Optimization of manufacturing processes in order to ensure the fastest possible fulfilment of the production plan

Atefeh Gooran Orimi
UOL, Department of medical physics and acoustics, atefehgooranorimi@yahoo.com
Seyyed Hamid Mousavi
FUM, Department of mathematics, hamidmousavi92@yahoo.com
Donatas Kavaliauskas
VU, Operational research department, donatas.worshipper@gmail.com
Narimantas Listopadskis
KTU, Applied mathematics department, narlis@ktu.lt
Kęstutis Lukšys
KTU, Applied mathematics department, kestutis.luksys@ktu.lt

Technical report for ESGI 142, 11-15 June 2018

Problem formulation

The furniture factory produces some articles which consist of several parts. Each article's structure and manufacturing sequence is defined by its bill of material (BOM). The necessary manufacturing operations are performed in specialized working centres with given capacity which may have alternatives. Every week a production quota is given, and it should be produced in optimal time with given restrains.

Task: Give some ideas how to automatically produce an optimal daily work centres production plan to meet the weekly production quota.

Mandatory conditions:

- At least 2 different article lots must be manufactured each day.
- The manufacturing sequence defined by the technological routing must be adhered to.
- If a finished article is combined out more than one packet, then these packets can`t be buffered.

How to determine if a plan is optimal (The higher the reason the more important it is):

- The least amount of parts/workpieces in the buffers between work centres.
- Shortest time to complete the production plan.
- One long idle is preferable to many short ones.

Terminology

- Article – A complete furniture piece, packed using flat pack methodology, sold to the customer.
- Packet – An assembly of finite number of parts.
- Workpiece – a part in an incomplete state of manufacture.
- Part – a finished piece out of which the whole article is assembled.
- Operation – A process which changes the properties of a raw material or a workpiece.
- Pallet capacity – Maximum number of parts, workpieces or packets that fits on a pallet.
- Work centre – a place where a raw material or a workpiece changes its properties.

- Buffer – a physical location before/after work centre, there parts/workpieces before the next operation are stored on pallets
- Alternative work centre - is used if a primary work centre is unavailable.
- Work centre capacity – Net duration of time (in a 24hour day) in which a work centre is capable of performing operations.
- Operation time – Time it takes to finish an operation in a given work centre.
- Technological routing – A sequence of operations done in work centres which results in a finished part.
- Bill of Material (BOM) – Article structure defined by raw materials, workpieces and parts.
- Pallet – Indivisible unit (Collection of identical parts or workpieces on a pallet) which follows the technological route one operation to the next.
- Technological pause – A mandatory period of time before the next operation can begin.
- Setup time – Time it takes to prepare the work centre for a different operation.

Flexible job shop problem

One branch of the industrial production scheduling problems is the job shop scheduling problem (JSP), which is among the hardest combinatorial optimization problems. The flexible job shop scheduling problem (FJSP) is a generalization of the classical JSP that allows to process operations on one machine out of a set of alternative machines. Hence, the FJSP is more computationally difficult than the JSP. Furthermore, the operation scheduling problem, the FJSP presents an additional difficulty caused by the operation assignment problem to a set of available machines [1].

In the given problem we have the case of FJSP since almost all work pieces has alternative work centres defined. Since we don't have preliminary knowledge on the how produced articles can be combined together and we want to offer flexible solution we decided to analyse genetic algorithm (GA) which. This methodology is efficiently applied for FJSP [1-4].

We used the chromosome representation encoded in two parts [2, 3]: operation sequence part (OS) and machine assignment part (MA). The first part OS is a vector with a length equal to the total number of operations, where each index represents a job which should be operated according to the predefined operations of the job set. The second part MA is a vector having the same length of OS, where each index represents the selected machine to process an operation indicated at chosen position.

Chromosome contains only the information about job sequence and on witch machines they will be processed. Time information is added during the decoding phase when the chromosome values are converted to an active schedule [1-4]. Startup time of each operation depends on selected machine availability and previous operations end time. In addition, setup times and operational pauses can be added at this point. Makespan value of the schedule is the finishing time of the last operation.

Chromosome fitness is calculated by the fitness function. In the simplest case, this function evaluates only the makespan of the obtained schedule. In more robust cases, machines idle times, setup, buffer loads can be evaluated, too. Fitness function is used to compare chromosomes and helps in evolving to optimal or close to optimal solution.

Predefined data

The task contains N jobs and M machines that will need to formulate a schedule. Also, the task contains a packaging operation. Each job or, in this case, an article will have to be assembled from

several parts. All jobs could be described by a collection tree, where each branch is production of parts and the final packaging operation is the root.

The main characteristics and limitations of this task are:

- Each job may have different number of operations.
- Each job has a certain order of execution of operations.
- The duration of the operations may vary for each job.
- Operations are assigned to one main machine and an alternative machine can be assigned, too.
- Also, there may be several identical machines in the working centre, which increases the number of machines that are used to execute the operation.
- The start of work is limited so that more than 2 different jobs cannot be started at a time.
- Each workpiece is operated in pallets. Pallets capacities are predefined.
- A new job operation j can begin with the j-1 operation already in progress executed one palette (the earliest possible start time of the operation j) but cannot finish before one operation takes one palette (the earliest possible end of the operation j).

For each article Bill of Material (BOM) is given together with information about primary work centre, operation time for 1 pcs in it, alternative work centre and time for 1 pcs, setup time for the operation, technological pause and pallet size. From the given weekly quota and maximal possible daily volume lots for each article are made. These lots are divided into parts of the articles and those are considered as separate jobs.

To store information about all operations on each machine four $\square \times \square \times \square$ format data matrixes , , , are used, where is the number of machines, is the number of jobs, is the total number of operations (without packaging).

For each job working times in given working centres are calculated multiplying 1 pcs time by the lot size. These times are stored in the matrix . The cells of the matrix are filled with the computed operational time, i.e. $_{ijk}$ cell shows how much time it takes to complete j^{th} operation of i^{th} job on k^{th} machine.

One pallet working times are calculated and stored in a matrix , which is of the same format as .

Matrix stores the setup times and matrix – technological pause times.

Chromosomes and makespan calculation

Chromosomes are defined by repetitive permutations coding. We create two vectors of $\square\square = \square \cdot \square$ length for OS and MA, respectively. Total operation count can be smaller if not all operations are processed for each job. This encoding consists of two parts, the sequence of operations depicts the job numbers and the sequence of the choice of machines. Let's say jobs can be up to operations. Then the job of $_i$, $\square \in [0, \square]$ and $_k$, $\square \in [0, \square]$, $\square \neq \square$ in the chromosome repeats after times. Meanwhile, the machine selection sequence consists of machine indices $_i$, $\square \in [0, \square]$, where machine indices are arranged according to the rule $\square = \square \cdot \square + \square$, where is the work number, is the operating operation number (different approach should be used if $\square\square \neq \square \cdot \square$).

The population is generated randomly in chromosomes by filling in genetic information. The transaction vector is randomly assigned a job number. Then one of the possible machines is randomly selected in the machine vector according to the changed information. Since there is a

limitation in the problem being solved that more than one job cannot start at a time, therefore, when generating a chromosome, it is considered that the first positions can only occupy the first job operations so that they meet the specified rule.

For evaluation of chromosomes in the test system we used a fitness function □(□) that evaluates makespan of the schedule for corresponding chromosome. For this purpose, additional matrices and are created. They are of rows and columns and its *il*-th cell captures the start and end time of the -th job's *l*-th operation ($0 \leq$ □ $<$ □, $0 \leq$ □ $<$ □). Also, two temporary arrays and are used. Array length is and its *j*-th cell records the end time of the -th machine ($0 \leq$ □ $<$). At the beginning all cells of are 0, i.e. all machines are idle, and specific cell is updated each time operation is performed on chosen machine according to the chromosome. Array length is and its *j*-th cell records the last operation for the *j*-th job.

According to the available chromosome encoding, in the vector we have a job number, and in the vector the machine number. When calculating the makespan each job in vector is considered in succession. This sequence can be started from the beginning or from the end. We analysed the situation starting from the beginning. For each element of we get the current job index □ = □□$_k$ ($1 \leq$ □ $\leq$ □□). Next, we calculate the index of operation for the chosen job □□ = □( ) + 1 (more robust rule might be used, if not all processes are needed for each job) and according to the rule described above, the machine index on which operation should be processed is obtained from vector : □ = □ □$_{e \cdot p + pr}$. By having these values and putting them into the operations time matrix, we get the duration □ = □□$_{s,e,pr}$ of current job operation. Start and ending times are calculated according these rules ($1 \leq$ □ $\leq$ □□):

- Setup time is considered if the last workpiece operated on the chosen machine was not the same (identified by workpiece code), i.e. □□□□□ = □□$_{s,e,pr}$. If it is the same workpiece then □□□□□ = 0.
- Delay time is considered in accordance with the previous operation of the current workpiece, i.e. starting from the second operation: □□□□□ = □□$_{s',e,P_e}$, where ′ is the index of the machine which last operated current workpiece and where technological pause might be needed.
- If it is the first operation of the given job, then

$$_{e,pr} = \square\,\square_s + \square\square\square\square\square,$$

$$_{e,pr} = \square\square_{e,pr} + \square$$

- If it is not the first operation, then we may have two situations:
- If current operation is slower (or the same) than the first one, then it can start as soon as one pallet is produced in the previous operation, thus:

$$_{e,pr} = \begin{cases} _{e,P_e} + \square\square_{s',e,P_e} + \square\square\square\square\square, \text{ if } \square\square_{e,P_e} + \square\square_{s',e,P_e} + \square\square\square\square\square - \square\square\square\square\square > \square\,\square_s \\ _s + \square\square\square\square\square, \text{ if } \square\square_{e,P_e} + \square\square_{s',e,P_e} + \square\square\square\square\square - \square\square\square\square\square \leq \square\,\square_s \end{cases},$$

- If current operation is faster than the first one, then it can start producing the last pallet as soon as it is finished in the previous operation, thus:

$$_{e,pr} = \begin{cases} _{e,P_e} + \square\square_{s,e,pr} + \square\square\square\square\square - \square\,\square\square\square_{e,P_e} + \square\square_{s,e,pr} + \square\square\square\square\square - \square - \square\square\square\square\square > \square\,\square_s \\ _s + \square\square\square\square\square, \text{ if } \square\square_{e,P_e} + \square\square_{s,e,pr} + \square\square\square\square\square - \square - \square\square\square\square\square > \square\,\square_s \end{cases},$$

- In both cases ending time of the operation is calculated the same:

$$_{e,pr} = \square\square_{e,pr} + \square$$

- After each step, arrays and are updated:

$$_{e} = \square\square,$$

$$_{s} = \square\square_{e,pr}.$$

The makespan of the schedule is the $\max_{1 \le i \le m}$ $_{i}$, i.e. the time when the last machine ends its work. The same value was used to evaluate the chromosome fitness in testing system (see Appendix 1): $(\ ) = \max_{1 \le i \le m}$ $_{i}$.

