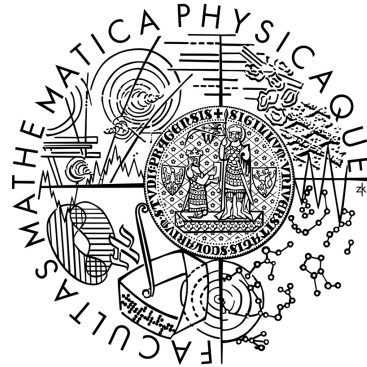


Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Bc. Peter Zvirinský

Social networks and data mining

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: doc. RNDr. Iveta Mrázová CSc.

Study programme: Software Engineering

Specialization: Software Systems

Prague year 2014

I would like to thank my supervisor doc. RNDr. Iveta Mrázová, CSc for her guidance, inspiration and time spent on consultations. I would also like to thank PhDr. Ing. Jiří Skuhrovec for the opportunity of working on the data from the Insolvency Register and his insights into the process of the insolvency proceedings.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In date

signature

Název práce: Sociální sítě a dobývání znalostí

Autor: Peter Zvirinský

Katedra / Ústav: Katedra teoretické informatiky a matematické logiky

Vedoucí diplomové práce: doc. RNDr. Iveta Mrázová CSc., Katedra teoretické informatiky a matematické logiky

Abstrakt: Aktuální techniky z oblasti dobývání znalostí představují moderní přístup vhodný pro analýzu velkého množství dat i extrakci potenciálně užitečných informací z těchto dat. Práce je věnována detailnímu studiu jednotlivých kroků procesu dobývání znalostí, včetně přípravy dat, jejich ukládání, čištění, analýzy i vizualizace získaných výsledků. Velký důraz je v práci kladen na efektivní analýzu dat veřejně dostupných z Insolvenčního rejstříku České republiky, který obsahuje údaje o insolvenčních řízeních zahájených v České republice po 1. lednu 2008. S ohledem na specifika uvažovaného typu dat se zaměříme zejména na popis, implementaci, testování a vyhodnocení vybraných metod dobývání znalostí. Mezi jinými budou studované techniky zahrnovat i analýzu nákupního košíku, Bayesovské sítě a metody pro analýzu sociálních sítí. Výsledky provedených analýz dokumentují některé ze sociálních vztahů patrných ve struktuře současné české společnosti.

Klíčová slova: sociální sítě, data mining, předspracování dat, dobývání znalostí, extrakce pravidel

Title: Social networks and data mining

Author: Peter Zvirinský

Department / Institute: Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: doc. RNDr. Iveta Mrázová CSc., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Recent data mining methods represent modern approaches capable of analyzing large amounts of data and extracting meaningful and potentially useful information from it. In this work, we discuss all the essential steps of the data mining process - including data preparation, storage, cleaning, data analysis as well as visualization of the obtained results. In particular, this work is focused on the data available publicly from the Insolvency Register of the Czech Republic, that

comprises all insolvency proceedings commenced after 1. January 2008 in the Czech Republic. With regard to the considered type of data, several data mining methods have been discussed, implemented, tested and evaluated. Among others, the studied techniques include Market Basket Analysis, Bayesian networks and social network analysis. The obtained results reveal several social patterns common in the current Czech society.

Keywords: social networks, data mining, data pre-processing, knowledge extraction, rule extraction

Contents

1. Introduction.....	3
2. Methodology.....	5
2.1. CRISP-DM.....	5
3. Insolvency Act.....	7
3.1. Procedural Bodies.....	7
3.2. Insolvency.....	9
3.3. Methods of resolution.....	9
3.3.1. Bankruptcy Order.....	10
3.3.2. Restructuring.....	10
3.3.3. Discharge.....	10
3.4. Exceptions from the Effects of the Insolvency Act.....	11
3.5. Moratorium.....	11
3.6. Information System.....	11
3.7. Insolvency States.....	13
4. Data Extraction.....	16
4.1. Web Application.....	16
4.1.1. Details Page.....	18
4.1.2. Web Scraper.....	21
4.2. Web Service.....	24
4.2.1. Data Structure.....	24
4.2.2. Querying.....	27
4.2.3. Web Service Scraper.....	27
4.3. Documents.....	29
4.3.1. Applications Of Receivables.....	29
4.3.2. Optical Character Recognition.....	32
4.4. Data storage.....	40
4.4.1. RDBMS.....	40
5. Data Analysis.....	46
5.1. Verification.....	46
5.2. Demography.....	49
5.3. State Transitions.....	54

5.3.1. Histogram Analysis.....	55
5.3.2. Bayesian Networks.....	64
5.3.2.1. Simple Model.....	66
5.3.2.2. Incorporating The Time Spent Information.....	73
5.3.3. Market Basket Analysis.....	78
5.3.3.1. Simple Model.....	80
5.3.3.2. Incorporating Insolvency Proceedings' Details.....	85
5.4. Subjects.....	91
5.4.1. Creditors.....	91
5.4.2. Social Network Analysis.....	97
5.4.2.1. Centrality.....	97
5.4.2.2. Prestige.....	100
5.4.2.3. PageRank.....	101
5.4.2.4. HITS.....	103
5.4.2.5. Community Discovery.....	106
5.4.2.6. Markov Cluster Algorithm.....	109
5.4.3. Social Network Of The Insolvency Register.....	111
6. Conclusion.....	129
6.1. Summary.....	129
6.2. Future work.....	130
7. Bibliography.....	131
Appendix A: Installation manual.....	134
Appendix B: Programmers manual.....	137
Appendix C: User manual.....	139
Appendix D: Contents of the enclosed CD.....	146

1. Introduction

On January the 1st, 2008 a new information system called Insolvency Register of the Czech Republic was launched by the government of the Czech Republic. The Insolvency Register contains huge amounts of data reflecting the social status of many subjects from the Czech society. An appealing approach capable of processing this type of data seems to represent modern data mining methods. Our work is devoted to their study and thorough analysis.

The Insolvency Register aimed at becoming a part of the Czech Open Data space along with systems like the Company Register [1] or the Public Contracts Register [2]. It is administered by the Ministry of Justice of the Czech Republic and it brings more clarity and publicity to the process of insolvency proceedings. On the same date and together with this information system, the Parliament of the Czech Republic adopted a new Act No. 182/2006 Coll. on Insolvency and Methods of its Resolutions, also known as the Insolvency Act. The new Act replaced the former Act No. 328/1991 Coll. and regulates the resolution of insolvency and imminent bankruptcy of debtors by court proceedings via one of the defined methods. For the purposes of the Insolvency Act, an insolvency proceeding means a court proceeding, the subject of which is the debtor's insolvency or imminent bankruptcy.

The Insolvency Act defines three methods of resolution in the insolvency proceeding: bankruptcy order, restructuring, and discharge. According to statistical reports¹ published by the Insolvency Register itself, 105 360 insolvency proceedings were commenced until the third quarter of 2013. Out of all those cases, 45 151 were resolved with discharge, 13 626 with bankruptcy order and only 83 with restructuring. In the rest 46 500 insolvency proceedings, the debtor was not found insolvent or the method of resolution is yet to be chosen. From these numbers, it is obvious that the most commonly used method of resolution is discharge.

The discharge may be performed by the liquidation of debtor's assets or through the performance of the payment calendar. From the 45 151 discharges 44 026 (97.5%) were resolved through the performance of the payment calendar. The Insolvency Act states the maximal duration for the performance of the payment calendar to be 5 years. This means that most of the proceedings which were resolved

1 Official statistical reports from the Insolvency Register can be found here <http://www.insolvencni-zakon.justice.cz/expertni-skupina-s22/statistiky.html>

with discharge in 2008 will finish in 2013. Currently there is thus no way of telling how successful or unsuccessful most of the insolvency proceedings were and what circumstances led to the corresponding results.

An insolvency proceeding was successful if the creditors' satisfaction was achieved and there is no reason to continue the insolvency proceeding. On the other hand if the creditors were not satisfied, then the insolvency proceeding was unsuccessful and the method of its resolution must be reconsidered.

Every insolvency proceeding's process is characterized by a series of states and transitions between them. These transitions directly relate to the way of addressing each respective insolvency. Based on all the collected data it is very interesting to analyze and predict the proceedings' results, next state transitions or their durations based on previous states and additional information about the debtors like age, gender, title, etc.

In this way our research can help the debtors to finish their insolvency proceedings successfully and give them better understanding of the whole process. It could be also helpful to creditors who could decide, whether it would be profitable for them to take part in the proceedings or not based on our results. It can also serve as a valid feedback to the Ministry of Justice of the Czech Republic for the forthcoming amendments of the Insolvency Act.

This thesis is organized as follows. In the next section we will describe the standardized methodology of data mining. In Section 3 we will focus on the Insolvency Act and its description in the extent required to understand all aspects of this work. In Section 4 we will describe the Insolvency Register with all the data it provides and present a full solution for extracting these data. In Section 5 we will first evaluate the quality of the extracted data, then we will focus on analyzing the data and applying specific data mining algorithms to it.

2. Methodology

Data mining is a creative process which requires a number of different skills, knowledge and resources. In the late 90-ties when the interest in data mining was mounting, no widely used approach to data mining existed. Because of that the success or failure of a data mining project was highly dependent on one particular person or a team carrying it out. Consequently, the successful practice of data mining could not be necessarily repeated across the whole enterprise.

There was a clear need for a standardized data mining process which would help the organizations to launch their data mining projects and finish them successfully. One of the proposed model which would standardize the methodology of data mining was the Cross Industry Standard Process for Data Mining (CRISP-DM) [3][4].

2.1. CRISP-DM

The CRISP-DM model organizes the data mining process into six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. These phases help organizations understand the data mining process and provide a road map to follow while planning and carrying out a data mining project. We shall now proceed to description of each phase.

The first and most important phase in any data mining project is the initial **business understanding** phase, that focuses on understanding the project objectives from the business perspective. The business perspective is then converted to a data mining problem definition together with a preliminary plan designed to achieve those objectives.

After the business understanding phase as the second comes the **data understanding phase**, that starts with an initial data collection. The analyst then proceeds to increase his familiarity with the data, to identify data quality problems, and to discover initial insights into the data.

The third phase is the **data preparation** phase, that covers all activities to construct the final data set or the data that will be fed into the modeling tools from the initial raw data. Tasks related to this phase include table, record, and attribute selection together with transformation and cleaning of data for the modeling tools.

In the fourth phase named **modeling** various modeling techniques are selected

and applied on the prepared data. Typically, several techniques exist for the same data mining problem type. Some techniques have specific requirements on the form of the data. Therefore, stepping back to data preparation may be necessary.

After all selected modeling techniques were applied the **evaluation** phase comes next. In this phase it is important to thoroughly evaluate the models and review whether the models achieved the business objectives. At the end of the evaluation phase, the project leader should decide whether the results are satisfactory and it is acceptable to move forward to the deployment phase, or whether some changes need to be done in any of the previous phases. These changes might include altering the business objectives, altering how the data is extracted and prepared, but also altering the data mining models used.

If it is decided that the results are sufficient at the end of the evaluation phase, the project moves to the **deployment** phase. The gained knowledge must be now organized and presented in a way that the customer can use it. Depending on the requirements, the deployment phase can be as simple as generating a report, or as complex as implementing a repeatable data mining process across the enterprise.

The complete process of CRISP-DM with all the phases is show in Figure 1 below.



Figure 1: The overview of the CRISP-DM model [5].

3. Insolvency Act

In this section, the Insolvency Act will be described to the extent required for a full comprehension of this work. The description is mostly based on the English translation of the Insolvency Act by Wolters Kluver ČR, a.s. [6] and is using the terminology established in this publication. Since it was published in 2011, it only contains amendments added until August 2011. Nevertheless, it can be still a valid source of more detailed information. The original up to date version of the Act can be found in the Collection of Laws of the Czech Republic. As a source of more practical information about the application of the Insolvency Act can serve [7] (available only in czech).

The Insolvency Act denotes the Act No. 182/2006 Coll. on Insolvency and Methods of its Resolution and all it's later amendments¹.

This Act regulates the resolution of bankruptcy and imminent bankruptcy of the debtor by court proceedings through one of the defined methods. The main principles of the insolvency proceedings are²:

1. The insolvency proceedings must be held so that none of the participants will be unfairly damaged and the highest possible satisfaction of the creditors will be achieved.
2. The creditors, who have similar position under this Act, have essentially equal opportunities in the insolvency proceeding.
3. The creditors cannot act towards their satisfaction outside of the insolvency proceeding.

An insolvency proceeding is commenced by filing an insolvency petition to the court. It can be filed either by the creditor or the debtor himself.

Certain disputes which may be caused by the insolvency proceedings and are defined in the Insolvency Act³ are called incidental disputes.

3.1. Procedural Bodies

Procedural bodies under the Insolvency Act are:

1 The up to date list of amendments can be found here:
<http://insolvencni-zakon.justice.cz/obecne-informace/zakon.html>

2 For details see Section 5 of the Insolvency Act.

3 For details see Part 1 Chapter 4 Division 8 of the Insolvency Act.

1. the insolvency court
2. the debtor
3. creditors who exercise their rights against the debtors
4. the insolvency administrator or another administrator
5. the Attorney General's Office which entered the insolvency proceedings or incidental dispute
6. the liquidator of the debtor

For the purposes of this, work only the first 4 procedural bodies are important. For more information about the Attorney General's Office or the liquidator of the debtor see Part 1 Chapter 2 of the Insolvency Act.

The insolvency court means a court before which the insolvency proceedings are held. This court issues decisions which are required or assumed by law and supervises the activities of other procedural bodies.

The creditor is a party (e.g. a legal entity or a natural person) that has delivered a product, service or loan, and is owed money by the debtor. The debtor on the other hand is a person or entity that owes money to the creditor. Creditors shall claim their receivables by the submission of an application of receivables⁴ and shall be satisfied depending on the method of resolution of the insolvency proceeding.

The main purpose of an insolvency administrator is handling assets of the debtor in order to achieve the highest possible satisfaction of the creditors. The insolvency administrators are appointed from the list of the insolvency administrators managed by the Ministry Of Justice⁵. The appointment of administrators is governed by a special regulation⁶. For certain situations special types of administrators exist. These are:

1. Insolvency Administrator Representative⁷
2. Separate Insolvency Administrator⁸
3. Special Insolvency Administrator⁹

4 For details see Part 1 Chapter 5 of the Insolvency Act.

5 The up-to-date list can be found here <https://isir.justice.cz/InsSpravci/public/seznamFiltr.do>.

6 Act No. 312/2006 Coll on Insolvency administrators, as amended under Act No. 296/2007 Coll.

7 For details see Section 33 of the Insolvency Act.

8 For details see Section 34 of the Insolvency Act.

9 For details see Section 35 of the Insolvency Act.

4. Provisional Insolvency Administrator¹⁰

Where appropriate, the insolvency court may appoint a representative of the insolvency administrator in the event that they could not temporarily perform their duties due to a serious reason.

If the insolvency administrator is excluded from certain acts due to his relationship to one of the debtor's creditors or to one of the representatives of the debtor's creditors, the insolvency court may appoint a Separate Insolvency Administrator for such acts.

In cases when it is necessary to deal with a special issue requiring professional expertise during the insolvency proceedings, the insolvency court may also appoint a Special Insolvency Administrator and amend their relationship with the insolvency administrator.

The insolvency court may appoint an insolvency administrator before the declaration of bankruptcy, in this case he is called the Provisional Administrator.

3.2. Insolvency

A debtor is insolvent if he has¹¹:

1. several creditors and
2. outstanding financial liabilities for more than 30 days overdue and
3. he is not able to fulfill such liabilities

It is believed that the debtor is not able to fulfill his financial liabilities if:

1. he stops the payments for more than 3 months overdue, or
2. the satisfaction of any outstanding financial receivables against the debtor may not be achieved by the enforcement of a decision or the execution

3.3. Methods of resolution

The method of the resolution of the insolvency or an imminent bankruptcy of a debtor in the insolvency proceeding means:

1. bankruptcy order
2. restructuring

¹⁰ For details see Section 84 of the Insolvency Act.

¹¹ For details see Section 3 of the Insolvency Act.

3. discharge
4. special methods of resolution for certain types of cases¹² (these will be not considered in this thesis because they are very rare and differ from case to case)

3.3.1. Bankruptcy Order

A bankruptcy order is a method of resolution based on the fact that the determined receivables of the creditors are essentially satisfied from the proceeds of the liquidation of assets. But also on the fact that non-satisfied receivables or any part thereof does not cease to exist unless the law stipulates otherwise.

For details see Sections 244 to 315 of the Insolvency Act.

3.3.2. Restructuring

Restructuring usually means the gradual satisfaction of creditors' receivables while preserving the operation of the debtor's company, secured by measures taken for the economic recovery of such a company under the restructuring plan approved by the insolvency court. This resolution is admissible only if the debtor is an entrepreneur and his total turnover for the last accounting period preceding the insolvency proceeding was at least 100 000 000 CZK or has more than 100 employees.

For details see Sections 316 to 364 of the Insolvency Act.

3.3.3. Discharge

Discharge is admissible only in case when the debtor is not an entrepreneur. It may be performed by the liquidation of the debtor's assets or through the performance of the payment calendar.

A discharge by the liquidation of the assets shall proceed similarly like the liquidation of the assets in case of the bankruptcy order.

In case of a discharge through the performance of the payment calendar, the debtor is obligated to pay the monthly installments to the creditors from his income for a period of 5 years. This resolution prefers social purpose over economical, it should allow the debtor a “fresh start” and motivate him to repay his debts[8].

For details see Sections 389 to 418 of the Insolvency Act.

¹² For details see Section 4 of the Insolvency Act.

3.4. Exceptions from the Effects of the Insolvency Act

This Act cannot be applicable if it is in regards to:

1. the State
2. the local government unit
3. the Czech National Bank
4. the General Health Insurance Company of the Czech Republic
5. the Deposit Insurance Fund
6. the Guarantee Fund of the Securities Traders
7. public non-profit institutional health facilities
8. a public college
9. Special cases¹³

3.5. Moratorium

The purpose of moratorium is to provide the debtor with a possibility of settlement with the creditors before the whole insolvency proceeding. This means that during a moratorium, bankruptcy cannot be declared. The debtor may file a petition to declare a moratorium to the insolvency court within 7 days of the submission of the insolvency petition.

For details about moratorium see Sections 115 to 127 of the Insolvency Act.

3.6. Information System

The Insolvency Act states that the Insolvency Register is an information system of the public administration, administered by the Ministry of Justice of the Czech Republic. The Insolvency Register contains a list of insolvency administrators, debtors and insolvency files. The Insolvency Register is publicly accessible, with some exceptions also stipulated by this Act. Everyone has the right to inspect it and make copies and extracts thereof.

If the debtor is a natural person, his name, surname, domicile, birth certificate number (if they do not have a birth certificate number, their date of birth) is recorded in the list of debtors.

¹³ For complete list see Section 6 the Insolvency Act.

If the debtor is a natural person who is also an entrepreneur, his place of business (if it is different from his domicile) and identification number is recorded in addition to the data in the previous paragraph.

If the debtor is a legal entity, its company name, registered office and identification number is recorded in the list of the debtors too.

The insolvency court shall publish the following information in the Insolvency Register chronologically, stating the time of the entry:

1. the decisions of the insolvency court issued in the insolvency proceedings
2. any submissions which are recorded in the judicial file kept by the insolvency court regarding the debtor
3. other information stipulated by the Insolvency Act

Upon the request of a natural person who made the relevant submission, the insolvency court may decide that some of the personal data of such natural person contained in the submission shall not be publicly accessible in the Insolvency Register. The insolvency court shall always publish the name and surname of such natural person in the Insolvency Register. Also submissions which are subject to confidentiality under special legal regulations shall not be recorded in the Insolvency Register.

After the expiration of 5 years since the date of the full force and effect of the decision by which the insolvency proceeding was completed, the insolvency court shall delete the debtor from the list of the debtors and render the details about them in the Insolvency Register inaccessible. There are also specific decisions on the insolvency petition which allow the debtor to request the removal of his insolvency proceeding from the Insolvency Register. These are¹⁴:

1. the refusal of an insolvency petition due to defects
2. the termination of the proceedings due to the lack of the conditions of the proceedings, which cannot be eliminated or which could not be eliminated, or due to the withdrawal of the insolvency petition
3. the dismissal of an insolvency petition

For more details see Sections 419 to 425 of the Insolvency Act.

¹⁴ These are points a.), b.) and c.) listed in Section 142 of the Insolvency Act

3.7. Insolvency States

The description of the Insolvency Act so far suggests that the insolvency proceeding is going through a series of states which describe its process. These states are also recorded in the Insolvency Register. In this section, all the possible states of an insolvency proceeding will be described. All possible states and transitions between them are illustrated in Figure 2¹⁵.

We will demonstrate how an insolvency proceeding is moving between states on the insolvency proceeding with reference number **KSBR 39 INS 1038 / 2008**. This insolvency proceeding was commenced on 14. 3. 2008, when it entered the initial state **Unresolved**. Then on 22. 4. 2008 it moved to the state **Bankruptcy Order**. After that on 23. 7. 2012 it moved to the state **Finished**, followed by the state **Effective** on 14. 8. 2012. Finally, on 12. 12. 2012 it moved to the state **Checked Off**, where the insolvency proceeding was finished successfully.

Incorrect Entry state represents an incorrect entry that was submitted to the Insolvency Register. This can be an error made by an official working with the register. It has no legal consequences.

Unresolved is the initial state of every insolvency proceeding. In this state the proceeding is waiting for a decision to be made by the insolvency court.

Unresolved – Advanced occurs when the petition for a insolvency proceeding was submitted to the incorrect insolvency court. This can be for example a insolvency court in different region etc. No decisions or changes are made in the insolvency proceeding in this state. The insolvency petition is only sent to the corresponding insolvency court.

Bankruptcy takes place when the Insolvency Court decides that the debtor is insolvent (based on the conditions defined in Section 3.2). In this state the method of resolution of the insolvency proceeding is determined. It may be one of the three defined in Section 3.3: bankruptcy order, restructuring or discharge. If the method of resolution is determined before the insolvency court decides that the debtor is insolvent, then this state may be skipped. In that case the insolvency proceeding moves directly to state which corresponds with the method of resolution.

Bankruptcy Order, Restructuring and Discharge states indicate that the

15 Figure 2 was adopted and translated from the official Insolvency Register documentation that can be found here: https://isir.justice.cz/isir/help/Popis_WS_v1_8.pdf

insolvency proceeding is being resolved with a method that has the same name as the corresponding state. These methods were described in Section 3.3.

Finished becomes an insolvency proceeding when the insolvency court issues a decision that will end the insolvency proceeding. In this state the decision did not attain legal effect yet.

Effective is a state which follows the *Finished* state and in which the decision about the insolvency proceeding's end attains legal effect. In this state it is not possible to submit a legal remedy anymore.

Checked Off means that the insolvency proceeding is removed from the list of active cases of the corresponding insolvency court.

Revived state occurs when the court in the appeal procedure declares the original court decision invalid. This causes that the insolvency proceeding is “revived” and the insolvency court must deal with it again.

Canceled by Supreme Court is a special state which takes place when the decision of the region court is canceled by the Supreme Court. In this case the decision made in the insolvency proceeding by the region court must be renegotiated.

Moratorium state may occur before the insolvency court decides whether the debtor is insolvent or not. During moratorium the debtor gets a chance of resolving his debts with his creditors before the whole insolvency proceeding starts. For more details see Section 3.5.

Bankruptcy Order after cancellation is again a special state when the insolvency court ends the ongoing bankruptcy order and the creditors or the administrator requests an appeal to restore it.

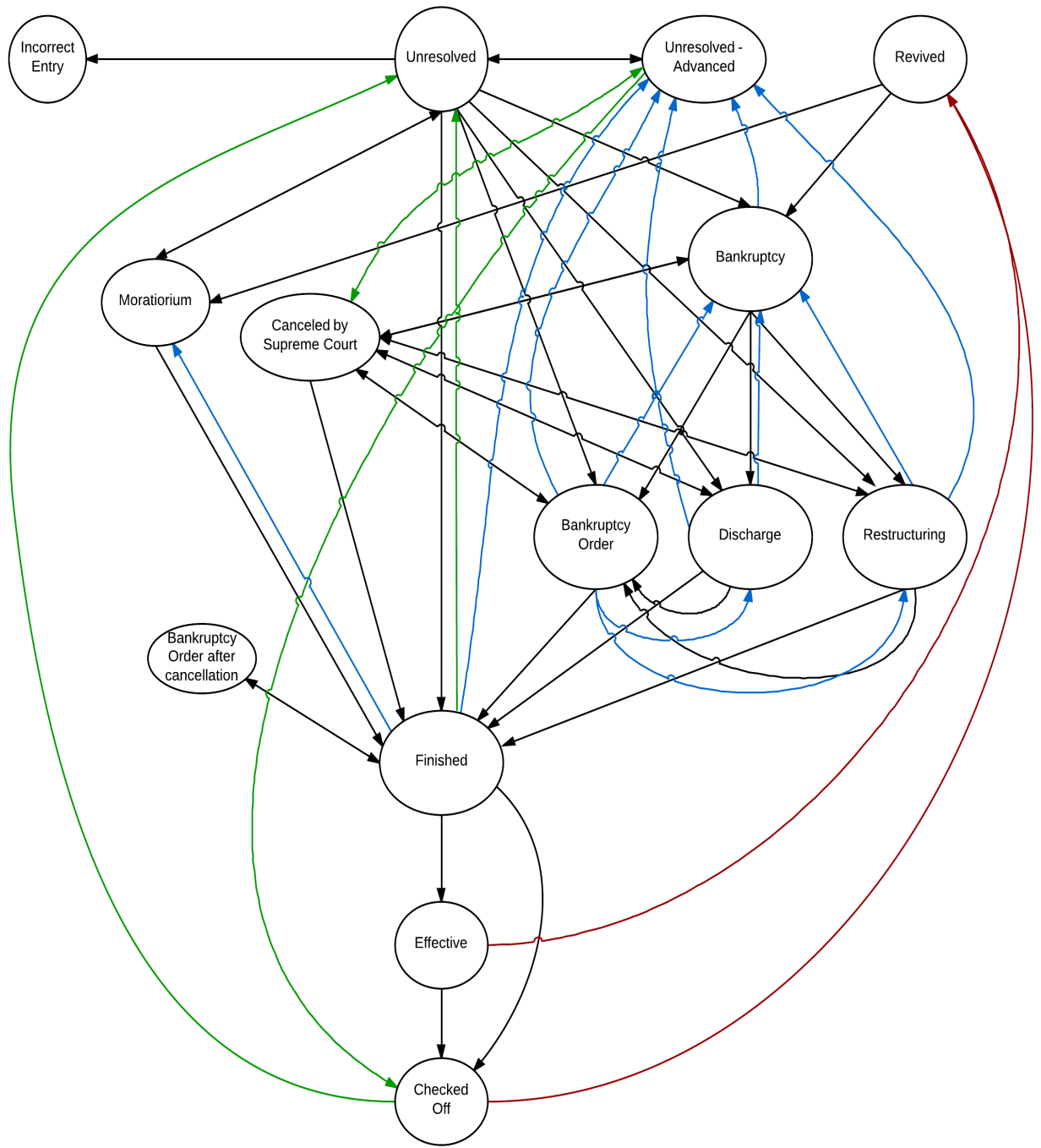


Figure 2: Insolvency States¹⁵

4. Data Extraction

Previous Section 3 dealt with legal aspects of the Insolvency Register. This section will focus on describing the two major parts of the Insolvency Register, the web application and the web service. Both these parts are designed for different purposes and provide different view on the data stored in the Insolvency Register. Therefore, both of them will be thoroughly described, from technical standpoint and also by the content of the data they provide. Then a complete solution for extracting and storing data from both of them will be presented. Since the Insolvency Register also provides data in different formats (Html, Xml, Pdf), all of them will be discussed and a method of data extraction will be proposed for each one of them. In the end we will summarize all the extracted data, and present the final data model that will be used later on for data analysis.

The Insolvency Register is located at <http://isir.justice.cz/> where the official documentation¹ can be found as well. Unfortunately, besides this work there is no documentation of the Insolvency Register available in English.

4.1. Web Application

The web application is the first major part of the of the Insolvency Register designed for the general public. The web application actually serves two purposes, it provides the possibility of searching through ongoing insolvency proceedings and showing detailed up to date information about them.

The main part of the Insolvency Register home page shown in Figure 3 is the search form. The form allows to search and filter ongoing insolvency proceedings by various attributes and preferences². The most important ones are:

1. name of the debtor
2. identification number in case of a debtor who is also an entrepreneur
3. date of birth or birth certificate number
4. domicile
5. insolvency proceedings number
6. commencement date of the insolvency proceeding

1 The official documentation of the Insolvency Register can be found here: <https://isir.justice.cz/isir/common/stat.do?kodStranky=NASTENKA>

2 Help and more details regarding the search form can be found here: <https://isir.justice.cz/isir/common/stat.do?kodStranky=NAPOVEDA>

V insolvenčním rejstříku lze vyhledat pouze dlužníky, proti kterým bylo zahájeno insolvenční řízení po 1. lednu 2008 a nebyli z rejstříku vyloučeni dle § 425 insolvenčního zákona. Dlužníky, proti kterým bylo zahájeno konkursní či vyrovnací řízení před 1. lednem 2008, lze vyhledat v Evidenci úpadků.

Příjmení/název

Vyhledat pouze dle začátku příjmení/názvu

Jméno fyzické osoby

IC

Datum narození

Roční číslo fyzické osoby

Oheč

Spisová značka INS /

vedená u

Stav řízení

v období od do

Aktuální řízení Aktuální i ukončená řízení

Akce

v období od do

Seřazení značka

Spisová značka incidentního řízení Icn /

Max. počet zobrazených položek 50 100 200 300 400

Monitoring insolvenčního rejstříku

Datum: Období:

Dne 2.12.2012 byl zde zveřejněn nový formulář přihlášky pohledávky. Vzhledy nového formuláře jsou popsány v sekci Aktuální info.

Upozorňujeme věřitele, že v souladu s ust. § 176 insolvenčního zákona, je možné podávat přihlášku pohledávky pouze na formulář, jehož podobu zveřejnil ministerstvo zvláštním úředním oznámením.

Figure 3: Insolvency Register's search form.

4.1.1. Details Page

After finding a particular insolvency proceeding, its detail containing all essential information is shown to the user. An example of such a detail is illustrated in Figure 4. Among others it shows all information mentioned in Section 3.6. For example the debtor's name, domicile and date of birth. Also the list of insolvency administrators together with the insolvency state is there. Information not mentioned yet is the table containing a list of documents visible in the bottom half and the insolvency identifier.

The identifier of the insolvency proceeding shown in Figure 4 is encoded as **KSCB 27 INS 21407/2011**. Below, let us explain what these numbers and letters actually mean. The first 4 letters **KSCB** are identifying the region of the court where the insolvency proceeding is being held. All Regional Courts together with their identifiers are listed in Table 1. Next in the insolvency identifier comes the number **27**, which is the Senate number. Judicial body which decides in the matters of the insolvency proceedings is called Senate in the Czech Republic. After that comes **INS** which is a shortcut for the Insolvency Register. In the remaining code **21407/2011** the number before the slash is the number of the proceeding and the number after that is the year of the proceeding's commencement.

Regional Court identifier	Region
KSJIMBM	Brno
KSJICCB	České Budějovice
KSVYCHK	Hradec Králové
KSVYCHKP1	Hradec Králové – branch office in Pardubice
KSSEMOSP	Ostrava
KSZPCPM	Plzeň
KSSTCAB	Prague
KSSCEUL	Ústí nad Labem
KSSECULP1	Ústí nad Labem – branch office in Liberec
MSPHAAB	City Court of Prague

Table 1: List of Regional Courts.

Detail insolvenčního řízení Emilie Citterbartové

Aktuální stav

Právní mocná věc

Základní identifikační údaje

Spisová značka:

KSCB 27 INS 21407 / 2011 vedená u Krajského soudu v Českých Budějovicích

Jmenný název:

Emilie Citterbartová

IČ:

65016491 (viz obchodní rejstřík)

Rodné číslo / Datum nar.:

615915/0888 / 15.09.1961

Adresa

Rozmítal pod Tremšínem, Brez 27, PSC 262 42

Bydliště:

Insolvenční správce

Historie insolvenčního řízení

Datum poslední zveřejněné události

31.01.2012

Oddíl A - Řízení do úpadku **Oddíl B - Řízení po úpadku** Oddíl C - Incidenční spory Oddíl D - Ostatní Oddíl P - Přílošky

Okamžik zveřejnění	Popis	Dokument	Vedlejší dokument	Datum právní moci	Senátní značka VSNS
1. 22.11.2011	Insolvenční návrh spojený s návrhem na povolení oddlužení	plný text (420 KB)			
2. 22.11.2011	Návrh - přílohy	plný text (3737 KB)			
3. 22.11.2011	Pověření vyššího soudního úředníka	plný text (164 KB)			
4. 22.11.2011	Pověření asistenta soudce	plný text (164 KB)			
5. 22.11.2011	Vyhlaška o zahájení insolvenčního řízení	plný text (168 KB)	plný text (1234 KB)		
6. 22.11.2011	Výzva na doplnění příloh insolvenčního návrhu	plný text (177 KB)	plný text (844 KB)		
7. 02.12.2011	Návrh - doplnění návrhu	plný text (20155 KB)			
8. 05.12.2011	Sdělení dlužníka	plný text (447 KB)			
9. 08.12.2011	Sdělení dlužníka	plný text (1419 KB)			
10. 03.01.2012	Usnesení o odmítnutí insolvenčního návrhu	plný text (162 KB)		30.01.2012	
11. 30.01.2012	Vzdání se práva odvolání	plný text (121 KB)			

Zobrazené záznamy: 1-11 z 11

Rozšířené zobrazení

Figure 4: Insolvency proceeding's details page.

The table at the bottom of the insolvency proceeding's details page contains a listing of various legal documents regarding the insolvency proceeding, that are divided into 5 sections. These sections are:

1. A – proceeding before declaring bankruptcy
2. B – proceeding after declaring bankruptcy
3. C – incidental disputes
4. D – others
5. P – applications of receivables

As the name suggests, section A and B respectively contains all legal documents related to the insolvency proceeding before and after declaring bankruptcy. They include mostly the decisions of the insolvency court in the matters of the insolvency proceeding, but also proposals from the creditors or the debtor etc. Section C is dedicated to incidental disputes that occurred during the insolvency proceeding and were described in Section 3. Then comes Section D which contains documents that do not belong to any other section, such as various requests from the creditors or the debtor, etc. At last comes Section P where the user can find all applications of receivables or their amendments. The table of Section P has one more column which holds the name of the creditor who submitted the corresponding application of receivables. This column however, may be blank in some cases and in order to find out the creditor's name it is necessary to find it manually in the application of receivables. One of these listings is shown in Figure 5, where the seventh column is dedicated to the creditors names and in two cases, it is blank.

Oddíl A - Řízení do úpadku			Oddíl B - Řízení po úpadku			Oddíl C - Incidenční spory			Oddíl D - Ostatní			Oddíl P - Přihlášky		
	Okamžik zveřejnění		Popis	Dokument	Vedlejší dokument	Datum právní moci	Platní věřitelé			Senátní značka VSNS				
P1 - 1.	26.10.2011	08:08	Přihláška pohledávky	plný text (759 kB)			Česká spořitelna, a.s.							
P2 - 1.	02.11.2011	07:37	Přihláška pohledávky	plný text (585 kB)			PROFI CREDIT Czech, a.s.							
P3 - 1.	18.11.2011	10:36	Přihláška pohledávky	plný text (694 kB)			SMART Capital, a.s.							
P4 - 1.	18.11.2011	10:48	Přihláška pohledávky	plný text (472 kB)			ESSOX s.r.o.							
P4 - 2.	26.01.2012	12:35	Vyrozumění	plný text (443 kB)										
P4 - 3.	01.02.2012	13:46	Usnesení o část. odmítnutí přihlášky	plný text (154 kB)	plný text (738 kB)	18.02.2012								
P5 - 1.	21.11.2011	15:29	Přihláška pohledávky	plný text (272 kB)			GE Money Bank, a.s.							

Figure 5: Section P of the documents' table from an insolvency proceeding's detail page.

Most of the documents' table columns are shared by all 5 sections. These columns are:

1. serial number representing the order of the document submissions
2. submission date and time

3. document type name
4. URL for downloading the document

There are also some columns which were not mentioned but are not important for our research and will be omitted.

4.1.2. Web Scraper

For the purpose of extracting data from the web application, a classic web crawler/scraper was implemented. The scraper is able to browse and scrape all the insolvency proceedings commenced within a given range of dates that is its input. The scraper extracts every information from the insolvency proceeding's details page and all related documents from all sections. For the purposes of extracting data from HTML XPath[9] was used. The algorithm expressing the process of scraping in detail is described in the section below.

