early methodology of creating the platform to detect cognitive biases using data-driven approach.

2. Description of experiments and applied methods

The experiment will be conducted in a form of investment game. Special stock market simulator is being written to provide specific functionalities. These include abilities for storing additional information in the database, collecting data using questionnaire,

displaying additional views for administration. In this project a software platform has been developed. The codename of this web application is French *Étoile*.

Stages and main elements of the experiment are the following:

1. Platform development.
2. Attachment of quizzes.
3. Data mining process using WEKA.
4. Interpretation of the results.
5. Integration of the results with cognitive biases and hints.
6. Continuation with the game, but add hints.
7. Users evaluate hints.
8. New users use the highest rated rules from the previous experiments.

Stage 1

As a platform a CodeIgniter framework will be chosen because of the small memory footprint, ease of coding, speed, and, of course, because it utilizes the MVC architecture. The software is going to be extended, so there is a need of using some easy to understand paradigm. MVC is a logic which enables programmer to separate application logic from presentation. The Model represents the data structures (usually database and mechanisms of communication, retrieving, inserting, selecting). The View is the information provided to the user. It can be HTML page, RSS, etc. Finally, the Controller is the mechanism that processes users' requests, and interacts with Model and View, to generate the web page to the user. The MVC architecture in this case is described by the scheme in Figure 2.

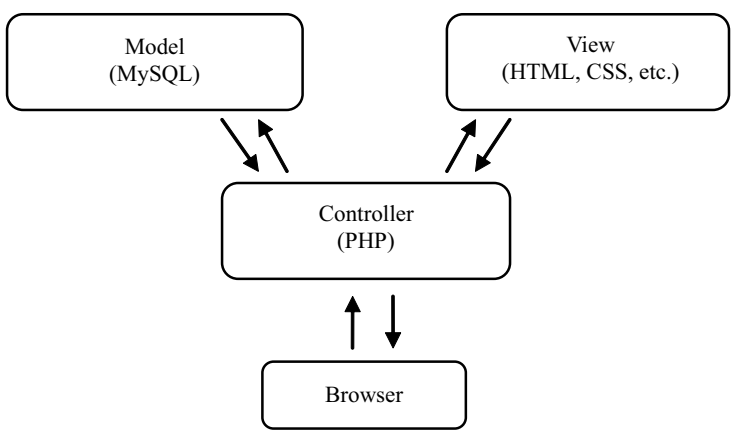

Figure 2. Model MVC

The main advantages of building the system in the Model-View-Controller way:

– display and processing are separated,
– improved security,

– higher coding standard (additional programmers can easily join),
– internationalization is simple,
– additional modules, classes and helpers,
– ORM (object-relational mapping) instead of SQL, which gives possibility of easy database migration,
– clean code,
– flexibility (this beta platform will be improved many times).

The game is intended for beginning traders, because they more often have less experience and thus are under higher impact of cognitive biases. Traders will be able to deal with many stocks through intuitive and easy interface. Trading data will be fetched by external applications executed by the cron daemon. After validating and parsing it will be stored in the database. Operating system of the host machine is Ubuntu. Extra packages needed by this project are:

– PHP – a very popular web programming language,
– Apache – a very common and widely used http server,
– MySQL – database of choice – CodeIgniter works best with MySQL,
– Java – needed by Data Mining software,
– WEKA – software package to mine data from the databases.

Étoile was designed as an interface for acquiring data from traders and bots using various methods. It is constructed on a well-known client-server architecture. Further, processing is done by external applications and imported back to *Étoile*. The application is used for:

1) data collecting (from brokers and bots/agents),
2) data cleaning (to filter, remove noise and inconsistencies),
3) data integration (to merge data from stocks and various databases),
4) data selection (export from database to data mining applications),
5) data transformation (data is being transformed to suitable formats).

Stage 2

As mentioned earlier, the main front-end is a simple Website. It can be accessed by any modern Web browser, palm device, tablet or a smart phone. User is able check here some basic information, such as current server time (UTC), quotes registered by the server, their capital and other. It also has buttons which one presses when buying or selling. Information displayed depends on the mode in which the program is operating, that is, anonymous, competition, or developer. Moreover, it is possible to change some basic settings.

First of all, the user is obliged to create an account, and then log in to the platform. Next, he or she should choose desired quotes and start ordinary trading by using own system and pressing buy/sell *Étoile*'s buttons. History of trading operations was also implemented. It displays information like: type of operation (buy/sell), time/date, quantity, price, volume, commission, total turnover. Current close is updated with AJAX so there is no need to reload Web page.

On the other hand, the server basically collects buy/sell signals from brokers. Then they are prepared to be parsed. This was done by filtering abnormal signals and checking for erroneous values. It is possible to attach any number of trading bots using various decisions and collect their signals, too. This is done by http GET method and it is simple to implement. This gives the system an opportunity to compare signals from brokers and bots and detect which algorithm has been used by the broker. Users can adjust the system's parameters so the given probability is achieved. Moreover, it is possible to search for similarities and patterns not only among brokers (real and bots) but also in a single session registered for user. Decisions can be later evaluated and explained. This gives one a chance to find patterns in trading behaviour and offer some advice in the future.