Packaging part is not encoded into the chromosomes. Packaging is done on separate working centres and it can begin only then all necessary parts are produced. Thus, this can be added to the schedule separately from the production of all parts but in the same manner of coding the necessary machines and their working time. Packaging start time depends on pallet operation times of all parts. It can begin when at least one pallet of the last part is produced but the last pallet cannot be packed before the last pallet of the last part is produced.

After filling all matrices, working schedule can be generated in a table form. This table can further be used in evaluating the buffers loads, working centres idle times and counts.

Implemented genetic algorithm

To show the potential of genetic algorithm we have used the test system. Used GA parameters:

Population. This parameter describes how many schedules we have in population. We used from 200 to 1000 schedules.

Iterations. Indicates how many times it will be attempted to improve the initial population by reproducing chromosomes and find the best schedule. We tried up to 500000 iterations.

Group size. This number indicates how many chromosomes will be selected for the reproduction in one iteration. In test system, these chromosomes were randomly selected and only two bests were used to generate two new child chromosomes.

Mutation probability defines the probability that mutation procedure will be applied for new child chromosome. It is recommended to keep mutation probability low (0,01-0,05) but we obtained the best and most stable results with 0,4-0,6 values. In the mutation function there is a random switching between the machine and work operation random swap operation or shift operation.

Crossover probability defines the probability that child chromosome will be reproduced from both parent chromosomes, otherwise new chromosome is equal to the first parent chromosome but can be affected by mutation. It is recommended to use high values (0,85-0,95) of this probability since this reproduction keeps the main trends of the population. We used 0,9-0,99 values and a single-point crossover strategy. The system allows cross-linking both parts of the chromosomes in the first operation, in the other operation it is allowed to cross only the part of machinery. If there is a possibility that two children will experience the same operation during the crossing, a control rule will be triggered to correct the problem.

Exemplary results

Example schedule was created for four articles each consisting of 6 parts. Each article was produced in single lot of 700 pcs. This forms us 24 jobs.

There are four main operations. These operations can be performed in 7 work centres, but the first one has double capacity, i.e. there are two machines in it, thus it is interpreted as to distinct work centres and it leads us to 8 work centres (machines) and two alternatives for each operation.

At first population of 200 chromosomes was generated randomly and the best chromosome's makespan was 5030 min. Repeatedly applying GA for this population, we got better schedules (Fig. 1). Average make span of the population constantly decreased. Although we are interested in the schedules with minimal makespan. The minimum rapidly decreased with the firsts iterations. Later the decrease appears only when GA is able to get out of local minimum area (mainly due to mutations).

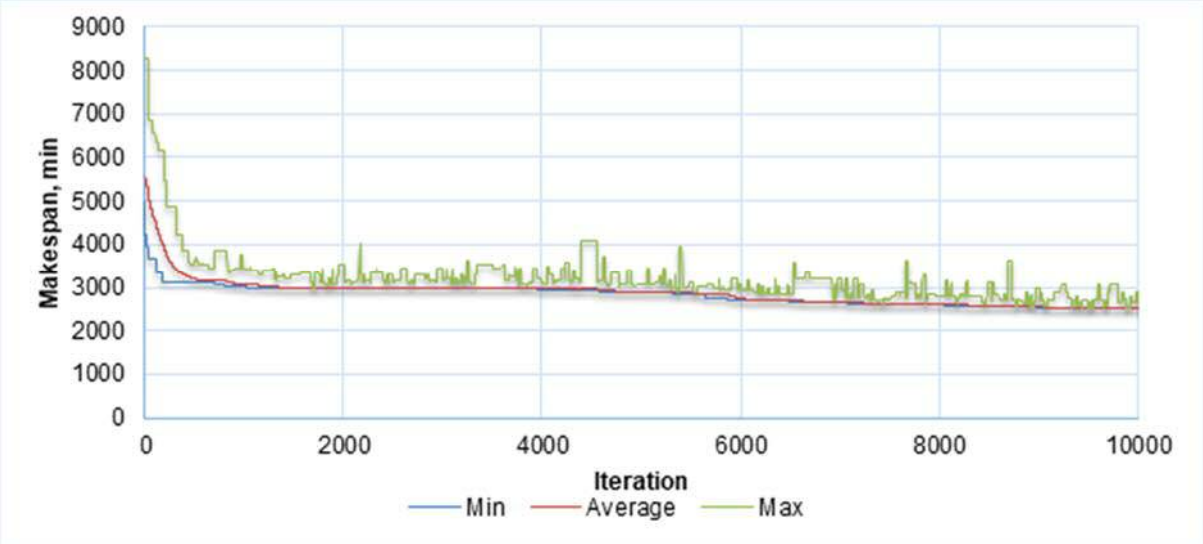


Fig. 1 Makespans of population during evolution

The best schedule was obtained after 10614 iterations with the 2508 min makespan, i.e. more than twice better than the best schedule of the initial population.

Since GA is a random process, every time it is applied different result are obtained. And these results can vary very much depending on generated initial population, evolution based on random crossover and mutations. In the given example, the best schedule was obtained after more than 20 tries. But this is quite fast process, since one try with 200 chromosomes population and 20000 iterations took less than 2 s.

One of the best schedule with the makespan of 2388 min was obtained with some extreme parameter values (40000 iteration, population 400, mutation probability 0,6, crossover probability 0,99).

For the whole week production plan with the same 4 articles testing system obtained the best schedule with the 10739 min makespan (initial chromosomes' makespan was over 22000 minutes). In this case there were 21 lots and 126 jobs in total (without packaging operations). Since there were more jobs 1000 chromosomes population was used to hold more genetic information and to help not to get stuck in local minimum. The best schedule was obtained with 500000 iterations, group of

Workshop of Mathematical Solutions in Business and Industry (ESGI)

Risk Assessment and Insurance Pricing Model Development for a Large Group of Automobiles

Technical report for ESGI 142, 11-15 June 2018

Keywords: claim analysis, budgeting modelling, regression, simulation

2018.06.11 - 2018.06.15
Palanga, Lithuania

Problem formulation

Motor operational leasing (long-term rental) is an alternative to owning vehicle. Usual rental term is 3 to 4 years. One of components of monthly payments is Risk premium. Widely used risk transfer solution is Insurance. First of all, insurance is solvency grant for bank financing fleet acquisition. Secondly, insurer has more capacity and financial instruments to cover high losses. However, traditional insurance is not a way for winning tough competition on the market and so for several years a step into risk self-management was taken by arranging insurance scheme with risk retention level while clients still are covered within a certain level of deductible. It means that company must act as an insurance company – predict amount of losses and collect enough funds from clients to cover it.

Goals and expectations

Running daily business brings in the trap of patterns and prejudice therefore fresh new view is most expected and appreciated. However, there are specific questions we would like to be answered:

- Optimal self-retention level
- Loss peak modelling
- Risk model for daily use calculating premiums for company's clients
- Monthly cash flow budgeting model

Description of a dataset used in the analysis

Dataset covered fleet information and claims over short time period:

- Time horizon: a half of one year
- Frequency: daily
- Three Baltic countries.

"Fleet" file contained a list of cars in the company's portfolio, which has been changing over time. In the "Claims" file, all claims incurred during the analysed period were registered. There was also an additional information about large losses over long-term period (5 years), irregular time periods for all Baltic countries.

Solutions

The challenge submitted by company was analysed applying different mathematical approaches, such as statistical modelling, time series technique, data mining, simulation modelling, and predictive analytics.

Due to the statistical data confidentiality and business rules, only the summary of proposed solutions is given.

First solution was based on the statistical modelling of Claim Size with the aim to estimate the necessary reserve for the following year. The dependency between Claim Size distribution and car price was analysed. The analysis of large claims allowed to estimate the average value, which is necessary per year / per car in order to keep reserve for large claims. The established linear models were proposed as the solution of the analysis performed.

Second solution was mostly oriented on the analysis of certain claim indicators in time. In total, five claim indicators were introduced. The time series were created for different frequency, such as daily, weekly, monthly. It allowed observing different patterns and characteristics in time. The experimental study was performed to explore and understand the historical dynamics of claim indicators, as well as to retrieve features (characteristics, patterns, trend, peaks, …) over time. During the analysis, the comparison among Baltic countries was also performed.

Third solution was based on the application of data mining methods. This technique can help to determine previously unknown and potentially useful information from data and find hidden patterns in it. Claim Size classification using features found in the dataset was carried out using Logistic Regression. The interested patterns found in the data were presented in the full report. After application of Decision Tree algorithm, the significant factors influencing Claim Size were determined. The application of clustering techniques to claims dataset let determine seven clusters, which were fully generalized by their characteristics by forming the profile for the cluster.

Fourth solution was oriented on the prediction analytics of certain claim variables. The linear regression models were established. The idea for the estimation of self-retention level was also proposed.

As five solution to the problem, the simulation approach was used to evaluate monthly cash flow of the company. Simulation is often applied to investigate difficult processes which are difficult to solve analytically. Various approaches of computer simulation, such as Monte Carlo method or discrete event simulation are often used for insurance business optimization. Basically, these methods are based on generation of random variables to mimic the stochastic behaviour or real-world-phenomena. Simulation allows one to generate possible scenarios and evaluate probability of relevant outcomes under given assumption. The possible simulation

algorithm was proposed under a certain abstraction level. The sensitivity analysis was demonstrated for given simulation parameters.