Web Scraper Algorithm

Input:

dateFrom ... date from which the scraping should start, in format

day.month.year, e.g. "01.01.2008"

dateTo ... date to which the scraping should be done, using the same format

as above

Step 1. Initialization and main loop

Initialize a new variable *currentDate* and set its value to *dateFrom*.

Step 1.1

Download the web page containing all insolvency proceedings commenced on *currentDate* by the following HTTP GET request:

```
https://isir.justice.cz/isir/ueu/vysledek_lustrace.do?spis_znacky_datum=${currentDate}
```

Instead of the place holder *\${currentDate}* insert the current date in the same format as was given on input (applies also in later steps). A concrete example of the full HTTP request:

```
https://isir.justice.cz/isir/ueu/vysledek_lustrace.do?spis_znacky_datum=01.01.2008
```

Store the HTML of the downloaded web page to the variable *currentPage*.

Step 1.2

Extract all URL-s pointing to the insolvency proceedings' detail pages and store them

to the list named *detailUrls*.

The extraction is done by the following XPath query:

```
//table[@class='vysledekLustrace']//tr[td//text()[contains(., '{currentDate}')] //a/@href
```

Step 1.3

Download the detail page from every URL in *detailUrls*. Extract all information regarding the actual insolvency proceeding and store them to the database. The extraction is basically just a series of XPath queries and because there is a lot of them they will not be presented here.

Step 1.4

If $currentDate < dateTo$ increase *currentDate* by one day and go to ***Step 1.1***.

The scraper is actually very simple and straightforward. Just to summarize what the scraper extracts, it is basically everything that is on the insolvency proceeding's details page with the exception of the birth certificate number and date of birth. This data cannot be stored by third parties due to the Act on personal data protection³. This law basically states that it is prohibited to store enough information about individuals so that it is possible for somebody to reliably identify them. The birth certificate number is of course unique for every person and therefore it is illegal to store it. Additionally, the same applies in case of storing the birth-date of individuals together with their names and addresses. Thus, only the year of birth is stored by the web scraper.

This Act however does not prohibit the processing of these informations and the birth certificate number in particular. This is very useful because in the Czech Republic the birth certificate number is generated in such a way that it is possible to determine whether it belongs to a male or a female. Basically, if the number on third position of the birth certificate number is larger than 5, then it belongs to a female individual. This may seem a little strange, therefore let's look at how to birth certificate numbers are being generated more precisely⁴.

The birth certificate number in the Czech Republic is a 9 or 10 digit number e.g. 1272127890. The first two digits stand for the year of birth, next two digits stand for

³ Act No. 101/2000 Coll on personal data protection.

⁴ The latest method for generating birth certificate number is defined by Act No. 133/2000 Coll.

month of birth and lastly the digits 5 to 6 represent the day of the month of the date of birth. Those 6 numbers are then followed by 3⁵ or 4 digits randomly selected to distinguish people born on the same day. However, the example birth certificate shows month of birth to be 72 , but there are only 12 months indexed from 1 to 12. This is because for all females 50 or 70 is added to their month of birth before creating the birth certificate number.

The administrators of the Insolvency Register did not set any access constraints, and basically it is possible to scrape with no limits. Nevertheless, it is still appropriate to prevent the server's overload by setting reasonable limits for the number of downloads required per time unit. This topic was heavily discussed in the late 1990s when web crawling became popular but it is still an issue. For example the WIRE crawler [10] uses 15 second intervals between downloads. The Mercator Web crawler [11] follows an adaptive politeness policy: if it took t seconds to download a web page, the crawler waits for $10 * t$ seconds before the next download. Dill et al. [12] uses 1 seconds between downloads. The crawler implemented for the Czech Insolvency Register uses the same adaptive policy as the Mercator Web crawler but the multiplying constant is adjustable. In the scope of this work we decided to use a constant equal to 2, which means that the crawler waits $2 * t$ seconds between downloads. While abiding this basic rule it is possible to safely scrape all insolvency proceedings in approximately 3 days. This seems to be a reasonable compromise between crawling speed and excessive usage of the system.

It is worth mentioning that there are some circumstances⁶ in which the insolvency proceeding is removed⁷ from the Insolvency Register e.g., 5 year after its end. This results to occasional invalid URL-s on the web and different sets of insolvencies scraped at different times. Estimations of how many insolvency proceedings were scraped and were available for the purposes of this thesis will be discussed in Section 5.

All the data found on the insolvency proceeding's details page should be also available through the Web Service interface. But based on our previous experience with public data sources it is difficult to guarantee that the data from either source

5 The 9 digit birth certificate number is obsolete and all new ones are created with 10 digits.

6 See Section 3.6 for more details.

7 An example of a insolvency proceeding removed from the Insolvency Register is [KSPH 36 INS 8831 / 2012](#).

will be complete at any time. Therefore, the best thing to do is to combine the data from both sources and increase the robustness of the system.

4.2. Web Service

The web application of the Insolvency Register described in the previous subsection provides a user oriented interface for browsing data about ongoing insolvency proceedings. On the other-hand the web service of the Insolvency Register is designed for automated machine processing purposes. This interface⁸ is implemented as a SOAP[13] web service on top of the classic HTTP protocol.

To secure the high availability of the web application, the web service is hosted on a different physical server even though it is accessible from the same domain⁹ as the web application. The availability of the web application has a higher priority than the web service simply because it is used by many more people. The access to the web service is not restricted by any security means and can be utilized at any time. Information about planned shutdowns can be found on the web of the insolvency register¹⁰.

Despite having no security regulations on the side of the web service, the amount of use is restricted to prevent system from overloading. It is recommended to wait at least 10 minutes between the queries from the same subject. This limit seems to be very restrictive when the large amount of data stored in the Insolvency Register is considered. The solution to this issue together with the description of how exactly the web service interface will be used is going to be described in Section 4.2.3.

4.2.1. Data Structure

The format of the data provided by the web service does not support similar methods for querying like the web application, based on the insolvency proceedings characteristics, e.g., reference number, debtor name, domicile etc. The interface was instead designed in more of a publish-subscription [14] manner rather than as a standard API with all possible methods for querying. To be able to query the insolvency proceedings a subscriber must thus create his own copy of the database and structure it in a way that fits his needs. Because of this, the data about the

8 The WSDL format of the interface definition may be found here:
https://isir.justice.cz:8443/isir_ws/services/IsirPub001?wsdl

9 The Insolvency Register's web service is hosted on the following URL: <https://isir.justice.cz:8443>

10 Information about planned shutdowns of the Insolvency Register's web service can be found here
<https://isir.justice.cz/isir/common/stat.do?kodStranky=NASTENKA>

insolvency proceedings are provided by the web service in the form of a so-called event. For every change, that occurred in an insolvency proceeding, an event is generated by the Insolvency Register and published by the web service afterwards. The first event which is shared by all insolvency proceedings, indicates its creation. Other ones may be a result of submitting documents, changing insolvency proceedings state and so on. One of these events in its original form is shown in Figure 6. It is important to say that an event cannot change once it is published, so its effect can only be changed by another event.

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header />
<soapenv:Body>
  <ns1:getIsirPub0012Responsexmlns:ns1="urn:IsirPub001/types">
    <result>
      <cas>2007-11-01T00:00:00.000Z</cas>
      <id>522</id>
      <idDokument></idDokument>
      <poznamka><?xml version="1.0" encoding="UTF-8"?><tns:udalost
        xmlns:tns="http://www.cca.cz/isir/poznamka"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        verzeXsd="1"
        xsi:schemaLocation="http://www.cca.cz/isir/poznamka
          https://isir.justice.cz:8443/isir_ws/xsd/poznamka.xsd">
          <idOsobyPuvodce>KSJIMBM</idOsobyPuvodce><vec>
            <druhStavRizeni>NEVYRIZENA</druhStavRizeni></vec>
          </tns:udalost>
        </poznamka >
      <spisZnacka>INS 86/2007</spisZnacka>
      <typ>3</typ>
      <typText>Insolvenční návrh</typText>
      <oddil>A</oddil>
      <poradiVOddilu>1</poradiVOddilu>
    </result>
  </ns1:getIsirPub0012Response>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 6: Example of an event published by the Insolvency Register's web service in its raw form.

The event structure consists of elements shared by all events such as **id** and then a variable part called **poznamka**¹¹ which has again the form of an XML snippet. As a result, the **note** XML part has its own definition in an XSD[15] format which can be found on the Insolvency Register's web page¹². Therefore, the definition of the **note** can change and evolve without affecting the appearance of the whole interface. The

11 The English translation of **poznamka** is **note** and will be adopted for the rest of the thesis.

12 The up to date XSD definition of the **note** can be found here:
https://isir.justice.cz/isir/help/poznamka_1_9.xsd

list of elements shared by every event, together with their descriptions can be found in Table 2.

Element Name	Description
Cas	Date and time when the event occurred
Id	Unique serial id of the event in the Insolvency Register
idDokument	URL of the document which is complementing the event. It is volatile and may be blank just like in Figure 6.
Poznamka	Note described in this section.
spisZnacka	Insolvency proceeding's number and year of commencement. Just like described in Section 4.1.
Typ	Event type id from the list of all existing event types ¹³ .
typText	Event type name from the list of all existing event types.
Oddil	One of the sections A,B,C,D or P (as defined in Section 4.1.1) to which the event belongs.
poradiVOddilu	Serial number of the event in its corresponding section.

Table 2: List of elements common for all events.

In Section 4.1, the documents on the insolvency proceeding's detail page are divided into 5 sections A, B, C, D, P now, also the events published by the web service are divided into the same sections. This is because the term event and document very often label the same thing. However, there are some events that do not represent any documents. For example, there are service events that announce changes done in the Insolvency Register by administrators or on special occasions. Furthermore, the documents are usually published by the web service in two stages by two different events. At first, the event describing the document with all meta-data (id, Typ, typText ...) is published, and then several days later the document's URL is published. This is because the digitalization of the documents may take some time, so the submission process is split into two parts. If we use the term document in this work, we are speaking about an event with a corresponding URL which can be found on the insolvency proceeding's detail page.

¹³ The list of all existing event types may be found here:
https://isir.justice.cz/isir/help/Cis_udalosti.xls

The note has its own structure which is much more complicated than the structure of the web service interface itself. Besides that it has already undergone many major changes which makes it even more difficult to grasp. Because of this the note's structure itself will be not described here, its details and full description is available at the Insolvency Register's web page¹⁴.

4.2.2. Querying

So far, only the data structure was described but not how the data can be queried. At the beginning of this section, it was stated that the interface does not provide querying options similar to the web application. Instead the SOAP interface provides only two methods with the same name *getIsirPub0012*, but different parameters. The first method takes an integer parameter representing the event's id. The second method takes a date parameter corresponding to the event's occurrence date.

The method with the integer parameter returns all events with an id larger than the input parameter in an ordered manner. Similarly, the second method returns all the events that occurred after the date which was taken as input, again in an ordered manner. It is important to note that not more than 1000 events are returned for either one of the methods. Below is one example for calling both methods, this URL may be simply used from a web browser.

1. https://isir.justice.cz:8443/isir_ws/services/IsirPub001/getIsirPub0012?long_1=1246
2. https://isir.justice.cz:8443/isir_ws/services/IsirPub001/getIsirPub001?Calendar_1=2004-10-18T00:00:00.000

4.2.3. Web Service Scraper

For data extraction from the Insolvency Register's web service, another scraper was implemented as a part of this work. Unlike the web application's scraper, this one works in two phases. At first it scrapes all the events to a local cache¹⁵ and once this is done, the second phase starts. In the second phase the scraper goes through every event from the local cache and extracts all essential data from those events. At last, the obtained data is merged with the data extracted by the Insolvency Register's web application scraper.

14 Full description of the note structure may be found here:

https://isir.justice.cz/isir/help/Popis_WS_v1_8.pdf

15 Local cache is created in the same database where all the other data is stored.

The web service provides some additional data, besides everything that is provided by the web application. This is for example the information about changing states of the insolvency proceedings. These possible states are the same that were described in Section 3.7. The Insolvency Register's web application is only showing the current state of the insolvency proceeding.

Furthermore, the web service is also providing more detailed information about subjects of the insolvency proceedings such as the creditors or the debtors. These informations are shown in Figure 7.

```
<osoba>
  <idOsoby>HERMA JAN 230581 3</idOsoby>
  <druhRoleVRizeni>DLUŽNÍK</druhRoleVRizeni>
  <nazevOsoby>Herma</nazevOsoby>
  <druhOsoby>F</druhOsoby>
  <jmeno>Jan</jmeno>
  <titulPred>Ing.</titulPred>
  <ic>123456</ic>
  <dic>DIC123456</dic>
  <rc>810523/2817</rc>
  <adresa/>
</osoba>
```

Figure 7: More detailed information about subjects of the insolvency proceedings provided by the Insolvency Register's web service.

Below we will summarize what information can be obtained about a subject from the web service. The first item is **idOsoby** which should be an unique identifier of the subject. Unfortunately, there might also occur cases when the same subjects have more than one different **idOsoby**. Because of that, we will not extract this information at all. The next item is **druhRoleVRizeni** which specifies the role of the subject in the insolvency proceeding, e.g., debtor, creditor or insolvency administrator. Then the name of the subject in a structured form is provided by the following three items: **nazevOsoby**, **jmeno**, **titulPred**, which stand for surname, name and title. Item **druhOsoby** distinguishes the subject to be either a natural person (**F**) or a legal entity (**P**). If the subject is also an entrepreneur, his identification numbers¹⁶ **IC** and **DIC** are provided as well. And lastly the birth certificate number **rc** and the address **adresa** is present. The restrictions¹⁷ about storing birth certificate numbers applies in this case as well.

The last problem we need to deal with, is the limited number of allowed requests

¹⁶ IC is an unique identification number which is given to every entrepreneur in the Czech Republic and DIC stands for tax identification number.

¹⁷ Act No. 101/2000 Coll on personal data protection.

that are required by the Insolvency Register's administrators. According to the documentation the limit is set to one request per 10 minutes. This restriction however, is not directly set in the system. Consequently, until a subject is not generating too many requests at once that causes server to overload, no problems should arise. Additionally, we will now demonstrate why the one request per 10 minutes requirement is just too restrictive and to be able to download the full history of the Insolvency Register in a reasonable time this restriction must be violated.

To this date approximately 10 000 000 events occurred. We said that with a single request it is possible to obtain a maximum of 1000 events, so to scrape all events roughly 10 000 requests are necessary. With the 10 minutes restriction it would take approximately 69 days to scrape all the data. Obviously that would take just too long. Therefore we kept reducing the interval between requests and we found out that one request per 30 seconds does not cause any undesired overloading and may be used safely. Using this restriction it is possible to scrape the complete history in roughly 4 to 7 days.

4.3. Documents

The Insolvency Register is providing URL-s to various documents through the web application and the web service. All these documents are stored and provided in PDF format. In the vast majority, the documents are created by scanning the originals as they were submitted. This results into the fact that it is not possible to easily extract any textual data from them. However, there are some cases when the documents have the form of a textual PDF files. Those documents were probably created in one of the existing Office Suites¹⁸ and then exported to PDF. An example of such a scanned document is shown in Figure 10.

4.3.1. Applications Of Receivables

Document Section P is mostly dedicated to documents called applications of receivables. These documents are crucial for this work since they determine the creditors that take a part in the insolvency proceedings. It was also mentioned that in some cases the name of the creditor is available in the documents table of the insolvency proceeding's detail page. But sadly, there are also cases when the creditor's names are not available, and the only way of finding out the creditor's

¹⁸ Popular Office Suites: MS Office, Open Office, Libre Office.

name is to open the application of receivables manually and look it up in the scanned document.

To summarize the status of the Insolvency Register, roughly 650 000 applications of receivables were submitted to the Insolvency Register to this date. In about 380 000 cases, the name of the creditor is available directly from the insolvency proceeding's details page. In approximately 270 000 (41%) cases the name of the creditor is thus not known. It is also very important to state that the number of cases with missing creditor names has a dropping rate. This problem occurred especially within the first 3 years of the Insolvency Register where only in 1% of cases the name of the creditor was present on the insolvency proceeding's detail page. This trend is shown in Figure 8 below.

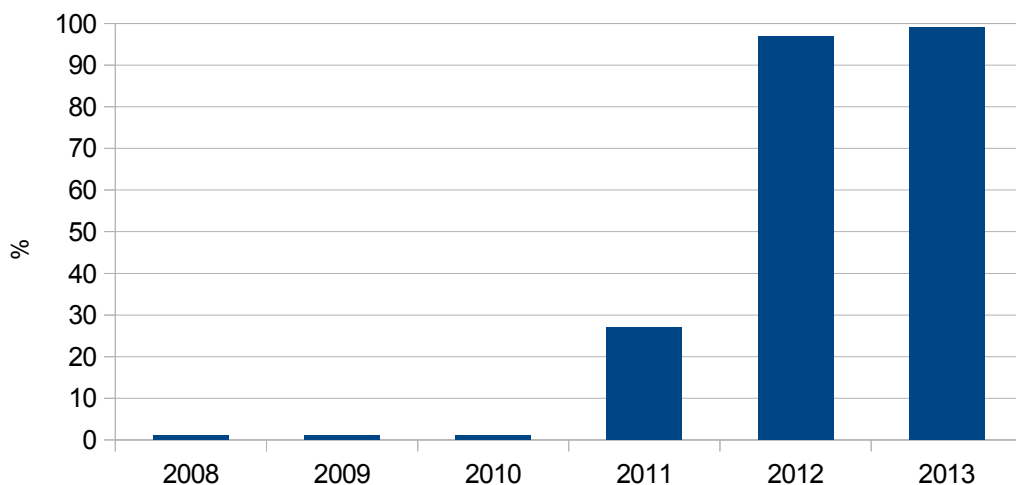


Figure 8: Number of cases in % where the creditor name is present together with the application of receivables.

Nevertheless, the number of cases with unknown creditor names is just too high and it is necessary to explore the possibilities of retrieving them automatically.

Since in roughly 59 % of all applications of receivables the name of the creditor is known, the idea of using machine learning approaches is very tempting. There is a large group of machine learning algorithms designed specially for text document classification [16][17][18]. The way the document classification algorithms (from now on just document classifiers) usually work is following. At first a group of documents labeled into several categories are given to the document classifier as an input. Then the document classifier uses those documents to learn itself so that is

able to automatically categorize new unseen documents into those categories.

In our case it may be possible to use those 59 % of applications of receivables labeled by creditor names as a training set for a document classifier. Then after training a document classifier, we could use it to automatically label the applications of receivables by creditor names. However, there might be one more problem with this approach which is, that we don't know whether the creditor names we know in those 59% of applications of receivables also occur in the rest 41%. There might be a fair chance that there is a group of creditors that is labeled in each case and then another group of creditors that is not labeled at all. If this was the case, then we would have no training documents to represent them while learning the classifier and the whole process would be useless.

However, in Figure 8 a very sharp change in the number of applications of receivables with present creditor names is visible. If the year 2011 would be examined with an even higher granularity let's say monthly a sharp increase from 1% to 95% would be visible in October. Based on this evidence, we can assume that for some reason the Insolvency Register started to publish the creditor names with a much higher emphasis. The Insolvency Register does not state why this change occurred anywhere. However, this observation is very important because this means that for whatever reason these names started occurring in the Insolvency Register in almost 100% of applications of receivables.

If we consider to what extent the creditors are participating in the insolvency proceedings, especially that a small group of creditors occur in vast majority of insolvency proceedings¹⁹, we can assume that the same frequency of occurrences applies also to applications of receivables submitted before October 2011. This assumption was also validated on a random sample of 100 applications of receivables submitted before October 2011. From Section 5.4.1 we also know that the top 50 most frequent creditors cover 58% of occurrences in the insolvency proceedings. Therefore, if we would only create a document classifier for these 50 creditors then in the ideal case another 24%²⁰ of creditors could be determined from applications of receivables with unknown creditors.

19 For statistics about the number of creditors' occurrences see Section 5.4.1 and especially Figure 34.

20 There remains 41% of applications of receivables without a creditor name so in the ideal case, when we assume that the classifier works with 100% accuracy we could determine $0.41 * 0.58 = 0.238$ more creditor names.

But to use a document classifier it is necessary to obtain some text documents first. Keeping in mind that the majority of documents are created by scanning original documents in their paper form, there is only one way of obtaining texts from them, and that is by using an Optical Character Recognition (OCR) tool.

4.3.2. Optical Character Recognition

For the purposes of this work the Tesseract OCR[19] tool was tested. Tesseract is a completely Open Source project and it also performs very well with comparison to commercial tools. More details about Tesseract together with its comparison with commercial tool Transym OCR is described in [20]. Tesseract was tested on 500 applications of receivables with known creditor names. The test consisted of extracting textual data from every application of receivables and then full-text searching the creditor name in the extracted text. The results were following:

- OCR processing of one page took in average 22 seconds
- the full creditor name was found in 47 % of applications of receivables
- at least one word of creditor name longer than 3 characters was found in 85% of applications of receivables (on any page)

These results were very promising if we take into account that it was used right out of the box without any special configuration nor document preprocessing.

However, the quality of the documents is poor, and there are ways of improving OCR accuracy by advanced image preprocessing steps [21]. The documents provided by the Insolvency Register are often skewed and contain various boxes and lines. Some of them are also colored and some texts occur on non white background. All these factors significantly reduce the accuracy of any OCR tool. Because of that the impact on OCR accuracy of the following three simple preprocessing steps has been tested:

- adjusting the document's perspective
- removal of all lines and boxes
- removal of all non white backgrounds from the document

From Figure 10, Figure 11 and their respective Tesseract output's Text 1, Text 2 it is clearly visible that in some cases these three simple preprocessing steps can improve the OCR accuracy significantly. Furthermore, all of these three steps may be performed automatically by using appropriate algorithms, e.g., the Hough Transform[22] for automatic line detection. But before reviewing methods for

automatic image preprocessing it is necessary to address the performance and complexity issues regarding the automatic OCR processing of all available applications of receivables.

There were approximately 650 000 applications of receivables submitted to the Insolvency Register to this date. Based on a random sample of 100 applications²¹ of receivables we observed that an application is in average 7 pages long. While OCR processing of one page by Tesseract takes in average 22 seconds. From these facts we can estimate the time required to OCR process all applications of receivable as it follows:

$$650\,000 * 7 * 22 = 100\,100\,000 \text{ seconds} = 3.17 \text{ years}$$

So it would take roughly 3,2 years to OCR process all applications of receivables. And it is worth mentioning that the time required for image preprocessing steps wasn't even considered in this estimate. Based on this unsatisfying estimation, several ways of OCR performance optimization techniques were examined. The first thing to do was checking the OCR documentation and trying to find a way to speed up the Tesseract itself. The first thing which can be done is to lower the DPI of the images, but all tests taken so far were already using the lowest recommended value 300 DPI. It was tested with even lower DPI 200 and 250 but this reduces the Tesseract accuracy heavily. The outputs in most cases were absolutely useless. Another tip mentioned was to increase the input documents quality, which was already done by the preprocessing steps applied in previous tests. But those steps did not reduce the time of OCR significantly.

After everything that was tried so far the only way left to reduce the time required for OCR processing is either by reducing the number of pages or by using more computational power.

21 This random sample was selected evenly from all applications of receivables submitted since 2008 till October 2013

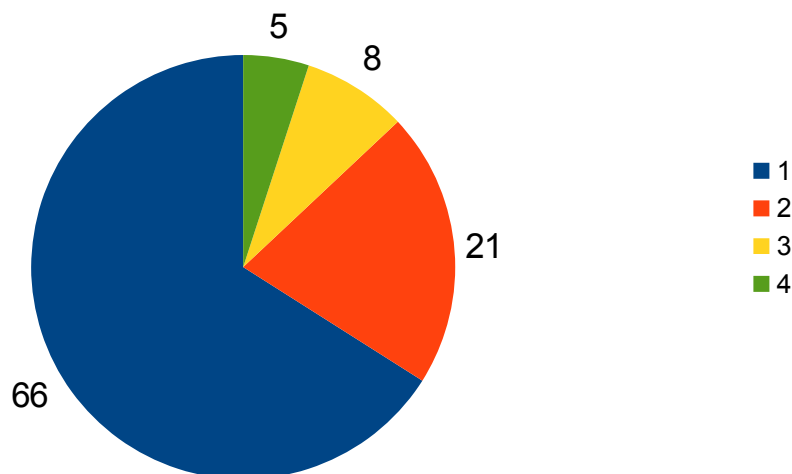


Figure 9: Page distribution of a random sample of applications of receivables on which the creditor's name appeared.

In case of reducing the number of pages to OCR process we've had to determine which pages are important for our research. Therefore, we selected another random sample of 100 applications of receivables and we examined on which page the name of the creditor appears. Basically, that is the most important piece of information for the goal of determining which creditor submitted the application of receivables. The result of the test is shown in Figure 9. Figure 9 shows that in 66 cases out of the considered 100 applications of receivables, the name of the creditor appeared on the first page. In 21 cases, the name of the creditor appeared on the second page and so on. As a result, we can assume that if only the first 4 pages of all applications of receivables would be OCR processed, then this should cover the vast majority of creditor names. Following this assumption, we could have reduced the time required to OCR process all applications of receivables to roughly 1,8 year as shows the updated formula:

$$650\,000 * 4 * 22 = 57200000 \text{ seconds} = 1.81 \text{ years}$$

The last option that has not been explored so far, is to increase the computational power used for OCR processing. All the estimations so far assume that OCR processing will work in a sequential manner, document after document and page after page. This is mostly because Tesseract runs in one thread only and does not support any level of parallelism. Therefore, a good approach would be to implement the parallelization explicitly and run Tesseract on more processor cores and maybe even on more than one computer. If for the purposes of this thesis, a server with 8 core processor would be available then the total time for OCR processing of all

650 000 applications of receivables would be:

$$\frac{650\,000 * 4 * 22}{8} = 14300000 \text{ seconds} = 0.23 \text{ years}$$

But it must be kept in mind that the time required by the preprocessing steps is not taken into account in this estimation. It is also assumed that the OCR processing would run 24/7 without any interruptions.

The extraction of textual data from all applications of receivables is a long term continuous work which could easily take months or even years to finish and requires investments in sufficient hardware equipment. Because of previous arguments this task is out of the scope of this work and may be a subject of future research.

PŘIHLÁŠKA POHLEDÁVKY

Soud:	Krajský soud v Ústí nad Labem	Sp. zn.:	KSUL 45 INS 150/2008
-------	-------------------------------	----------	----------------------

DLUŽNÍK

01 Typ: Fyzická osoba

Státní příslušnost:¹

Osobní údaje	Příjmení:		Jméno:	
	Titul za jm.:		Titul před jm.:	
	Dat. narození: ²		Rodné číslo:	
Údaj o podnik. ³	IČ:		Jiné registr.č.:	
Trvalé bydliště	Obec:		PSČ:	
	Ulice:		Č. p.:	
	Stát:			

02 Typ: Právnícká osoba

Právní řád založení:¹

Právnícká osoba	Název/obch.firma:	3D Plast, spol. s r.o.		
	IČ:	60915137	Jiné registr.č.:	
sídlo	Obec:	Liberec	PSČ:	46007
	Ulice:	Volgogradská	Č. p.:	17
	Stát:	Česká republika		

VĚŘITEL

03 Typ: Fyzická osoba

Státní příslušnost:¹

Osobní údaje	Příjmení:	Pluhař	Jméno:	Jiří
	Titul za jm.:		Titul před jm.:	
	Dat. narození: ²		Rodné číslo:	750307/2566
Údaj o podnik. ³	IČ:	65633326	Jiné registr.č.:	
Trvalé bydliště	Obec:	Liberec	PSČ:	46006
	Ulice:	Dobiášova	Č. p.:	854/2
	Stát:	Česká republika		

04 Typ: Právnícká osoba

Právní řád založení:¹

Právnícká osoba	Název/obch.firma:		Jiné registr.č.:	
	IČ:			
Sídlo	Obec:		PSČ:	
	Ulice:		Č. p.:	
	Stát:			

05 Korespondenční adresa:⁴

Korespondenční adresa	Obec:		PSČ:	
	Ulice:		Č. p.:	
	Stát:			

Elektronická adresa:

Akreditovaný poskytovatel certifikačních služeb:

¹ vyplní se pouze u zahraničních osob

² datum narození se vyplní pokud nebylo přiděleno rodné číslo

³ vyplní se pouze u dlužníka podnikatele

⁴ vyplňte pokud se liší od sídla či trvalého bydliště



Soud: KS Ústí n.L.

11-2008 11-2008

Figure 10: Original application of receivables without any preprocessing steps applied.

PŘIHLÁŠKA POHLEDÁVKY

Soud: Krajský soud v Ústí nad Labem E KSUL 45 INS 150/2008

DLUŽNÍK

01 Typ: Fyzická osoba Státní Příslušnost

-
Liberec
' = '17

VĚŘITEL

Osobní údaje
_ 750307/2566

65633326

Trvalé Liberec
bydliště ' : ' "

Korespondenční
adresa

Elektronická adresa: . Alcreditovanř' Ěoslíyl'-ovatel certítikačních služeb:

ävypkúsepouzeuzmhnuúüüchosob
"H datum narození se vyplní pokud nebylo přiděleno rodné číslo

00001001000001»u\lmmuwuuuuu

Soud: KS Usti n.L.

n-_l-. an ru onnn 11-14

Text 1: Tesseract output for application of receivables shown in Figure 10.

PŘIHLÁŠKA POHLEDÁVKY

Soud: **Krajský soud v Ústí nad Labem**

Sp. zn.: **KSUL 45 INS 150/2008**

DLUŽNÍK

01 Typ: Fyzická osoba

Státní příslušnost:¹

Osobní údaje	Příjmení:	Jméno:
	Titul za jm.:	Titul před jm.:
	Dat. narození: ⁱⁱ	Rodné číslo:
Údaj o podnik.	IC:	Jiné registr.č.:
Trvalé bydliště	Obec:	PSC:
	Ulice:	Č. p.:
	Stát:	

02 Typ: Právnícká osoba

Právní řád založení:¹

Právnícká osoba	Název/obch.firma:	3D Plast, spol. s r.o	
sídlo	IC:	60915137	Jiné registr.č.:
	Obec:	Liberec	PSC: 46007
	Ulice:	Volgogradská	C. p.: 17
	Stát:	Česká republika	

VĚŘITEL

03 Typ: Fyzická osoba

Státní příslušnost:¹

Osobní údaje	Příjmení:	Pluhař	Jméno:	Jiří
	Titul za jm.:		Titul před jm.:	
	Dat. narození: ⁱⁱ		Rodné číslo:	750307/2566
Údaj o podnik.	IC:	65633326	Jiné registr.č.:	
Trvalé bydliště	Obec:	Liberec	PSC:	46006
	Ulice:	Dobiášova	Č. p.:	854/2
	Stát:	Česká republika		

04 Typ: Právnícká osoba

Právní řád založení:¹

Právnícká osoba	Název/obch.firma:		
Sídlo	IC:		Jiné registr.č.:
	Obec:		PSC:
	Ulice:		C. p.:
	Stát:		

05 Korespondenční adresa:^{iv}

Korespondenční adresa	Obec:	PSC:
	Ulice:	C. p.:
	Stát:	

Elektronická adresa:

Akreditovaný poskytovatel certifikačních služeb:

ⁱ vyplní se pouze u zahraničních osob

ⁱⁱ datum narození se vyplní pokud nebylo přiděleno rodné číslo

ⁱⁱⁱ vyplní se pouze u dlužníka podnikatele

^{iv} vyplňte pokud se liší od sídla či trvalého bydliště

Figure 11: The same application of receivables from Figure 10 but with adjusted perspective, removed lines and adjusted black color threshold to produce "more white" background.

PŘIHLÁŠKA POHLEDÁVKY

Soud: Krajský soud v Ústí nad Labem Sp. zn.: KSUL 45 INS 150/2008

DLUŽNÍK

01 Typ: Fyzická osoba Státní příslušnost?

Osobní Příjmení: Jméno:

údaje Titul za jm.: Titul před jm.:

Dat. narození" Rodné číslo:

U o odnik. IC: Jiné registř.:

Trvalé Obec: PSC:

bydliště Ulice: C. p.:

Stát:

02 Typ: Právní osoba Právní řád založení?

Právní Název/obch. firma: 3D Plast, s ol. s r.o.

osoba IC: 60915137 Jiné re ' mě.:

sídlo Obec: Liberec PSC: 46007

Ulice: Volgogradská C. p.: 17

Stát: Česká republika

VĚŘITEL

03 Typ: Fyzická osoba Státní příslušnost?

Osobní Příjmení: Pluhař Jméno: Jiří

údaje Titul za jm.: Titul před jm.:

Dat. narození" Rodné číslo: 750307/2566

U o dník. IC: 65633326 Jiné r ' tr.č.:

Trvalé Obec: Liberec PSC: 46006

bydliště Ulice: Dobiášova C. p.: 854/2

Stát: Česká republika

04 Typ: Právní osoba Právní řád založení?

Právní Název/obch. firma:

osoba IC: Jiné registř.:

Sídlo Obec: PSC:

Ulice: C. p.:

Stát:

05 Korespondenční adresa

Korespondenční Obec: PSC:

adresa Ulice: C. p.:

Stát".

Elektronická adresa: Alcreditovanř' Ě0s%0vatel certíflkačních služeb:

Ívypkúsepouzeuzmhnuúüüchosob

"H datum narození se vyplní pokud nebylo přiděleno rodné číslo

mmmmulužtmwmmmaxtuuuuuurt»mmmuu

Soud: KS Ulli n.L.

nA n. nnnn .na an

Text 2: Tesseract output for document shown on Figure 11.

4.4. Data storage

So far only the Insolvency Register was described in detail with all the data it makes available and how the data from the Insolvency Register can be extracted. In this subsection we will summarize all the extracted data and more importantly define their structure, which will be used in the following sections. It was mentioned several times that the structure in which the data is being provided is not suited for our analysis. Because of that it is necessary to transform the data and then store it in a structure that will meet the needs of this work. Furthermore, this work will strongly focus on the relationships between the subjects participating in the insolvency proceedings such as creditors, debtors and administrators. Consequently, these subjects and their relationships will be the main blocks of the proposed data model structure.

4.4.1. RDBMS

Considering the low number of different entities and their simplistic relationships (see Figure 12 and Table 3) the classical relational model and RDBMS was chosen to store the extracted data. If we also consider the estimated amount of available data from Table 3, then we now that these numbers are much lower than what today's relational databases are capable of storing. Because of these conditions there is no need to look for alternative databases such as NoSQL etc., which are designed for much more complicated data structures and much larger amounts of data.

Entity	Number estimate	Size estimate
Insolvency	120 000	100 MB
Document ²²	4 000 000	6000 MB
Administrator	1 000	1 MB
Subject	100 000	13 MB
Insolvency State	400 000	2000 MB

Table 3: Estimations for the extracted stored data.

²² Only documents' Meta-data and URL-s are stored, not the actual content.

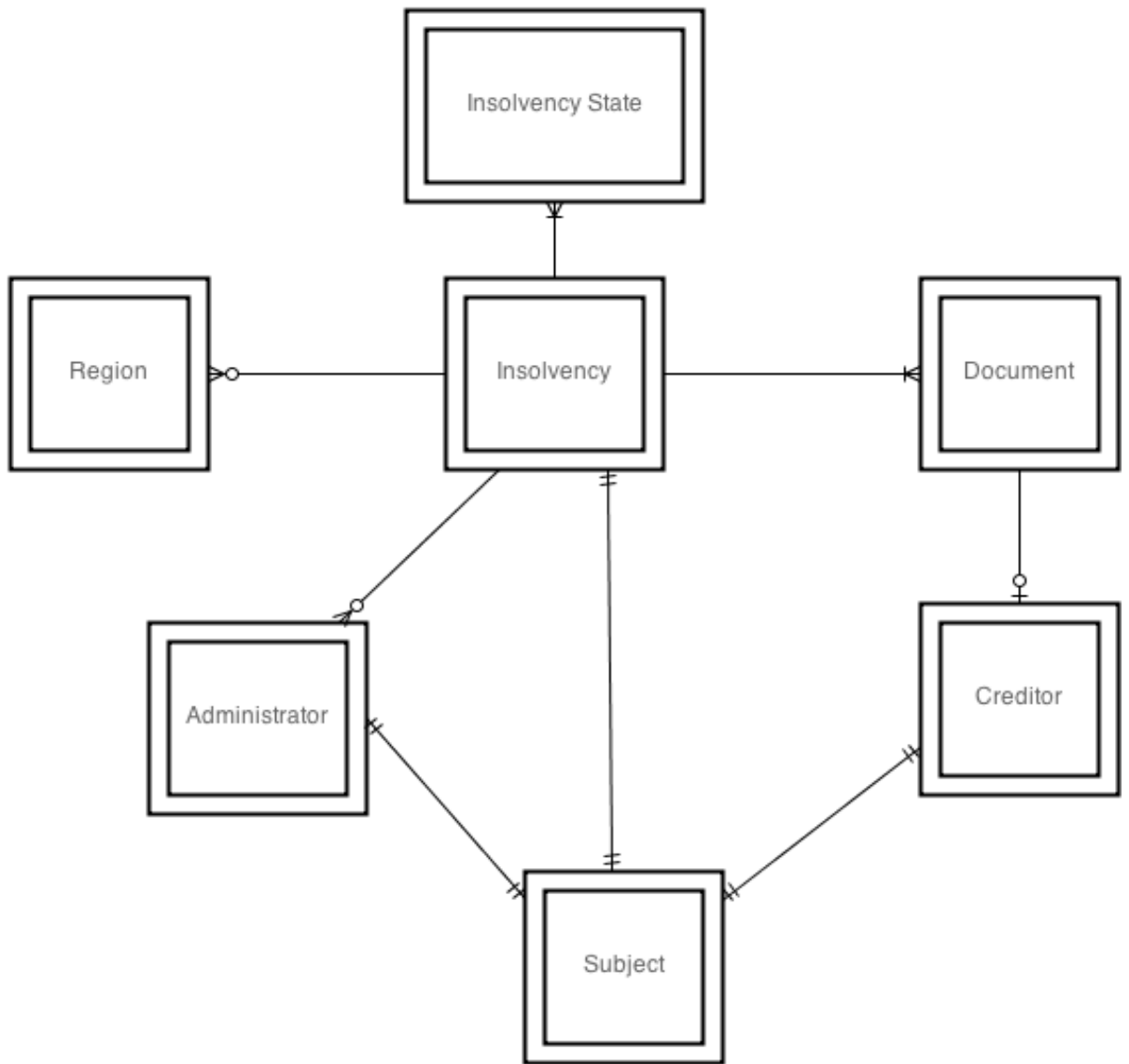


Figure 12: Entity-relationship(ER) model of the extracted data.