To build a matrix of stock data and human behaviour, a bit of cognitive science was implemented: psychological responseness and reaction time. It was achieved with the use of questionnaire. Questions use various input methods, for example: yes/no/don't know, text input, multiple choice, scoring (1-5). After every operation (i.e. buy/sell) user is being forced to answer five simple questions. This is a work in its very beginning and needs further investigation.

It is worth mentioning that quizzes have been arranged in a new way. After every Buy/Sell operation the system displays relevant quiz randomly. There are three types for each operation, so that makes 6 in total:

1. Buy/pick up items which describe situation (used for association discovering).
2. Sell/pick up items which describe situation (used for association discovering).
3. Buy/select actions which make this decision good (used for decision tree construction).
4. Sell/select actions which make this decision good (used for decision tree construction).
5. Buy/evaluate the risk (used for clustering, gathers more info in background, such as transaction amount, etc.).
6. Sell/evaluate the risk (used for clustering, gathers more info in background, such as transaction amount, etc.).

Welcome to Étoile

Your decision has been registered. Please fill the form below to continue.
All questions must be answered.

Q 1. Check words which correctly describe this operation and/or your decision:

[ ] risky	[ ] a little bit late	[ ] it's a guess	[ ] I feel lucky
[ ] safe	[ ] just in time	[ ] too fast	[ ] proved by algorithms

(...)

[ ] my friend's choice [ ] based on media info [ ] seen this situation before

Q 2. Select options which prove your decision right:

I have seen similar pattern in the diagram before I have made this decision. [yes/no/uncertain]
(...)
I have been selling too much. [yes/no/uncertain]

Q 3. How do you evaluate your decision (1 = risky, 5 = sure):

1 2 3 4 5
[Submit button]

Listing 1. Étoile – experimental platform for behavioural pattern recognition – questionnaire for the user after investment decision

Stage 3

Using WEKA in batch mode is very well documented. It is not clear which algorithms of data mining will provide the best results. That is why this section will be described in detail after the first run of the game. Three main applications are planned: clustering, inductive decision trees, and looking for associations. After manual tests WEKA will be put in the batch mode. It is handy to copy commands from The Explorer log window or to use classes directly. WEKA can be also run directly, as shown on Listings 2 and 3

```
java weka.classifiers.meta.AttributeSelectedClassifier \
  -t <training.arff> \
  -E „weka.attributeSelection.CfsSubsetEval -M" \
  -S „weka.attributeSelection.BestFirst -D 1 -N 5" \
  -W weka.classifiers.trees.J48 \
  -- \
  -C 0.25 -M 2
```

Listing 2. Using WEKA directly – example of using CfsSubsetEval and BestFirst

Source: http://weka.wikispaces.com/Performing+attribute+selection.

```
...
 === Run information ===

Scheme: weka.clusterers.SimpleKMeans -N 2 -A „weka.core.EuclideanDistance -R first-last"
-I 500 -S 10
...
kMeans
======
```

```
Number of iterations: 4
Within cluster sum of squared errors: 26.0
Missing values globally replaced with mean/mode

Cluster centroids:
                        Cluster#
Attribute     Full Data         0         1
                   (14)      (10)       (4)

Clustered Instances

0      10 ( 71%)
1       4 ( 29%)
```

Listing 3. Sample clusterer output

Listing 3 shows that sample output will have to be parsed by some external applications. This is needed to perform live calculations during the interaction with the users.

Stage 4

The interpretation of the results will be made by connecting rules found at the Stage 3 with the information provided by the psychology. It is clear that some assumptions will be more accurate and will better reflect the reality than the others. Moreover the margin of the error is not very narrow because user can agree with the system, even when it is not 100% compliant but rather falls into "I can say so" category.

Stage 5

Implementation of rules found during the Stage 4 will be performed manually. These will be implemented to react on users decision. Useful information will be displayed for him or her when in scope of a given rule. A brief illustration of this – we can construct a rule:

```
If [user-describes-situation-as (I sell too much; Risky; I feel lucky)]; then
// he owns basket of consciences that makes him consider the world incorrectly
        User_is_under_the_effect(Gambler's Fallacy);
Display_useful_information_on(Gambler's Fallacy);
```

Stage 6

Guessing rules would be useless without user evaluation. This is why we need to extend above "If rule" and add to this pseudo-code one more line:

```
Evaluate_usefulness_of_this_hint( (I sell too much; Risky; I feel lucky) => Gambler's Fallacy);
```

This will enable us to clean rules which have been assumed by mistake during the Stage 4. It will also automatically bump the rules that are precise to the Stage 7.

Stage 7

Finally, proven to be good, rules start to be used during this stage. New users that will join the game are provided with them from the early beginning. They can be diagnosed faster but also can evaluate the decisions (like during the Stage 6). This enables better rules (more common) to be detected earlier and leave the exotic ones in the tail and for special occasions.