Sixth approach used for this challenge was to apply the predictive analytics to determine the probability of occurrence of an accident and the amount of total insurance payment. Firstly, a new binary variable was introduced. Logistic model was learned, then the neural network was used in order to improve the precision of model. To determine the expected loss during the accident, a linear regression model was constructed. Then, the case study presents the simple idea (algorithm) which is used for budgeting and estimating reserves. The algorithm consists of 4 steps that were explained graphically.

Finally, the conclusions and recommendations were formulated.

During the workshop, the following tools were used in the analysis:

- Excel add-ins;
- Server Data Mining Add-ins;
- Statistical analysis tool R with useful API – Rstudio;
- Matlab.

The limitations of the study: we conclude that all of the results are valid based on the assumptions of short period 6 month period.

Workshop of Mathematical Solutions in Business and Industry (ESGI)

“Developing a model for resources and productivity relations to various factors”

Members of the workgroup:
Arūnas Strazdas (DPD, Operations director)
Mantas Landauskas (KTU, Department of mathematical modelling, mantas.landauskas@gmail.com)
Kristina Poškuvienė (UNI, Department of mathematical modelling, kristina.lukoseviciute@ktu.lt)
Daiva Petkevičiūtė (KTU, Department of applied mathematics, daiva.petkeviciute@ktu.lt)
Daniel Howard (Howard Science Limited, dr.daniel.howard@gmail.com)
Andrej Bugajev (VGTU, Department of Mathematical Modelling, andrej.bugajev@vgtu.lt)

Technical report for ESGI 142, 11-15 June 2018

Abstract

Commercial software dedicated for optimal route finding could determine the most optimal path of how the parcels need to be delivered. The optimization criterion could be time, cost, other constraints. But there is still a question how a particular set of parameters interact one with each other as well as with the overall productivity. In fact this is a kind of inverse problem: not to determine optimal path, but to determine the influence or significance of the factors in a predetermined optimal path. It is not a completely trivial task and depends not only on the parameters themselves, but also on the overview of how we understand the resources which need to be allocated.

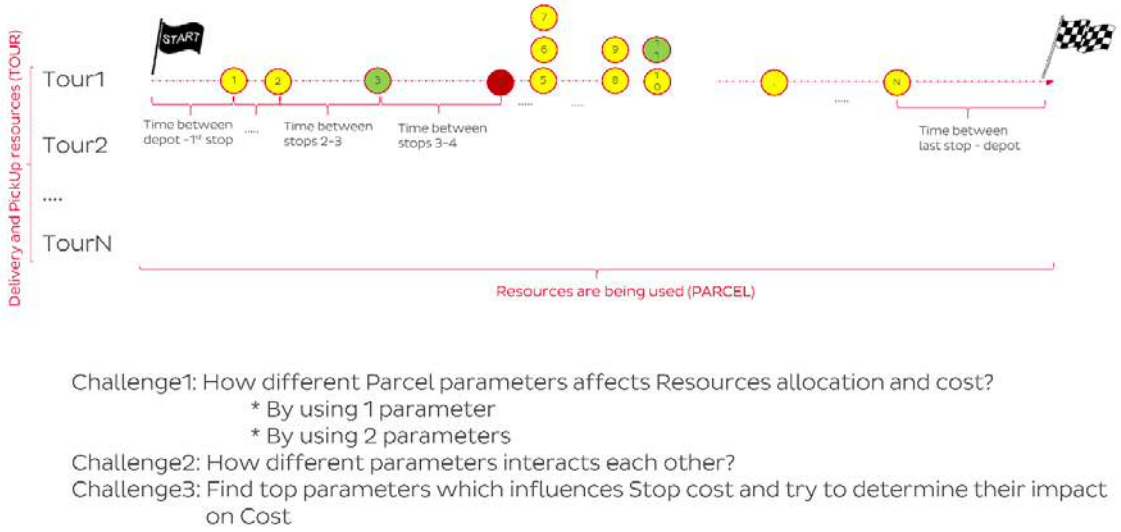
Keywords: *resource allocation, model fitting, significance of factors*

2018.06.11 - 2018.06.15
Palanga, Lithuania

Contents

Introduction

The work described in this report was done at the Workshop of Mathematical Solutions in Business and Industry (ESGI) in Palanga on the 11-15th of June 2018.



Challenge1: How different Parcel parameters affects Resources allocation and cost?
* By using 1 parameter
* By using 2 parameters
Challenge2: How different parameters interacts each other?
Challenge3: Find top parameters which influences Stop cost and try to determine their impact on Cost

Figure 1. Concept of problematics

The goal of our project, as formulated by representatives of DPD, was to:

- explore the dependency of the delivery cost to various parameters of the route.
- Explore possible interconnections between separate factors.
- Predict the cost of the point in the route in respect of different parameters.

Methodology of the solution

Data description

The data was provided in Excel format. In the figure below we can see a list of the parameters used by the team.

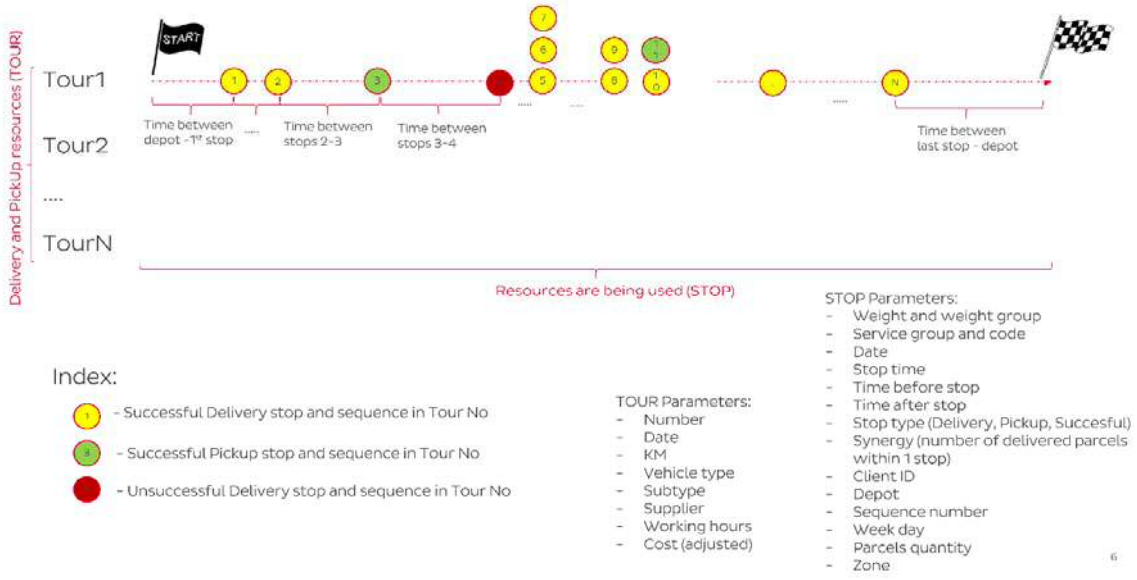


Figure 2. Structure of the data provided

Constructing response/target variables

After an agreement the group decided to create two new target variables which assigns fractional part of the resources to a particular stop.

Some of the relationships between these factors and different tour parameters were investigated. All other possible combinations (which do not exist in the report provided to the company) of possible interconnections could be investigated by using the App created.

Descriptive analysis of the data & dependency check

We started our research from investigation of possible values and distribution of each factor/variable.

To understand what type of model can be applied to model target variables, we have checked:

- Scatter plot for newly created variables versus each parameter.
- Any linear/nonlinear relations and their strength.
- Spatial relationships in respect of which variable.

If case we find anything in the above tasks - usual regression techniques such as [1] and/or PCA will do. But if not: Artificial neural network could be a possibility to model relations.

During the week the work group has tested classical ordinary least squares linear regression (OLS regression), advanced regression techniques, artificial neural networks (ANN) and geographically weighted regression (GWR) [2].

Results

Relationships between factors and target variables

Analysis showed that models obtained by advanced regression techniques are explaining several percent more variance in the dataset. These models train longer but are more suitable due to low coefficient of determination in many cases. They also give more stable results.

The workgroup provided a ranked list of 7 most influential factors for each of the target variables as if complete dataset were used.

It must be emphasized that the coefficients will be different if the model would be fitted on different data (different filter, date range etc.). The analyst should train the model every time he is interested in different dataset.

Software solution

Many of the results obtained (and more tasks in fact) can be produced by using the App created. Of course, one can select different filters and different variables to investigate possible

interconnections inside different subsets of data. The tool of choice is so called Shiny Web App joined together with R (language for statistical computing) and is a part of RStudio project. RStudio is a popular interface for R.

Besides the main value of the App of being a small tool for exploration of the dependencies between various factors there is also a another benefit. For many analysts R is a tool of choice which makes the App created a possible start for much bigger software tool.

Resource allocation between different stops

Background and application challenges

The time resource dedicated to reach each stop is different. It is unclear how much do different stops contribute to the total time of the route. The Shapley value is a solution concept in cooperative game theory [3]. It lets distribute the total gains to the players, assuming that they all collaborate.

Thinking about it, when resources are shared the attribution of resources to each participant can be rather arbitrary. Yet, allocations must be seen to be fair. In 1953, Lloyd Shapley, the 2012 recipient of the Nobel Prize in Economics, devised a cooperative gaming algorithm that achieves a fair allocation of resources.

Applicable to a wide variety of problems of resource allocation, e.g. the Airport Problem, it is known as the Shapley value.

When the logistics department plans the route for each driver then both distances and forecasts are accounted for. Normally, an optimization of the journey enjoys aspects of game theory and a solution to some version of the Travelling Salesman problem.

However, of importance is what actually transpired during the actual journey. Drivers may take longer but quicker roads in response to timely information about accidents or traffic congestion or unscheduled road works, or may experience unavoidable delay to backtrack for access to an uncongested road. They may also swap deliveries responding to a request for an urgent delivery on route. Events may transpire to cause delay or to be ahead of schedule and unforeseen actions taken.

The historical record of the route and in particular time, the duration of each leg of the journey, rather than distance or route, is of paramount importance in the historical analysis. It is not unreasonable for most cases to assume that the final journey was time optimal in some sense.

Reference [2] describes a Scandinavian case study of a cost allocation exercise in a related industry, discusses many formula and useful ideas and concepts. Many are relevant to this problem.