Insolvency

Insolvency proceeding abbreviated as Insolvency is the main entity of the proposed model. This entity contains all the basic information related to an insolvency proceeding most of which can be found on the insolvency proceeding detail's page. It has the following attributes:

Attribute name	Description	Example value
id	Unique insolvency proceeding's identifier generated from its reference number.	msphins9997/2010

debtor_name	Name of the debtor.	TAKING s.r.o.
ico	If the debtor is also an entrepreneur his identification number.	27204316
reference_number	Reference number of the insolvency proceeding.	MSPH 93 INS 9997 / 2010
proposal_timestamp	Date and time when the insolvency proceeding was commenced.	2010-09-01 16:42:00
url	Insolvency proceeding detail's url.	https://isir.justice.cz/isir/ueu/evidence_upadcu_detail.do?id=b8bdca1d-a8f9-4a52-901b-ff40602075cc
gender	Debtor's gender (if available).	M
debtor_address	Debtor's address (if available).	Praha 10 Vršovice, Moskevská 617/58, PSČ 101 00
year_of_birth	Debtor's year of birth (if available)	1960
region_id	Debtor's region id (extracted from his address).	1

Table 4: Insolvency entity attributes and values.

The internal id of an insolvency proceeding was chosen as a sub part of the reference number²³. The whole reference number was not selected because it is not directly provided by the Insolvency Register's web service and putting it together manually is too complicated. As a result only the Insolvency Court's region identifier together with the insolvency proceeding's number and year is used. This id is sufficient for a unique identification of any insolvency proceeding.

²³ The structure of the insolvency proceeding's reference number was described in Section 4.1.1.

Insolvency State

The Insolvency State entity represents the states through which the insolvency proceedings are moving. One instance of the Insolvency State entity refers to a change of state of exactly one insolvency proceeding. It has the following attributes:

Attribute name	Description	Example value
insolvency_id	Id of the insolvency proceeding the change of state refers to.	msphins9997/2010
state	Id of the state which refers to one of the states shown in Figure 2.	1
state_change_timestamp	Date and time when the insolvency proceeding entered this state.	2010-09-01 16:42:00
action_id	Id of the event published by the SOAP interface which moved the insolvency to this state.	1243

Table 5: Insolvency State entity attributes and values.

Document and Creditor

The documents related to the insolvency proceedings from all sections are represented by the entity Document. In case of an application of receivables with a known creditor name the Creditor entity is associated with the corresponding application.

Attribute name	Description	Example value
id	Automatically generated id of the document.	1
file_type	Specifies the type of the document, e.g. application of receivables.	Příhláška pohledávky
url	URL from which the document can be downloaded.	https://isir.justice.cz/isir/doc/dokument.PDF?id=1592027
insolvency_id	Id of the insolvency proceeding to which the document belongs.	msphins9997/2010

publish_date	Date and time when the document was submitted to the Insolvency Register.	2010-09-01 16:42:00
creditor	Name of the creditor (if known).	HOLUB Roman s.r.o.
document_section	Document section to which it belongs.	A

Table 6: Document entity attributes and values.

Administrator

With every insolvency proceeding one or more administrators represented by entity Administrator are associated. There are only very limited informations available regarding the administrators. We are basically only able to obtain their name and what kind of administrator they represent in a specific insolvency proceeding.

Attribute name	Description	Example value
id	Administrator's identifier (generated automatically).	1
administrator	Administrator's name.	Ing. Jana Vodrážková
administrator_type	Administrator's type	Insolvenční správce

Table 7: Administrator entity attributes and values.

Subject

Subject is an aggregation entity which may represent any subject of an insolvency proceeding (debtor, creditor or administrator). Its main purpose is to provide more detailed information about subjects that are accessible only from the Insolvency Register's web service. It has the following attributes:

Attribute name	Description	Example value
name	Subjects name	Ing. Jana Vodrážková
ico	If the subject is also an entrepreneur then this is its identification number.	27204316
form	Specifies mostly whether the subject is a natural	P

	person or a legal entity.	
legal_form	If the subject is a legal entity then it's actual legal form.	a.s.
address_form	Specifies what kind of address is associated with this subject e.g. permanent residence, temporary residence etc.	TRVALÁ
address_city	City part of the address.	Praha
address_street	Street part of the address.	Štěpánska
address_description_number	Description number of the address.	112
address_country	Country part of the address.	Česká Republika
address_zip_code	Zip code part of the address.	150 00

Table 8: Subject entity attributes and values.

Region

Based on the debtor's address one of the 14 regions are assigned to the debtors.

These regions include:

- Jihočeský kraj
- Jihomoravský kraj
- Karlovarský kraj
- Královéhradecký kraj
- Liberecký kraj
- Moravskoslezský kraj
- Olomoucký kraj
- Pardubický kraj
- Plzeňský kraj
- Praha
- Středočeský kraj
- Ústecký kraj
- Kraj Vysočina
- Zlínský kraj

5. Data Analysis

So far this work was concentrating on describing the problem domain and everything that needs to be understood and done before starting the analysis. Finally, this section will focus on all the obtained data. At first quality of the data (mostly from completeness standpoint) and potential problems will be discussed. Since all the data used are not available as a prepared dataset for research and only a very little research was done in this area, it is necessary to verify its consistence. The verification process will consist mostly of various visualizations of the data. If the visualization shows some suspicious irregularities, it may indicate a problem in the data extraction process. Similarly, some statistics obtained from the extracted data will be compared to those officially available on the Insolvency Register's web page.

If the verification process will be successful, then the more sophisticated data mining tools will be applied. This analysis will mostly focus on the insolvency proceeding's state transitions and relationships between the entities occurring in the insolvency proceeding such as the creditors, debtors, etc.

5.1. Verification

The first thing to check is the number of successfully scraped insolvency proceedings. This is important due to the fact that they are being removed from the Insolvency Register after some period of time¹. For comparison (Figure 13) the official statistics[23] from the Insolvency Register are used.

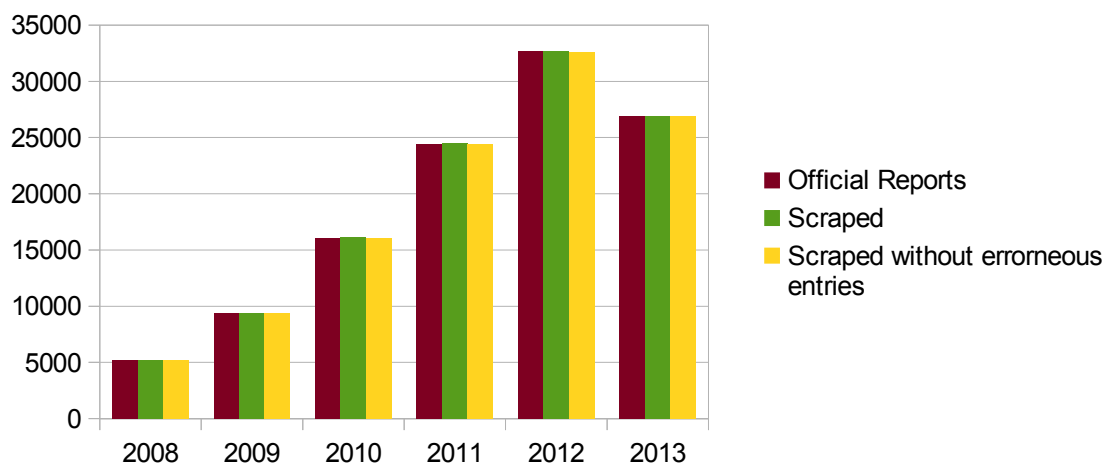


Figure 13: Comparison of insolvency proceeding numbers.

¹ For details see Section 3.6.

Surprisingly, the chart above shows that our database contains even slightly more insolvency proceedings that were commenced according to the official statistics. This is due to the fact, that the numbers from the official statistics do not count erroneous entries. These are insolvency proceedings that entered the state Incorrect Entry² and have no real legal consequences. If all erroneous entries are removed from the statistics we get almost identical numbers which are provided by the official statistics. However, even these results are very surprising because according to Figure 13 we managed to extract data about almost every insolvency proceeding.

We assume that this is mostly because of the fact that scraping was started in late 2012. The Insolvency Act defines that most insolvency proceedings are removed from the Insolvency Register not sooner than 5 years after their end. Consequently, they started being removed in 2013 and we managed to successfully scrape most of the insolvency proceedings before that happened. Another possible reason is that some information about insolvency proceedings remains available in the Insolvency Register even after their removal. The removal of the insolvency proceeding results in making its detail page unavailable. However, some information is also provided in the listings of the Insolvency Register's search results. The reader may simply look up insolvency proceedings that started in the first week of January 2008. One of the insolvency proceedings that will show up in the search result is the one with reference number KSOS 8 INS 1/2008 (Figure 14). Even though its URL is invalid, the listing shows the debtor name, identification number and birth certificate number.

Zadaná vyhledávací kritéria:

Údaje za období: 29.12.2007 - 00:00:00 - 11.01.2008 - 23:59:59
 Údaje platné ke dni: 09.11.2013 - 19.33
 POČET NALEZENÝCH ZÁZNAMŮ 31

[Export do Excelu](#)

Spisová značka	Vedená u	Datum zahájení řízení	Jméno/název	IČ	Rodné číslo
KSOS 8 INS 1 / 2008	Krajský soud v Ostravě	02.01.2008 - 12:30	KOTYSGLOBAL s.r.o.	25865366	
KSOS 13 INS 2 / 2008	Krajský soud v Ostravě	02.01.2008 - 13:51	Lucie Vrbková		715802/5523

Figure 14: Insolvency Register's web page search result listings with invalid URL-s to insolvency proceeding's details.

² For details see Section 3.7.

We could not find out whether this is intentional or a bug in the system. Not to mention that all the informations are still accessible from the Insolvency Register's web service. Once again the reader can simply confirm this fact by calling the web service³ with the date parameter 01/01/2008 which lists the first 1000 events which occurred in the Insolvency Register. The first event ever will again belong to the first insolvency proceeding 1/2008.

The next thing to check is how many times different methods of resolution were applied (Figure 15 and Figure 16). These are namely: discharge, restructuring and bankruptcy order.

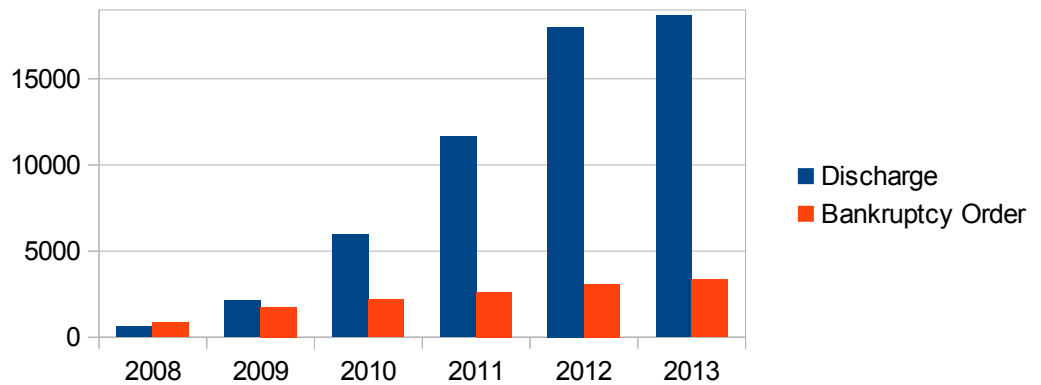


Figure 15: Yearly number of discharge and bankruptcy order applications.

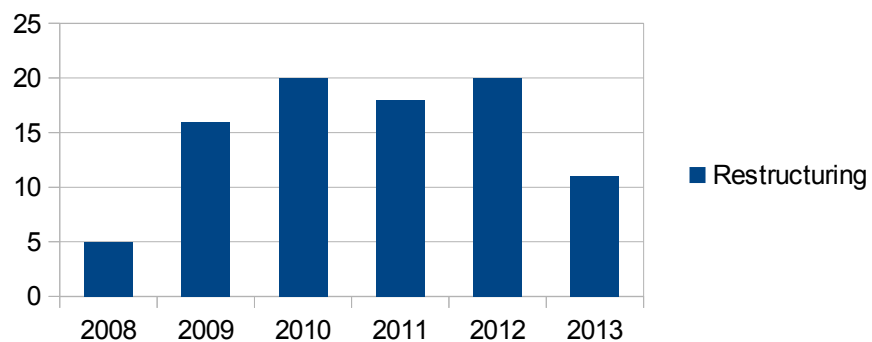


Figure 16: Yearly number of restructuring applications.

The numbers have had to be put into two charts because the numbers of restructuring are too small. This is mainly due to the fact that restructuring is

3 The web service may be simply called from any web browser by submitting the following URL:
https://isir.justice.cz:8443/isir_ws/services/IsirPub001/getIsirPub001?Calendar_1=2008-01-01T00:00:00.000

applicable only in case when the debtor is an entrepreneur with a yearly turnover at least 100 000 000 CZK. Again, all these numbers are very similar to the ones obtained from the official statistics of the Insolvency Register.

As we can see from the three charts above (Figure 13, Figure 15, Figure 16), the numbers of insolvency proceedings were steadily rising yearly by an average of 59% between year 2008 and 2012. From the statistics so far it seems that in 2013, this trend slowed down significantly and the number of insolvency proceedings hit its peak. Another obvious trend is that the most frequently applied method of insolvency proceeding's resolution is discharge.

5.2. Demography

We managed to successfully extract demographical data like gender, year of birth and address of the debtors from the Insolvency Register. Therefore, we can simply visualize the age and gender distribution of the debtors (Figure 17).

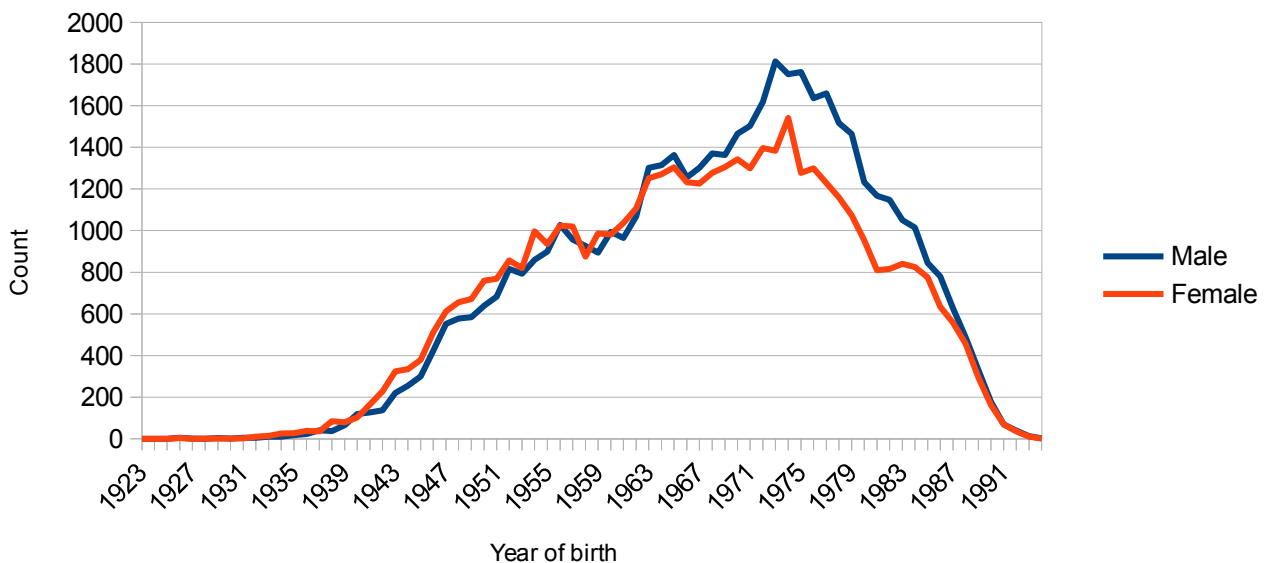


Figure 17: Debtors year of birth and gender distribution

From Figure 17 it is obvious that most of the debtors are between their 40-ties and 50-ties and that the number of male and female debtors is very similar. However, a small change of trend is visible around year 1965 when the number of male debtors started to slightly dominate over the female ones.

To this date we managed to scrape roughly 120 000 insolvency proceedings and in roughly 100 000 (83%) of them, we managed to extract the ZIP code from the

debtors address. From the ZIP code we can easily determine the region in which the debtor lives and visualize this data. However, the number of citizens in each region of the Czech Republic may vary quite a lot, therefore we will normalize the absolute numbers of the debtors by the number of citizens from each region before visualization. We will examine the overall number of the insolvency proceedings and then, we will examine each method of the insolvency proceeding's resolution⁴ individually. Lastly, we will compare the results with the unemployment rates in the Czech Republic. With attention to clarity, the visualization is done by a map of the Czech Republic with a color overlay which expresses the numbers for each region by its darkness. The darker is the color, the larger is the number in the context of the current map. The exact numbers which were used to create the maps shown in Figure 18, Figure 19, Figure 20, Figure 21 are aggregated in Table 9.

From the first map shown in Figure 18 we can see that most insolvency proceedings with respect to the number of citizens were commenced in *Královéhradecký kraj* (166.2)⁵ followed by *Ústecký kraj* (156.1). Furthermore, *Ústecký kraj* has also the overall largest number of commenced insolvency proceedings from all regions. On the opposite end of the scale are *Středočeský kraj* (74.5) and *Kraj Vysočina* (71.8) with the lowest number of commenced insolvency proceedings. As an illustration of how misleading the absolute numbers in this context may be is *Karlovarský kraj* where its 3 368 commenced insolvency proceedings is the lowest number of all. However, it is also the region with the least citizens (only 301 075) and so the normalized number of insolvency proceedings is the third largest in the Czech Republic.

The next map (Figure 19) shows the number of discharges which are again with respect to the number of citizens. In comparison with Figure 18 it looks very similar. Again, *Královéhradecký kraj* (103.4) is followed by *Ústecký kraj* (101.7) as the regions with the most discharges. However, this time on the very tail is region *Praha* (Prague) with only 14.77 discharges.

4 The methods of insolvency proceedings are: discharge, bankruptcy order and restructuring. For details see Section 3.3.

5 All the numbers in the brackets in this subsection are counted as occurrences per 10 000 citizens.



Figure 18: Insolvency proceeding numbers distribution per region.



Figure 19: Discharge numbers distribution per region.

We will continue with Figure 20 showing the bankruptcy order numbers distribution per region. This one is different from the previous two figures because the most⁶ of bankruptcy orders were commenced in *Královéhradecký kraj* (20.0) and *Prague* (15.8). In contrast, least bankruptcy orders were commenced in *Olomoucký kraj* (6.5) and *Kraj Vysočina* (7.0).

We are not showing the map for commenced restructurings because the numbers are just too small. However, these numbers can still be found in Table 9.

Lastly, we drew a similar map of unemployment rates from the the data obtained from the Czech Statistical Office [24] shown in Figure 21. We can see that the map of discharges shown in Figure 19 fits the unemployment rates in most regions. In fact we can clearly see that not only *Ústecký kraj* has the largest unemployment rates, but the number of discharges as well. Conversely *Prague* has the lowest rates of both unemployment and discharges. However, there is also one visible difference and that is represented by *Královéhradecký kraj* where the unemployment rates are average, yet the number of discharges is quite high. This observation is very interesting and brings up a question whether the number of discharges might be a consequence of high unemployment rates or whether it is the other way around. Although we are not qualified to answer this question, this information may be valuable when we will try to predict the insolvency proceedings outcomes.

6 The numbers are again normalized with respect to the number of citizens in each region.



Figure 20: Bankruptcy order numbers distribution per region.



Figure 21: Unemployment in Czech Republic per region.

Regions / Numbers	Insolvency proceedings	Discharges	Bankruptcy Orders	Restructuring	Number of citizens	Unemployment (%)
Jihočeský kraj	6 314	3 041	805	5	636 459	6.17
Jihomoravský kraj	11 774	3 591	1 277	7	1 168 975	8.04
Karlovarský kraj	3 368	1 906	346	2	301 075	8.89
Kraj Vysočina	3 665	1 673	357	1	510 520	6.89
Královéhradecký kraj	7 289	4 535	877	2	438 527	8.02
Liberecký kraj	4 898	2 395	497	3	552 099	6.7
Moravskoslezský kraj	12 650	7 954	948	11	1 224 044	9.75
Olomoucký kraj	5 278	2 569	411	2	636 677	8.81
Pardubický kraj	4 564	2 773	446	2	515 804	6.59
Plzeňský kraj	5 703	3 240	661	3	572 859	6.02
Praha	10 728	1 840	1 970	7	1 246 176	5.12
Středočeský kraj	9 662	4 775	1 215	7	1 297 044	6.51
Ústecký kraj	12 896	8 399	1 045	7	826 037	11.4
Zlínský kraj	4 869	1 946	531	0	586 626	7.56

Table 9: Absolute numbers used to create maps shown in Figure 18, Figure 19, Figure 20 and Figure 21.

5.3. State Transitions

In this section we will examine the state transitions in the insolvency proceedings. First, we will examine the number of each possible transition and then we will consider also the time spent in each state. All possible states were described in Section 3.7. From Figure 2 we could conclude that there is a large number of states and that there are also many different ways of how an insolvency proceeding can move to another state. For this reason, we would like to simplify the diagram in Figure 2 so that the analysis will become feasible. Consequently, this will allow us to use more sophisticated methods for analysis such as the Bayesian Networks or the Market Basket Analysis.

5.3.1. Histogram Analysis

There are several general approaches available to simplify a graph of the above type(Figure 2). However, which approach is the best depends strongly on the actual application. Usually, we want to simplify the graph as much as possible while preserving most of the information we need. One of the simplest option available would be to remove nodes or edges that contain no or just little information we are interested in. In such a case we need to define a specific relevance measure that can be calculated for each node or each edge in the graph. Then, we can remove successively the least relevant nodes and edges until we reach an acceptable compromise between the graph simplicity and information preservation.

However, there are many cases when we cannot simply remove a node or an edge from the graph. In this case we may however, inspect the graph more closely and possibly replace a group of nodes by just one node. It is very common that in a given application a group of nodes might represent the same piece of information in each of its nodes and thus all of them may be redundant.

Another common problem might represent cycles in the graph. There are many algorithms that simply cannot deal with cyclic graphs. Because of that it would be often an advantage to remove them. Any cycle in the graph could be removed by assuming that the cycle allows only a finite number of passings. Essentially, this can be done by removing the edge closing the cycle and introducing new nodes representing new passings through the cycle. For each new passing we need to duplicate each node in the cycle and connect it to the the nodes representing the previous passing. Also, we need to make sure that each additional passing we create is optional, by creating an edge from the end of each passing to the rest of the graph. We are showing the whole process on Figure 22 and Figure 23 with allowing the example cycle two passings at most.

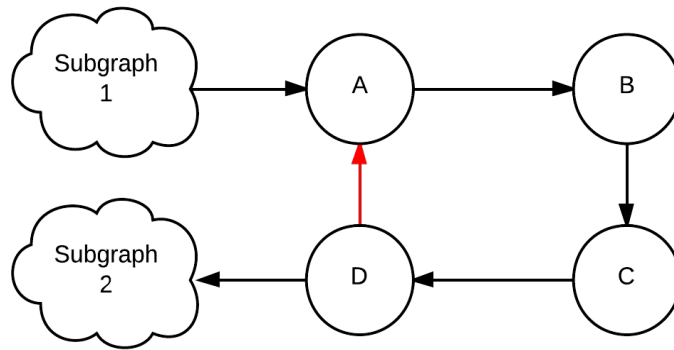


Figure 22: An example of a directed graph with a cycle. The edge marked red is closing the cycle.

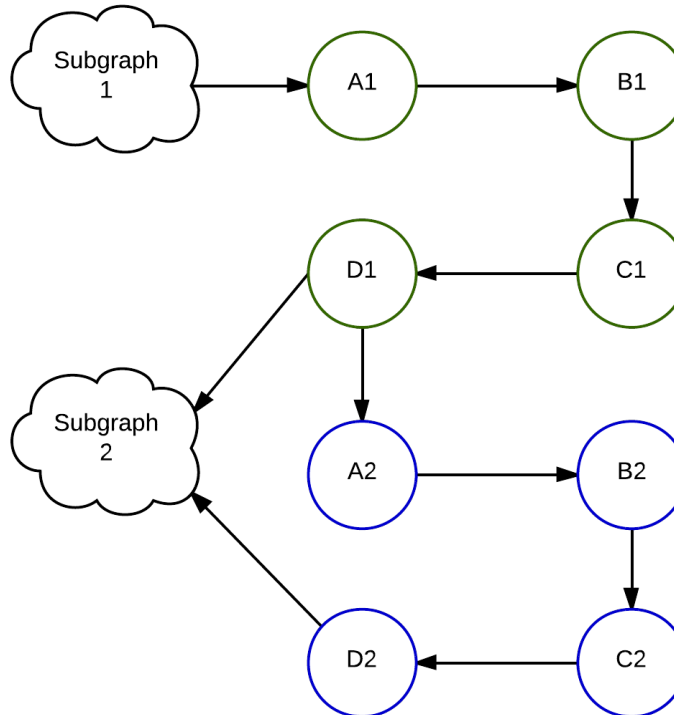


Figure 23: The same graph from Figure 22 but allowing only two passings (marked green and blue) at most through the cycle.

In the following paragraphs and sections we will show how each of these methods can be applied on our task represented by the graph shown in Figure 2.

We will start our investigation by analyzing the frequencies of each state transition. We expect to find some edges or even nodes from Figure 2 which are used

very rarely and therefore do not seem to be important for the upcoming more sophisticated analysis. As the graph in Figure 2 contains many edges, a classic bar chart histogram would not be very insightful. In this case we would like to determine nodes or even entire subgraphs that are only exceptionally entered by the insolvency proceedings. Therefore, we will divide the edges of the graph from Figure 2 into several categories. Each category will represent an interval of the occurrence frequency. The resulting graph with labeled edges can be found in Figure 24.

First of all we dropped the **Incorrect Entry** node, because this state has no legal meaning with respect to the Insolvency Act and is therefore not worth examining. Further, let us remind that to this date roughly 120 000 insolvency proceedings have been commenced. We can clearly see that the vast majority of transition edges are used quite rarely. To be precise less than 0.1% of all insolvency proceedings, corresponds roughly to tens of cases. Furthermore, there exist nodes that are connected to the rest of the graph only by such rarely occurring edges. Consequently, these nodes are also quite rarely entered and because of that, they are marked gray in Figure 24. One of these states is for example **Moratorium**⁷, which means that the debtors use the option of moratorium just very rarely. In our case, we decided to prune all the edges and nodes that are entered in less than 0.1% of all insolvency proceedings. Their number is just too small and statistically irrelevant for the purposes of this work. Furthermore, we decided to discard the node **Unresolved-Advanced** as well. This is because the **Unresolved-Advanced** state is entered before any important decisions are made in the insolvency proceeding and so the legal consequences of this state are not relevant.

Let us continue with simplifying the Figure 24 even further. We can clearly see that the states **Finished**, **Effective** and **Checked Off** are redundant for the upcoming analysis. For example the state **Effective** is often left out and the insolvency proceedings goes from **Finished** right to **Checked Off**. Also, the insolvency proceedings can go from the states **Finished** or **Checked Off** back to the state **Unresolved** or **Revived**. So consequently we could simply merge these three states **Finished**, **Effective** and **Checked Off** into one.

A similar effect can be observed for the state **Bankruptcy** which is quite often omitted and the insolvency proceedings go directly from the state **Unresolved** right

⁷ For details about **Moratorium** see Section 3.5.

to the states **Bankruptcy Order** or **Discharge**. Obviously, it would be reasonable to prune the state **Bankruptcy** without significantly changing the process of the insolvency proceeding as a whole. In such a case, we would then have to create new transitions from the state **Revived** to the states **Bankruptcy Order** and **Discharge**. A direct transition from **Revived** to **Discharge** is already present in the data even though it is not specified in the official documentation⁸. So technically we will only create one new transition edge from **Revived** to **Bankruptcy Order**. The transition from **Discharge** to **Bankruptcy Order** and vice versa will still be possible, so we will not lose any important information.

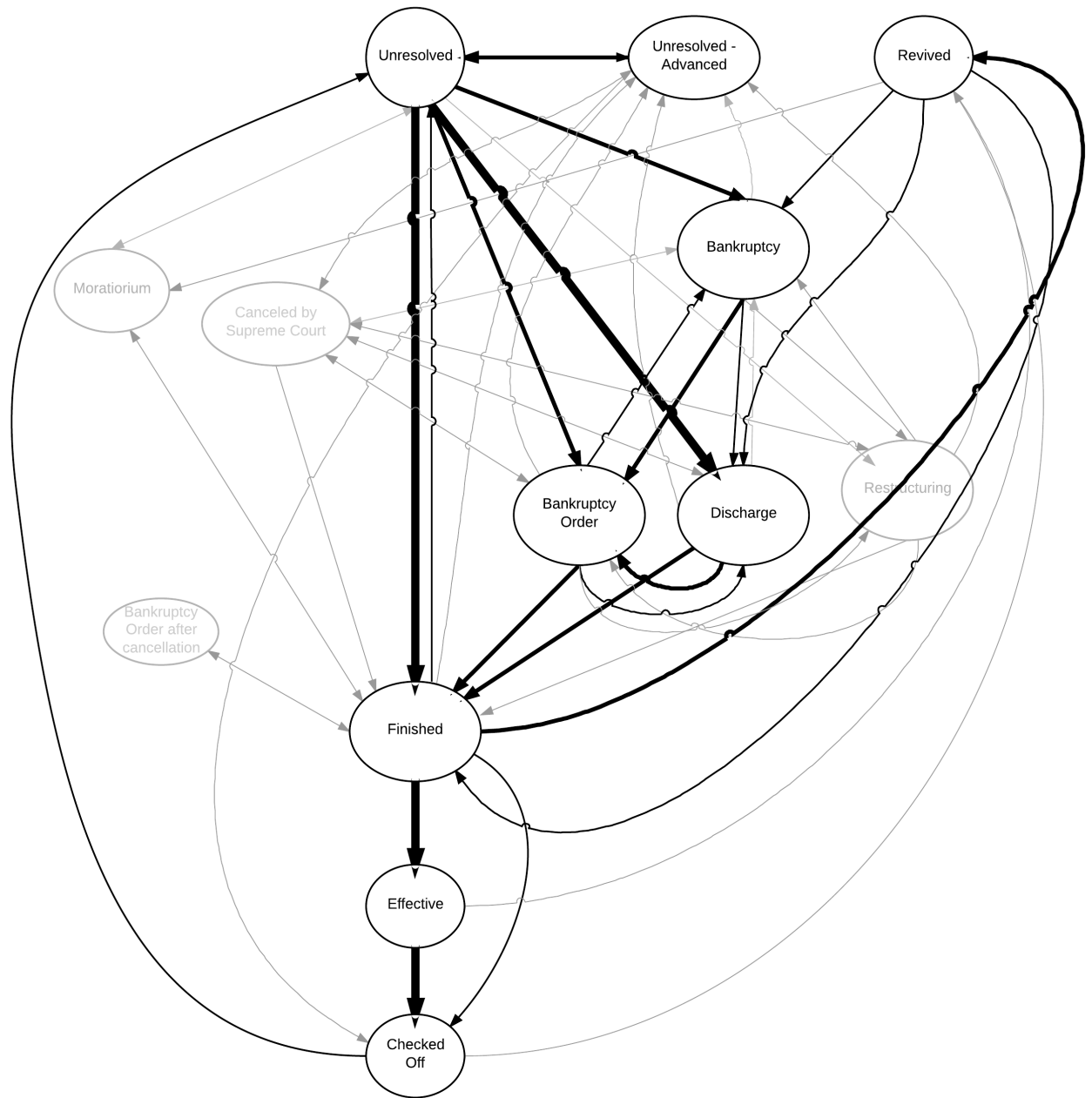
The final simplified transition model can be found in Figure 25. We can see a thick transition from the state **Unresolved** to the state **Discharge**, which occurs in 46% of all insolvency proceedings. But interestingly enough there is no such transition from the state **Discharge**. This is again a result of the insolvency proceedings being stuck in state **Discharge** right now because of the 5 years limit of discharge duration⁹. At the same time, only 1.5% of all insolvency proceedings were transferred from the state **Discharge** to the state **Bankruptcy order**, meaning that discharge method of resolution was unsuccessful. Similarly, 1.5% of all insolvency proceedings changed their state from **Discharge** to **Finished**. But in the latter case we still cannot tell whether the insolvency proceedings were successful or not because another 1.3% of insolvency proceedings moved from state **Finished** to state **Revived**. This means that the insolvency proceedings are starting over and might end up in **Discharge** or **Bankruptcy Order** once again.

We also decided to omit **Restructuring** from the state transition analysis because of its rare occurrence¹⁰ and from now on we will focus more closely on **Bankruptcy Order** and **Discharge**.

8 An example of an insolvency proceeding going from the state **Revived** right to the state **Discharge** is MSPH 89 INS 5348 / 2012.

9 For details about Discharge method of insolvency proceeding's resolution, see Section 3.3.3.

10 Restructuring occurred only in approximately 100 of insolvency proceedings since 2008. For details see Figure 16.





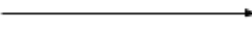

- Transitions that occurred in at least 10% of all insolvency proceedings 
- Transitions that occurred in 1% to 10% of all insolvency proceedings 
- Transitions that occurred in 0.1% to 1% of all insolvency proceedings 
- Transitions that occurred in less than 0.1% of all insolvency proceedings 

Figure 24: Insolvency proceedings state transitions frequencies.