Stages one to seven should be sufficient to build this enhanced version of trading platform. The plan has some drawbacks, like plenty of points, where programmers need to interact, but at this early stage there is no other way, than to coordinate it manually. Another improvement which should be included in the next release is an extended mechanism of taxonomy of events. These include: users reactions, timings, background data (this involves technical analysis). Anyway it is more reliable to build this kind of system in simple to understand and evaluate steps.

3. Discussion of results

The prototype *Étoile* has been already tested. This was achieved during the game organized for students of the Wrocław University of Economics. They could operate only on France Telecom stock. During the experiment time FTA has not experienced significant change. December 2009 was in falling trend until 21st, then a small rise followed. Users operated in mixed conditions. The game lasted about one month and ended on 11th of January 2010. The sample collected in previous run with the usage of *Étoile* shows some encouraging information [Fafuła 2010]. Thus it is clear that wider experiments should be carried out. Results show that the decisions of the users, supported and related to answers to simple survey, can indeed create patterns. This is why the improvements are planned and implemented in the current version of prototype.

It is also worth mentioning that patterns appearing as a result of the data collection process itself can be used for various purposes. For example, they can be reprocessed using different criteria, which will be discovered in the future. Samples can be stored in databases, reused, discussed, validated and improved. However, the most significant achievement is to use it for searching for the patterns in behaviour.

Conducting this experiment with newbie users also brings a few interesting ideas. They appear to be easier to "tag". This is because of the fact they do not operate on the investment markets professionally. Moreover they tend to use the same tools like web resources and applications. It is also likely they use similar technical analysis tools. This is also confirmed by the results of the survey.

It is clear that the group of beginning investors is the largest and needs more decision-supporting tools. The experiment conducted with prototype shows that they are influenced by similar and simple cognitive biases [Fafuła 2010]. Finally it is possible that perhaps sharing this knowledge with them is going to improve their behaviour and increase the value of portfolio. This makes justification for students and newbie investors to give this simulator a try.

4. Conclusions

Étoile is a work in progress and still under heavy development. It is necessary to discuss this experiment with the wider audience, involving especially: psychologists, financial specialists and traders. This will enable the system to be extended and improved. Technical plans for the future direct project towards its usability. In the area of psychology there is still a lot of improvements on the TODO list. Hopefully traders will enjoy the game and benefit from eliminating cognitive biases, one by one, playing the *Étoile*.

Another run of the game is going to take place since December 2010. Until then the platform is being developed and tested. Many points of failure and bugs have been fixed. Finally questions displayed to the user are more mature. Moreover mechanisms are upgraded so it is now easy to swap questions with the more sophisticated ones.

Acknowledgements

I would like to express the deepest appreciation to professor Jerzy Korczak, whose encouragement, guidance and support helped me to write the paper. His knowledge and experience in the research area, artificial intelligence and stock markets were invaluable.

References

Bar-Hillel M. (1980), The base-rate fallacy in probability judgments, *Acta Psychologica*, Vol. 44, pp. 211-233.

Fafuła A. (2010), A prototype of platform for data-driven approach to detection of cognitive biases, [in:] *Data Mining and Business Intelligence*, Ed. J. Korczak, Research Papers of Wrocław University of Economics No. 104, Business Informatics 16, Wrocław University of Economics, Wrocław, pp. 71-78,.

Gilovich T. (1991), *How We Know What Isn't So: The Fallibility of Human Reason in Everyday Life*, The Free Press, New York.

Leibenstein H. (1950), Bandwagon, snob, and Veblen effects in the theory of consumers' demand, *The Quarterly Journal of Economics*, Vol. 64, No. 2, pp. 183-207.

O'Neill B., Puza B.D. (2004), *Dice Have No Memories but I Do: A Defence of the Reverse Gambler's Belief*, reprinted in abridged form as: O'Neill B., Puza B.D., In defence of the reverse gambler's belief, *The Mathematical Scientist*, Vol. 30, No. 1, pp. 13-16.

Pallier G. et al. (2002), The role of individual differences in the accuracy of confidence judgments, *The Journal of General Psychology*, Vol. 129, No. 3, pp. 257-299.

PROJEKT PLATFORMY DO WYKRYWANIA BŁĘDÓW POZNAWCZYCH POPRZEZ PODEJŚCIE ZORIENTOWANE NA DANE ZE WSPARCIEM

Streszczenie: Błędy poznawcze wpływają często na procesy związane z podejmowaniem decyzji inwestycyjnych na giełdach papierów wartościowych. Do błędów tych zalicza się złożone problemy postrzegania oraz rozumienia rzeczywistości. Artykuł opisuje propozycję metodologii utworzenia systemu, którego przeznaczeniem jest detekcja błędów poznawczych. Wykrywanie tychże oparte zostało na metodach drążenia danych, a o poprawności detekcji decyduje sam użytkownik. Wybrano następujące błędy poznawcze: paradoks hazardzisty, iluzję grupowania, zaniedbywanie miarodajności, efekt polaryzacji i zasadę podczepienia. Do prowadzenia badań skonstruowano symulator giełdowy, który oprócz standardowych funkcji rozbudowany został o mechanizm prowadzenia ankiet psychologicznych oraz ich ewaluacji. Rezultaty badania są obiecujące i zasługują na pogłębienie studiów na ten temat.