Strategy and implementation

A strategy is to incorporate the Shapley values algorithm to help to temper or quantify the contribution of resources needed to be allocated. When the tour is additive then Shapley numbers

are easily computed as in the Airport Problem. However, if the journeys offer possibilities of shortcuts then the computation of Shapley numbers is time consuming. If a journey involves n stops then $n \times (n-1) \times (n-2) \times \ldots \times 1$ rows of possibilities must be computed.

The computation of Shapley values grows significantly with the number of stops. However, and in practice, a number of numerical techniques can be made help to considerably reduce the algorithmic cost, or one can rely on good approximations.

Conclusions and recommendations

Main findings and conclusions of the work group:

1. A very simple approach to evaluate time resource was agreed on the first day.
2. The target variables do vary as a functions of other variables and is stronger if investigated in filtered data.
3. The ability to choose any factor as a response variable in our App enables to explore various inter-factor relations.
4. Several formulas (models) were constructed for some of the data subsets. The subsets were selected in respect to the results from problems 1 and 2. It must be noted that for best results it is advisable to rerun the training of the method if new data is present or new filter is applied.

Recommendations:

1. Further improve the quality of prediction using an artificial intelligence techniques.
2. Apply the Shapley numbers for the allocation of resources between stops.

References

[1] Zou, Hui; Hastie, Trevor (2005). "Regularization and Variable Selection via the Elastic Net". Journal of the Royal Statistical Society, Series B: 301–320.

[2] Geographically Weighted Regression:
https://cran.r-project.org/web/packages/spgwr/vignettes/GWR.pdf.

[3] Engevall, S. (1996). “Cost allocation in distribution planning. Division of optimization”, Department of mathematics, Linköping University.
http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/28/047/28047371.pdf

Sustainable Income Algorithm: Finpass

Lina Dindiene, Audrius Kabasinskas, Armin Krupp,
Victoria Pereira, Bogdan Toader

Technical report for ESGI 142, 11-15 June 2018

1 Introduction and problem overview

In 2011 the Bank of Lithuania first introduced the Responsible Lending Regulations (with later amendments) seeking to encourage the practice of responsible lending by credit institutions, maintain the market's discipline and ensure transparency of operations, decrease the systemic risk of the credit institutions sector, imbalanced changes in real estate prices, rapid credit growth, concentration of surplus risk, wanting to protect consumers from the excessive burden of financial liabilities and to develop responsible lending practices, thus helping to ensure the stability of the financial system. Responsible Lending Regulations obligate credit institutions to fully assess the ability of credit borrowers to return credit in the long term and pay all related contributions, define the largest permitted loan-to-value ratio as well as the largest debt-service-to-income ratio, defines the highest possible repayment duration and other factors of responsible lending.

The average amounts of instalments of the principal and payments of interest of the borrower calculated dividing the total amount of instalments of the principal and payments of interest by credit maturity for all liabilities may not be in excess of 40% of the person's (household's) income recognized by a credit institution to be sustainable.

Although sustainable income is an integral component of DSTI, Bank of Lithuania does not provide an exact definition of sustainable income and leaves it to lenders to decide what income can be recognized as sustainable. Without sustainable income lender cannot calculate DSTI ratio and, consequently issue financing for his customer.

In general, sustainable income is income that lender expect borrower to receive over a period of a loan.

Bank of Lithuania talks about sustainable income in these three publications:

1. Responsible lending regulations (RLR)
2. Responsible lending regulations for consumer credit (RLR for CC)
3. Guidelines on issuing consumer credit (guidelines)

More about them you can find on BoL website.

Guidelines provide the most information regarding what income should be considered sustainable. Therefore, this document should be your primary source of information. However, don't expect to get a clear cut answer there!

RLR and RLR for CC have a minor, yet very important difference. RLR define sustainable income in case borrower applies for a mortgage. Income from at least 6 previous months should be taken in account when determining sustainable income. RLR for CC define sustainable income in case borrower applies for a consumer loan. Income from at least 4 previous months should be taken in account when determining sustainable income.

Please note that although there is only a minor difference it has significant income on calculations. For example consider a person that only has employment history of 5 months with his first employee. His income would be considered sustainable for consumer loan, but not for mortgage.

1.1 Current practice

According to the information about Composition of the monthly disposable income in cash provided by the Lithuanian Department of Statistics, the income from hired work is about 60 percent of all disposable income (per household member in 2014, 2015 and 2016 years). Separately, this number in town is approximately 65 %, in countryside the income from hired work is about 50 %. Therefore, a significant part of the income is derived from non-hired employment. The bank does not assess such income when it analyzes the borrower eliglibility (some information about borrower eliglibility in USA https://www.usdaloans.net/wp-content/uploads/2012/12/USDA-Home -Loan-Handbook-Chapter-4.pdf). Consequently, it is logical that the bank is too strict for those people who have other stable sources of income.

As there is not set definition, each lender can use its own proprietary methodology to calculate sustainable income. This is a time consuming and labor intensive task, that should be standardized and automated as much as possible.

As there is no uniform methodology on calculating sustainable income, lenders have a considerable amount of leeway with regards to what type of income should be treated as sustainable income. However, it is also up to lender's to prove that their calculations conform to regulatory requirements if they are inspected by the Bank of Lithuania.

The only automated process that can be used to check for sustainable income is social security (i.e. SoDra) registry, which holds information on income that is used for social security taxation purposes. However, this registry does not keep income records for those individuals who are self employed, or receive income from other sources, such as rent or dividends (i.e. income not used is social security taxation calculation).

Some lenders also use raw or categorized client's bank statement data in order to determine what types of income their client earns in addition to those that are used for social security calculations. Yet, this source provides only additional information, but no actual solution

A number or lenders use and pay for both sources of income information: social security registry and bank statement data. However, in most cases social security income information is already present in bank statement data. Thus lenders are paying twice for essentially the same information and receive no complete answer.

Income that should be included:

1. Pension
2. Salary

Regular/recurring bills should be included:

1. Bills
2. Dividends received
3. Interest received

Can be included if they are regular/sustainable. Regularity/sustainability has to be determined:

1. Alimony and child support
2. Online payments
3. Other incoming payments from employer
4. Per diem
5. Rental income
6. Royalty fee
7. Scholarships
8. Securities
9. Social security benefits
10. Social security benefits
11. Tax return

Generally such income should not be treated as sustainable :

1. Advance payment
2. Cash deposit
3. Currency exchange
4. Deposits

5. Gambling
6. Insurance indemnity
7. Loans (student loans, consumer loans, leasing, etc.)
8. Online money transfer
9. Other incoming payments
10. Personal transfers (between accounts, from relatives, other)
11. Returned payments
12. Savings
13. State aid payments

1.2 Task and expectation

Task: to create a sustainable income calculation algorithm that complies with the Bank of Lithuania RLRs & Guidelines and uses information only from a categorized bank statement (i.e. categorized transactions).

Algorithm should be forward looking:

1. It should not be limited to calculating average historic sustainable income (although this is a good start).
2. It should be able to forecast sustainable income for the near future, i.e. duration of a consumer loan. (for obvious reasons income forecast for mortgages is not required).

Algorithm should have the following calculation options:

1. Sustainable income calculation for mortgages (minimum of 6 month income information used)
2. Sustainable income calculation for consumer credit (minimum of 4 month income information used
3. Sustainable income calculations where some types of incomes are excluded
4. Sustainable income calculations based on all types of income

Finding a way to determine which income can be treated as sustainable, firstly we made the exploratory data analysis. We looked through the data and analyzed each type of income separately. We searched for the clusters of persons. We constructed the shape of the formula which could evaluate the stable income. Further, we tried to apply some predictive models: weighted mean based model, confidence interval based model, autoregressive model, tree-based prediction model. We submitted some cases of prediction and real data.

It is understandable that no solution will be perfect, however, a good solution should catch situations, where lender should be alerted that expert judgment on client's income is needed.

2 Exploratory Data Analysis

2.1 General overview: How are the incomes distributed

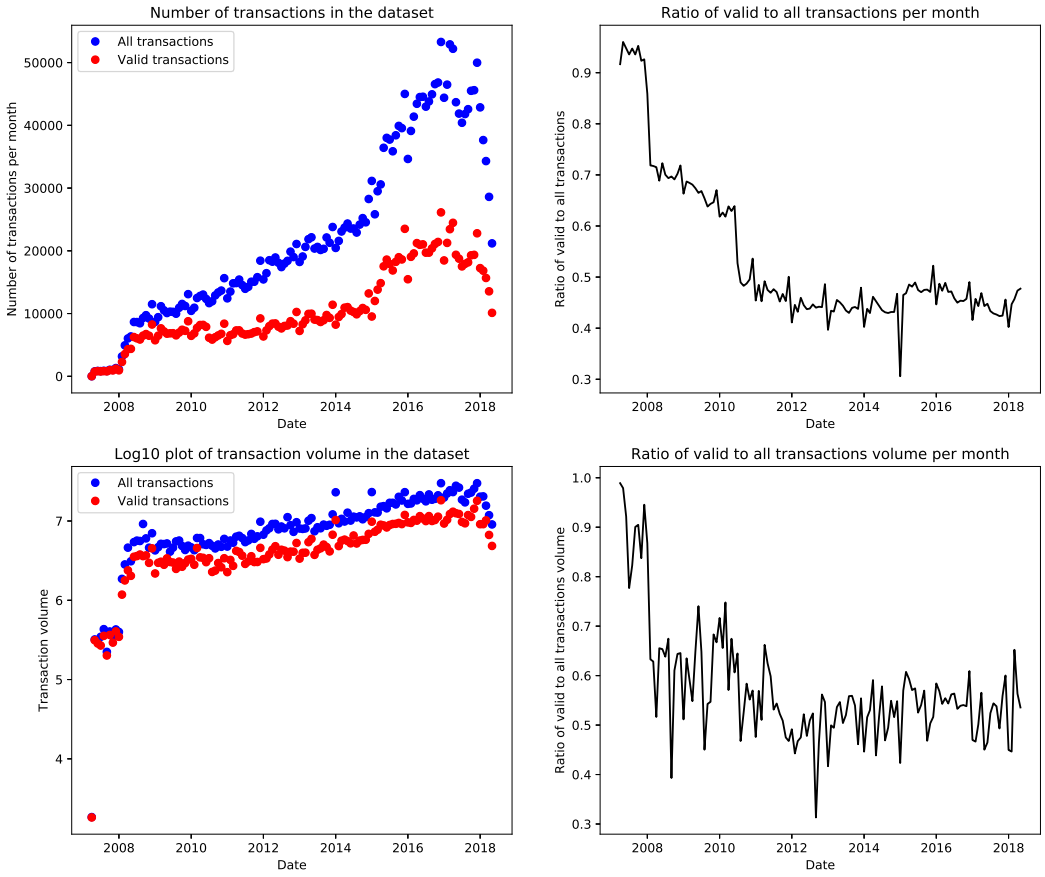


Figure 1: Distribution of number of transactions and transaction volume

The distribution of the median monthly income does not really seem to change over time, the standard distribution stays the same. We can fit a t-distribution to both the $\log_{10}$ of the mean income and standard deviation.