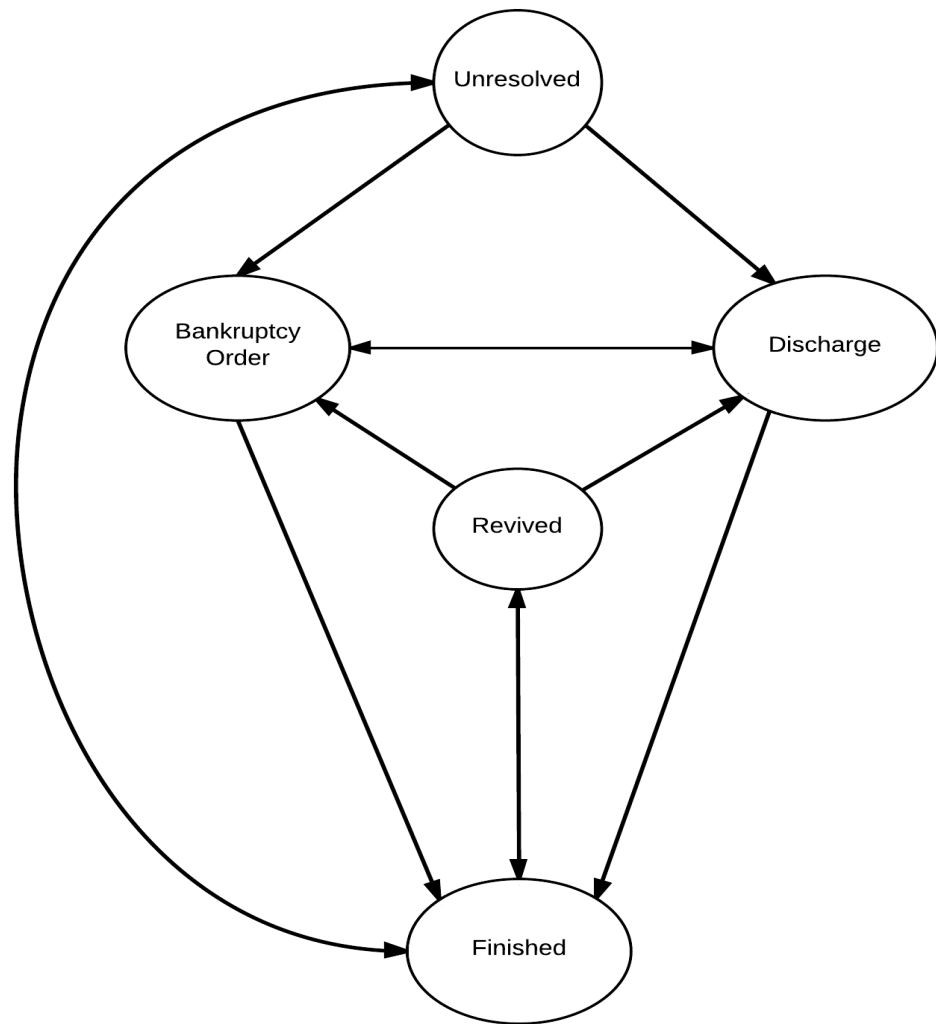


Figure 25: Model A, Simplified insolvency proceeding's state transitions.

From Figure 25 it is obvious that we managed to radically simplify the insolvency proceedings' state transition model without losing essential information.

The obtained transition graph is cyclic and has 5 nodes with 12 directed edges in total. This brings up an interesting question, whether the cycles really occur during the insolvency proceedings. Essentially, we would like to know how often the insolvency proceedings enter a specific state more than once. The answer to this question is summarized in Table 10, which shows that even though the repetitions are possible they do not occur very often. The initial and final states (Unresolved, Finished) are repeated the most times, but even they occur in only around 1.2% of all insolvency proceedings. This observation will be essential for the following analysis where we will need to deal with the graph's cyclicity.

State	1 - occurrence	Preceding state	2 - occurrences	Occurrence 3 and more times
Unresolved	130 821	Finished	1 631	176
Bankruptcy Order	14 737	Unresolved	225	114
		Revived	12	3
		Discharge	47	8
Discharge	63 544	Unresolved	648	72
		Revived	12	4
		Bankruptcy Order	101	13
Revived	1 702	Finished	121	10
Finished	51 262	Unresolved	2 245	203
		Revived	411	95
		Discharge	73	5
		Bankruptcy Order	269	43

Table 10: Insolvency proceedings state occurrence summary.

The last thing we will explore in this section is how much time the insolvency proceedings spent in each node of model A. There might exist a connection between how long an insolvency proceeding remains in one state and the state it will transfer to next. If it is the case, we can use this information in the upcoming analysis when we will try to predict the next states in the insolvency proceedings.

This time we will use classic bar chart histograms with months on the x-axis and transition counts on the y-axis. As it has been already mentioned there are together 12 directed edges in Figure 25, which is a lot and therefore we will show only the 5 transitions we found the most interesting. These are the transitions that decide whether the debtor is insolvent and if he is, then by which method the insolvency proceeding will be resolved. All 5 histograms can be found in Figure 26, where the red bar represents the average duration.

The first histogram from Figure 26 is for the transition from **Unresolved** to **Finished**, which occurs when the court rejects the insolvency petition and the insolvency proceeding ends right away. Surprisingly, this decision is usually done within the first month. We can thus assume that these cases are very straightforward. The second histogram shows transitions from **Unresolved** to **Discharge**. During that

period of time. the court decides that the discharge method of insolvency proceeding's resolution will fit the current case best and will result in creditors' full satisfaction. We can see that making this decision takes 1 to 3 months in general.

The next (third) histogram shows the duration distribution for transition from **Unresolved** to **Bankruptcy Order**. Similarly, as in the previous case in this period of time the court has to decide that **Bankruptcy Order** will be the best method of resolution above the other two (discharge, restructuring). We can see that this decision takes the most time so far, from 1 to 3 months. In all of the previous three cases, the distribution is very uneven and most of the transitions take a similar amount of time. This is however, not the case for the following two.

For example in case of the transition from **Discharge** to **Finished** we can see two trends. The first one starts within the 3-th month and continues to rise with its peaks around the 14-th and 22-th month. Then, the trend slowly starts to decline with the minimum at the 61-th month. However, in the second trend, the counts start rising again around the 65-th month. This is also the consequence of the 5 year limit for the discharge duration.

We will end the discussion with the last histogram for transition from **Bankruptcy Order** to **Finished**. This one doesn't show anything surprising and is very uniformly distributed with a slowly declining trend starting with the 20-th month. We can see that Bankruptcy Order can take from 3 to 30 months to finish. Just from the first 3 charts in Figure 26 we can deduce for example, that if an insolvency proceeding is in the state **Unresolved** for 2 months then there is a $18905 \div (18905 + 2002 + 8498) = 0.64$ probability that it will move to the state **Discharge**. The numbers in the formula simply represent the counts in the second bar of the first three charts in Figure 26. This illustration shows that simple observations like this truly have the potential to increase the prediction precision of the next state that the insolvency proceeding will move to very much.

In this sub-section we managed to significantly simplify the state transition model of the insolvency proceedings. Based on all the observations so far we were able to provide an overview of the ongoing insolvency proceedings. We have also shown that it might be worth of incorporating the transition durations in the upcoming analysis. All the observations so far will serve as a basis for the following two sub-sections.

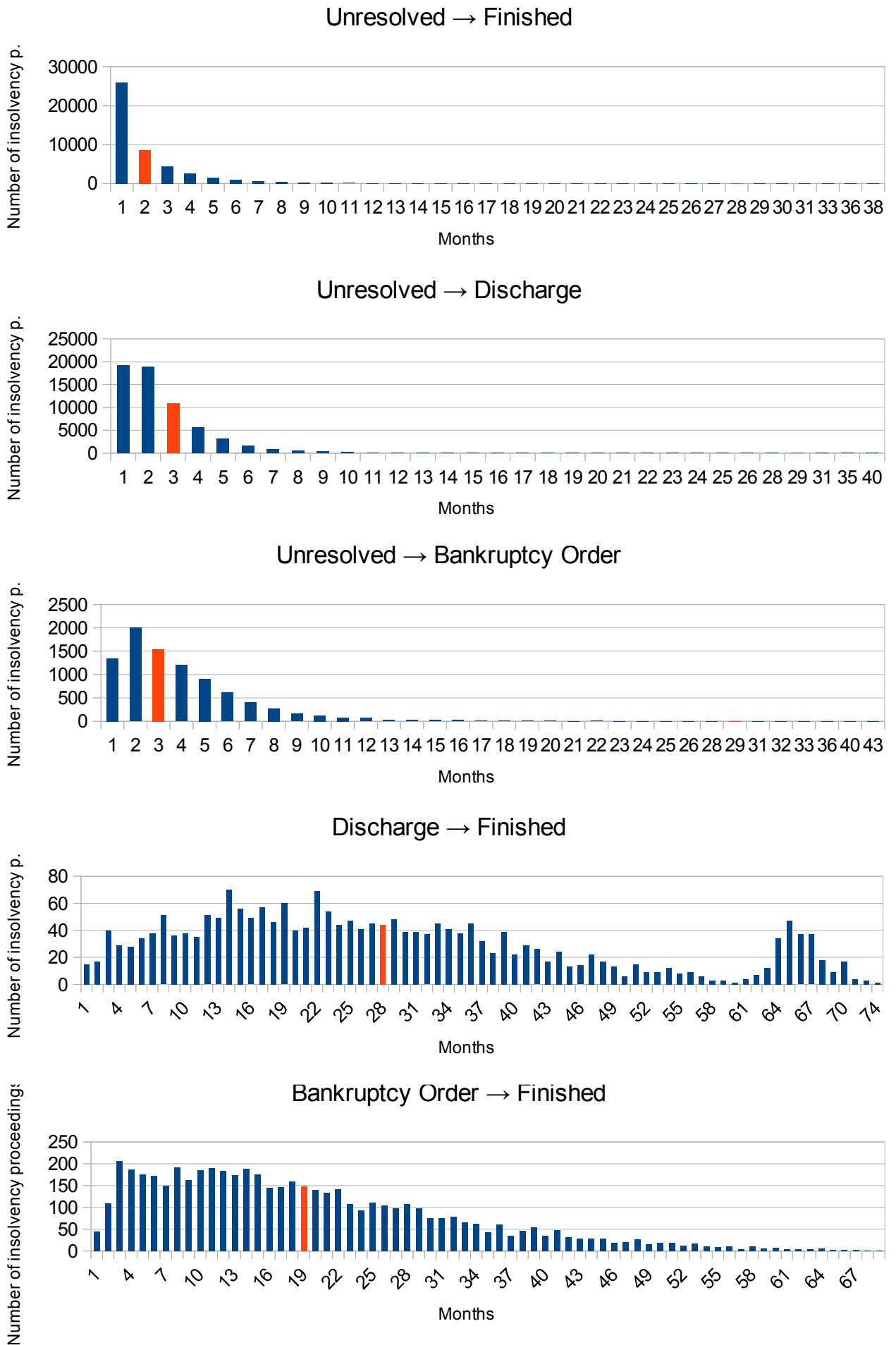


Figure 26: Insolvency proceedings' state transitions duration distributions. The column marked red represents the average duration.

5.3.2. Bayesian Networks

Bayesian networks [25][26] are directed acyclic graphs (DAG) that allow efficient representations of the joint probability distribution over a set of random variables. Further, the nodes of the graph need to have a topological ordering. A topological ordering of a directed graph is a linear ordering of its nodes so that every directed edge uv from node u to node v , u comes before v in the ordering. This is of course possible if and only if the graph has no directed cycles.

Each node in the graph represents a random variable, and each edge represents a direct influence of the outgoing variable on the incoming variable. Basically, the network encodes the following conditional independence relationship: each variable is independent of its non-descendants in the graph given the value of all its parents. This kind of independence is then exploited to reduce the number of parameters needed to estimate the joint probability distribution, and to efficiently compute the posterior probabilities. All probabilistic parameters can be encoded by means of a set of tables, one for each variable, that expresses the considered conditional distribution of variables given their parents.

Efficient algorithms exist that perform inference and learning in Bayesian networks [25][27]. There are three main inference tasks for Bayesian networks: inference for not observed variables, parameter learning, structure learning.

Because a Bayesian network is a complete model for the variables and their relationships, it can be used to answer probabilistic queries about them. For example, the network can be used to find out updated knowledge of the state of a subset of variables when other variables (evidence) are observed. This process of computing the posterior distribution of variables given some evidence is called probabilistic inference and will be used in this thesis.

In regards of learning the Bayesian networks, one needs to specify both, the graph topology (structure) and the parameters of all the conditional probability distributions considered. It is possible to learn both of them from the data provided, however, learning the network structure is more complex than learning the parameters. When some nodes are hidden or there are missing data, also the learning process is more difficult than in the case when everything is observed. Consequently, this results into 4 cases based on whether the structure is known or not and whether the data is fully or just partially observed. In each case, of course, a different

approach must be used. However, the analysis of the insolvency proceeding's state transitions represents the simplest case when both the structure is known and all data are fully observed as well. The structure of the Bayesian network will copy the structure of the simplified state transitions model from Figure 25 to a large extent. Obviously the graph shown on Figure 25 is breaking some assumptions which are required by the Bayesian networks such as acyclicity, but these issues will be addressed later in this section.

When we assume that all the data were fully observed, then the simplest method of learning conditional probabilities can be used and that is learning the probabilities directly from the data. Another more sophisticated and most common method is called Maximum likelihood estimation(MLE)[28].

The goal of MLE is maximizing the (log) likelihood of the training dataset. Let us assume that the dataset $T = \{x_1, \dots, x_m\}$ contains m mutually independent cases, where $\vec{x}_l = (x_{l1}, \dots, x_{ln})^T$. Then let us define a parameter set $\Theta = (\theta_1, \dots, \theta_n)$ where θ_i is the vector of parameters for the conditional distribution of a random variable X_i (represented by one node in the graph). The set of parents of x_i will be denoted as π_i . Then, the log-likelihood of the training dataset then can be defined for each node as:

$$\log L(\Theta|T) = \sum_{i=1}^n \sum_{l=1}^m \log(P(x_{li}|\pi_i, \theta_i)) \quad (1)$$

The log-likelihood function decomposes according to the graph structure, basically one can maximize the contribution to the log-likelihood function of each node independently. Let each discrete variable x_i have r_i possible states with probabilities

$$p(x_i = k | \pi_i = j, \theta_i) = \theta_{ijk} > 0 \quad (2)$$

where $k \in \{1, \dots, r_i\}$, j is the state of x_i -th node parents and $\theta_i = \{\theta_{ijk}\}$ specifies the parameters of the multinomial distribution for every combination of π_i . Given the training data D , the MLE of $\{\theta_{ijk}\}$ can be computed as

$$\theta_{ijk} = \frac{N_{ijk}}{N_{ij}} \quad (3)$$

where N_{ijk} is the number of cases in D in which $x_i=k$ and $\pi_i=j$ and

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} .$$

In case of the known structure and partial observability, when some of the nodes are hidden, we can use Expectation Maximization (EM) algorithm to find a locally optimal MLE of the parameters. The basic idea behind EM is that, if we knew the values of all nodes, learning (the M step) would be easy as described above. Because of that, in the E step, we compute the expected values of all nodes using an inference algorithm, and then treat these expected values as if they were observed.

The third scenario when the structure of the Bayesian network is not known but the data is fully observed, the goal is to learn a DAG that best explains the data. This problem however is NP-hard, since the number of DAG-s on N variables is super-exponential in N . One approach is to proceed with the simplest assumption that the variables are conditionally independent given a class, which is represented by a single common parent node to all the variable nodes. Consequently, the obtained structure corresponds to the naïve BN, which surprisingly is found to provide reasonably good results in some practical problems.

When both the structure is not known and the data is only partially observable, one has to marginalize out the hidden nodes as well as the parameters. Since this is often intractable, usually an asymptotic approximation to the posterior called *Bayesian information criterion* is used. In this case one considers the trade-off effects between the likelihood term and a penalty term associated with the model complexity.

In the following 2 sub-sections we will describe exactly how the Bayesian network model will be used to analyze insolvency proceedings' state transitions. We will introduce two models. The first one will be a simple model focusing just on the state transitions and the second will be a natural extension of the first one where we will also take into account the time spent in each node into consideration.

5.3.2.1. Simple Model

The Bayesian network model for the insolvency proceedings' state transitions will be mostly based on the graph in Figure 25. We will simply start by adopting all the nodes from Figure 25. Unfortunately, we cannot adopt the edges just as simply,

because the graph in Figure 25 is cyclic and the Bayesian networks have to be acyclic.

To be able to create an acyclic graph for our model, we need to make some more simplifications. Let us revisit Table 10 and the states' repetitions counts. From Table 10 we can see that 3 or more occurrences of the same state are very rare and occur in less than 0.2% of all insolvency proceedings. Thus, we can start the simplification by assuming that any insolvency proceeding may enter the same state at most 2 times. Now we have to identify which transitions are causing the graph to be cyclic. These transitions are: from **Finished** to **Unresolved**, from **Finished** to **Revived** and finally from **Discharge** to **Bankruptcy Order** and back.

The next step is to modify our graph so that it will become acyclic but still, we will not lose much information. With regard to the above assumption, that each state may occur at most 2 times, we will start by removing the transition from **Finished** to **Unresolved**. Now we will duplicate our graph and name all the copied nodes with superscript 2. We will refer to this subgraph as the 2-subgraph from now on. Next we connect the original graph with the 2-subgraph by a directed edge from **Finished** to **Unresolved²**. This edge now represents the deleted transition from **Finished** to **Unresolved**. Lastly, we will delete the state **Revived²** with which we will deal later. So now we allow the insolvency proceeding to start over once more and go over the states again. But it has to end in the **Finished²** state and cannot go back anymore.

In case of the transition from **Finished** to **Revived**, we will proceed in a similar way, but at first we will remove the transitions from **Revived** to **Bankruptcy Order** and **Discharge**. Once again, we duplicate the original graph, but without the state **Revived** and name all the new nodes with superscript r. We will refer to this subgraph as the r-subgraph from now on. Finally, we create two directed edges from the state **Revived** to the states **Bankruptcy Order^r** and **Discharge^r** that are the replacement for the two we deleted before.

Now an insolvency proceeding may go from the state **Finished** to the state **Revived** and then to the subgraph with r-nodes or to state **Unresolved²** and then to the subgraph with 2-nodes. To finalize these changes we have to cover the most important case and that is when the insolvency proceeding ends in the state **Finished**. To represent this case explicitly we create a new node denoted as **End** and create a directed edge from the state **Finished** to it.

Lastly, we have to deal with the transition from state **Discharge** to state **Bankruptcy Order** and back. To deal with this cycle we create two more nodes named **Discharge after Bankruptcy Order** and **Bankruptcy Order after Discharge**. We will also create two more directed edges from the two new nodes to the node **Finished**. Next, we delete the cycle between **Discharge** and **Bankruptcy Order** and finally we create directed edges from **Discharge** to **Bankruptcy Order after Discharge** and from **Bankruptcy Order** to **Discharge after Bankruptcy Order**. We will of course make the same changes in the r-subgraph and the 2-subgraph as well. This way we have removed the cycle but covered both cases when the insolvency proceeding moves from the state **Bankruptcy Order** to **Discharge** or vice versa. Again, to simplify the model we assume that once the insolvency proceeding moves from state **Bankruptcy Order** to state **Discharge** it cannot go back to the **Bankruptcy Order** again. For the opposite transition we have the same assumption.

The final model denoted B with all changes is shown in Figure 27. It is important to state that we are able to apply model B on 99.8% of all insolvency proceedings. The only insolvency proceedings on which this model cannot be fully applied are the ones which were resolved by restructuring or started over for the third time or more. But these two cases only represent less than 0.2% of all insolvency proceedings.

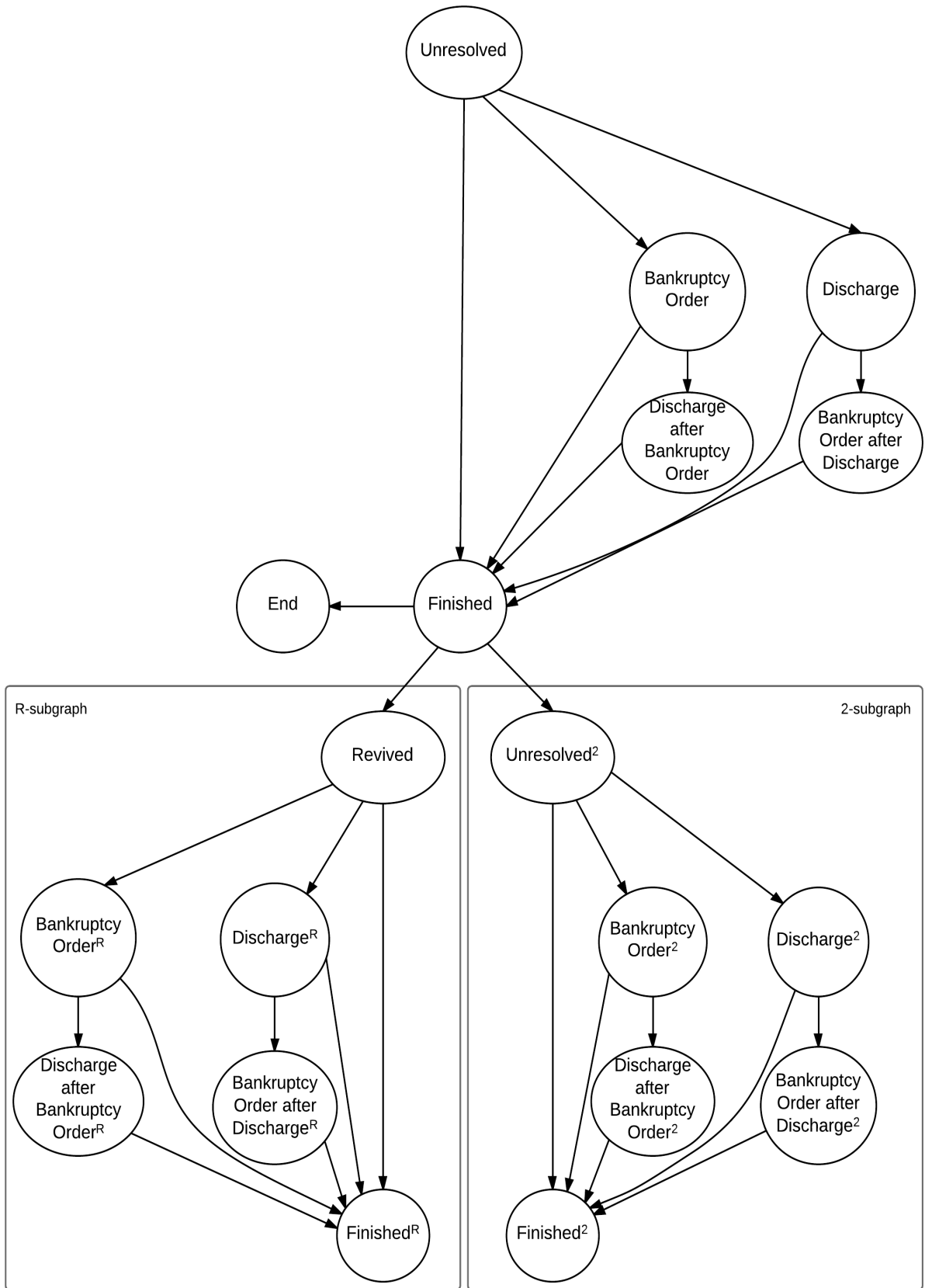


Figure 27: Model B, insolvency proceeding transitions' Bayesian network model.

To learn this model we used an open source machine learning toolkit named Weka[29], which fully supports modeling and learning of Bayesian networks. We exported the state transitions from all insolvency proceedings (around 120 000) and transformed them to model B. In addition, we stored the result as a CSV file where each column represented one node in model B and every line represented one insolvency proceeding. The column contained a **yes** value or a **no** value based on whether the insolvency proceeding passed through the respective node or not.

Then, we re-created model B in Weka's Bayes Network Editor[29]. After that we finally inferred the network parameters based on the data in the exported CSV file. The results obtained from both tested approaches (direct estimation of probabilities and MLE), were identical. The results are presented in Figure 28 below, where we show the probabilities of moving to the next node for each node of model B.

Please note that we are showing only direct probabilities of the transitions to the following nodes, which means that we do not consider all previous states. The later approach to the visualization of Bayesian networks is supported by the Weka Bayesian Network Editor, that also provides an easy way to compute the exact probabilities for each case.

There are several interesting observations which can be noticed in Figure 28. For example there is a 44% probability that the debtor is not considered insolvent and the insolvency proceeding finishes right away. We can also deduce that if this happens then right now there is just a very small probability(5%) that the insolvency proceeding starts over either by moving back to state **Unresolved** or **Revived**. It could be interesting to investigate these insolvency proceedings further, because these debtors might rather be victims of groundless charges of being insolvent with the aim to damage the debtors name. Moreover, it would be interesting to know how many of those 44% really are potentially damaging accusations. Sadly, there is no easy way of distinguishing these cases.

The majority of the insolvency proceedings are resolved by discharge (47%) and only 9% by bankruptcy order. This shows that discharge is the most acceptable solution for the creditors and the debtors which are natural persons. Now we would like to highlight the huge difference between the probability of the insolvency proceeding to move from **Bankruptcy Order** to **Discharge** and vice versa. The first mentioned is just around 2.6%, however the probability of moving from **Discharge**

to **Bankruptcy Order** is 41%. Consequently, this means that 41% of the discharges that were finished so far were not successful and that the creditors demanded resolution by bankruptcy order after that.

We also know that because of the 5 year limit for discharges that they only started expiring in 2013. There is a large probability that the 41% ratio will change overtime. Approximately 3% of insolvency proceedings start over through the state **Revived** and approximately 2% through the state **Unresolved²** . It is interesting how much the probabilities differ in the corresponding subgraphs. For example there is a 90% probability that a insolvency proceeding moves from **Unresolved²** to **Finished²** but only a 44% that it moves from **Revived** to **Finished^R**.

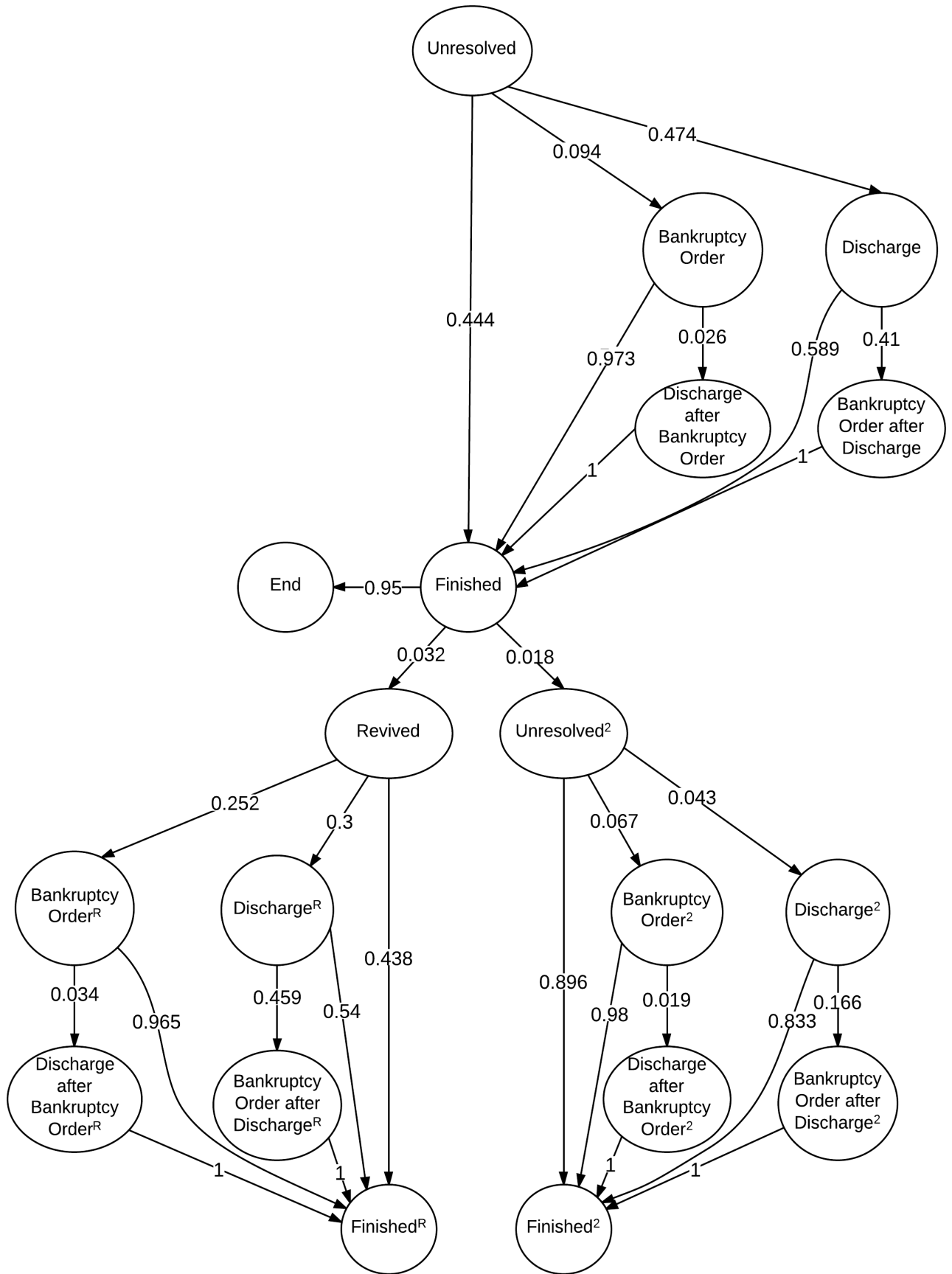


Figure 28: Model B with estimated probabilities by Bayesian networks.

5.3.2.2. Incorporating The Time Spent Information

In the previous section we have introduced and inferred a Bayesian network model which gave us a good overview of how the insolvency proceeding is moving between states. Now we will modify this model so that it will also contain information about the time spent by the insolvency proceedings in each state. This means that we will not only know which transition will come next, but could also estimate how long the insolvency proceeding will remain in that state.

In Figure 26 we have shown, that an insolvency proceeding may remain in one state from few days (e.g. state **Unresolved**) to several years (e.g. state **Discharge**). Since the time spent in a state is a continuous variable and we want to keep our model simple, we will divide these times into 3 groups: short, medium and long. We will set the intervals of these groups for each node differently. Furthermore, we will select the intervals so that each group will cover approximately the same number of insolvency proceedings. This is because, as we have seen in Figure 26, the time spent in most of the nodes does not tend to be uniformly distributed.

For the above reason, each node in our Bayesian network will be replaced by 3 nodes, each of them representing one time group. As a base for these changes we could directly use the network from model B. However, using the first part of the network from model B, to be specific only the subgraph ending in state **Finished** and excluding the r-subgraph and 2-subgraph will do just as fine. Both omitted subgraphs are indicating the fact that the insolvency proceeding is starting over. We can simply represent the second iteration as a new pass through our Bayesian network. To explore the time spent in the **Revived** state, we will keep this node, but it will have no directed edges coming from it. The final model (named C) is shown in Figure 30, where the nodes colored blue will be replicated three times, each representing one time group: short, medium, long.

Note that we also removed the node **End** because it does not represent any state of the insolvency proceeding so it does not make sense to study the time spent in this node. To avoid starting the insolvency proceeding from 3 possible states we created a new node **Start** that represents the new start of the insolvency proceeding and is connected to the 3 respective states **Unresolved**.

In case of learning model C we proceeded very similarly like in case of learning model B. Using Weka's Bayes Network Editor [29] we recreated model C and

exported the required data from the database. Since in this case we are focusing mainly on the time course of the insolvency proceedings, we decided to only export data which included finished insolvency proceedings (approximately 54 000). For the active insolvency proceedings we don't have complete informations about the time spent for each state and including them would only make the predictions less accurate.

It may not be obvious from Figure 29, but the resulting model is very large, it contains 19 nodes and 93 directed edges. For that reason we are not presenting the complete model, but only the most important part for this research which deals with the insolvency proceeding's resolution method. The result is shown in Figure 31, where the intervals under the name of the nodes indicate the insolvency proceeding's time spent (in days) in the corresponding state. For example the node **Unresolved [11, 47]** represents insolvency proceedings that were in state **Unresolved** from 11 to 47 days before they moved to any other state.

The probabilities from the node **Start** to the nodes **Unresolved** are uniform as expected, based on how we selected the intervals. Next, the probabilities from all three **Unresolved** nodes to the **Finished** node representing the interval 5 to 57 days are very similar (around 0.75). Consequently, most of the insolvency proceedings end between the first and eighth week after entering the state **Finished**. Similarly, the probabilities to the state **Finished** within 5 days (0.4) and in at least 57 days (0.12) are very similar as well. This shows that moving to the state **Finished** is almost independent of the time spent in node **Unresolved**.

One obvious observation regarding states bankruptcy order and discharge is that the probabilities of moving to them are very small in comparison to moving to the state **Finished**. This is due to the fact that we only consider finished insolvency proceedings. Most of all insolvency proceedings (48%) at the time of writing were in the state **Discharge**, so we did not include them in the data we exported. Bankruptcy order was used in only 11% of all insolvency proceedings, so in comparison to the number of insolvency proceedings that moved from the state **Unresolved** directly to the state **Finished** it leads to such small probabilities. Nevertheless, we can see that there is a very small (less than 0.01) probability that an insolvency proceeding moves to **Bankruptcy Order** within the first 11 days in the state **Unresolved**. This probability increases for insolvency proceedings which remained in the state

Unresolved from 11 to 47 days to 0.023. Similarly, the probability of moving to the state **Bankruptcy Order** increases even more (from 0.04 to 0.12) if the insolvency proceeding remains in state **Unresolved** for more than 47 days. From this observation we can assume that the decision whether a insolvency proceeding will be resolved by bankruptcy order takes quite some time and usually at least 47 days.

On the other hand the probabilities of moving to the states **Discharge** have quite different distributions. For example the largest probability (0.011) is for moving from the state **Unresolved - short** to the state **Discharge – long**. The same applies for moving from **Unresolved – medium** to the state **Discharge – long** with the probability 0.03. But, in case of the state **Unresolved – long** the largest probability (0.017) is for moving into state **Discharge – short** and the second largest (0.014) is for moving to the state **Discharge – medium**. From this we could deduce that the time spent in the state **Unresolved** may have a large influence on whether the discharge will be successful or not. Please notice that states **Discharge - short** and **medium** represent discharges that were finished in approximately 2.5 years. There are two scenarios which could explain these cases. The first could mean that the debtor managed to repay his debts to the creditors and then the insolvency proceeding moved to the state **Finished**. The second however, that the creditor might have stopped paying his monthly installments and the insolvency proceeding moved to the state **Bankruptcy Order**.

Even though Figure 31 does not show the probabilities of moving to the next states we learned that the probability of moving from the state **Discharge – short** to the state **Bankruptcy Order After Discharge (BAOD)** is 0.47. In contrast the probability of moving from **Discharge – medium** to **BAOD** is only 0.19 and from **Discharge – long** only 0.08. This could really lead us to the conclusion in the previous paragraph that the time spent in node **Unresolved** may have a positive impact on whether the discharge will or will not be successful.

Please note that in the data for building this model we included only around 3000 insolvency proceedings which moved to the state **Discharge**. This represents only 4% of all discharges that were started but not finished yet. Consequently, the conclusions of this section might not be accurate and might change in the near future. As more and more discharges will finish our model will give more accurate probabilities. To be able to analyze the time course of discharges we need more data

which would provide us with definitive conclusions. Because of that we will leave this problem open.

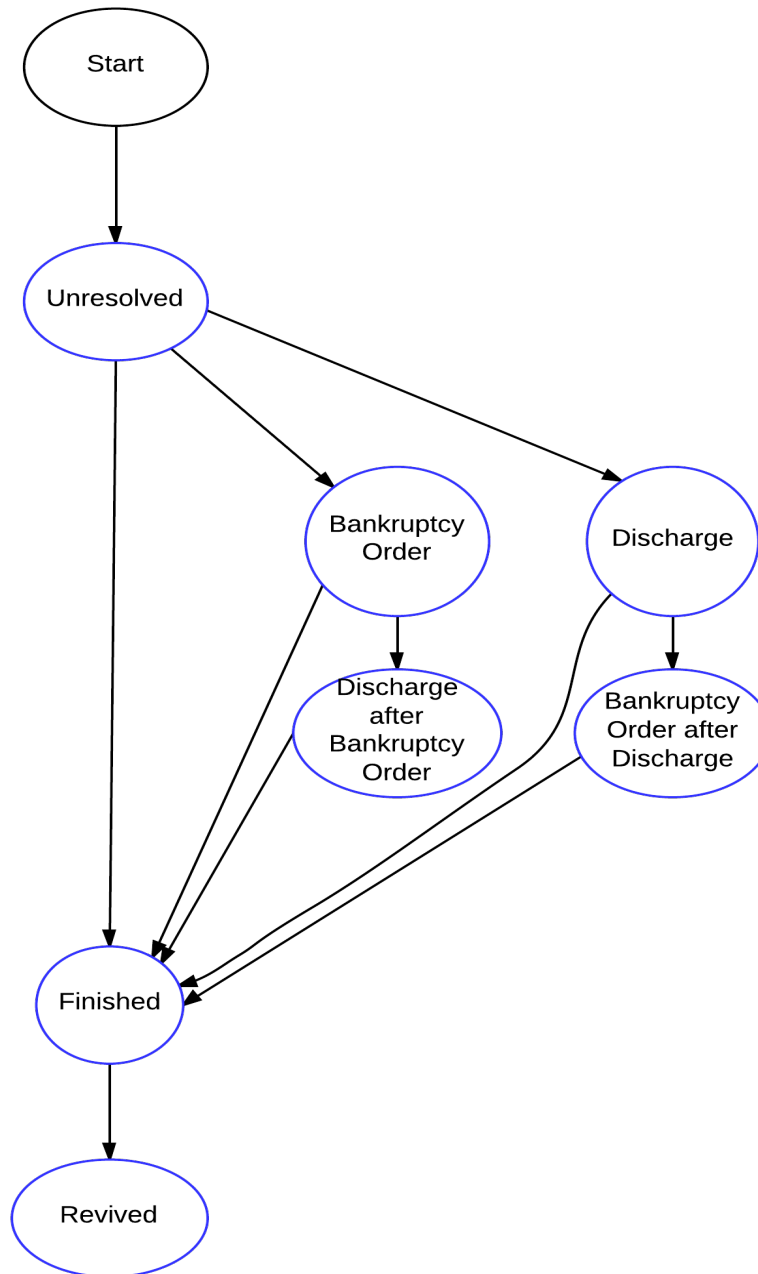


Figure 30: Model C, Bayesian network for analyzing how much time insolvency proceedings spend in each state. The nodes marked blue are replicated three times, each replication representing one time spent group: short, medium and long.