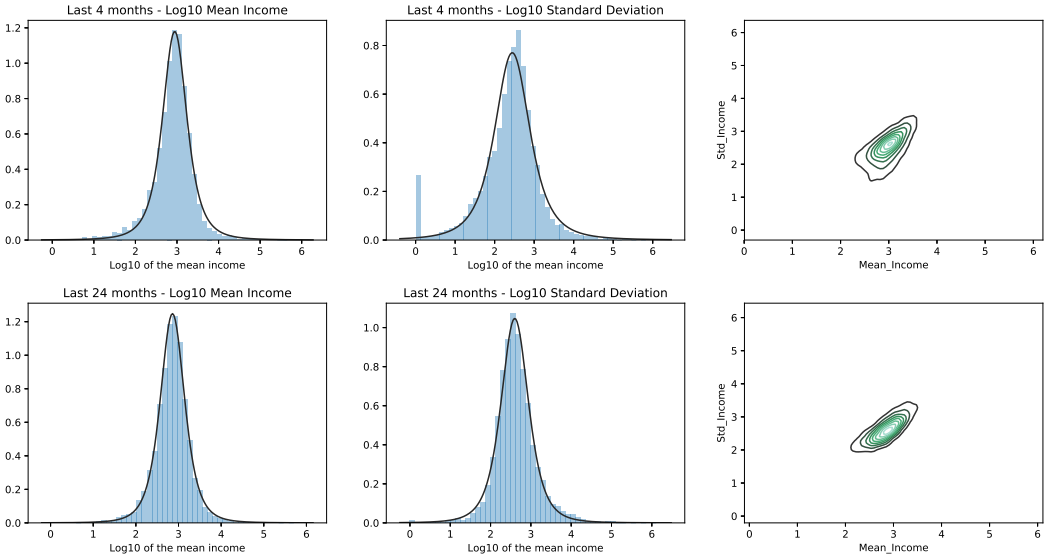


Figure 2: Mean, standard distribution, and kde-plot of their combination

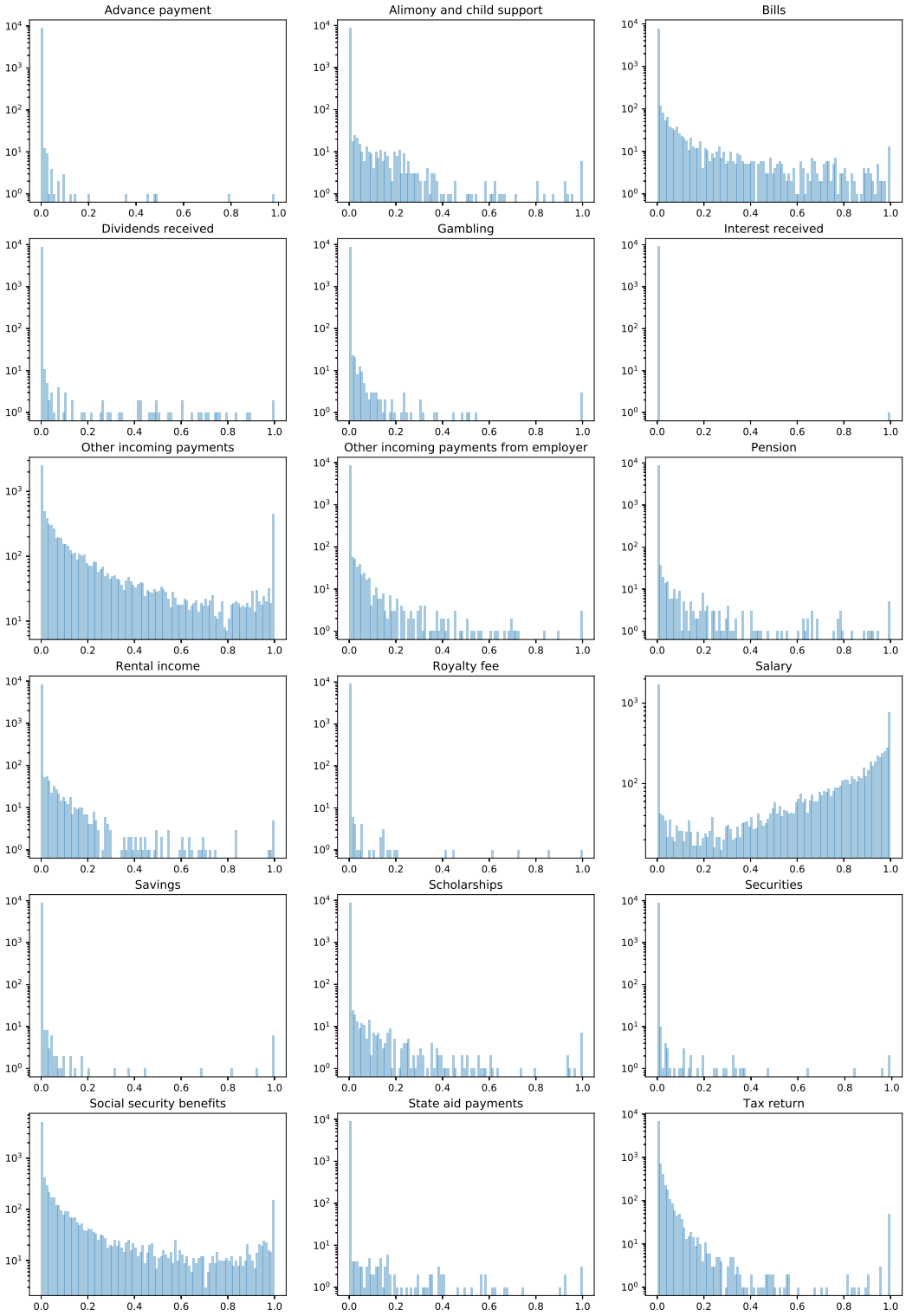


Figure 3: Log frequency of

2.2 Individual time series

- Example for being paid twice a month, large payments before vacation
-

2.3 Clustering

Clustering is used to partition the data into groups, where data points in the groups are closer to each other than they are to other groups. In this data set, we can consider clustering in many ways:

1. Cluster on the client level over all their transactions across different categories.
2. Cluster on the client level over their transaction within a single category.
3. Cluster over all the clients based on transactions within a single category.

We ran k-means clustering for these three clustering objectives, but found little cluster structure. For example, see figure 4. This shows the similarity matrix ordered by clusters for client 422. If there was successful clustering, clear block matrices would be seen. The results suggest that clustering the data in these ways is not helpful to reduce the size of the problem.

Workshop of Mathematical Solutions in Business and Industry (ESGI)

"Developing a model for PIT (point in time) migration matrices forecasting using general macroeconomic indicators (GDP growth, etc.) as explanatory variables"

Technical report for ESGI 142, 11-15 June 2018

Abstract

The aggregate market-wide rating transition trends, such as upgrade/downgrade ratios, as well as the realized default rates (we consider default as a particular form of rating transition), are widely followed by portfolio managers as a gauge of the credit cycle. Robust estimates, and, if possible, forecasts of rating transition risks are important ingredients of their macro investment strategies. The aim of this report is to investigate possible modelling approaches for PIT (point in time) migration matrices with the aim to forecast the default probabilities for some period in time taking into consideration general macroeconomic indicators (GDP growth, etc.) as explanatory variables.

Table of Contents

Introduction

During the recent years, the financial markets have been unusually volatile. In the turmoil of the last financial crisis that began in 2008, many companies defaulted on their debt and thereby caused huge credit losses to their counterparties. Furthermore, among the companies that did not default there were many credit rating downgrades. A lower credit rating implies that the probability of default has increased [1].

The aggregate market-wide rating transition trends, such as upgrade/downgrade ratios, as well as the realized default rates (we consider default as a particular form of rating transition), are widely followed by portfolio managers as a gauge of the credit cycle. Many long-term credit investors, including those managing bank loan portfolios, insurance assets and CDOs, often pose a question: "what is the current estimate of the N-year rating transition probability?"

(where the typical values of N can be 1, 5 or 10 years). Robust estimates, and, if possible, forecasts of rating transition risks are important ingredients of their macro investment strategies [2].
The aim of this report is to investigate possible modelling approaches for PIT (point in time) migration matrices with the aim to forecast the default probabilities for some period in time taking into consideration general macroeconomic indicators (GDP growth, etc.) as explanatory variables.
Several different datasets were presented for the group. The methods presented in this report span time series modelling, continuous and discrete time Markov chain analysis, multivariate regression, parameter space reduction techniques.

Methodology and analysis

Coupled ARIMAX model technique for the forecasting of risk class transition matrix time series

A sequence of matrices $^{(t)}; \square = 1, \dots, \square$ is given:

$$^{(t)} = \begin{bmatrix} ^{(t)}_{11} & \cdots & \square\,^{(t)}_{1n} \\ \vdots & \ddots & \vdots \\ ^{(t)}_{n1} & \cdots & \square\,^{(t)}_{nn} \end{bmatrix},$$

Elements $^{(t)}_{jk}$ denote the number of clients that transitioned from risk class to risk class at time . Denote $^{(t)}_{j} = \sum_{k=1}^{n} \; ^{(t)}_{jk}$ and define:

$$^{(t)}_{jk} = \frac{1}{^{(t)}_{j}} \; ^{(t)}_{jk}; \quad \square, \square = 1, \dots, \square.$$

The above equation results in a sequence of matrices $^{(t)}; \square = 1, \dots, \square$ that satisfies the properties:

$$(\;) \; 0 \le \square^{(t)}_{jk} \le 1; \;\; \square, \square = 1, \dots, \square; \;\; \square = 1, \dots, \square;$$

$$(\;) \sum_{k=1}^{n} \square^{(t)}_{jk} = 1; \;\; \square = 1, \dots, \square; \;\; \square = 1, \dots, \square.$$

Consider one row of matrix (1):

$$^{(t)}_{j1}, \square^{(t)}_{j2}, \dots, \square^{(t)}_{jn}; \;\; \square \in \{1, \dots, \square\}; \;\; \square = 1, \dots, \square. \qquad (2)$$

We propose using the same order model ARIMAX($\square, \square, \square$) for all elements of matrix (1) – and are kept constant, but the parameter is selected using the augmented Dickey-Fuller test for time series stationarity.
The analytical expressions of the models are as follows:

$$\nabla^{d} \; ^{(t)}_{jk} = \square_0 + \square_1 \nabla^{d} \; ^{(t-1)}_{jk} + \cdots + \square_p \nabla^{d} \; ^{(t-p)}_{jk} + \square_1 \; _{t-1} + \cdots + \square_q \; _{t-q} \qquad (3)$$
$$+ \square_1 \; ^{(1)}_{t} + \square_2 \; ^{(2)}_{t} + \square_3 \; ^{(3)}_{t} + \square_4 \; ^{(4)}_{t} + \square_t,$$

where $_t$ is white noise; $\nabla \square^{(t)}_{jk} := \square^{(t)}_{jk} - \square^{(t-1)}_{jk}$, $\nabla^{d} \; ^{(t)}_{jk} := \nabla^{d-1} \; ^{(t)}_{jk} - \nabla^{d-1} \; ^{(t-1)}_{jk}$.
Coefficients $_0, \dots, \square_q; \; \square_1, \dots, \square_p; \; \square_1, \dots, \square_4$ depend on and , but the orders and are independent of these indices.
The model (3) is used to obtain a series of fitted values for each matrix row:

$$\square^{(t)}_{j1}, \square^{(t)}_{j2}, \dots, \square^{(t)}_{jn}; \;\; \square \in \{1, \dots, \square\}; \;\; \square = 1, \dots, \square.$$

Obviously, the fit cannot satisfy (□) and (□) in the general case, which means that the following refinements are needed:

$$(\ )\ \widehat{+}_{jk}^{(t)} = \square_{jk}^{(t)} - \min\left(0, \min_{k=1,\dots,n} \square_{jk}^{(t)}\right)$$

$$(\ )\ \square_{jk}^{(t)} = \frac{1}{\sum_{k=1}^{n} \square_{jk}^{(t)}}\ \widehat{+}_{jk}^{(t)}.$$

The algorithm described above is applied for forecasting. The sample is split into two parts: the model fitting sample $^{(1)}, \square^{(2)}, \dots, \square^{(T_{train})}$ and validation data $^{(T_{train}+1)}, \dots, \square^{(N)}$. The former is used to fit the coefficients of models (3) and the latter is used to evaluate the quality of the forecasts given by the model. Note that the exogenous variables are considered to be known for the time period of validation.

To evaluate the general quality of the forecast, the root mean squared error between the forecasts $\tilde{}^{(T_{train}+1)}, \dots, \tilde{\square}^{(N)}$ and validation data is computed.

The methods described above were applied to different datasets. Note that the results can be compared across different sequences of matrices, because they have been normed to have values ranging from 0 to 1. Also, the computation of the errors takes the size of the matrix into account, allowing the comparison between different types of product in different countries. This is an important factor in deciding modelling (and forecasting) accuracy, as the volume of

A plot of the forecast and real values of transition probabilities for the validation period is given in Figure 1 for C3 p4. It can be observed that the forecasts are generally capable of capturing the general dynamics of the transition probabilities. It seems that the order □, □ of the best-fit models does not vary greatly from product-to-product with the exception of C1 p4 in which the model did not perform well.

For a single product, the probabilities of going from any risk class to any other class are modeled by independent time series models that are of the same complexity and also incorporate macro variables (GDP, unemployment, house pricing index, inflation). The approach is tested by dividing the data set into two parts: the last two years (8 quarters) are set aside to check the accuracy of the models, which are constructed using the remaining data. The quality of the forecasts varies slightly from product to product, but generally the models seem to predict the probabilities reasonably accurately for a two-year horizon – the average errors have an approximate magnitude of 0.01 in all analyzed datasets.

The idea of this analysis could be successfully applied even if ARIMAX modelling would be discarded. Any forecasting method incorporating lagged values of the transition probability time series and macro variables could potentially be used in the same role that ARIMAX played in this analysis. With larger datasets, application of more sophisticated techniques such as neural networks have potential for much more accurate results.

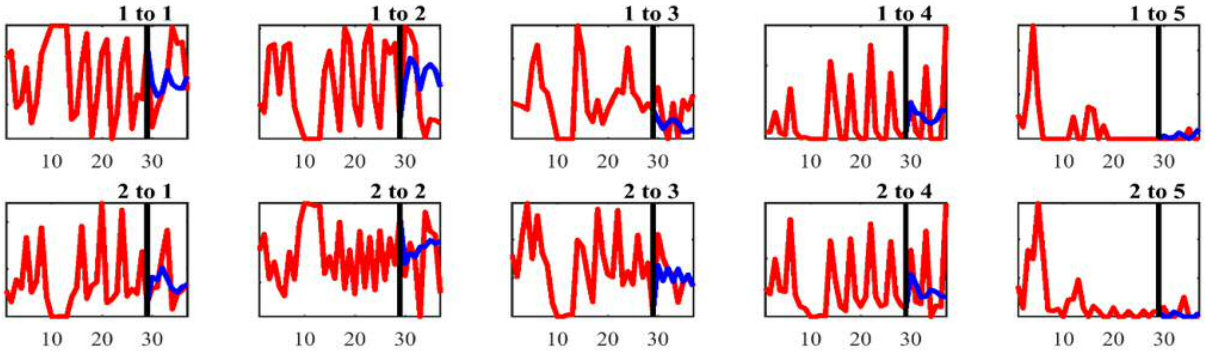


Figure 1. The observed transition probabilities (red line) and forecasts of these probabilities for the validation period (blue line). The title of each subplot indicated what probability is depicted.

Continuous time and discrete time Markov chain parameter estimation

Consider a general continuous time Markov chain with transition rates being non-zero for every possible connection of different chain states. Each transition matrix is observed over equal periods of length Δ. The observed transition matrix (for n states) has the following form:

$$\left(\begin{array}{ccc} _{1,1} & \cdots & _{1,n} \\ \cdots & \cdots & \cdots \\ _{n,1} & \cdots & _{n,n} \end{array} \right).$$

Each i[th] row of this matrix is assumed to follow multinomial distribution:

$$_{i,1}, \ldots, _{i,n} \sim \left(_{i,1}( ), \ldots, _{i,n}( ) \middle| _{i} \right),$$

where $_{i} = \sum_{j=1}^{n} _{i,j}$ is the total number of observations, is the transition matrix for the Markov chain

Transition probabilities $_{i,j}$ are related by the following expression:

$$= ^{\Delta G}.$$

The log-likelihood function of this model can be expressed as follows:

$$\mathrm{L} = \sum_{i=1}^{n} \sum_{j=1}^{n} _{i,j} \cdot \log\left(_{i,j}( ) \right)$$

with transition matrix containing unknown model parameters.

Estimation of the parameters was carried out by using Markov Chain Monte Carlo stochastic optimization method [5].

Above defined model assumes that every transition probability related to every other probability. This is determined by the equation $= ^{\Delta G}$. However, assume that each transition matrix row is generated by independent stochastic processes, where row elements follow multinomial distribution as above. However, no dependence on the transition matrix G is assumed. And log-likelihood function is expressed as follows:

$$\mathrm{L} = \sum_{i=1}^{n} \sum_{j=1}^{n} _{i,j} \cdot \log\left(_{i,j} \right),$$

so that unknown parameters are simply a matrix of probabilities $_{i,j}$. This row independence clearly breaks the continuous time Markov chain definition and we have discrete time Markov chain.

Estimation of parameters was carried out by using the same MCMC stochastic optimization as above. However, before that, parameterization of probability matrix was performed. Since each entry must be within interval [0; 1] and the sum of each row must be exactly 1, it is an optimization problem with constraints. However, it is well known that MCMC works better if parameters can be any real number. In order to have unconstrained problem we considered the following parameterization:

$$\left[_{i,1}, \ldots, _{i,i}, \ldots, _{i,n} \right] = \left[\frac{x_{i,1}}{1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{i,j}}, \ldots, \frac{1}{1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{i,j}}, \ldots, \frac{x_{i,n}}{1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{i,j}} \right]$$

The dataset that were considered contains many entries without any observations over an entire period. This hinders estimation of probabilities at those positions. Without loss of generality, assume we have observed such transition matrix row:

$$_{i} = [_{i,1}, _{i,2}, \ldots, _{i,n-2}, 0, 0]$$

i.e. no transitions from i[th] state to the last two was ever observed. Since such property of transition matrix makes it impossible to properly estimate transition probabilities at those positions it was assumed that no transition can occur to those states, i.e. there are no connection

with those states from i^{th} state. We will refer to this construction as **cutting-links**. Consequently, the number of unknown parameters is reduced.

The first data set that was considered was C transition matrices with low number of observations. The testing period was always 2 years, and entire dataset is observed quarterly. As for the training data, it was used a subset left when 2 last years was excluded. Comparison of the method was done with naïve estimator: i.e. a prediction is assumed to be equal to the last observed frequency matrix. From the training dataset visualization (Figure 2) it is clear, that many entries had no observed data.

The estimation of entire intensity matrix G proven to be extremely hard and no stable estimates were obtained. Regressing on time or macroeconomic covariates would increase the number of unknown parameters and in this case would make the problem even harder. Relaxing the continuity assumption and dropping the dependence on intensity matrix, enabled estimation of transition probability. However, it was only with a strong assumption of cutting-links.

Testing RMSEs was 0.085 and 0.076 for multinomial model and for Naïve estimator respectively.

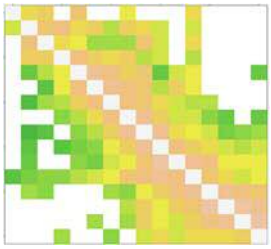


Figure 2 C average transition frequency (logarithms) visualization. White positions except the diagonal elements have no observations during an entire training dataset period.