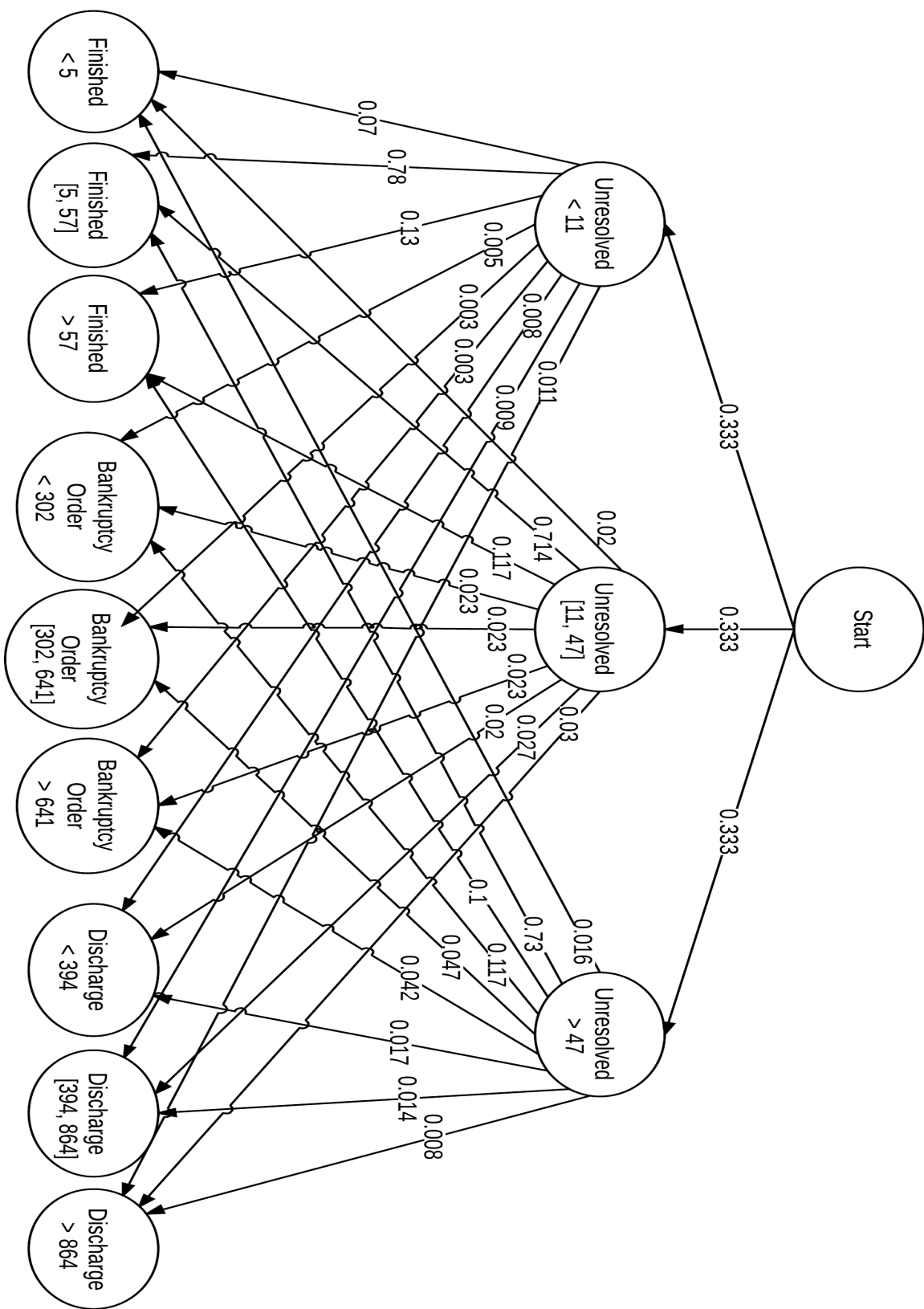


Figure 31: Model C with estimated probabilities by Bayesian networks. The number below the nodes' names indicate the time intervals the corresponding nodes represent.

5.3.3. Market Basket Analysis

Market basket analysis (MBA [30]) deals with the problem of discovering association rules between items in a large database of sales transaction. A concrete example of an association rule in the context of retail business may be: $[butter, milk] \rightarrow [bread]$ which indicates that if a customer buys butter and milk together then he or she is likely to buy also bread.

Almost every retail organization these days stores massive amounts of sales data, which are referred to as the market basket data. One record usually consists of the transaction date and the items bought in the transaction. These data are of course a very important part of the marketing infrastructure. Basically, they enable the marketers to develop and implement customized marketing strategies which are based exactly on the personalized customers needs. The problem of mining association rules from the basket data was firstly introduced in [31]. An example of a mined association rule may be that, if a customer buys bread then in 98% of cases he also buys milk. These findings then may be used for cross marketing, attached mailing applications, catalog design, store layout and much more.

The problem of MBA can be formalized as it follows. Let $I = \{i_1, \dots, i_m\}$ be a set of items and D be a set of transactions where each transaction T is a set of items such that $T \subseteq I$. With each transaction a unique identifier called TID is associated. Consequently, an association rule is an implication $X \Rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ has confidence c if $c\%$ of transactions in D that contain X also contain Y . Similarly, the rule $X \Rightarrow Y$ has support s if $s\%$ of transaction in D contain $X \cup Y$. Thus, the problem of mining association rules consists from generating all association rules, that have support and confidence greater then specified by *minsupport* and *minconfidence*. There are several algorithms designed for these types of tasks, such as AIS[31], SETM[32] and FP-tree[33]. However, one of the simplest and most common for this task is the Apriori algorithm proposed in [30]. That is also why the Apriori algorithm was also adopted for the purposes of this work. The notation that will be used in the following section is described in Table 11. Next we will describe the Apriori algorithm in detail.

k-itemset	An itemset having k items.
L_k	Set of k-itemsets (those with minimum support). Each member of this set has two fields: itemset and support count.
C_k	Set of candidate k-itemsets. Each member of this set has two fields: itemset and support count.
\overline{C}_k	Set of candidate k-itemsets when the TIDs of the generating transactions are kept associated with the candidates.

Table 11: Apriori algorithm notation

The algorithms for discovering the relevant itemsets make multiple iterations over the data. In the first iteration they count the support and confidence of individual itemsets and determine which of them have the minimal required support and confidence (are frequent enough). Each of the following iteration starts with the itemsets found in the previous iteration. These itemsets are then used for generating new potentially larger itemsets, called candidate itemsets.

The Apriori algorithm generates the candidate itemsets to be counted in an iteration by using only frequent enough itemsets found in the previous iteration, without considering the transactions in the database. The basic intuition behind this is that any subset of a frequent itemset must be frequent as well. Consequently, the candidate itemsets having k items can be generated by joining frequent enough itemsets having $k-1$ items and removing those that are not frequent enough. Therefore, the algorithm generates much less candidate itemsets and improves its performance significantly. The exact description of the Apriori algorithm is below.

Apriori

Input:

items ... set of all items which can occur in a transactions

transactions ... set of transactions based on which the association rules will be generated

minSupport ... minimal required support of each generated rule

minConfidence ... minimal required confidence of each generated rule

Step 1. Initialization

Initialize a new set L_1 containing one itemset for each item in *items*, that has a *confidence* larger than *minConfidence* and *support* larger than *minSupport*.

Initialize a new variable $k = 2$.

Step 1.1 Candidate generation

Create the candidate set C_k as follows:

```
insert into  $C_k$ 
select  $p.item_1, p.item_2, \dots, p.item_{(k-1)}, q.item_{(k-1)}$ 
from  $L_{(k-1)} p, L_{(k-1)} q$ 
where  $p.item_1 = q.item_1, \dots, p.item_{(k-2)} = q.item_{(k-2)}, p.item_{(k-1)} < q.item_{(k-1)}$ 
```

Set the counter $c.count$ of each candidate $c \in C_k$ to 0.

Step 1.2 Pruning the candidates set

Delete all itemsets $c \in C_k$ such that some $(k-1)$ subset of c is not in $L_{(k-1)}$.

Step 1.3 Updating the candidates' counters

For each t in *transactions* get the set of candidate itemsets C_t which are a subset of t .

For every candidate c in C_t increase its counter $c.count$ by one.

Step 1.4 Filtering the candidates

Calculate the *support* and *confidence* of each candidate c in C_t .

Create a new set L_k containing all candidates c from C_t which have *support* larger than *minSupport* and *confidence* larger than *minConfidence*.

Step 1.5 Loop condition

Increase k by one.

If $L_{(k-1)}$ is not empty go to **1.1**:

Step 1.6 Returning the result

Return the union of each generated set of itemsets: $\cup_k L_k$

5.3.3.1. Simple Model

In this section we will describe the insolvency proceedings' transitions and states by means of a MBA model. MBA models work on a set of items and transactions. Furthermore, the transactions represent various subsets of the items set. To be able to apply this approach we will think of insolvency proceeding states as items and the

respective insolvency proceedings as different transactions. Similarly as in the case of Bayesian networks¹¹ the standard model cannot cover the cyclicity of the transition's model. This is a simple consequence of using sets to represent the transitions, which gives no possibility to encode a cyclic passing through a subgraph.

However, we have successfully dealt with the cyclicity problem in previous section and we came up with a solution shown in Figure 27. In this graph we have no cycles and therefore were able to use the Bayesian network model successfully. Additionally, we can take this representation and encode it by using items and transactions also for the MBA model. For every node in Figure 27, we will create one custom item. Each insolvency proceeding will be then represented by one transaction which may contain the state item exactly once or not at all. So basically if we would take an insolvency proceeding which passed through the graph in Figure 27 as follows: **Unresolved** → **Finished** → **End**, then we would represent it by the transaction: [**Unresolved, Finished, End**].

Lastly, we will perform one more optimization that consists of replacing each node with superscript R by a corresponding node with superscript 2. By doing so we will not lose any information and we will reduce the number of required items to represent model B from 21 to 16. We can carry out this optimization safely because we know that in model B the insolvency proceedings go through either the state **Revived** or **Unresolved**² but never both. Consequently, if a rule contains the item representing the state **Revived** and e.g. **Discharge**², we now that this rule is representing the path in the r-subgraph of model B. Naturally, the same applies for the 2-subgraph.

Notice that the same optimization cannot be performed in case of the Bayesian networks without losing their accuracy. By doing so we could not distinguish between the paths in the R-subgraph and 2-subgraph. This would happen simply because the probability of moving from both states **Revived** and **Unresolved**² would be equal to one.

The Weka machine learning toolkit[29] we used for training the Bayesian networks can be used also for MBA tasks and includes the Apriori algorithm. However, we found out that the implementation of Apriori in Weka is not very efficient and consumes too much computing memory and time. Because of that we

¹¹ See Section 5.3.2.1.

have had to look for an alternative implementation. We found and used a tool called The Sequential Pattern Mining Framework (SPMF) [34] with a very efficient implementation of the Apriori algorithm.

The SPMF toolkit requires an input format which consists of one file in which each line represents exactly one transaction. Furthermore, the transaction is an ordered set of integers representing different items. We numbered the nodes from 0 to 15 and exported the data from the database in the required format. Before starting the Apriori algorithm it is necessary to set additional two parameters and that is the minimal required confidence and the minimal required support.

To test the performance of the system, we started by setting confidence to 0.1 and support to 0.1. The results returned by SPMF were in the order of seconds so we kept decreasing the values and finished at both confidence and support equal to 0.0001. Even with this small parameters SPMF returned the value in less than 10 seconds which we found very impressive in comparison with the Weka's implementation. Weka was able to return the same results in order of minutes and in case of confidence and support set to 0.0001 it did not return the results even after 60 minutes. We set the parameters to such small values in order to obtain the rules for most of the possible paths in model B. With the last mentioned parameters SPMF generated approximately 3500 rules.

The application of the MBA model and the Apriori algorithm in particular was very appropriate. Even though there are together $2^{16} - 1$ possible itemsets (excluding the empty set which is not a valid itemset) Apriori managed to efficiently filter out most of them. This was possible thanks to the fact that most of the possible itemsets do not represent any path in model B and thus have support 0 and are filtered out almost right away. However, there is also a disadvantage in using MBA for this type of task. The rules with very little support get ignored and so we are unable to get the probability for each path. We could fix this by lowering the support even more but the more we lower the support the more rules we get. Possibly also rules that we are not interested in. Consequently, it is necessary to find a balance between the number of generated rules and the model's coverage. In this case we only have had 16 possible items so this problem was not very noticeable, but as the number of possible itemsets grows exponentially with the number of items, it sure could be a problem for larger models.

In Figure 32 we are showing the results in the same fashion as in Section 5.3.2.1 so that we can directly compare the MBA model with the Bayesian networks model. We can clearly see that the results are almost identical with the results obtained by the Bayesian networks, especially to the state **Revived** and **Unresolved**². The difference is less than 0.01 most of the time, the exception with the largest difference (0.03) is the transitions from **Unresolved** to **Finished**.

However, in Figure 32 there are also transitions with 0 probabilities which did not occur in Figure 28. This is because the rules representing these transitions have very small support and are ignored by the Apriori algorithm. For these cases we provide the maximum error that could have been neglected based on the support of the rules representing the previous nodes.

Notice that only around 5 % of insolvency proceedings go to either state **Revived** or **Unresolved**². Then as the process of insolvency proceedings gets spread over the following possible states, the transitions with 0 probability get a support less than 0.0001 (in order of tens of cases). In comparison with figure Figure 28 the error caused by this effect is less than 0.04 in case of transitions from **Bankruptcy Order** to **Discharge after Bankruptcy Order** in both the R-subgraph and 2-subgraph. Nevertheless, the largest error (0.17) occurs in case of the transition from **Discharge**² to **Bankruptcy Order after Discharge**², this is however still much smaller than the maximum error we estimated.

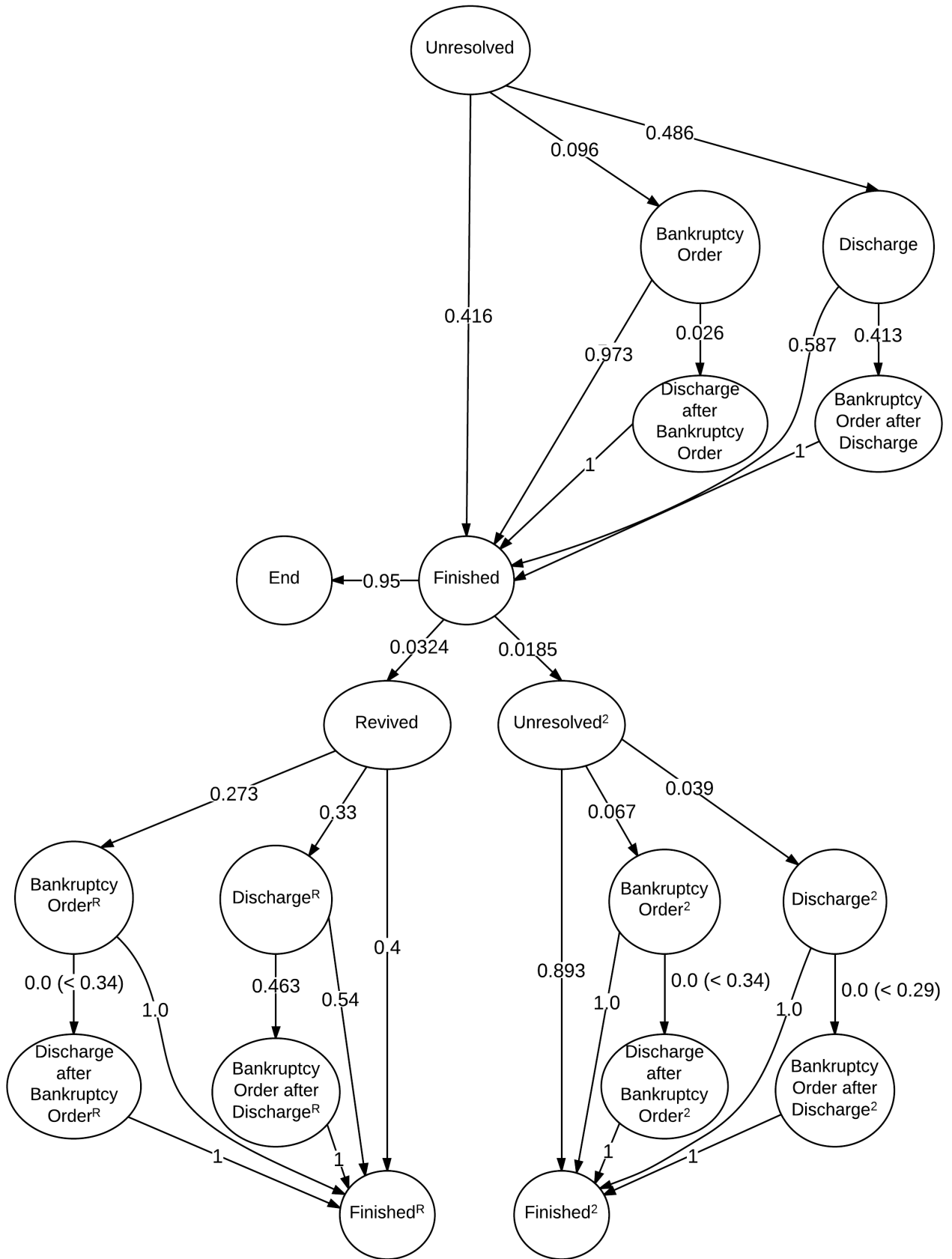


Figure 32: Model B with estimated probabilities by MBA.

5.3.3.2. Incorporating Insolvency Proceedings' Details

In Section 5.3.2.2 we managed to extend model B so that it would also contain information about the time course of the insolvency proceedings' process. However, we have also shown that at the time of writing there is not enough data to be able to come up with definitive conclusions. Because of that, in this section we decided to extend model B in a different way, by incorporating more detailed informations about the insolvency proceedings. These details include: gender, education, region, age, the information whether the debtor is an entrepreneur or a natural person and finally creditors participations. For each additional detail we will create a custom item representing its presence. We will denote this model as model D.

In the case of education we will divide the persons into two groups, those who have a title (indicating a university degree) and those who do not have any title. The title is usually a part of the debtor's name in the Insolvency Register, so it is simple to detect their presence. In case of age we will divide the people into 4 groups: under 30, between 31 and 40, between 41 and 50 and after 51. Lastly, we will include the information about the presence of the 10 most occurring creditors (see Figure 34). With this data we will be able to tell whether the demography or creditors presence has any impact on the insolvency proceedings' process.

In the resulting model we decided to omit all states from model B that come after the states **Revived** and **Unresolved**². We omitted them because the rules containing these states with the combination of the additional information would cause them to have even smaller support and probably would get filtered out by Apriori anyway. In the end this leaves us 43 different items, which is almost 3 times more as in the previous model. If we consider that this number of items can generate at most $2^{43} - 1$ different itemsets, then this will be a real test of the Apriori algorithm's efficiency.

In regards of the the minimal support and the minimal confidence we started with both parameters set to value 0.01 and kept decreasing it. Unfortunately we stopped at the same number as in previous section and that is 0.0001. Because all the previous values did not result in a sufficient coverage of all possibilities in our model. For example the generated association rules were missing almost all the rules including a state and the debtor being an entrepreneur.

However, with the minimal confidence and the minimal support set to 0.0001 the Apriori algorithm ran for 3 days straight and generated approximately 213 000 000

association rules. This number is really high especially in comparison with the 3500 rules generated in the previous section. From this we can see that the additional items really have a huge impact on the size of the learned model. But this is simply due to the fact that basically for each transition we need to know its probability for every combination of the additional items.

To be able to work with such a huge model in a compendious manner we needed to be able to search though the generated association rules efficiently. Consequently, we needed an effective storage for them as well. We decided to use the same RDBMS system we used for storing the extracted data from the Insolvency Register. However, there are different ways how the rules can be stored in the relational database.

At first we wanted to do it the correct way, which is compliant with the ER modeling principles. Based on these we should create two entities: *Rule* and *Item*. Then we should join these two entities by a $m:n$ association, because one *Item* can be a part of n rules and one *Rule* can contain m items. This relation would be then realized by 3 database tables. The first one (*t_items*) would contain the items, the second would contain the rules (*t_rules*) and the third (*t_items_rules*) would join them together. More precisely the *t_items_rules* table would contain two columns, the first one for the *Item* id and the second for the *Rule* id. Sadly, this representation is not suited if the *t_rules* table contains 200 million rows. Basically if one wants to find an association rule containing specific items, he needs to join those three tables and group the results by the *Rule* id. Then he can look up the association rule containing all the items he wants. This operation is very inefficient and one lookup takes a lot of time (in order of tens of minutes).

In our second attempt we dropped the *t_items_rules* table and created a column in the *t_rules* table that contained an ordered list of *Item* ids. This list was represented as a string value where the ids were simply separated by commas. This made it possible to directly lookup the association rule without needing to join any tables. But even this method turned out not to be efficient enough, in terms of both speed and disk space consumption. The file which contained all the 213 million association generated by SPMF was 11 GB large. When we imported the rules to the database the *t_rules* table was approximately 11 GB large as well. However, to speed up the lookups we needed to build an index over this table and that was approximately as

large as the table itself. Basically to store all these rules we needed over 20 GB of disk space. Even with the created index one lookup took in order of tens of seconds which we still found too slow and forced us to consider yet another method of representations.

In our third and last attempt we tried the following representation as our last resort. We stored every rule into 2 integers (64 bit large), the first (*rule_left*) one represented the left side of the rule and the second (*rule_right*) represented the right side. Since we have together 43 items, we were able to number them from 1 to 43. Then we set the n-th bit of *rule_left* of the corresponding rule to 1 if the n-th item on the rule's left side was a part of it. We did the same with *rule_right* and the right side of the rules. Finally, we created indexes on both columns *rule_left* and *rule_right* which resulted in lookups that took usually under 1 second.

With searches so fast we were able to create a web application called IPredictor (shown in Figure 33) which allows the user to search through the association rules in a very simple and interactive manner. In this application the user can select the current state of the insolvency proceeding and all the other parameters (gender, age etc.) and IPredictor will return the probabilities of moving to the next states.

Thanks to IPredictor and the extended model D we were able to improve the predictions' accuracy. For example if we consider that the debtor is a natural person then he has a 0.71 probability of moving to state the **Discharge** and only 0.03 probability of moving to the state **Bankruptcy Order**. In comparison if the debtor is an entrepreneur he has a 0.19 probability of moving to the state **Bankruptcy Order** and only 0.24 to the state **Discharge**. Here we would like to remind that based on the Insolvency Act the debtors which are also entrepreneurs should not be able to go through the state **Discharge** because the discharge method of insolvency proceeding's resolution is not applicable in their cases. This is very probably a result of our simplified method of distinguishing entrepreneurs from natural persons. We denote a debtor as an entrepreneur if he has an identification number (IČO). However, this is not always true, because self-employed people do have an IČO as well and are considered a natural person in context of the Insolvency Act and as debtors can go to discharge.

There is also a huge difference in probabilities of moving from the state **Unresolved** to the state **Finished** if we consider whether the debtor is a natural

person or an entrepreneur. In case of entrepreneurs this probability is 0.57 and in case of natural persons only 0.26. Interestingly enough there is also a large difference between the estimated probabilities if we consider that a debtor who is natural person has a university degree or not. The probability of moving to the state **Bankruptcy Order** in case of a natural person with an university degree is 0.09 and only 0.03 in case of a natural person without an university degree. The same applies for moving to the state **Discharge** where the probabilities are 0.48 (university degree) and 0.71 (without an university degree).

We also found out that gender usually does not play an important role in the decision of moving to the next state. The same applies for age. However, the region may have a large influence on the probabilities. We summarized these results in the Table 12 below.

Region	Discharge	Resolved	Bankruptcy Order
Jihočeský	0.77	0.2	0.03
Karlovarský	0.76	0.2	0.04
Vysočina	0.63	0.34	0.03
Královohradecký	0.84	0.13	0.03
Liberecký	0.69	0.28	0.03
Moravskoslezský	0.85	0.13	0.01
Olomoucký	0.67	0.31	0.02
Pardubický	0.81	0.16	0.03
Plzeňský	0.74	0.22	0.05
Praha	0.53	0.42	0.04
Stredočeský	0.67	0.29	0.05
Ústecký	0.79	0.19	0.02
Zlínský	0.58	0.39	0.03
Jihomoravský	0.52	0.45	0.03

Table 12: Probabilities of the insolvency proceeding's resolution states depending on the region (assuming that the debtor is a natural person).

The largest probability of a debtor who is natural person to go through the state **Discharge** is in region Moravskoslezský (0.85) and Královohradecký (0.84). On the other hand the smallest is this probability in regions *Jihomoravský* (0.52) and *Praha* (0.53).

Lastly, we will explore the influence of the creditors' participation on the method

of resolution. We will again assume that the debtor is a natural person. The result can be simply summarized as follows. If any of the 10 most participating creditors (see Figure 34) is present in an insolvency proceeding then the probability of the insolvency proceeding being resolved by discharge is at least 0.9. For example in case of the creditor *GE Money Bank, a.s.* this probability is 0.95. This definitively shows then among the most frequent creditors the discharge method of the insolvency proceeding's resolution is the most preferred.

In the social network analysis (section 5.4.3) we managed to naturally divide the debtors and creditors into two communities. The conclusion of the community discovery analysis was that these two groups of debtors differ for some reason but we could not find any other characteristics than a different group of creditors. To study the difference between these communities we decided to adjust the selection of creditors in the current model to contain the top 5 creditors from communities I. and II. listed in Table 17.

After we generated a new set of rules we started examining the probabilities depending on each creditor's participation. Most of the time the probabilities were similar for creditors from both groups. However, we noticed a slightly higher probabilities of the insolvency proceedings going from the state *Discharge* to the state *Bankruptcy Order After Discharge* in case of creditors from community II. *Telefónica ČR a.s.* (0.66), *T-Mobile ČR a.s.* (0.65) and *Všeobecná zdravotní pojišťovna ČR* (0,73). For comparison the probabilities of the same transition for the creditors from community I. were: *GE Money Bank a.s.* (0.56), *Cetelem ČR a.s.* (0,58) and *Česká spořitelna a.s.* (0.55).

The creditor that differed the most from all the other is *Všeobecná zdravotní pojišťovna ČR* because all the insolvency proceedings it is involved in have a much higher probability of being resolved by bankruptcy order (0,14). For every other of the 10 creditors this probability is less than 0,05.

In the previous paragraphs we were mostly focused on the debtors which are natural persons and the methods of resolution. The reason for this is that the natural persons group of debtors is the largest and we find the method of resolution the most important in the process of an insolvency proceeding. However, we published IPredictor as a public web application accessible on <http://zviri.cz/ipredictor> and anyone is free to explore also the other parts of the insolvency proceedings' process.

By doing so we have successfully joined the Open Data initiative of the Czech Republic and showed how the publicly available data can be reused.



Insolvency proceeding analysis

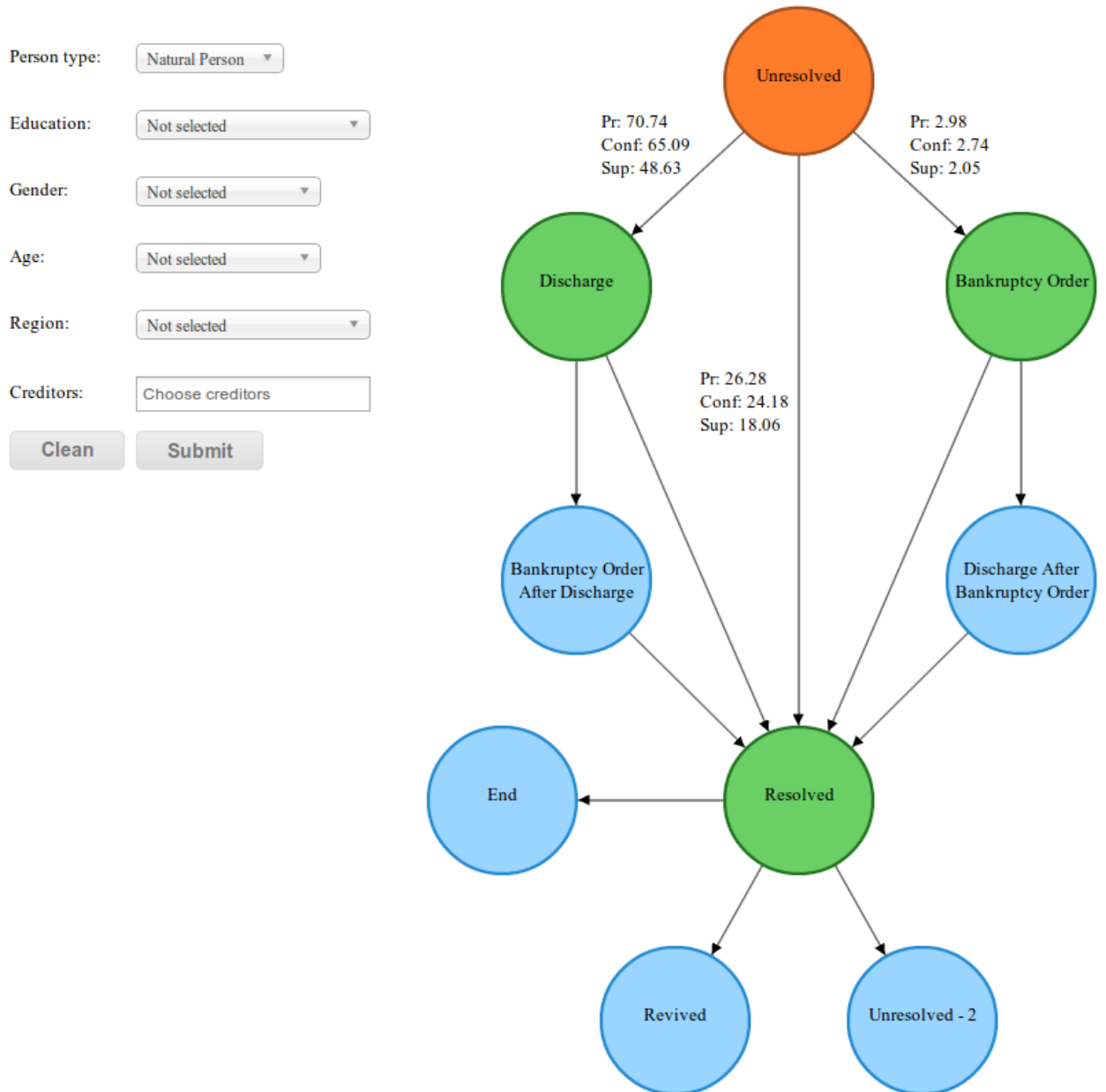


Figure 33: IPredictor web application for browsing the generated association rules of model D.

5.4. Subjects

In the previous section we have successfully analyzed the insolvency proceedings and now we have a solid understanding of the entire process. However, there is another important standpoint of the insolvency proceeding and that is how the subjects involved in it are related. In the scope of this work we will distinguish between three types of subjects, the debtors, insolvency proceeding administrators and creditors. In this section we will focus on finding interesting relationships between these subjects.

Based on the results of Section 5.1 and Section 5.2 we have a basic overview of the debtors. From Section 3.1 we know that the insolvency proceeding administrators are selected from a list of administrators managed by the Ministry of Justice. This means that there is just a limited number of certified administrators.

However, so far we have basically no information about the creditors. We can only assume that the creditors are mostly institutions providing loans, but we don't know which ones and we don't know how often each creditor participates in the insolvency proceedings. In addition, we don't even know the number of different creditors that occurred in the Insolvency Register. Because of that we will start the analysis by taking a closer look at the creditors.

5.4.1. Creditors

To be able to estimate the number of different creditors in the Insolvency Register, we must at first find a reliable identifier for them. Theoretically their names should do just fine but the problem is that some subjects occur in the Insolvency Register under different names. For example the creditor GE Money Bank occurs in the Insolvency Register under the following names:

- *GE Money Bank*
- *GE Money Bank a.s.*
- *GE Money Bank a.s..*
- *GE Money Bank, a. s.*
- *GE Money Bank, a.s.*
- *GE Money Bank,a.s.*
- *GE MONEY BANK a.s.*
- *GE Money Bank, a. s. GE Money Bank, a. s.*

As we can see there might be a quite large range of names for the same subject. We will fix most of this redundancy by the following simple preprocessing steps: transforming the name to lower case, removing white spaces and removing the subject form the end of its name. After this preprocessing there are 49 493¹² different creditors in the Insolvency Register at this time.

Now we would like to get the distribution of creditors over the insolvency proceedings. In simple words, we would like to know how often each creditor participates in the insolvency proceedings. The distribution of the 50 most frequent creditors is shown in Figure 34. These subjects are mostly banks, insurance companies and non-bank lending companies. The most dominant creditor is however GE Money Bank that is participating as a creditor in 21 474 different insolvency proceedings. If we take into consideration that there are roughly 120 000¹³ insolvency proceedings in the Insolvency Register right now, then *GE Money Bank* is a creditor in 18% of them. The second most frequent creditor is another bank *Česká spořitelna* with 14 032 (12%) occurrences. The next 6 subjects starting with *Cetelem ČR* (13 457) and ending with *Essox* (8 676) are non-bank lending companies. Now we would like to know the ratio of occurrences of the first 50 creditors which represent less than 0.1% of all creditors in comparison to all creditors' occurrences. By counting the occurrences of the first 50 creditors we get a number 226 709 and by counting all the occurrences we get 393 374. So the ratio we were looking for is equal to 0.58, which means that 58% of all creditors' occurrences are formed by the group of 50 creditors listed in Figure 34.

From the last paragraph we know that there is a small group of creditors that participate in insolvency proceedings very frequently. This brings up another question and that is whether there are creditors that co-occur in the insolvency proceedings and if yes then how often. For the purpose of this task we will use again the MBA model from Section 5.3.3. We will select the 100 most frequent creditors and represent them as items and consequently every insolvency proceeding will be considered to be a transaction. For example if there is an insolvency proceeding with the following creditors: *GE Money Bank*, *Essox*, *Provident* then we will simply represent it as a transaction [*GE Money Bank*, *Essox*, *Provident*].

12 Please note that this number and also the rest of the analysis in this section is based just on the set of known creditors we managed to extract. Therefore, the real number might be different. For details see Section 4.3

13 For more statistics about insolvency proceedings see Section 5.1.

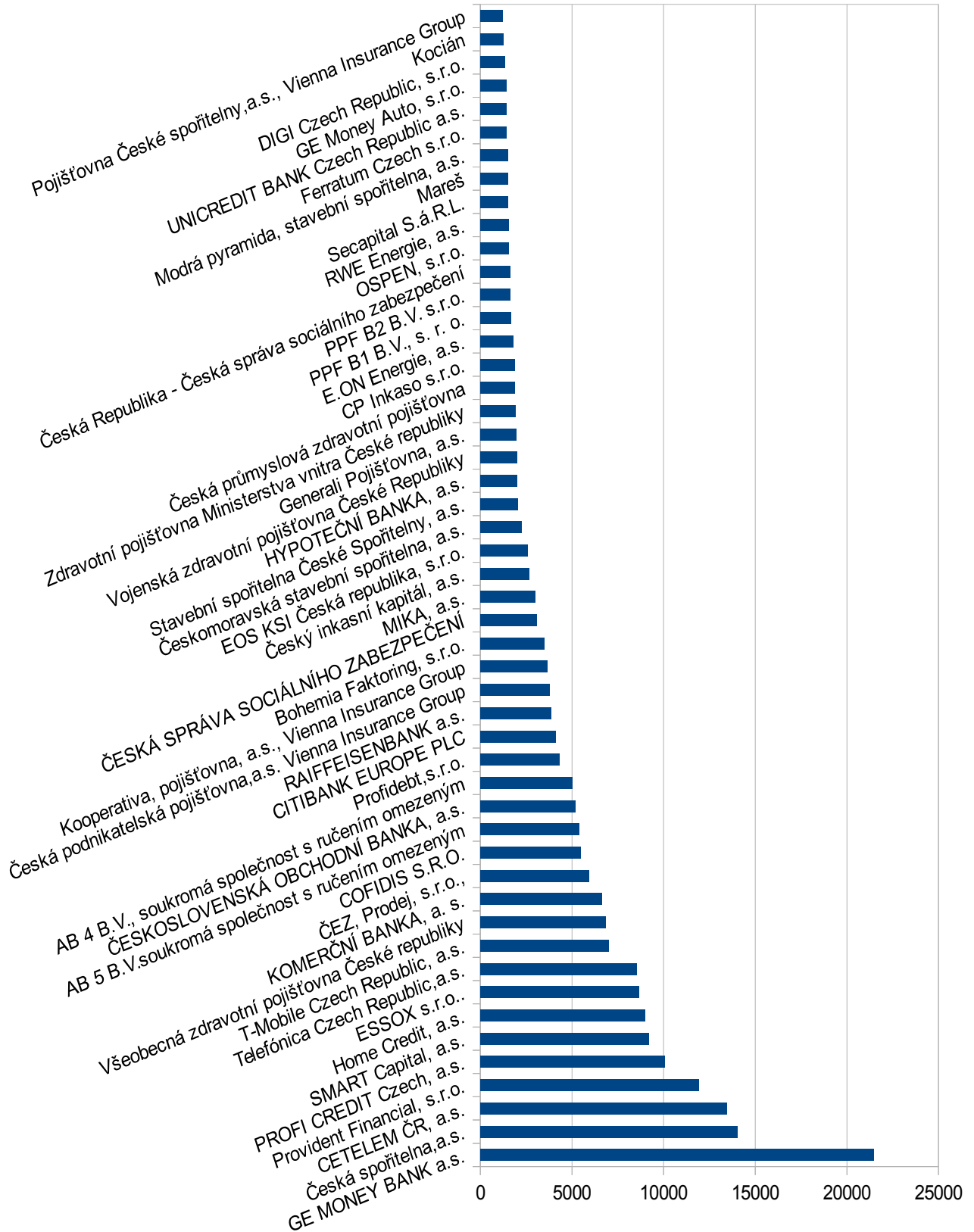


Figure 34: Number of insolvency proceedings creditors participated in.