This data set was eventually dropped from further analysis because from times series of total number of observations at each time step it was observed a significant "switch" of the entire dynamics. Next C1, C2, C3 datasets were considered. Only multinomial distribution was fit to the data, as it contained even more empty positions in the transition matrix and the estimate of full transition matrix G would be unstable.

RMSE errors reported below:

RMSE	C1	C2	C3
Multinomial	**0.012**	**0.036**	0.087
Naive	0.064	0.073	0.083

As one can observe, multinomial distribution fitting has some superiority over the naive estimator. Multinomial fit and Markov chain rate intensity matrix estimation was carried out for C3 (3 products), C2 (4 products) and C1 (4 products) data sets with 5 possible states. In this case for every transition some data was observed. RMSE errors are presented in table below:

Table 2. Rate intensity matrix estimation

Model	Product 1	Product 2	Product 3	Product 4	Country
Multinomial	0.034	0.034	-	0.106	C3
Full Markov	0.032	0.040	-	0.108	
Naive	**0.013**	**0.0250**	-	**0.130**	
Multinomial	0.024	0.019	0.017	0.028	C2
Full Markov	0.024	0.018	0.019	0.025	
Naive	**0.011**	**0.012**	**0.017**	**0.027**	
Multinomial	0.018	0.102	0.046	0.150	C1
Full Markov	0.016	0.081	0.039	0.133	
Naive	**0.009**	**0.073**	**0.03**	**0.121**	

Below is the comparison of predicted transition probabilities and observed (presented logarithms instead of original values for better visualization efficiency).

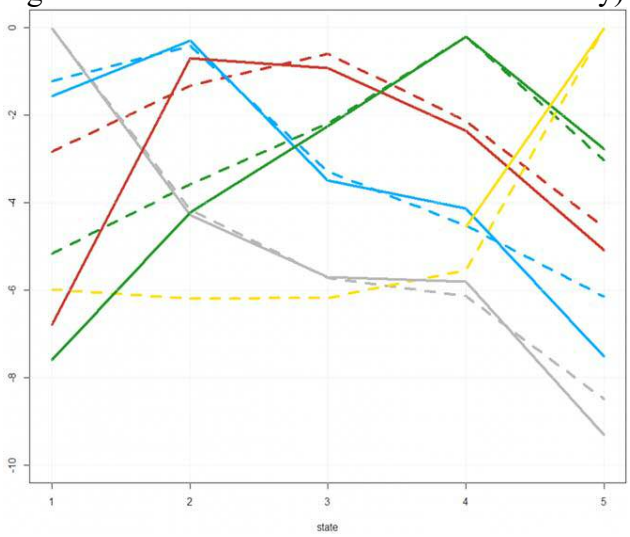


Figure 5 Representation of predicted (dashed lines) and observed transition probability matrix (C3 case). Gray, blue, red, green and yellow represents 5 states from 1 to default state 5.

For the C1, C2, C3 data set, regression on time covariate was considered on some of the transition matrix entries. Though, no improvement was observed. At this point it is difficult to assess the usefulness of estimating an entire Markov transition rate matrix with regression "hooked" on some of its parameters. Further analysis is required. On the other hand, multinomial distribution fit for the transition matrix rows showed some promise and should be further investigated for data set with small observation samples for some states. A fully Bayesian analysis might be considered, i.e. putting prior distributions, elicited by field experts, on those transitions where no data were observed.

Embedding of transition matrices into lower dimension spaces

Assume there are *n* risk classes. The transitions between these classes are defined by the so called migration matrix of size *n* by *n.* The last column is associated with the default event, which occur when a loan recipient fails to make payments for 90 consecutive days. These matrices are used to model the lifetime of a certain loan which is significant for allocating reserves in case of default. It is expected that the migration matrix is influenced by macro-economic dynamics, like GDP, HPI, unemployment, inflation, and possibly other factors.
In this approach the first step is to represent the whole matrix by a smaller number of parameters. It can be observed that the biggest row elements are the diagonal ones (if not considering extreme cases, which might be the result of data collection policy changes etc.). A visual inspection of the transition matrices it looks lead to an idea, that probabilities of migration from a certain class to other classes follows a type of exponential decay. Therefore, it was decided to specify migration probabilities by exponential functions:

$$-\lambda_k x_i$$

where $_i$ is the number of positions between kth and ith states plus one. Since transition probabilities to classes towards default might differ from transition probabilities towards more higher rating classes, these probabilities was approximated by two different exponential functions for each row (except for the first one, and the last one), which are specified by the parameters $\overset{(a)}{k}$, $\overset{(b)}{k}$ (for kth row). Zero-value matrix entries were ignored, as they cannot be approximated by exponential function.

Given a list of *lambda* parameters, the migration matrix can be reconstructed in the following way. Initially we set diagonal elements of the matrix to ones. Then we use the exponential functions from diagonal elements to compute other elements of the matrix. Finally, the matrix is normalized, by computing the sums of rows and dividing each row element by a respective sum.

Next, the task is to find out how these lambdas depend on macro-economic variables. To do that we have computed correlation coefficient between each lambda series and each macro-economic variables. It is reasonable to expect a lag in the relation of macroeconomic variables and transition matrices, to investigate that we have plotted the dependency of correlation versus lag. For predicting lambda values, we used a neural net trained by a back-propagation algorithm. Two data sets were investigated. The simulated one with 6 risk classes and the one of C1, i.e. p4 and part of p2 data set listed at the introduction.

First of all, we look at correlation of each lambda with GDP. A strong correlation was observed. More specifically, we have a positive correlation for lambdas which describe exponential approximations for probabilities of migrating to classes towards default. It has to be noted that the higher the lambda is the smaller probabilities are for migrating to more distance classes. So, the results are reasonable, because when GDP rises the probabilities of migrating to more risky classes gets smaller and probabilities of migrating to safer classes increase. HPI (house price index) have a very similar impact on lambdas compared to one of GDP. Data suggests that unemployment before 4-8 quarters positively impacts the lambdas (b), i.e. higher unemployment makes smaller probabilities of migrating towards more risky classes, which is not that reasonable.

Since there is some kind of correlation we can move on with the lambda prediction. We construct a neural net with one hidden layer of 100 neurons. The input layer is given all macro-economic variables of up to 4-th lag. The output layer gives us the predicted lambdas. The learning rate is 0.0025, and momentum is 0.85. We have trained the neural net by a back-propagation algorithm for 100000 epochs. And tested the ANN (artificial neural net) by using it for predicting migration matrices for four quarters, which have not been included into the training set.

Example of lambdas prediction is presented in Figure 3. Lambdas $\lambda_1^{(b)}, \lambda_2^{(a)}, \lambda_2^{(b)}, \ldots, \lambda_5^{(a)}, \lambda_5^{(b)}, \lambda_6^{(a)}$ are denoted by ‘lam-0’, ‘lam-1’, …,’lam-9’.

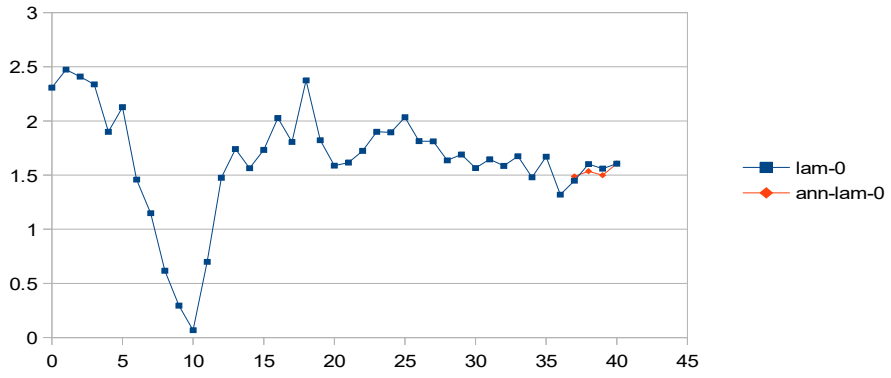


Figure 3 Prediction of lam-0 for the simulated data set.

In some cases the neural net predicts lambdas quite well. However, we need to use the actual probability matrices.

Parametric approach

This section defines problem analysis based on parametric approach.

Parametric model of transition matrix

The original dataset consist of time series of clients' transition matrices between risk categories. These matrices were normalized to be a stochastic matrices, i.e. its elements represents transition probabilities between risk classes and each matrix row sums to 1. A transition matrix is visualized in Figure 4.

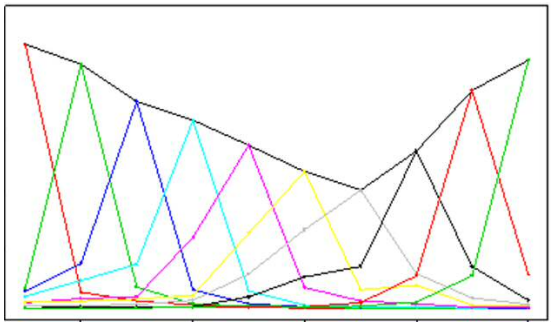


Figure 4 Transition matrix of C dataset. Color curves represent matrix lines. The top black line represents main diagonal of transition matrix.

We can see that diagonal elements of matrix are the largest elements and values are decreasing fast with increase or decrease of risk class. Let us try to find the shape of transition probability decrease. Transition probabilities decreases exponentially. We define parametric model based on this property of transition matrices :

$$\begin{aligned} p_{ij} &= p_{ii} / q_{Li}^{i-j}, \qquad j < i \\ p_{ij} &= p_{ii} / q_{Ri}^{j-i}, \qquad j > i \\ p_{ii} &= 1 - \sum_{j \neq i} p_{ij} \end{aligned} \tag{6}$$

where parameters $q_{Li} > 1$ and $q_{Ri} > 1$ defines decrease rate of transition probability left slope (migration to class of lower risk) and right slope (migration to class of higher risk). Elements p_{ii} and p_{ij} are defined recursively, but effective calculation of transition probabilities based on model parameters can be done by assuming p_{ii} is equal to 1 and transition probabilities are normalized after calculation of the remaining elements of the matrix.

We defined here a parametric model containing a separate slope parameters q_{Li} and q_{Ri} for each row of transition matrix. Let us call this model a *multi* parameter model. Using assumption that slopes decrease rate for various risk classes are the same, we define a *single* parameter model

$$q_L = q_{Li}, \; q_R = q_{Ri}, \; \forall i \tag{7}$$

Slope parameters of multi parameter model were estimated using linear regression model for log-probabilities. Estimates for single parameter model were obtained from corresponding multi parameter model by taking a median of left and right slope parameters. Median (instead of mean) was taken because left slope parameters of second lowest risk class and right slope parameters for last before the highest risk class (both calculated based on a single transition probability) tend to be outliers. Additional model fine-tuning can be performed to select a better single slope parameter selection method like mean of parameters with outliers removed, weighted mean, etc.

Multi parameter model gives quite precise approximation of transition matrix. The largest drawback seen so far is that this approximation does not take into account the probability increase for the last risk class (default class). Transition probabilities has deceasing slope, but the last probability (migration to default class) has much higher values than predicted by estimated decreasing slope. Model extension taking into account this increase should

improve precision. Additional parameter may be implemented to estimate amount of this increase

Figure 5 presents time series of macroeconomic variables and model parameters $_L$ and $_R$. A clear relation between economic parameters and model parameters can be seen on the plot. Increase of GDP and HPI is followed by increase of parameter $_R$. We remind that $_R$ is decrease rate of right slope of transition probabilities.

Variable selection and signs of regression coefficients looks very chaotic. Ex., unemployment has positive coefficients for lags 0 and 2, but negative coefficients for lags 1 and 3 in the last presented model. This can be explained by seasonal behavior of unemployment, see Figure 11. Macroeconomic time series should be seasonally adjusted before it is used for building linear models. Lagged values of parameters $_L$ and $_R$ may also be included, making the model to be autoregression model with independent variables. Non-linear dependencies can be checked. Other time series models can be applied. Residuals check for models should be performed. An alternative approach to forecast values of parameters of multi parameter model can also be used. It is a good chance for well adjusted models to explain 60 to 90% of total variance for the analysed dataset.

So, if GDP and HPI increases, the probabilities of migration to higher risk class become smaller. A similar relation can be seen between inflation and $_L$ – decrease of inflation is followed by decrease of $_R$, i.e. probability of migration to lower risk class increases.

Linear regression models of parameters on macroeconomic variables were analysed. C dataset was used. This model explains 52.53% of variance of $_L$.

Regression of . Stepwise procedure selected HPI as the only statistically significant variable for regression of $_R$This model explains 82.98% of variance of $_R$.

The initial idea was to use linear regression model described in previous subsection for forecasting of transition matrix model parameters, and then to estimate the values of this matrix. The theoretical precision of this approach was tested using estimated (not forecasted based on previous values) matrix parameters and calculation the error of transition matrix approximation using these parameters. This test mimics the case if values of parametric model can be forecasted with zero error. Root mean square error (RMSE) measure for all matrix elements was used. Results were compared to Naïve forecast method which uses the last observed value as future forecast. The obtained errors are presented in Figure 6.

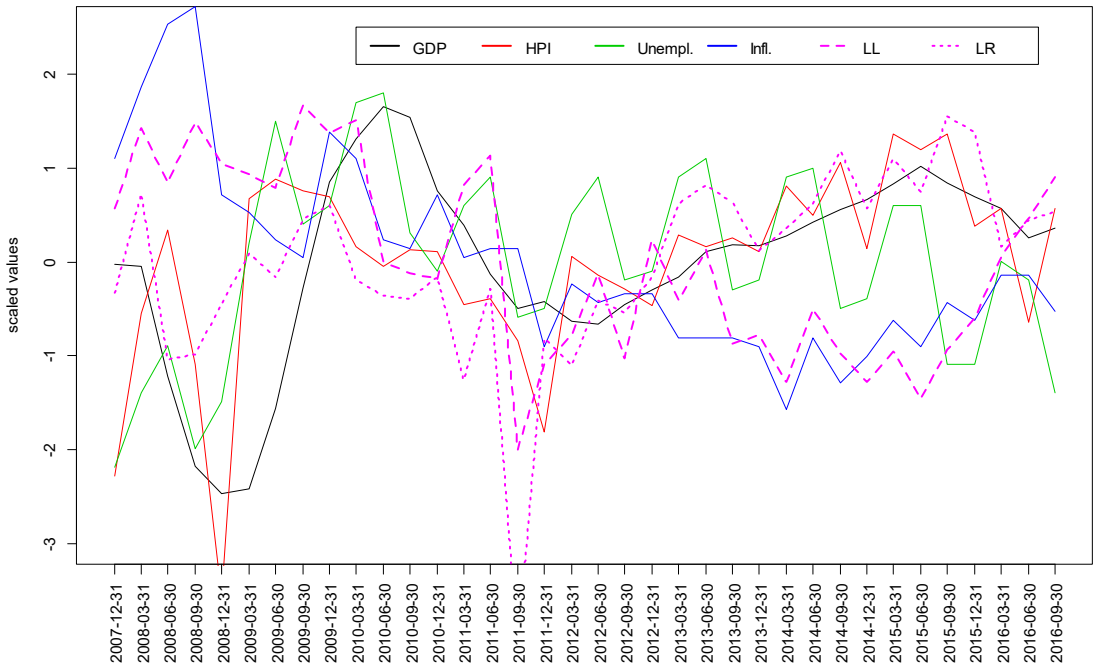


Figure 5 Time series of macroeconomic variables (GDP, HPI, unemployment, inflation) and model parameters ($_L$ and $_R$). All variables are scaled to be presented on the same plot.

Another suggested approach: we have to estimate difference between two transition matrices and adjust this difference using parametric model. Research revealed that estimation of difference creates some problems. Matrix adjustment may contain values larger than values of the past transition matrix, thus, making estimate to contain negative values. Additional fix and reweight of matrix row helps, but the properties of the final estimate diverges from expected. We propose to use ratio of matrices instead of difference for adjustment. Thus, the main idea of semi-parametric model is defined by equation

$$\widehat{\ }_{t+1} = \square_t / \square(\square_t) \cdot \square(\hat{\square}_{t+1}) \quad (8)$$

where $\widehat{\ }_{t+1}$ is the future estimate of transition matrix, $_t$ – observed value of transition matrix, $\square(\square_t)$ and $\square(\hat{\square}_{t+1})$ denote parametric model of observed matrix $_t$ defined by estimated parameter $_t$ and the forecast of parametric model based on forecasted parameter value $\hat{\ }_{t+1}$. Matrix multiplication and division is made element-wise. Both single and multi parameters can be used.

The semi-parametric model defined by equation 8 was analysed. It has a precision similar to Naïve estimator. We implemented moving average smoothing with exponentially decreasing weights. The remaining value of weights (weight values sums to 1) was assigned to matrix of ones.

We propose the semi-parametric model with smoothing

$$\widehat{\ }_{t+1} = \left(\square_t / \ (\ _t) \cdot \square + \square_{t-1} / \ (\ _{t-1}) \cdot \square^2 + \cdots + \square_{t-p} / \square\left(\square_{t-p}\right)\square^{p-1}\right) \cdot \square\left(\hat{\square}_{t+1}\right) + (1 - \square - \square^2 - \cdots - \square^{p-1}) \cdot \square \quad (9)$$

where $$ is the order of moving average, $0 < \square < 1$ is smoothing weight, and $$ is a matrix of ones. Matrix multiplication and division is made element-wise.

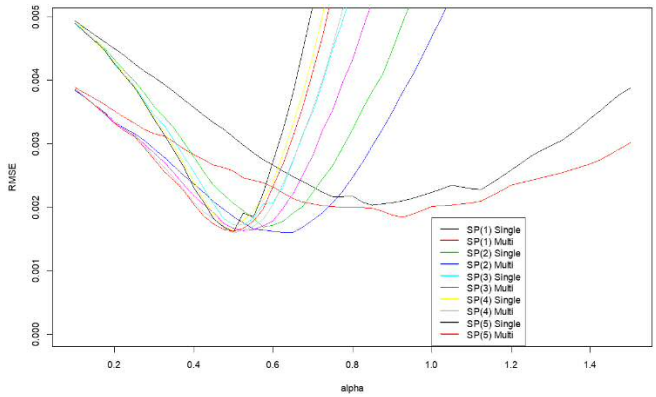


Figure 6 Root mean square error dependency of semi-parametric model on smoothing parameter . Semi-parametric model of order is denoted by SP(p).

Conclusions and recommendations

1. The probabilities of going from one risk class to another are modelled by independent ARIMAX models of the same order. The forecasts seem to predict the probabilities reasonably accurately for a two-year horizon – the average errors have a magnitude of 0.01. The developed approach could be applied using more sophisticated techniques, such as neural networks etc., to obtain better forecasts.
2. Multinomial distribution applied to transition matrix each row separately produced lower RMSE error as compared to Naïve estimate and full Markov transition matrix estimate for dataset with many zero-count transitions. It is worth to investigate it further for these types of problems. Estimation of full Markov transition rate matrix gave no advantage over Multinomial distribution estimate and Naïve estimate.
3. In case of simulated data set, we have observed a clear correlation structure; as for C1 data set, the correlations were weaker. Neural network approach produced predictions for the first simulated data set which were more accurate in comparison with Naïve estimates (RMSE: 0.033, 0.029, 0.037, 0.038), while predictions for the C1 data set were RMSE: 0.112, 0.096, 0.097, 0.121.
4. The smoothed semi-parametric model of order 3, performs better than Naïve estimator for the periods of stable economy growth periods observed for time index 18-32) and have similar performance to Naïve estimator.

References

1. https://www.math.kth.se/matstat/seminarier/reports/M-exjobb14/140908.pdf
2. A. M. Berd. Dynamic Estimation of Credit Rating Transition Probabilities. arxiv.org/pdf/0912.4621.pdf
3. A Ng (2000). "CS229 Lecture notes" (PDF) CS229 Lecture notes 1 (1), 1-3.
4. *Yuan, Ya-xiang (1999). "Step-sizes for the gradient method" (PDF). AMS/IP Studies in Advanced Mathematics. Providence, RI: American Mathematical Society.* ***42*** *(2): 785.*
5. Hastings, W.K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". Biometrika. 57(1): 97–109.