We managed to extract 57 049 different insolvency proceedings with at least one known creditor. From those we have filtered out insolvency proceedings where none of the 100 hundred most frequent creditors have participated. The remaining 52 138 insolvency proceedings or transactions were used as input for the Apriori algorithm. We set only the minimal support parameter of Apriori, at first to 0.01 (473) to get a larger number of rules from which we can select the ones with the highest accuracy. After that we increased the minimal support to 0.1, which resulted in a much smaller number rules, however much more frequently applicable.

With minimal support set to 0.01 we managed to mine 4 470 association rules. To summarize these rules, it was necessary to rank the association rules in some way. Obviously *support* nor *confidence* is not suited for ranking because we want to see the best rules from both of the two standpoints. For this purpose usually the measurement called *lift* is used. For an association rule $X \Rightarrow Y$ *lift* is defined as follows:

$$lift(X \Rightarrow Y) = \frac{support(X \cup Y)}{support(X) * support(Y)} \quad (4)$$

In essence the *lift* of an association rule is the factor by which the confidence exceeds the expected confidence. The larger the *lift* is the stronger the association between the items is. Furthermore, if the lift value is greater than 1 then the rule's left and right side appear more often together than expected and it means that the occurrence of the rule's left side has a positive effect on the occurrence of the rule's right side. On the other hand if the lift is smaller than 1 then the rule's left side and rule's right side appear less often together than expected. Moreover it means that the occurrence of the rule's left side has a negative effect on the rule's right side.

Sadly, the SPMF framework does not provide the *lift*-s of the generated association rules, but Weka does. Furthermore, Weka can also sort the generated association rules by their *lift*-s. Since for this case we don't want to generate a large number of rules as in Section 5.3.3.1 and 5.3.3.2, we decided to use Weka instead.

The top 12 rules of the 4 470 generated by Weka sorted by their *lift*-s in descending order are shown in Table 13 below. Please note that *lift* is a symmetric measure, meaning that rules $X \Rightarrow Y$ and $Y \Rightarrow X$ have the same *lift*. Consequently, the rule $X \Rightarrow Y$ comes right after $Y \Rightarrow X$ in our results. To be able

to show more different association rules we only show the rule with the larger confidence from the two in Table 13.

The rules can be interpreted as follows: if creditors *Všeobecná zdravotní pojišťovna ČR* and *Vojenská zdravotní pojišťovna ČR* are participating in the same insolvency proceeding, then there is a 0.57 probability that also creditor *Zdravotní pojišťovna Ministerstva vnitra ČR* is participating.

Next we will present the results generated by the Apriori algorithm with minimal support set to 0.01. With this setup we obtained together only 30 rules with very short left and right sides. We sorted the rules by their *lifts* again and for the same reason as in case of previous Table 13 (skipping the reversed rules) we are only showing 15 rules in Table 14. It is obvious that Table 14 is full of the most frequent creditors from Figure 34 and based on the rules' properties we can say with confidence that they often participate in the same insolvency proceedings.

Rule's left side	Rule's right side	Support	Confidence	Lift
Všeobecná zdravotní pojišťovna ČR Vojenská zdravotní pojišťovna ČR	Zdravotní pojišťovna Ministerstva vnitra ČR	1.2% (630)	57%	13.48
Všeobecná zdravotní pojišťovna ČR Česká průmyslová zdravotní pojišťovna	Zdravotní pojišťovna Ministerstva vnitra ČR	1.1% (588)	56%	13.24
Všeobecná zdravotní pojišťovna ČR Zdravotní pojišťovna Ministerstva vnitra ČR	Vojenská zdravotní pojišťovna ČR	1.2% (630)	54%	12.52
PPF B2 B.V. s.r.o.	PPF B1 B.V., s. r. o.	1.1% (582)	39%	12.33
Všeobecná zdravotní pojišťovna ČR Zdravotní pojišťovna Ministerstva vnitra ČR	Česká průmyslová zdravotní pojišťovna	1.1% (588)	5%	12.23
Všeobecná zdravotní pojišťovna ČR Česká průmyslová zdravotní pojišťovna	Vojenská zdravotní pojišťovna ČR	1% (535)	51%	11.8
Zdravotní pojišťovna Ministerstva vnitra ČR	Vojenská zdravotní pojišťovna ČR	1.4% (741)	36%	8.29
Česká Republika - Česká správa sociálního zabezpečení	Česká průmyslová zdravotní pojišťovna	1% (573)	33%	8.13

Česká průmyslová zdravotní pojišťovna	Zdravotní pojišťovna Ministerstva vnitra ČR	1.3% (694)	34%	8.13
Zdravotní pojišťovna Ministerstva vnitra ČR	Telefónica Czech Republic,a.s.	1.3% (693)	33%	6.25
Profi Credit Czech, a.s. AB 5 B.V. s.r.o.	Provident Financial, s.r.o. AB 4 B.V., s.r.o.	1% (559)	36%	6.2
GE Money Bank a.s. Česká spořitelna,a.s. AB 5 B.V. s.r.o.	Cetelem ČR, a.s. AB 4 B.V., s.r.o.	1% (520)	34%	6.17

Table 13: Top 12 Creditors' co-participation association rules generated by the Apriori algorithm with minimal support set to 0.01. The first column represents the left side of the association rule and the second rule represents the right side of the association rule.

Rule's left side	Rule's right side	Support	Confidence	Lift
AB 4 B.V. s.r.o.	AB 5 B.V. s.r.o.	2.3% (3 042)	52%	4.16
GE Money Bank a.s. Smart Capital a.s.	Profi Credit Czech a.s.	1.9% (2 497)	46%	2.13
GE Money Bank a.s. Profi Credit Czech a.s.	Smart Capital a.s.	1.9% (2 497)	4%	2.07
Smart Capital a.s.	Profi Credit Czech a.s.	3.1% (4 126)	43%	1.98
Všeobecná zdravotní pojišťovna ČR	Telefónica ČR a.s.	2% (2 638)	36%	1.94
AB 4 B.V. s.r.o.	Provident Financial s.r.o.	2.1% (2 847)	49%	1.86
GE Money Bank a.s. Cetelem ČR a.s.	Essox s.r.o.	2.3% (3 058)	34%	1.83
GE Money Bank a.s. Essox s.r.o.	Cetelem ČR a.s.	2.3% (3 058)	52%	1.82
AB 5 B.V. s.r.o.	Provident Financial s.r.o.	2.1% (2 796)	45%	1.71
Cofidis s.r.o.	Cetelem ČR a.s.	2% (2 748)	48%	1.68
GE Money Bank a.s. Provident Financial s.r.o.	Profi Credit Czech a.s.	2% (2 575)	36%	1.68
Smart Capital a.s.	Provident Financial s.r.o.	3.2% (4 237)	44%	1.67
GE Money Bank a.s. Česká spořitelna a.s.	Cetelem ČR a.s.	2.7% (3550)	47%	1.66
AB 4 B.V. s.r.o.	Cetelem ČR a.s.	2% (2747)	47%	1.65
Essox s.r.o.	Cetelem ČR a.s.	3.2% (4313)	47%	1.64

Table 14: Top 15 Creditors' co-participation association rules generated by the Apriori algorithm with minimal support set to 0.01.

5.4.2. Social Network Analysis

Social network analysis is the study of social entities (called actors), and their interactions and relationships. These relationships can be represented by a graph (or a network) where each node represents an actor and each edge (or link) represents a relationship. Based on this network we can study the properties of its structure, the roles, positions and prestiges of each actor in it. We can also look for various kinds of sub-graphs (e.g. communities) formed by the groups of actors.

In this section we will try to adapt the study of social networks in the context of the subjects participating in the insolvency proceedings. We will start by analyzing the properties of our network and try to look for prominent nodes in it. The prominence of a node however can be derived from many different characteristics by many different methods. As a part of this work we shall review the most common approaches and apply them to the social network of the Insolvency Register. The output of this part should be a ranking of nodes based on their prominence and reviewing the position of the most prominent nodes in the network.

The second part of this section will focus on community discovery. A community in a network can be simply characterized as subgroup of nodes that share similar properties and naturally belong to each other. There are again many different methods for this task and we shall similarly review the most popular ones and apply them to our social network.

One of the typical examples of a social network is the Web where each web page represents an actor and each hyperlink represents a relationship. Many of the results from social networks study can be adapted and extended for use in the Web context. The ideas from social network analysis are the main reason of the Web search engines success. That is also why most of the algorithms are defined in the context of the Web and why we will keep this assumption in the definition of these methods.

In the following 2 sections, we will introduce two types of prominence measures, *centrality* and *prestige*. Both of these measures are based on different characteristics and may lead to different results. The following sub-sections are mostly based on the following two books [35][36].

5.4.2.1. Centrality

The important actors included in the social network are those that are involved

(linked) with other actors extensively. For example in the context of an organization, a person with extensive contacts to many people is considered more important than a person with a smaller number of contacts. We denote the actors involved with many other actors as *central actors*. There might be different types of links (relationships) between the actors, thus there might exist several types of centrality in one network as well. In the rest of this subsection we will discuss the three most popular types of centralities[35]. The definitions of these centralities differ when we assume a directed or an undirected graph, thus each centrality will be defined for both cases individually.

Degree Centrality

In this case the central actors are the ones that have most links with other actors. Let us denote the total number of actors (nodes) in the network as n .

Undirected Graph: In an undirected graph, the *degree centrality* $C_D(i)$ of an actor i is simply the node degree of the actor's node, denoted as $d(i)$, normalized by the maximum possible degree $n-1$:

$$C_D(i) = \frac{d(i)}{n-1} \quad (5)$$

Directed Graph: In case of directed graphs, we need to distinguish between incoming links (in-links) of an actor i and outgoing links (out-links). The *degree centrality* $C'_D(i)$ is then defined based only on the out-degree (the number of out-links) $d_o(i)$:

$$C'_D(i) = \frac{d_o(i)}{n-1} \quad (6)$$

Closeness Centrality

This type of centrality is based on the closeness or distance. Basically, if an actor i is central it can easily interact with all the other actors. Moreover, its distance to all other actors is short. Thus, we can use the shortest distance to compute this measure. Let us denote the shortest distance from actor i to actor j as $d(i, j)$.

Undirected Graph: The closeness centrality $C_C(i)$ of an actor i is defined as:

$$C_c(i) = \frac{n-1}{\sum_{j=1}^n d(i, j)} \quad (7)$$

The denominator is simply the sum of the shortest distances from actor i to all other actors. Note that this equation is only meaningful for a connected graph.

Directed Graph: The same equation can be used for a directed graph. The computation of distances however needs to consider the directions of the edges.

Betweenness Centrality

If two non-adjacent actors j and k want to interact and actor i is on the path between j and k , then i may have some control over their interactions. Betweenness measures this control of i over other pairs of actors. Naturally, if i is on the paths of many such interactions, then i is an important actor.

Undirected Graph: Let p_{jk} denote the number of shortest paths between actors j and k . The betweenness $C_B(i)$ of an actor i is then defined as the number of shortest paths that pass i (denoted as $p_{jk}(i)$) divided by the total number of shortest paths of all pairs of actor not including i :

$$C_B(i) = \sum_{j < k} \frac{p_{jk}(i)}{p_{jk}} \quad (8)$$

In case when there are multiple shortest paths between actor j and actor k , then we assume that all paths are equally likely to be used. Since, the betweenness centrality is a little harder to imagine, let us demonstrate it on the example social network in Figure 35. For example actor 1 is the most central actor in this case, because it lies on all 15 shortest paths connecting the other 6 actors. All the other actors (besides 1) have the betweenness centrality equal to 0.

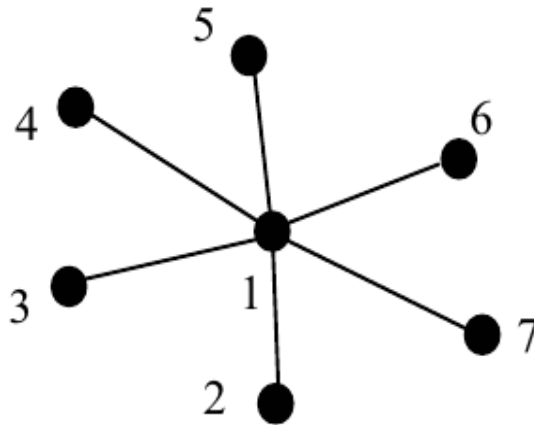


Figure 35: An example of a social network illustrating the betweenness centrality[35].

Directed Graph: We can use the same equation for the directed graphs as well, but it must be multiplied by 2 because now the path from actor j to k is different from the path from actor k to j . Likewise, p_{jk} must consider paths from both directions as well.

5.4.2.2. Prestige

The *prestige* is a more sophisticated measure of prominence of an actor than a centrality. We need to distinguish between the in-links and out-links of the actors. In this case a prestigious actor is the one who has the most in-links. Moreover, the prestige cannot be computed unless the social graph is directed. Consequently, the main difference between the prestige and the centrality is that the centrality focuses on out links while prestige focuses on in-links.

There exists more types of prestige [35][36], but we will concentrate on one in particular called the *rank prestige*, which forms the basis of most link analysis algorithms, such as PageRank and HITS.

The *rank prestige* measure considers the prominence factor of individual actors who do the “voting” or “choosing”. More precisely, an actor i chosen by an important actor is more prestigious than one chosen by a less important actor. In real world this could be for example, that a company CEO voting for a person is much more important than a worker voting for the same person. Similarly, if the actor's circle of influence is full of prestigious actors, then his prestige is high as well. Because of that, the rank prestige $P_R(i)$ of an actor i is defined as a linear combination of links that point to i :

$$P_R(i) = A_{1i}P_R(1) + A_{2i}P_R(2) + \dots + A_{ni}P_R(n), \quad (9)$$

where $A_{ji} = 1$ if j points to i , and 0 otherwise. Basically, the equation 9 says that the actor's rank prestige is a function of the ranks of the actors who vote or choose the actor. For each n actors we have n equations so we can write them in a matrix notation. We will define a column vector P containing all the rank prestige values, i.e. $P = (P_R(1), P_R(2), \dots, P_R(n))^T$. Then we use matrix A to represent the adjacency matrix of the network, where $A_{ij} = 1$ if i points to j , or 0 otherwise. This results in equation:

$$P = A^T P \quad (10)$$

Equation 10 is the characteristic equation used for finding the eigensystem of matrix A^T , where P is an eigenvector of A^T . We will use this equation as a basis in the next two sections where we will describe 2 of the most well known ranking algorithms which are used for Web Search (PageRank and HITS). Since both of these algorithms were defined mainly for the use in Web Search engines we will describe it in this context as well. Nevertheless, we will then describe how they can be applied on our task of studying relationships between the subjects in the Insolvency Register.

5.4.2.3. PageRank

The PageRank algorithm [35][37] is a static ranking algorithm of Web pages in the sense that the rank value is computed for each page off-line and it does not depend on the search queries. Since PageRank is based on the measure of prestige in social networks, its value for each page (actor) can be represented as its prestige.

Before deriving the PageRank formula we will start by defining some main concepts in the Web context.

In-links of page i are all the hyper links that point to page i from any other pages.

Out-links of page i are all the hyperlinks pointing out to other pages from page i .

According to the rank prestige in social networks, the importance of page i (PageRank score) is determined by summing up the PageRank scores of all pages that point to i .

To formulate the PageRank algorithm, we will treat the Web as a directed graph $G=(V, E)$, where V is the set of vertices or nodes (all pages), and E is the set of directed edges (hyperlinks). Let the total number of nodes (pages) in graph G be $n=|V|$. Then we define the PageRank score $P(i)$ of the page i as:

$$P(i) = \sum_{(j,i) \in E} \frac{P(j)}{O_j} \quad (11)$$

where O_j is the number of out-links of page j . We can formulate this mathematically by a system of n linear equations with n unknown variables. Consequently, we can use a matrix to represent all these equations. Let us define P as a n -dimensional column vector of PageRank values $P=(P(1), P(2), \dots, P(n))^T$.

Then let A be the adjacency matrix of our graph, where $A_{ij} = \frac{1}{O_i}$, if $(i, j) \in E$ and 0 otherwise. Similarly, as in Section 5.4.2.2 we can write the system of n equations as:

$$P = A^T P \quad (12)$$

This is the characteristic equation of the eigensystem, where the solution to P is an eigenvector with the corresponding eigenvalue of 1. A well known mathematical technique called *power iteration*[38] can be used to find P . The power iteration algorithm is described below.

PageRank-Iterate

Input:

G ... input graph representing the Web

d ... damping factor between 0 and 1 (usually set to 0.85 [38]).

ε ... threshold parameter for convergence testing. If the norm of the residual vector is less than this threshold the algorithm assumes it has successfully converged.

Step 1. Initialization

Initialize a new set $P_0 = \frac{e}{n}$, where $e = (1, \dots, 1)^n$ and n is the number of nodes in graph G .

Initialize a new variable $k = 1$.

Step 1.1 Main loop

Create a new set $P_k = (1-d)e + d A^T P_{k-1}$

Step 1.2 Loop condition

If $\|P_k - P_{k-1}\| > \epsilon$ increase k by one go to 1.1.

Step 1.3 Returning the result

Return P_k

In the context of the Web, PageRank has several major advantages. One of them is its ability to fight spam. A page is important if the pages pointing to it are important as well. Since it is not that easy for a Web page owner to add in-links into his page from other important pages, it is thus not simple to influence the resulting PageRank. Another advantage of PageRank is that it is a global measure independent from the query. This allows the PageRank-s to be precomputed and stored off-line rather than computed at query time allowing large efficiency during query time.

5.4.2.4. HITS

The acronym HITS [35][39] stands for *Hypertext Induced Topic Search*. The main difference between HITS and PageRank is, that HITS is query dependent and thus not static. Everytime the user sends a query, HITS first expands the list of relevant pages returned by a search engine and then produces two rankings, *authority ranking* and *hub ranking*.

HITS distinguishes between two types of pages (nodes), the first one is *authority* which is a page with many in-links. We assume that authority pages may be a good and reliable source of information and thus many people trust it and link to it. On the other hand the *hub*-s are pages with many out-links. These pages serve as an organizer of information and point to many good authority pages regarding a specific topic. The basic idea behind HITS is that a good hub should link to many good authorities and a good authority is linked to by many good hubs. Consequently, authorities and hubs have a mutual reinforcement relationship.

Before we start describing the HITS algorithm, it is worth mentioning how HITS collects pages to be ranked. Let us assume that an user submits a query q . Then HITS collects t (typically set to 200 [39]) highest ranking pages which we assume are highly relevant to the search query q . We will call this set of pages as the root set W .

Then W is expanded by including any page pointed to by a page in W and any page that points to a page in W . This results in a larger base set called S . However, there is a large possibility that S becomes too large and thus the algorithm restricts the size of set S by allowing each page in W including at most k (usually 50) pages.

Then HITS starts processing every page in S by assigning it an *authority* and *hub score*. Let us denote the number of pages in S to be n . We will again use $G=(V, E)$ to denote the link graph of S , where V is the set of pages (nodes) and E is the set of links (directed edges). We will denote the adjacency matrix of G as L and set $L_{ij}=1$ if $(i, j) \in E$ and 0 otherwise. Next we will denote the authority score of page i as $a(i)$ and the hub score of page i as $h(i)$. We will represent the mutual reinforcing relationships of the two scores as follows:

$$a(i) = \sum_{(j,i) \in E} h(j) \quad (13)$$

$$h(i) = \sum_{(i,j) \in E} a(j) \quad (14)$$

We will combine equation 13 and 14 to get the matrix form of these equations. We will use \vec{a} to denote the column vector with all the authority scores, $\vec{a}=(a(1), a(2), \dots, a(n))^T$ and \vec{h} to denote the column vector with all the hub scores, $\vec{h}=(h(1), h(2), \dots, h(n))^T$. The matrix form can be then expressed as:

$$\vec{a} = L^T \vec{h} \quad (15)$$

$$\vec{h} = L \vec{a} \quad (16)$$

The computation of the authority and hub scores is basically the same as the computation of the PageRank scores. Thus, we will again use the power iteration algorithm. We will denote \vec{a}_k and \vec{h}_k the authority and hub scores at the k -th iteration. The iterative processes for generating the final solutions are

$$\vec{a}_k = L^T L \vec{a}_{k-1} \quad (17)$$

$$\vec{h}_k = L L^T \vec{h}_{k-1} \quad (18)$$

starting with

$$\vec{a}_0 = \vec{h}_0 = (1, 1, \dots, 1). \quad (19)$$

Notice that equation 17 does not use the hub vector due to substitution of equation 16. The same applies for the authority vector in equations 18. Also both the authority and hub score vectors are normalized to sum up to 1. The final description of the HITS algorithm is below.

HITS-Iterate

Input:

G ... input graph representing the Web

ϵ_a, ϵ_h ... threshold parameters for convergence testing. If the norm of both residual vectors is less than the corresponding threshold the algorithm assumes it has successfully converged.

Step 1. Initialization

Initialize 2 new vectors $\vec{a}_0 = \vec{h}_0 = (1, 1, \dots, 1)$.

Initialize a new variable $k = 1$.

Step 1.1 Main loop

Create a new vector $\vec{a}_k = L^T L \vec{a}_{k-1}$ and $\vec{h}_k = L L^T \vec{h}_{k-1}$

Normalize both vectors to sum up to 1:

$$\vec{a}_k = \frac{\vec{a}_k}{\|\vec{a}_k\|}$$

$$\vec{h}_k = \frac{\vec{h}_k}{\|\vec{h}_k\|}$$

Step 1.2 Loop condition

If $\|\vec{a}_k - \vec{a}_{k-1}\| > \epsilon_e$ or $\|\vec{h}_k - \vec{h}_{k-1}\| > \epsilon_h$ increase k by one and go to 1.1.

Step 1.3 Returning the result

Return \vec{a}_k and \vec{h}_k .

The main strength of the HITS algorithm is its ability to rank pages according to the query topic, which may provide more relevant results to the users. However, HITS has several disadvantages as well. First of them is its inability to filter spam like PageRank. It is quite easy to influence the HITS scores by adding out-links from one's own page to point to many good authorities. This can significantly increase the

hub score of the page.

Another common problem of HITS is topic drift. During the expansion of the root set, HITS can easily collect many pages (including hubs and authorities) which have nothing to do with the search topic. For example the out-links of these pages may not point to pages that are relevant and thus include spam in the result.

Also HITS cannot be precomputed and stored as PageRank and its evaluation may take significant time during the query time.

5.4.2.5. Community Discovery

Intuitively, a community simply represents a group of entities that share a common interest or are involved in the same activity. Apart from the Web, communities also exist in emails and text documents. There are many motivations for discovering these communities. For example in the context of the Web Kumar et al. [40] Listed three reasons:

1. Communities provide valuable and the most reliable and up-to-date information resources for a user interested in them.
2. They represent the sociology of the Web: studying them gives insights into the evolution of the Web.
3. The enable target advertising at a very precise level.

We can define the concept of communities formally as follows [35]. Given a finite set of entities $S = \{s_1, s_2, \dots, s_n\}$ of the same type a community is a pair $C = (T, G)$, where T is the community theme and $G \subseteq S$ is the set of all entities in S that shares the theme T . If $s_i \in G$ then s_i is said to be a member of community G . Based on whether we allow one entity or node to be a member of one or more communities, we differentiate between overlapping [41] and non-overlapping communities.

Our social network of the subjects in the Insolvency Register however will be much smaller in terms of both number of nodes and number of relationships in comparison to the traditionally studied networks (e.g. the Web). Because of that we will only be assuming that each node (in our case subject) can be a member of at most one community. Thus, we will focus on the discovery of non-overlapping communities (partitioning).

Methods for studying community discovery or partitioning in social networks are a very popular topic of research. There exist several applicable methods, the more early ones are CONCOR [42] proposed in 1977 and BURT [43] proposed in 1980. Examples of the more recent ones are the *Exact-Flow* algorithm [44] and the modularity based algorithms like the *Louvain method* [45] and the *Method of Optimal Modularity (MOOM)* [46]. The later one will be adopted for the purposes of this work.

Suppose that we are given a social network and want to determine whether there exists any natural division of vertices into non-overlapping communities. Furthermore, these communities may be of any size. Let us approach this problem in stages. We will start by the problem of whether any good division of the network exists into only two communities. The most obvious approach in this case would be to split the nodes into two sets so that we will minimize the number of edges between the two sets. This is the so called “minimum-cut” approach adopted most often in the graph partitioning literature. The problem in our case however is that we do not know the sizes of the communities beforehand.

The main idea behind *MOOM* is that a good division into communities is not simply one in which there are few edges between communities, but the one in which there are fewer edges between communities than expected. If the number of edges between two groups is only what one would expect from a random distribution then this decision has a low information value. On the other hand if the number of edges between the two groups is much smaller than we expect by chance then we can conclude that it might be a natural division.

We can express this idea by the statistical measure called modularity. Modularity is simply the number of edges falling within groups minus the expected number in an equivalent network with edges selected at random. The modularity can attain positive values indicating the possible presence of a community or a negative value and not indicating an existence of a community. If we assume the above to be true then by simply choosing the division which maximizes the modularity would provide us with the communities we are looking for.

Let us assume that our network contains n nodes and is represented by a graph $G=(V, E)$ where V is the set of all nodes and E is the set of all edges. For a

particular division into two sets let $s_i=1$ if node i belongs to Set 1 and $s_i=-1$ if it belongs to Set 2. Let us represent the edges once more by an adjacency matrix A , where A_{ij} is 0 or 1 based on whether there is an edge between node i and j or not. Next we assume that the expected number of edges between node i and j is equal to

$$\frac{k_i k_j}{2m}, \quad (20)$$

where k_i and k_j are the degrees of the corresponding vertices and m is the total number of edges in the network. Then the modularity Q is defined simply as

$$Q = \sum_{(i,j) \in E \wedge (i=j=1 \vee i=j=-1)} A_{ij} - \frac{k_i k_j}{2m}. \quad (21)$$

The sum in equation 21 is simply across all edges that belong to the same set (meaning both nodes of the edge belong to the same set). When considering that

$\frac{1}{2}(s_i s_j + 1)$ is equal to 1 if both i and j are in the same set and 0 otherwise, then we can express the modularity as

$$Q = \frac{1}{4m} \sum_{(i,j) \in E} (A_{ij} - \frac{k_i k_j}{2m})(s_i s_j + 1). \quad (22)$$

We can rewrite equation 22 in the matrix form as

$$Q = \frac{1}{4m} \vec{s}^t B \vec{s}, \quad (23)$$

where s is the column vector of the s_i values and B is a symmetric matrix with real elements

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}. \quad (24)$$

which we call the *modularity matrix*.

Notice that the elements of each row and each column of the modularity matrix sum up to zero, meaning that it always has an eigenvector $(1,1,1,\dots)^n$ with its corresponding eigenvalue 0. We will continue by rewriting \vec{s} as a linear

combination of the normalized eigenvectors u_i of B

$$Q = \frac{1}{4m} \sum_i a_i \vec{u}_i^T B \sum_j a_j \vec{u}_j = \frac{1}{4m} \sum_{i=1}^n (\vec{u}_i^T \cdot \vec{s})^2 \beta_i, \quad (25)$$

where β_i is the eigenvalue of B corresponding to eigenvector \vec{u}_i .

Let us reorder the eigenvalues in a descending order so that β_1 is the largest and β_n the smallest. We will remind that we want to maximize the modularity by choosing an appropriate division of the network, more precisely choosing the value of the index vector \vec{s} . Again, if there were no other constraints on our choice of \vec{s} , the solution would be very simple, we would choose \vec{s} proportional to the eigenvector \vec{u}_1 . By doing so we would place all of the weight in the term involving the largest eigenvalue β_1 . This would make the other terms equal to zero, because the eigenvectors are orthogonal.

The other constraint we have is the restrictions of the elements to the values $+1$ or -1 , which means \vec{s} cannot be chosen parallel to \vec{u}_1 . We will instead make \vec{s} as close to parallel as possible and maximizing the dot product $u_1^T \cdot \vec{s}$. This is achieved by setting $s_i = 1$ if the corresponding element of \vec{u}_1 is positive and $s_i = -1$ otherwise.

We can summarize the algorithms as follows: we compute the leading eigenvector of the modularity matrix and divide the vertices into two groups according to the signs of the elements in the vector. For dividing the nodes into 3 or more communities we can simply call the algorithm recursively on each generated community.

5.4.2.6. Markov Cluster Algorithm

The problem of discovering non overlapping communities or partitioning resembles the problem of finding good clusters in the network in many ways. However, the general graph partitioning algorithms usually require to know the number of partitions or the partition sizes beforehand. We have already mentioned that this poses a problem for community discovery applications, when we don't have any informations regarding the potential communities in the network.

On the other hand clustering methods are used to naturally group similar objects

(in our case nodes) together to form the final clusters. The number of clusters does not have to be known in advance. As in case of graph partitioning there are several methods proposed for graph clustering as well [47][48]. However for the purposes of this work we adopted one of the best performing algorithms called the *Markov Chain Algorithm (MCL)* [49] with many effective implementations.

Natural clusters in networks are characterized by the presence of many edges between the members of clusters. Consequently, we assume that the number of longer paths¹⁴ between two nodes in one cluster is high as well. This number should be also high in comparison to pairs of nodes which do not belong to the same cluster. We can formulate this idea with the concept of random walks, where we expect that the probability of a random walk to go from one cluster to another is much lower than the probability of it going through nodes of the same cluster.

The MCL algorithm simulates random walks within a network or a graph by alternation of two operators called *expansion* and *inflation*. Expansion corresponds to taking the power of a stochastic matrix using matrix squaring. Inflation corresponds to taking powers of the matrix entry-wise. These two steps are then followed by a scaling step, such that the resulting matrix is stochastic again.

A column stochastic matrix is a non-negative matrix with the property that each column sums up to 1. Given a stochastic matrix M and a real number $r > 1$, we define the inflation operator with the power coefficient r as $\Gamma_r(M)$ where

$$\Gamma(M_{ij}) = \frac{M_{ij}^r}{\sum_{r,j} (M)}. \quad (26)$$

The denominator of equation 26 corresponds to the summation of all the entries in column j of M after taking powers (normalization step). Each column j of matrix M corresponds to one node j in our graph. Row i of column j represents the probability of going from node j to node i . Expansion operation corresponds to computing random walks of higher length, it associates new probabilities with all pairs of nodes. Since longer paths are more common within a cluster than between different clusters, the probabilities associated with the pairs of nodes from the same cluster should be larger as well. Then inflation has the effect of boosting the

14 We define the path as a sequence of directed edges connecting a sequence of nodes. The length of a path is the number of edges it uses.

probabilities of intra-cluster walks. This is the effect of the cluster structure being present.

The MCL algorithm consists of iteratively applying expansion and inflation which results into separation of the graph into different segments (clustering). At the end of the algorithm no paths between the segments should be present anymore. Increasing the inflation parameter r makes the inflations “stronger” and increases the granularity of the clusters. The whole algorithm is summarized below.

MCL

Input:

G ... input graph

r ... inflation parameter affecting clusters granularity

Step 1. Initialization

Initialize a new matrix M_1 as the adjacency matrix of graph G .

Step 1.1 Main loop

Perform expansion: $M_2 = M_1 * M_1$

Perform inflation: $M_1 = \Gamma(M_2)$

Step 1.2 Loop condition

If M_1 is not idempotent (difference between matrices M_1 and M_2 is less than ϵ) go to *1.1*

Step 1.3 Returning the result

Return M_1 .

5.4.3. Social Network Of The Insolvency Register

In the previous section we have thoroughly described the theoretical aspects of the social networks. We introduced several popular methods used in the study of the social networks. In this section we will construct the social network of the subjects participating in the insolvency proceedings and use the introduced methods to explore its final structure.

For the purposes of this section we decided to use an open-source toolkit for

social network visualization and exploration, named Gephi[50]. Gephi basically contains the implementation of each method described in Section 5.4.2 and many different methods as well. Another feature of Gephi is its user friendly interface, which allows interactive social network analysis and exploration. It can also handle quite large networks with more than 20 000 nodes[50].

We decided to include three types of subjects (nodes) in our social network, debtors, creditors and administrators. To this date the Insolvency Register contains approximately 120 000 different insolvency proceedings, 60 000 creditors and 2000 administrators. Thus, the resulting social network would contain together approximately 180 000 different nodes. This number is sadly too large for Gephi because it only hardly fits into a memory of one computer. That is why we decided to only study a small subset of subjects in the Insolvency Register.

We will focus only on the largest and most interesting group of insolvency proceedings which were or are being resolved by discharge. However, this would still leave us with 66 000 insolvency proceedings. To lower the number even more we decided to only include the insolvency proceedings from the region with the highest rate of discharges (*Ústecký kraj*¹⁵). With the previous step we managed to cut down the number of insolvency proceedings to a reasonable number of 4 500.

Notice, that we could also pick approximately 4 500 insolvency proceedings at random from all insolvency proceedings resolved by discharge, so that we would get a sample with region distribution similar to the original one. However, this kind of sampling would not suit our interest of also studying debtors sharing the same address (see later in this section). The probability of choosing two debtors sharing the same address at random is very small and it would be useless to study their relationships. On the other hand by choosing all the debtors from one region we make sure that we will include all debtors possibly sharing their address.

Furthermore, from Section 5.4.1 we know that a small group of creditors are participating in the vast majority of insolvency proceedings. That is why we only included the 1 000 most frequent creditors covering 81% of all creditors' participations in all insolvency proceedings. Finally, we included only the administrators (approximately 200) which were participating in the 4 500 insolvency proceedings we selected earlier.

15 For the the ratio of discharges per region see Section 5.2.

The next step is naturally including edges between the approximately 6 000 nodes (subjects) we selected in the previous paragraph. At first we created a directed edge between the node of debtor d and the node of creditor c if and only if creditor c was participating in the insolvency proceeding of debtor d . We will refer to the later relation simply as *owes*. Similarly, we created a directed edge between the node of debtor d and the node administrator a , if and only if administrator a was participating in the insolvency proceeding of debtor d (relation *administers*). Lastly, we created a bidirectional edge between all the debtors sharing the same address (relation *share address*). This should make us possible to discover for example married couples or family members in the Insolvency Register.

One of the ways of importing an existing graph into Gephi is by two CSV files, the first one containing the nodes of the graph and the second its edges. We exported our graph in this format from our database and successfully imported it into Gephi. The first thing we did after import was coloring the three types of nodes and edges as follows: debtors – green, administrators – red and creditors blue. The nodes of the network are very simply structured in a grid layout right after import. This layout however is not very good for exploring a network. What we wanted to do was to layout the graph so that similar nodes would be nearby. Gephi supports several layouting algorithms such as Fruchterman Reingold[51] and Force Atlas 2 [52]. We used the later one because it has a parallel implementation making it much faster than Fruchterman Reingold. Lastly, we wanted to simply highlight the nodes with the largest degrees. We did so by ranking the nodes by their degrees and set the sizes of nodes and labels depending on its ranking. The result denoted as *Network A* is shown in Figure 36.

From Figure 36 we can see that most of the edges (78%) represent the relation of owing (edges marked blue). The second most common edge represents administering (17%) marked red and surprisingly edges share address represent 5% of all edges. We can see that the layouting algorithm pushed the largest groups of debtors sharing the same address outside of the main cluster (mostly at the top). The two largest groups of debtors are on the top and we can see that these groups are quite large containing 10 to 20 members indicating a possible community.

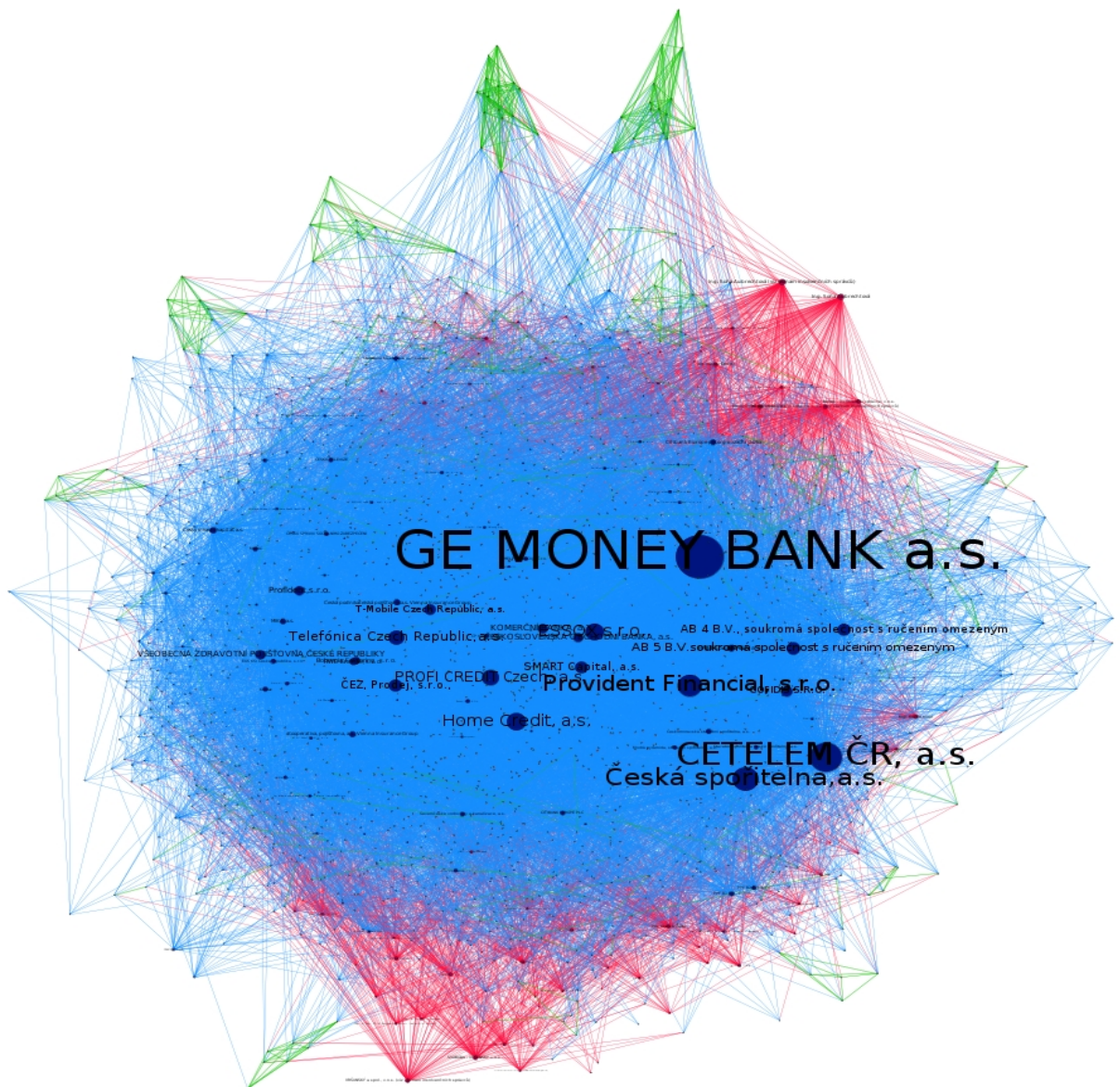


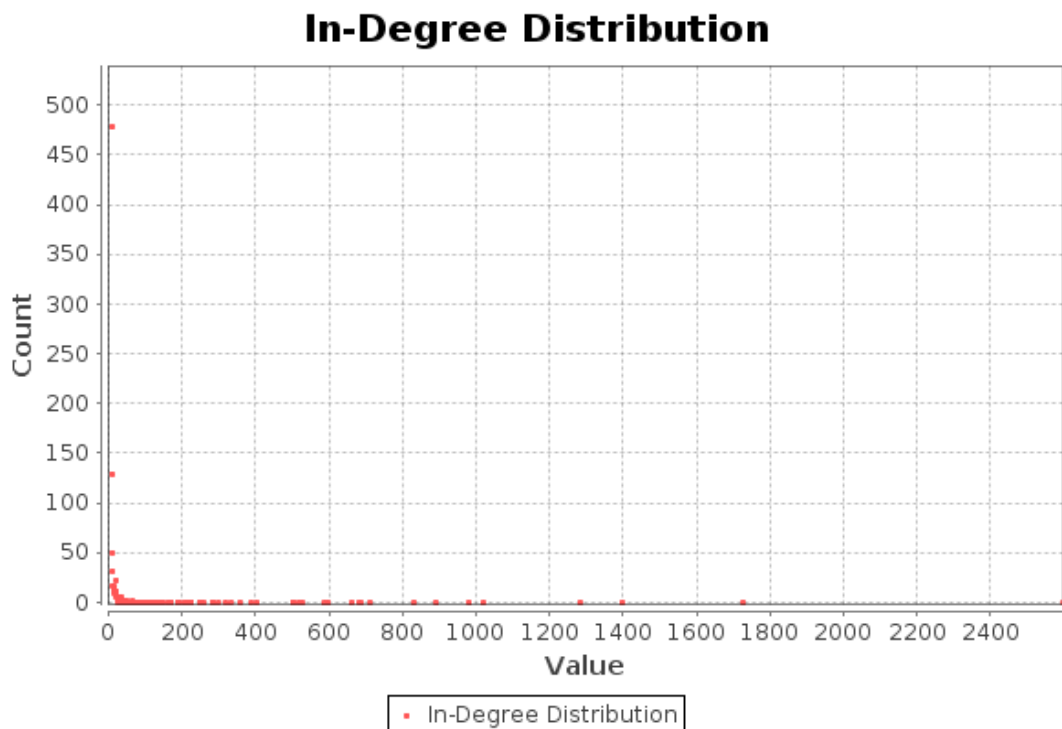
Figure 36: Network A, Raw network of the insolvency proceedings' subjects.

Before we start the main analysis of our social network from Figure 36 we will summarize its basic properties. Please note that even though Network A looks very complex it is made of very simple directed paths and sometimes it makes more sense to assume that our graph is undirected. However, if not stated otherwise we are always assuming a directed graph.

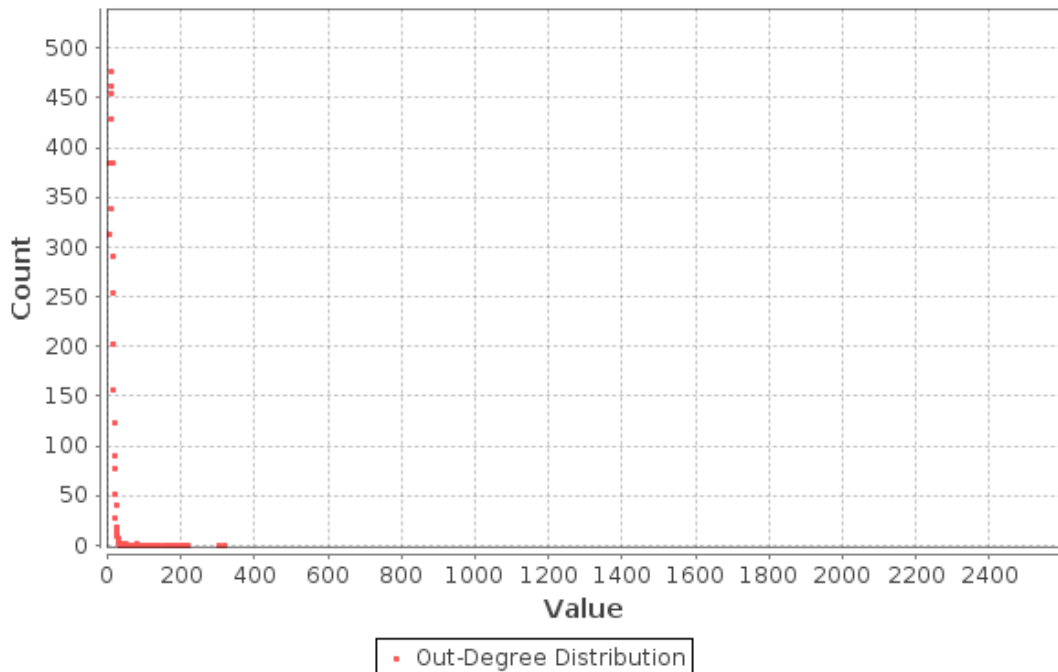
The average degree (assuming undirected for graph) of a node is equal to 6.8. The distribution of the in and out degrees is shown in Figure 37 and Figure 38. Note that if we are speaking about in-degrees of Network A then these are mostly formed

by the *owing* relation directed to the creditors. Consequently, the in-degree distribution is basically telling us the distribution of creditors over the insolvency proceedings. We can see that most of the creditors participate in under 200 insolvency proceedings. As the in-degree grows larger the number of participations grows as well. The top three most frequent creditors correspond to the same top three in Figure 34, namely *GE Money Bank a.s.*, *Cetelem a.s.* and *Česká spořitelna*.

The out degree distribution consists again mostly of the *owing* relationship but now it is pointing out of the debtors. The rest is made by the *administrators* and *share address* relationships so the interpretation is not completely accurate. However, since the number of administrators is quite small in comparison to the number of debtors, most of the administrators have quite large out-degrees. The top 50 nodes with largest out-degrees all represent administrators. The top three administrator nodes with the largest out degree are *Ing. Soňa Aubrechtová*, *Ing. Václav Dlouhý* and *Beneš v.o.s.* Consequently, we can state that most of the debtors are involved with less than 10 creditors.



Out-Degree Distribution



It may not be obvious from Figure 36 but Network A consists of 543 connected components (assuming undirected graph). However, most of these components are made of single nodes representing debtors which are not involved with the top 1000 creditors we included in Network A. These are not essential for this section and we will only focus on the largest connected component.

The diameter of a network is the longest path among all the shortest paths. It is obtained by first computing the shortest paths between all the nodes and then selecting the largest one. In case of Network A the diameter is equal to 4, which shows that it consists of very short paths. To be precise the average path length is only 1.37. In previous section we introduced two types of centralities based on path lengths, Closeness Centrality and Betweenness Centrality. Their corresponding distributions among nodes of Network A are shown in Figure 39 and Figure 40.

Figure 39 shows that the vast majority of nodes have a very small closeness equal to 1 (these are mostly creditors) and it only rarely goes beyond 2. This again shows that all nodes are relatively close to each other. There is also a large group of nodes that have a closeness 0, which contains mostly single nodes not connected to the main subgraph.

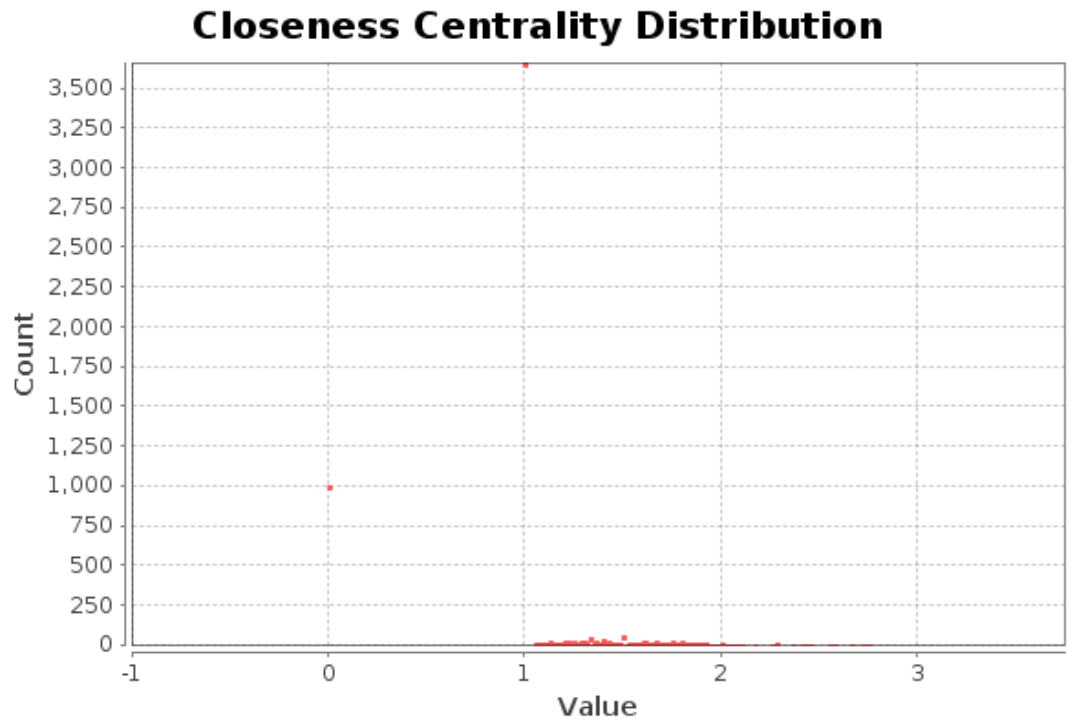


Figure 39: Closeness Centrality distribution of nodes in Network A.

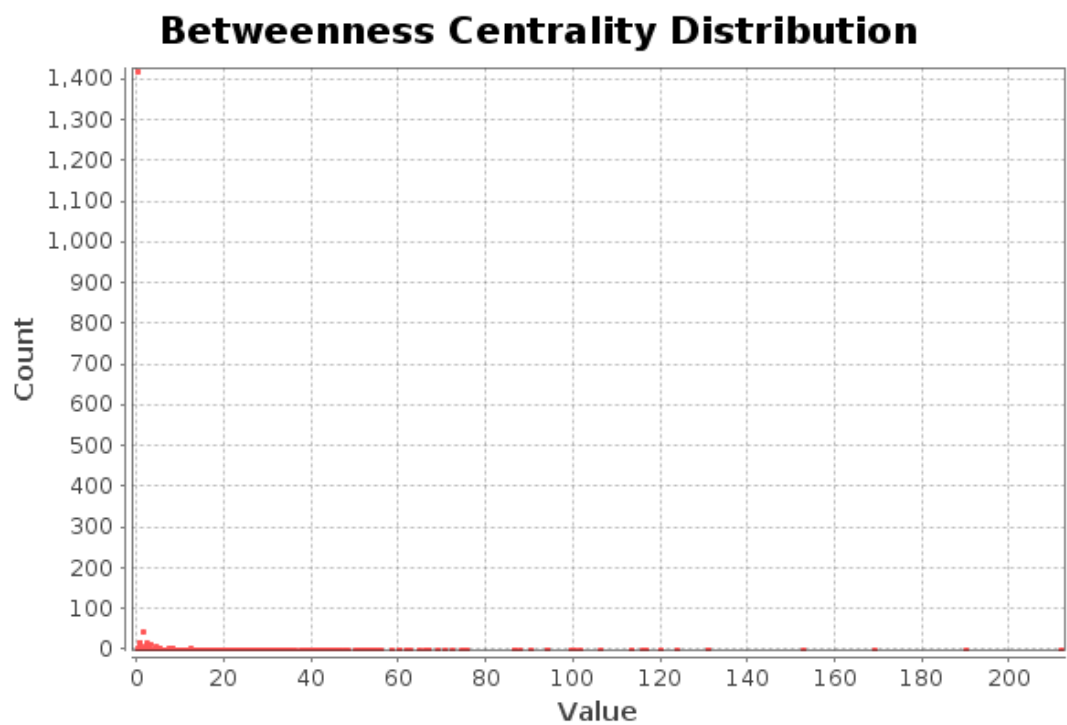


Figure 40: Betweenness Centrality distribution of nodes in Network A.

Most of the nodes have betweenness 0 as shown in Figure 40. These nodes are basically mostly all the creditors and all the administrators, since they have only

in-links or out-links, thus cannot lay on any path between two other nodes. However the rest of the nodes (debtors) have betweenness distributed quite uniformly mostly in range from 1 to 80. The debtors with the smaller betweenness are those who do not share their address. Similarly, the debtors with a larger betweenness are those who share their address (visible on the top of Figure 36).

We also tried to compute the undirected version of both the Closeness Centrality and the Betweenness Centrality, but since they are computed in a very similar manner they also yielded very similar results. Because of that we don't think they are worth including.

Next we will explore the prestige of each node in Network A by first using PageRank and then HITS. The PageRank distribution among nodes is shown in Figure 41. As we can see the PageRank values are very small ranging from $2.84 \cdot 10^{-10}$ (creditor *Úrad práce České republiky*) to 0.03 (creditor *GE Money Bank*). The top ten nodes with the highest PageRanks (due to the large number of in-links) are all frequent creditors from Figure 34. We list these top 10 creditors with their corresponding PageRanks in Table 15.



Figure 41: PageRank distribution of nodes in Network A

Creditor	PageRank
GE Money Bank, a.s.	0.031
Cetelem ČR, a.s.	0.019
Česká spořitelna, a.s.	0.017
Provident Financial, s.r.o.	0.014
Home Credit, a.s.	0.012
Essox s.r.o. Smart Capital, a.s.	0.010
Profi Credit Czech, a.s.	0.009
Telefonica CR, a.s.	0.008
AB 5 B.V., s.r.o.	0.008
AB 4 B.V., s.r.o.	0.008

Table 15: Top 10 creditors with the largest PageRank from Network A.

By using HITS we obtained very similar results to the ones ones from PageRank. Especially the authority distribution(Figure 42) is very similar to the PageRank distribution. The top 10 nodes are almost identical to those from PageRank, the only difference is on the 10th place which now belongs to *ČEZ, Prodej s.r.o.*. The top ten authorities of Network A are shown in Table 16.

The distribution of Hubs(Figure 43) is however different, the nodes with the highest hub score are debtors which share the same address. Since all the debtors are natural persons we don't want to include their names because of privacy concerns. Instead we will visualize the distribution by setting the node size depending on its hub score. This way the largest nodes in the network will represent the largest hubs. This visualization is shown in Figure 44.

From Figure 44 we can clearly see that the largest Hubs are nodes from the largest group of debtors which share their address (on the top). This is because the *share address* edge is going between each debtors that share their address. Consequently, if the largest group has approximately 15 nodes than both the number of out and in links of one particular node within that group is 14, which is a lot in the context of Network A. Since all debtors are pointing to creditors which represent the nodes with the highest Authority, they become the largest Hubs.

Notice that the previous is not true for administrators. Even though they are pointing to a large number of debtors they do not have a large hub score. This is simply because debtors have low authority scores.

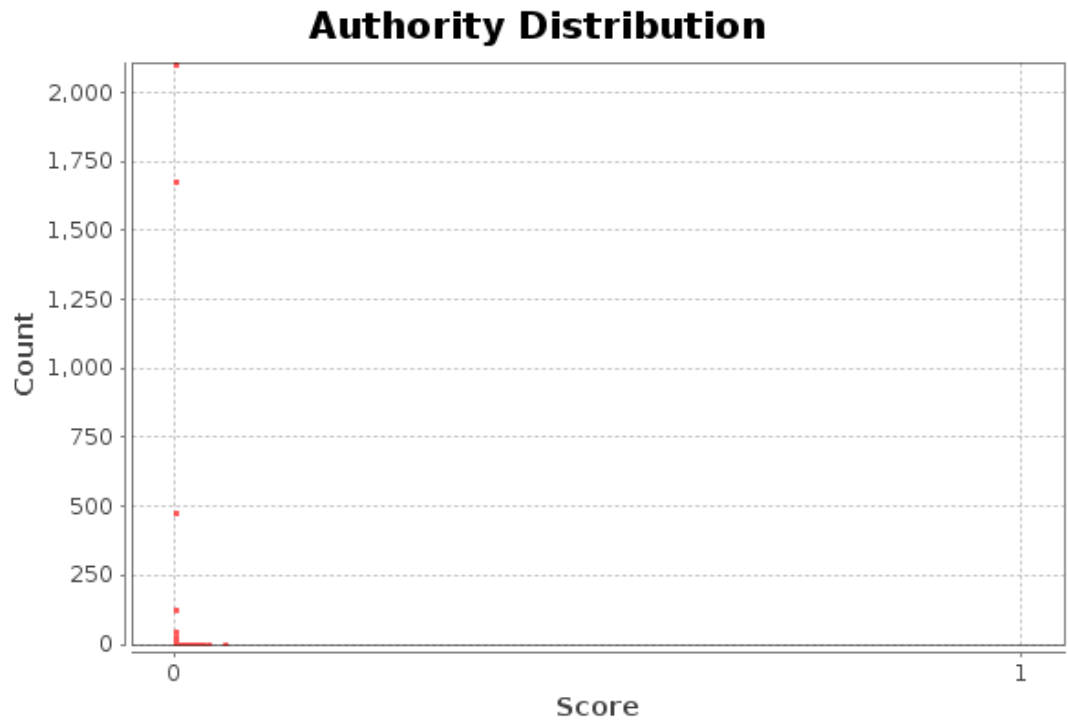


Figure 42: HITS Authority distribution among nodes of Network A.

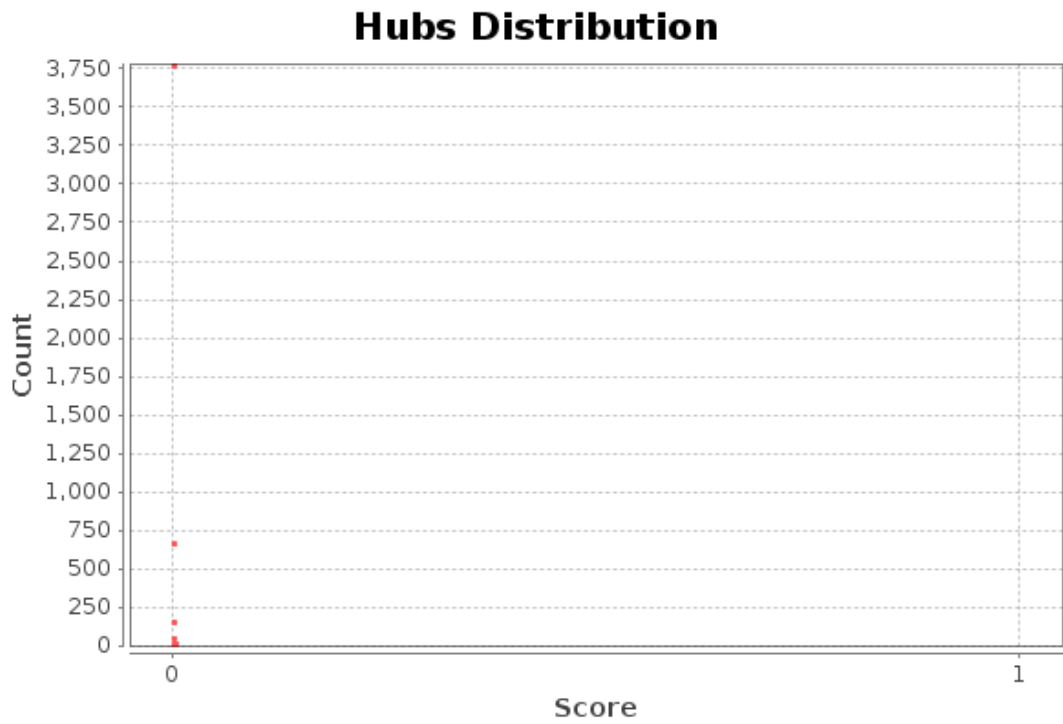


Figure 43: HITS Hub distribution among nodes of Network A.

Creditor	Authority
GE Money Bank a.s.	0.059
Cetelem ČR, a.s.	0.039
Česká spořitelna,a.s.	0.032
Provident Financial, s.r.o.	0.029
Home Credit, a.s.	0.023
Essox s.r.o. Smart Capital, a.s.	0.022
Profi Credit Czech, a.s.	0.019
Telefonica CR, a.s.	0.016
AB 5 B.V., s.r.o.	0.016
ČEZ, Prodej, s.r.o.	0.015

Table 16: Top 10 creditors with the largest Authority from Network A.

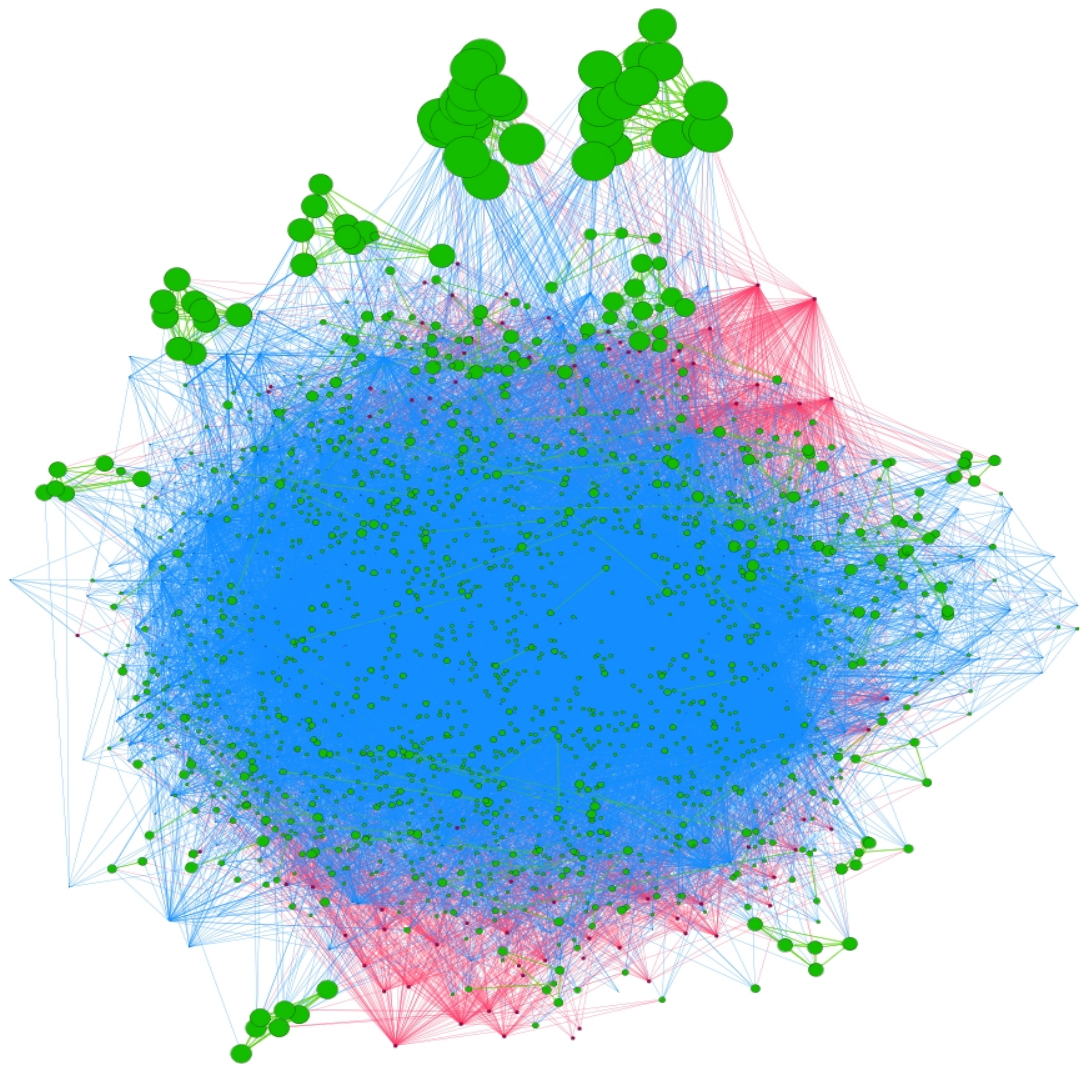


Figure 44: Network A Hubs distribution among nodes.

The last thing to explore in Network A are its communities. We started by the MOOM algorithm which divided the nodes in 557 different communities (groups). However most of the communities (520) were very small containing only 1 or 2 nodes. Gephi allows to adjust the resolution parameter of the partitioning algorithm to produce smaller or larger communities. We tried to set this parameter to values in range from 0.5 to 25, while the smaller the parameter is the more communities the algorithm discovers. The best results with sufficient granularity and meaningful communities were obtained by setting the parameter to 1.1. The smaller values failed to divide Network A to more than one large community and the larger values only grouped together debtors with their creditors. The result of partitioning with resolution 1.1 is shown in Figure 45, where the nodes and edges of each community have their own unique color.

In Figure 45 we can see 4 large communities colored blue, green, orange and purple. Each of the 4 communities contains all three types of nodes, debtors, creditors and administrators. We are listing the nodes with the largest prestige from each of these communities in Table 17.

The largest is the I. community marked green, which contains most of the frequent creditor from Figure 34 (the top 6 are all in there). Now we would like to point out the similarity of community I. with the rules from Table 14. For example they confirm that creditors *GE Money Bank a.s.*, *Essox s.r.o.* and *Cetelem ČR a.s.* truly belong to the same community because they participate frequently in the same insolvency proceedings.

The most prestigious creditors in the second largest community (II.) are mobile operators *Telefónica ČR a.s.* and *T-Mobile ČR a.s.* This community does not contain many of the frequent creditors but quite oppositely a larger number of less frequent creditors. Interestingly, the debtors in community II. have much less relationships with the frequent creditors than the debtors from community I (this is true because of the division by modularity optimization). This suggests that these two large communities of creditors should differ from each other for some reason. To examine these two communities more closely, we adjusted the selection of creditors in model D of Section 5.3.3.2 so that it contains 5 most frequent creditors from both groups.

The last two communities III. and IV. consist mostly of creditors sharing the same address. We believe that these highly interlinked clusters are the main reason

why these communities even emerged and included some of the larger creditors such as *Home Credit a.s* which they share. The two largest clusters (SA1, SA2) are highlighted on the top Figure 45. Cluster SA1 and SA2 both contain 15 debtors sharing the same address. Again for privacy concerns we do not want to include their names, instead we list their insolvency proceedings' reference numbers in Table 18. However, these debtors do not seem to be related because no last name occurs in each cluster more than once. Unfortunately, since all these debtors are natural persons it is almost impossible to find more detailed information about them.

Out of curiosity however, we looked up the addresses of both communities (III. and IV.) and found out that one address belongs to the town-council of *Chomutov* and the other to the town-council of *Most*. This led us to the conclusion that these debtors lost their homes due their debts and found shelter in the building of their local town-council. We looked into the smaller groups of debtors sharing their address as well to confirm our theory and found out that their addresses belonged to the town-councils of *Ústí nad Labem* and *Lovosice*.

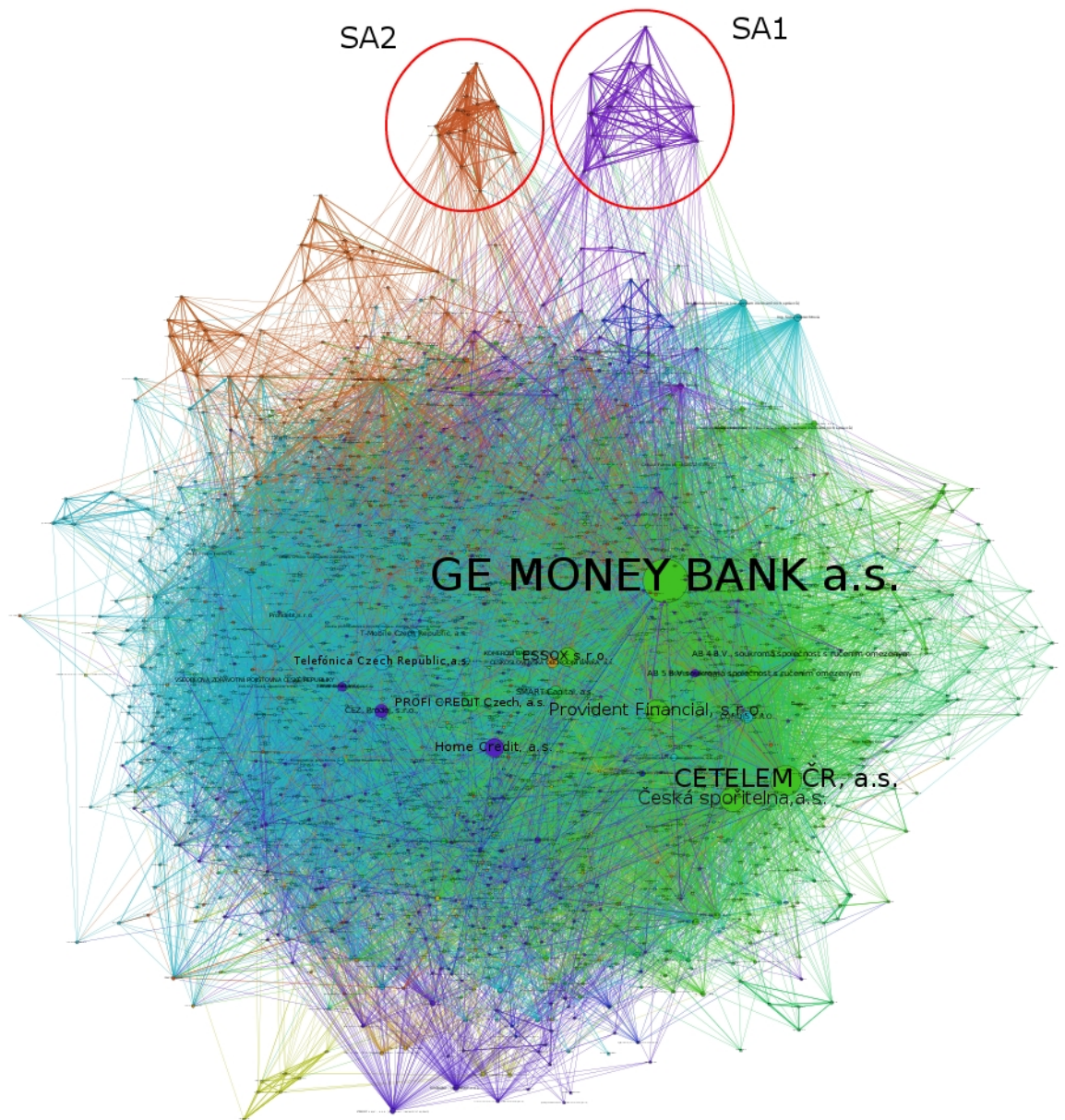


Figure 45: Communities of Network A obtained by modularity optimization.

Community	Color	Size	Node	Node Type
I.	Green	23.5%	GE Money Bank a.s.	Creditor
			Cetelem ČR a.s.	Creditor
			Česká spořitelna a.s.	Creditor
			Provident Financial s.r.o.	Creditor
			Essox s.r.o.	Creditor
II.	Blue	13.1%	Telefónica ČR a.s.	Creditor
			T-Mobile ČR a.s.	Creditor

			Profidebt s.r.o.	Creditor
			Všeobecná zdravotní pojišťovna ČR	Creditor
			Cofidis s.r.o.	Creditor
III.	Orange	12.2%	ČSOB a.s.	Creditor
			Statutární Město Ústí nad Labem	Creditor
			Košina	Administrator
			Sdružená konkurzní v.o.s.	Administrator
			JUDr. Bohumil Vintrich	Creditor
IV.	Purple	10.3%	Home Credit a.s.	Creditor
			ČEZ Prodej s.r.o.	Creditor
			Energie s.r.o.	Creditor
			Ing. Václav Dlouhý	Administrator
			Vršanský a spol. v.o.s.	Administrator

Table 17: The 4 largest communities of Network A found by modularity optimization and their most prestigious nodes.

Group	Insolvency proceedings
SA1	KSUL 44 INS 33710 / 2013
	KSUL 74 INS 32363 / 2012
	KSUL 74 INS 3385 / 2012
	KSUL 44 INS 17135 / 2013
	KSUL 44 INS 32865 / 2012
	KSUL 81 INS 22185 / 2012
	KSUL 44 INS 24558 / 2012
	KSUL 79 INS 25435 / 2012
	KSUL 70 INS 18618 / 2012
	KSUL 46 INS 4692 / 2012
	KSUL 45 INS 20959 / 2012
	KSUL 70 INS 6487 / 2012
	KSUL 69 INS 8783 / 2013
	KSUL 77 INS 3431 / 2013
	KSUL 44 INS 33710 / 2013
SA2	KSUL 46 INS 1025 / 2014
	KSUL 77 INS 15691 / 2012

KSUL 74 INS 37136 / 2013
KSUL 74 INS 18726 / 2011
KSUL 69 INS 31561 / 2012
KSUL 77 INS 3020 / 2013
KSUL 74 INS 7310 / 2012
KSUL 44 INS 28140 / 2013
KSUL 44 INS 23514 / 2012
KSUL 81 INS 26664 / 2013
KSUL 44 INS 17306 / 2013
KSUL 46 INS 10362 / 2013
KSUL 69 INS 13326 / 2012
KSUL 70 INS 13198 / 2013
KSUL 44 INS 9791 / 2013

Table 18: Listing of insolvency proceedings sharing the same domicile.

The next step in our community exploration was applying MCL clustering on Network A and comparing the found communities with those obtained by modularity optimization. In this case we tried different inflation parameters (r) ranging from 1.5 to 5. We obtained the best division into communities with inflation parameter equal to 2.0. However, even with the best result MCL was not able to divide the main cluster of nodes in Network A. The obtained communities are shown in Figure 46.

The largest community (marked green) obtained by MCL contains 81% of all nodes. The only smaller communities MCL was able to detect were those containing debtors sharing the same address (size around 1%). In comparison to Figure 45 we can see that MCL put each group of debtors sharing address into one community. However this division can be very easily accomplished by modularity optimization as well by adjusting the granularity parameter.

We can confidently say that modularity optimization method of community discovery is superior to MCL when applied to Network A. We believe that this is because Network A is very dense, meaning that it contains a large number of very similar edges and MCL just fails to divide the main cluster into smaller ones. Nevertheless, MCL also takes much longer to compute (approximately 10 times longer than modularity optimization).

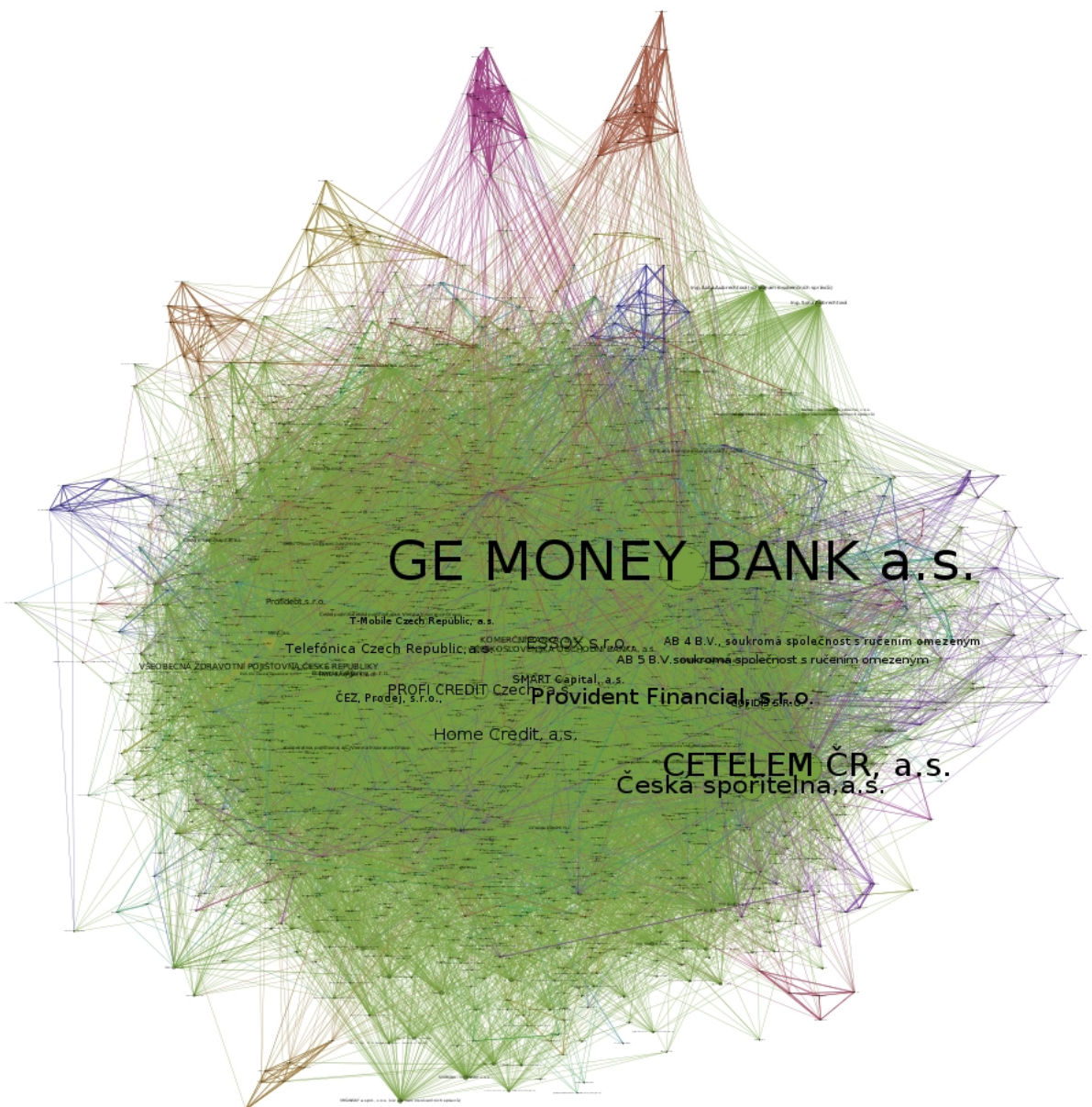


Figure 46: MCL clustering of Network A with inflation parameter 2.0.

In this section we have applied several methods on the social network of the subjects in the Insolvency Register. Some of these methods were more successful than others but even though we were able to explore the data from the Insolvency Register once more from a different point of view. In some cases (community discovery) we were able to come to similar conclusion than in the previous section. Furthermore, social network analysis provided us with a better overview of the relationships between the subjects of the insolvency proceeding. We could use this valuable information to adjust our previous model which dealt with predictions of the insolvency proceedings.

However we think that social network analysis has a much larger potential was not utilized completely in this work. The main problem is that our social network does not cover relationships between debtors. Basically the only thing we now is whether two debtors share the same administrator or creditor. We believe that it would be far more interesting to focus this research on legal entities (companies etc.), because there is a lot of information publicly available about them.

For example the Register of Economic Subjects of the Czech Republic[53] contains various information about each economic subject including the names of people involved in them. This is a huge source of additional relationships. There is also a project called the Visual Business Register[54] which provides an interactive network visualization of these subjects and their relationships. This network could be easily combined with our network of the Insolvency Register. By doing so we could study the insolvency proceedings in much more detail. For example we could explore the impact of one company going into insolvency on the other involved companies and so on.

6. Conclusion

6.1. Summary

This work is focused on the analysis, implementation, testing and evaluation of recent data mining methods. In particular, we have studied the process of insolvency proceedings in the Czech Republic and the Insolvency Register. First we defined the problem domain in detail and set possible objectives of our work. Next, we have proposed and implemented several methods for data extraction from the Insolvency Register. The extracted data was then cleaned and transformed into a form suited for data analysis. We have also assessed the quality of the data according to official statistics published by the Insolvency Register. From the evaluation we concluded that we managed to extract almost all data available in the Insolvency Register.

After the successful data extraction phase we thoroughly analyzed the obtained data. We have focused on analyzing various features of the insolvency proceedings, such as state transitions, time course and various demographical information. Based on the obtained results we divided our research into two directions, state transition analysis and social network analysis.

In the area of state transition analysis, our goal was to assess the probabilities of the insolvency proceedings moving to next states, using two popular data mining approaches, Market Basket Analysis and Bayesian networks. We have also investigated the effect of various additional features on these assessments such as the demographical data, participating subjects and time course information. The main results of this section were that the probability of the discharge method of insolvency proceedings' resolution is the largest (71%) among debtors who are natural persons. For example in the same group of debtors, the probability that bankruptcy order method of resolution will be used is only 3% . We have also shown that the information in which region the insolvency proceeding is being held has a large impact on these assessments. For example the probability of the discharge method of resolution in the region *Moravskoslezský* is 85% and in the region *Jihomoravský* only 52%. On the other hand age and gender have almost no effect on these assessments.

In the second part of our analysis we focused more on the subjects participating in the insolvency proceedings, namely administrators, creditors and debtors. In this section we constructed a social network containing these three types of subjects and

their relationships. Then, by using popular methods for social network analysis (PageRank, HITS) we were able to determine the most important subjects in the Insolvency Register (creditors *GE Money Bank a.s.*, *Cetelem ČR a.s.*, *Česká spořitelna, a.s.*).

However, the main purpose of the social network analysis was community discovery. In this work we discussed two different methods used for community discovery, one based on network clustering called Markov Cluster Algorithm and second based on network modularity optimization called Method of Optimal Modularity. We have concluded that the latter was superior for our social network of the Insolvency Register. Based on this approach we discovered that the creditors and debtors form two large groups having different characteristics. We could then use these results to adjust and improve our previous prediction models.

6.2. Future work

Despite of the good results we obtained, we also believe that there is a much larger potential in the exploration of the Insolvency Register. This applies especially for the social network analysis, where the potential is the largest. We believe that focusing more on legal entities (e.g., companies) and including additional subjects and relationships in our social network would uncover more interesting facts about the insolvency proceedings. The most positive fact about this extension is that this data is publicly available for example via the Register of Economic Subjects of the Czech Republic[53].

Also the prediction model for insolvency proceedings could be improved. The truth is that the process of the insolvency proceedings is much more complicated than what the state transition model captures. The documents published in the Insolvency Register describe the process in much more detail, but this process is also very complicated and defining it precisely would require help of a professional with a thorough understanding of the Insolvency Act.

7. Bibliography

- [1] The Company Register of the Czech Republic, Ministry of Justice of the Czech Republic, 2012 [viewed 13 March 2014]. Available from: <https://or.justice.cz/>
- [2] The Public Contracts Register of the Czech Republic, Ministry for Regional Development of the Czech Republic, 2012 [viewed 13 March 2014]. Available from: <http://www.vestnikverejnychzakazek.cz/>
- [3] C. Shearer The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 2000, Vol. 5, pp. 13-22.
- [4] IBM web documentation, IBM, [viewed 15 March 2014]. Available from: <http://pic.dhe.ibm.com/>
- [5] IBM SPSS Modeler Information Center, IBM, 2012 [viewed 13 March 2014]. Available from: <http://pic.dhe.ibm.com/infocenter/spssmodl/v15r0m0/index.jsp>
- [6] Wolters Kluwer ČR, a.s. *Insolvency Act*. Prague: Wolters Kluwer ČR, a.s., 2011
- [7] E. Kislingerová and T. Richter, and L. Smrčka *Insolvenční praxe v České republice v letech 2008 až 2013*. Prague: C.H.Beck, 2013
- [8] M. Paseková Personal Bankruptcy and its Social Implications. *International Advances in Economic Research*, 2013, Vol. 19, pp. 319-320.
- [9] XML Path Language (XPath) 2.0, W3C, 2011 [viewed 13 March 2014]. Available from: <http://www.w3.org/TR/xpath20/>
- [10] R. Baeza-Yates and C. Castillo. Balancing Volume, Quality and Freshness in Web Crawling, *In Soft Computing Systems - Design, Management and Applications*, Santiago, Chile, 2002, pp. 565-567
- [11] A. Heydon and M. Najork Mercator: A Scalable, Extensible Web Crawler. *World Wide Web*, 1999, Vol. 2, pp. 219-229.
- [12] S. Dill et al. Self-similarity in the web. *ACM Trans. Internet Technol.*, 2002, Vol. 2, pp. 205-223.
- [13] B. Suda. *SOAP Web Services*. MSc. thesis, School of Informatics University of Edinburgh, 2003
- [14] A. Virgillito. *Publish/Subscribe Communication Systems: from Models to Application*. PhD. thesis, Università La sapienza, 2003
- [15] H. S. Thompson et al., *XML Schema Part1: Structures (Second Edition)*. W3C, 2004, pp.
- [16] F. Sebastiani and C. N. D. Ricerche Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 2002, Vol. 34, pp. 1-47.
- [17] J. Thorsten. Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of the 10th European Conference on Machine Learning*, London, UK, 1998, pp. 137-142
- [18] K. Gayathri and A. Marimuthu. Text document pre-processing with the KNN for classification using the SVM, *Intelligent Systems and Control (ISCO), 2013 7th International Conference*, Tamil Nadu, India, 2013, pp. 453-457
- [19] Tesseract OCR, Google, 2006 [viewed 13 March 2014]. Available from: <https://code.google.com/p/tesseract-ocr/>
- [20] C. Patel and A. Patel Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. *International Journal of Computer Applications*, 2012, Vol. 55, pp. 50-56.
- [21] W. Bieniecki and S. Grabowski, and W. Rozenberg. Image Preprocessing for Improving OCR Accuracy, *Perspective Technologies and Methods in MEMS Design*, Lviv-Polyana, 2007, pp. 75-80
- [22] D. H. Ballard Generalizing the Hough Transform to detect arbitrary shapes.

- Pattern Recognition*, 1981, Vol. 13, pp. 111 - 122.
- [23] Insolvency Register's official statistical reports, Ministry of Justice of the Czech Republic, 2008 [viewed 13 March 2014]. Available from: <http://www.insolvencni-zakon.justice.cz/expertni-skupina-s22/statistiky.html>
- [24] Czech Statistical Office, Administration of the Czech Republic, 2007 [viewed 13 March 2014]. Available from: <http://www.czso.cz>
- [25] N. Friedman and D. Geiger, and M. Goldszmidt Bayesian Network Classifiers. *Mach. Learn.*, 1997, Vol. 29, pp. 131-163.
- [26] J. Pearl *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988
- [27] W. Buntine A Guide to the Literature on Learning Probabilistic Networks from Data. *IEEE Trans. on Knowl. and Data Eng.*, 1996, Vol. 8, pp. 195-210.
- [28] T. L. Griffiths and A. Yuille A primer on prob-abilistic inference. *Trends in Cognitive Sciences*, 2006, Vol. 10, pp. 1-11.
- [29] M. Hall et al. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 2009, Vol. 11, pp. 10-18.
- [30] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases, *Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1994, pp. 487-499
- [31] R. Agrawal and T. Imielinski, and A. Swami Mining Association Rules Between Sets of Items in Large Databases. *SIGMOD Rec.*, 1993, Vol. 22, pp. 207-216.
- [32] M. Houtsma and A. Swami Set-oriented Data Mining in Relational Databases. *Data Knowl. Eng.*, 1993, Vol. 17, pp. 245-262.
- [33] V. Vidya Mining Weighted Association Rule using FP - tree. *International Journal on Computer Science & Engineering*, 2013, Vol. 5, pp. 741-752.
- [34] P. Fournier-Viger, *A Sequential Pattern Mining Framework*. 2009, <http://www.philippe-fournier-viger.com/spmf/>, pp.
- [35] B. Liu *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Berlin: Springer, 2007
- [36] S. Chakrabarti *Mining the Web: Discovering Knowledge from Hypertext Data*. San Francisco, Ca.: Morgan Kaufmann, 2003
- [37] S. Brin and L. Page The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst.*, 1998, Vol. 30, pp. 107-117.
- [38] A. Arasu et al. PageRank computation and the structure of the web: Experiments and algorithms, *Proceedings of the Eleventh International World Wide Web Conference*, New York, NY, USA, 2002, pp. 107-117
- [39] J. Kleinberg Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 1999, Vol. 46, pp. 604-632.
- [40] R. Kumar et al. Trawling the Web for Emerging Cyber-communities. *Comput. Netw.*, 1999, Vol. 31, pp. 1481-1493.
- [41] X. Li and B. Liu, and S. Philip. Discovering Overlapping Communities of Named Entities, *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases*, Berlin, Germany, 2006, pp. 593-600
- [42] J. E. Schwartz. An Examination of CONCOR and Related Methods for Blocking Sociometric Data, *D. Heise (Ed.), Sociological Methodology*, Jossey-Bass, San Francisco, USA, 1977, pp. 255-282
- [43] R. S. Burt Models of Network Structure. *Annual Review of Sociology*, 1980, Vol. 6, pp. 79-141.
- [44] G. W. Flake and S. Lawrence, and C. L. Giles Self-Organization of the Web and Identification of Communities. *Computer*, 2002, Vol. 35, pp. 66-71.

- [45] V. D. Blondel and Vincent D, and J. L. Guillaume, and R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks, , , 2008, pp. 10008-10020
- [46] M. E. Newman. Modularity and community structure in networks, *Proc. Natl Acad. Sci.*, USA, 2006, pp. 8577–8582
- [47] G. W. Flake and R. E. Tarjan, and K. Tsioutsoulouklis Graph Clustering and Minimum Cut Trees. *Internet Mathematics*, 2004, Vol. 1, pp. 385-408.
- [48] R. Gorke et al. Dynamic graph clustering combining modularity and smoothness. *J. Exp. Algorithmics*, 2013, Vol. 18, pp. 1-29.
- [49] S. V. Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000
- [50] M. Bastian and S. Heymann and M. Jacomy. Gephi: an open source software for exploring and manipulating networks, *Proceedings of the Third International ICWSM Conference (2009)*, San Jose, California, 2009, pp. 361-362
- [51] T. M. Fruchterman and E. M. Reingold Graph Drawing by Force-directed Placement. *Softw. Pract. Exper.*, 1991, Vol. 21, pp. 1129-1164.
- [52] M. Jacomy et al., *ForceAtlas2, A Continuous Graph LayoutAlgorithm for Handy Network Visualization.* , 2009, pp. 1-21
- [53] ARES: Access to Registers of Economic Subjects, Ministry of Finance of the CR, 2013 [viewed 22. March 2014]. Available from: <http://www.info.mfcr.cz/ares/ares.html.en>
- [54] Visual Business Register, AliaWeb, spol. s r.o., 2000 [viewed 22. March 2014]. Available from: <http://obchodni-rejstrik.podnikani.cz/>

Appendix A: Installation manual

This appendix contains installation steps of each module created for the purposes of this work and all tools used for data analysis. We will start with the database, continue with the scrapers, IPredictor web application, data extraction scripts and finally third party tools.

Database

In this work we have used the popular open-source database PostgreSQL (9.1).

Installation Windows:

1. Go to <http://www.postgresql.org/download/> and download the installation package for windows.
2. Start the downloaded installer and install the PostgreSQL database and *pgAdmin* database management tool

Installation Linux:

Software repositories of most popular Linux distributions contain PostgreSQL installation packages. In a Debian based distribution install PostgreSQL by command:

```
apt-get install postgresql-9.1
```

If your distribution does not contain PostgreSQL packages download the binary from <http://www.postgresql.org/download/> and follow the installation steps.

Also install the *pgAdmin* database management tool. Again, this tool is present in the software repository of the most Linux distributions and can be installed in Debian based distributions by command:

```
apt-get install pgadmin3
```

If your distribution does not contain *pgAdmin* package you have to go to <http://www.pgadmin.org/download/>, download the *pgAdmin* sources and compile them yourself.

Database Initializaton:

1. Start the installed database.
2. Start *pgAdmin* and connect to the database, the default connection

informations are:

- hostname: localhost
 - port: 5432
 - username: postgres
 - password: postgres
3. Create a new database called *isir_prod_db* via the context menu of *pgAdmin* and connect to it. Make sure you choose the utf-8 default encoding.
 4. Open a new scripting window, then open and run the included initialization SQL script from *install/db/init_main.sql*

Scrapers

1. Download and install Java SE Runtime Environment (JRE) of version 1.7 from <http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>.
2. Java command must be set in the *PATH* variable of your system so that it can be used directly from the command line.
3. Copy the folder */install/scrapers* somewhere to your computer.
4. Read the included *README.txt* file containing details about how to start the scrapers.

IPredictor web application

The installation of the IPredictor web applications consists only from extracting */install/ipredictor/ipredictor-1.0.zip* somewhere in your computer and starting it with the included *bin/ipredictor(.bat)* script. The web application will run by default on URL <http://localhost:9000/ipredictor>.

Data Extraction Scripts

To be able to analyze our data by a specific tool, it is necessary to extract the data from the database in a format required by the specific tool. All the extraction scripts used in this work are included in *install/data_extraction*. These scripts are written in the Python language. To run them you will have to download and install Python 2.7 from <https://www.python.org/downloads/>.

It is also necessary to install the *psycopg2* python module to run these scripts.

Note for Linux users: Python 2.7 and *psycopg2* module are available in the software repositories of most popular Linux distributions.

For the exact descriptions of each script and how to use it please read the included *README.txt* file.

Data Mining Tools

We have used the following three data mining tools in this work:

- SPMF [34]
- WEKA [29]
- Gephi [50]

All three are Java based so the installation process is usually very simple consisting only from extracting archives containing the binaries.

SPMF

Download the SPMF binary from:

<http://www.philippe-fournier-viger.com/spmf/spmf.jar>.

WEKA

Download and extract the developer version (3.7) of Weka from:

<http://www.cs.waikato.ac.nz/~ml/weka/downloading.html>

Gephi

Download the Gephi installer for your operating system from <https://gephi.org/users/download/> and follow the included installation instruction.

Appendix B: Programmers manual

In this appendix we will briefly describe the modules from which the scrapers are built as well as the *IPredictor* web application. We will also provide instructions how to compile all modules from source code. However, we will not describe the source code at class level, since this description is available in the included JavaDoc that can be found in *src/scrapers/*.

The scrapers and IPredictor are all standard Java applications and to build them, Oracle JDK 1.7 is required (in addition to JRE 1.7). The JDK can be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/> and installed the very same way as JRE 1.7 in the previous section.

Scrapers

All scraper modules are covered by Unit and Integration tests (running them is not mandatory and if not interested this step can be skipped). To run all of them it is necessary to create another database called *isir_test_db* and initialize the database with the same script as in the previous section (*install/db/init_main.sql*). Please do not use the *isir_test_db* for any other purposes, before and after each test the database is cleared.

The scrapers are built from three modules: *isir.shared*, *isir.html-scraper* and *isir.ws-cache-scraper*. The first one *isir.shared* is as the name suggests a module containing shared classes among the other two modules. This module contains mostly the data access layer for working with the database. The later two represent the Insolvency Register web application and web service scrapers.

Before building the scraper modules it is necessary to build the *isir.shared* module. All three modules are built by the standard Java Maven (version 3) build tool. Maven can be downloaded from <https://maven.apache.org/download.cgi> and installed according to the installation instructions. After Maven is installed, we assume that it can be used from the command line by using the *mvn* command.

To build *isir.shared*, extract the included sources from *src/scrapers/isir.shared-1.0.0-src.jar* somewhere in your computer. The *jar* file type denotes nothing else than a regular zip file and can be extracted with any archive manager. It can be also extracted by the *jar* utility included in the JDK by command:

jar xf isir.shared-1.0.0.jar. After it was extracted go to the module's root directory (the one that contains the *pom.xml* file) and run command: *mvn clean install*¹. After the build is finished you should see BUILD SUCCEEDS on the standard output. If that is the case then everything went alright and all tests succeeded.

After *isir.shared* was built you can proceed to building the other two modules (scrapers). Again extract *src/scrapers/isir.html-scrapers-1.0.0-src.jar* and *src/scrapers/isir.ws-cache-scrapers-1.0.0-src.jar* somewhere in your computer and navigate to the root directory of each module. Then run *mvn clean install assembly:assembly*. With this command you will build the project and create the zip distribution archives which contain the same files as can be found in *install/scrapers*.

IPredictor Web Application

In order to build and work with *IPredictor* it is necessary to install *Play Framework*, a Java framework used for creating web applications. The installation is very simple, just go to <http://www.playframework.com/download> and download the latest standard distribution. Then, extract the distribution somewhere in your computer. Lastly, append the *playframework/bin* folder to your *PATH* variable so that command *play* is available from your command line.

To build *IPredictor* from source code just go to */src/ipredictor* and use command *play clean compile*. If you want to run the application just use command *play run* and it will start on port 9000. To create the same distribution package as in */install/ipredictor/ipredictor-1.0.zip* use command *play clean dist*.

¹ If you want to skip tests during build use *mvn clean install -DskipTests*.

Appendix C: User manual

In this appendix we will give an overview of the data mining tools used in this work (*SPMF*, *Weka*, *Gephi*) and how to use them to analyze our data.

SPMF

The *SPMF* tool is the simplest from all three and we used it only to generate the association rules for model *B* and model *D* in Section 5.3.3. The datasets we used are included in folder *data/spmf*. If we assume that both the *spmffjar* and the data are in the same folder we can generate the association rules with the *Apriori* algorithm with the following command:

```
java -jar spmf.jar run Apriori_association_rules modelB.spmf \
rules.txt $minSupport $minConfidence
```

By doing so the association rules will be generated to the *rules.txt* file. Before running the above command replace the place holders *\$minSupport* and *\$minConfidence* with minimal required support and confidence of the generated rules (e.g. 0.1 for both).

Note that *spmff* requires the items in the transactions to be represented as integers. Because of that it also outputs the rules containing only integer items. The mapping from integers to items can be found in *_item2name* files, for example *data/modelB_item2name*. We have also included a python script which is able to convert the integers in the generated *rules.txt* to item names in folder *install/data_extraction/*. You can do the conversion as follows:

```
python ids2names.py rules.txt modelB_item2name rules_with_names.txt
```

Weka

We have used *Weka* for the training of Bayesian network models and also for generating association rules for the most frequent creditors. We will show how these tasks can be performed in *Weka*.

Go to the *Weka* installation folder and start it simply by:

```
java -jar weka.jar
```

However, note that our data and our models are quite large and it might be necessary to allow *Weka* to use more memory. We can allow *Weka* to use e.g. 4048

MB (which should be sufficient) of memory by running it with the following parameters:

```
java -Xms1024m -Xmx4048m -jar weka.jar
```

The *Xms* parameter sets the initial memory of the Java process and the *Xmx* parameter the maximum allowed. The above command launches the *Weka GUI Chooser*(Figure 47) which servers as *Weka's* starting point.



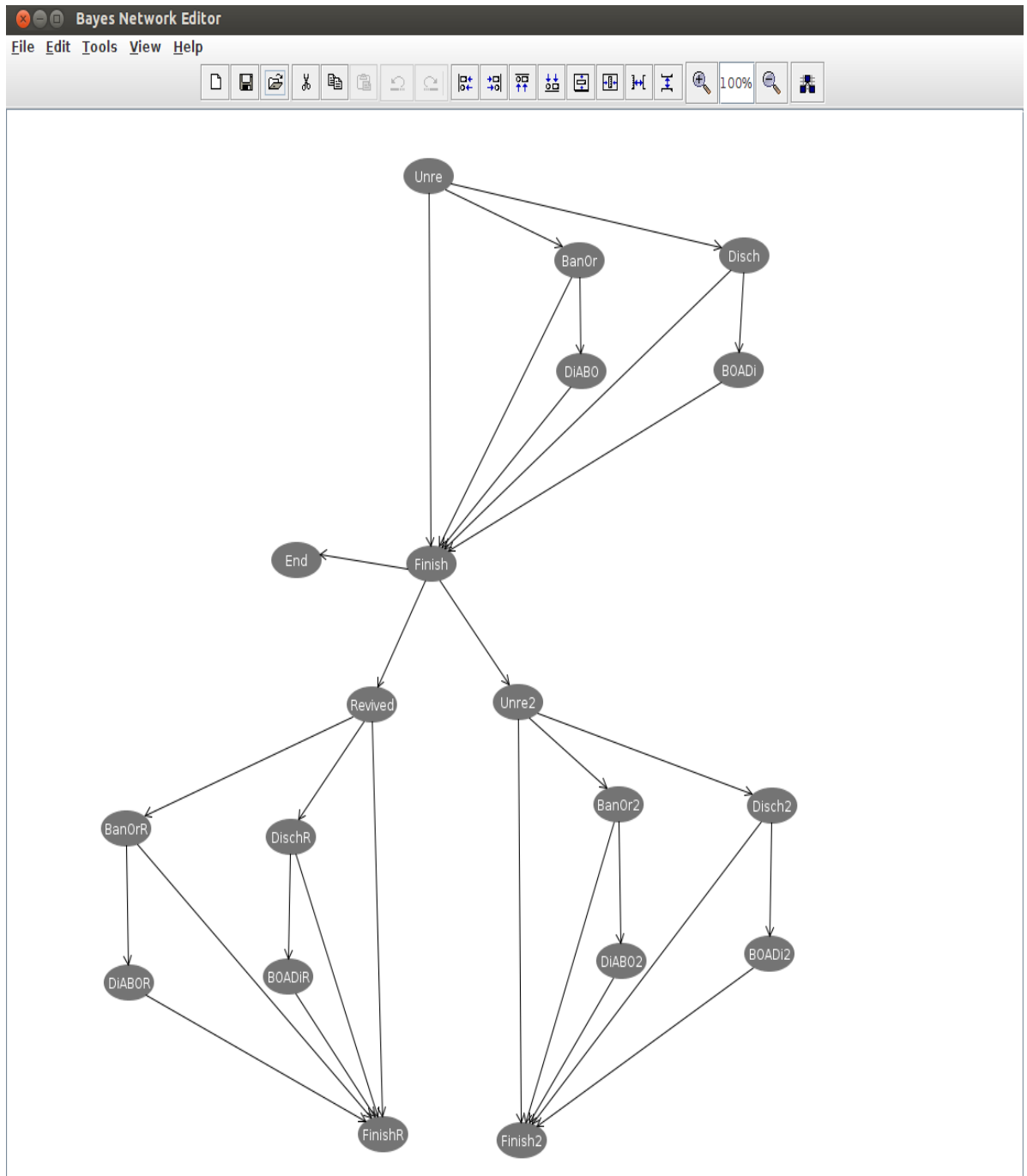
Figure 47: Weka GUI Chooser

Bayesian Networks

Weka has a separate module called the *Bayes Network Editor* which is dedicated specially for working with Bayesian networks. It can be started from the top menu of the *Weka GUI Chooser* in section *Tools*.

With the opened *Bayes Network Editor* we can load for example the Bayesian Network of model B via. menu: *File > Load* and selecting the included file */data/weka/modelB.xml*. After the the model of the Bayesian network was loaded you should see something similar as is shown in Figure 48. This is only the clear network without the learned conditional probabilities.

To learn the Bayesian network it is necessary to set the training data via. *Tools > Set Data*. The training data for model B can be found in */data/weka/modelB.arff*. Then you can proceed to learning of the loaded Bayesian network via *Tools > Learn CPT*.



To explore the learned Bayesian network proceed by setting the *Tools > Show Margins* to true. This will show the probabilities of each node right next to them in percents. You can right click on any node and set its evidence to *yes*, which in case of model B means that the insolvency proceeding passed via the corresponding node. The probabilities of each nodes will be recalculated based on the new evidence. Please note that rounding errors may occur during the calculation of the probabilities.

Association Rules

To work on association rules select the *Explorer* in the the *Weka GUI Chooser*. This component is dedicated to classification, clustering and association rule generation tasks. Each of these tasks has its own tab in the *Explorer* as can be seen in Figure 49.

We used *Weka's* association rules generation algorithms only to perform the creditors participations analysis. To perform this analysis click *Open file* in the *Preprocess* tab of the *Explorer* and select */data/weka/creditors.arff*. The *Preprocess* tab then shows you some basic statistics about the data. For example on the right side of Figure 49 we can see that *GE Money Bank a.s.* participated in 22 756 insolvency proceedings. Please note that the names of the creditors are normalized.

To generate the association rules click on the *Associate* tab and select the *Choose* button to select the algorithm. Select *Apriori* which will be shown in the list of the available algorithms. You can then click on the algorithm's name and adjust its parameters. We have used the default parameters with the exception of *lowerBoundMinSupport* and *minMetric*. The first mentioned denotes the minimal support of the generated association rules and the *minMetric* the minimal confidence. Notice, that you can change the *metricType* parameter from *confidence* to e.g. *lift*. If you select *lift* for instance, then *Apriori* will filter rules with lower *lift* than specified in parameter *minMetric* instead of confidence.

After all parameters are set click *Start* to generate the association rules. This can take a while, but after it is finished you will see them in the *Explorer's* output.

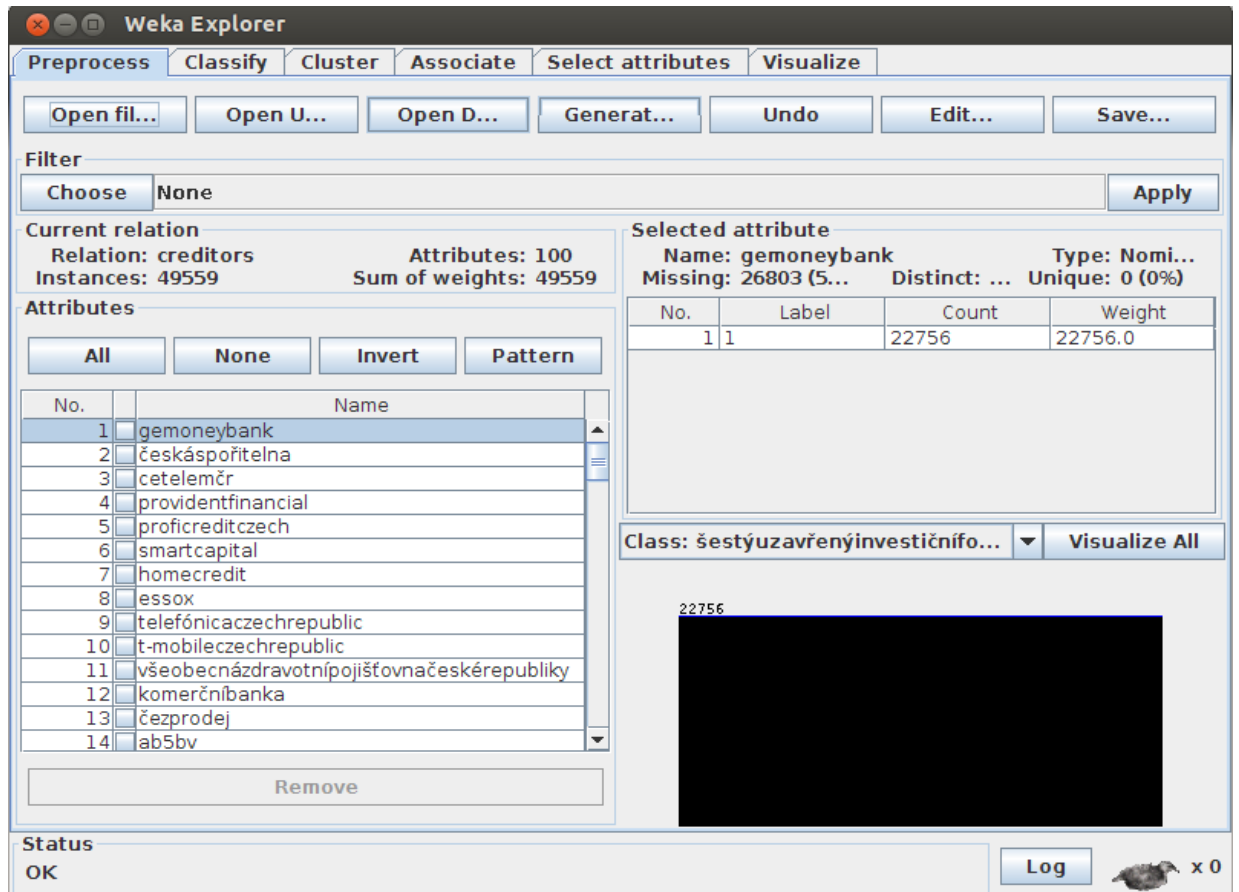


Figure 49: Weka Explorer

Gephi

The last tool we used in this work is the social network analysis tool called *Gephi*. You can start *Gephi* by the start script included during installation *bin/gephi*.

There are two ways how a network can be loaded into *Gephi*, either by a network definition in a *.gephi* file or a two CSV files containing network nodes and edges. We included files for both ways in */data/gephi*.

To load network A from a *.gephi* file go to the *File* menu and select *Open* and navigate to */data/gephi/networkA.gephi*. After this you should have network A loaded as shown in Figure 50.

To load network A via the list of nodes and edges goto *File* menu and select *New Project*. Then, go to the *Data Laboratory* and the *Data Table*. Select *Import Spreadsheet* and first select *As table: Nodes table* and navigate to file */data/gephi/nodes.tsv*. Select *Tab* as separator and click *Finish*. Repeat the whole process for the */data/gephi/edges.tsv* file with the exception of selecting *As table:*

Edges table.

The *Statistics* menu shown on the right side of Figure 50 contains most of the social network analysis algorithms we used in this work. To run them just click the *Run* button of the corresponding algorithm and after the computation is finished the results will be presented to you. On the left side of Figure 50 you will find the *Layout* menu which can be used to select a specific layouting algorithm, set its parameters and run it by clicking on the *Run* button.

Gephi is a very complex tool and its description can fit into one whole book. That is why we will not describe it any further. We find the official tutorials which can be found on <https://gephi.org/users/> to be very useful and easily understandable. We would recommend them as a starting point for working with Gephi.

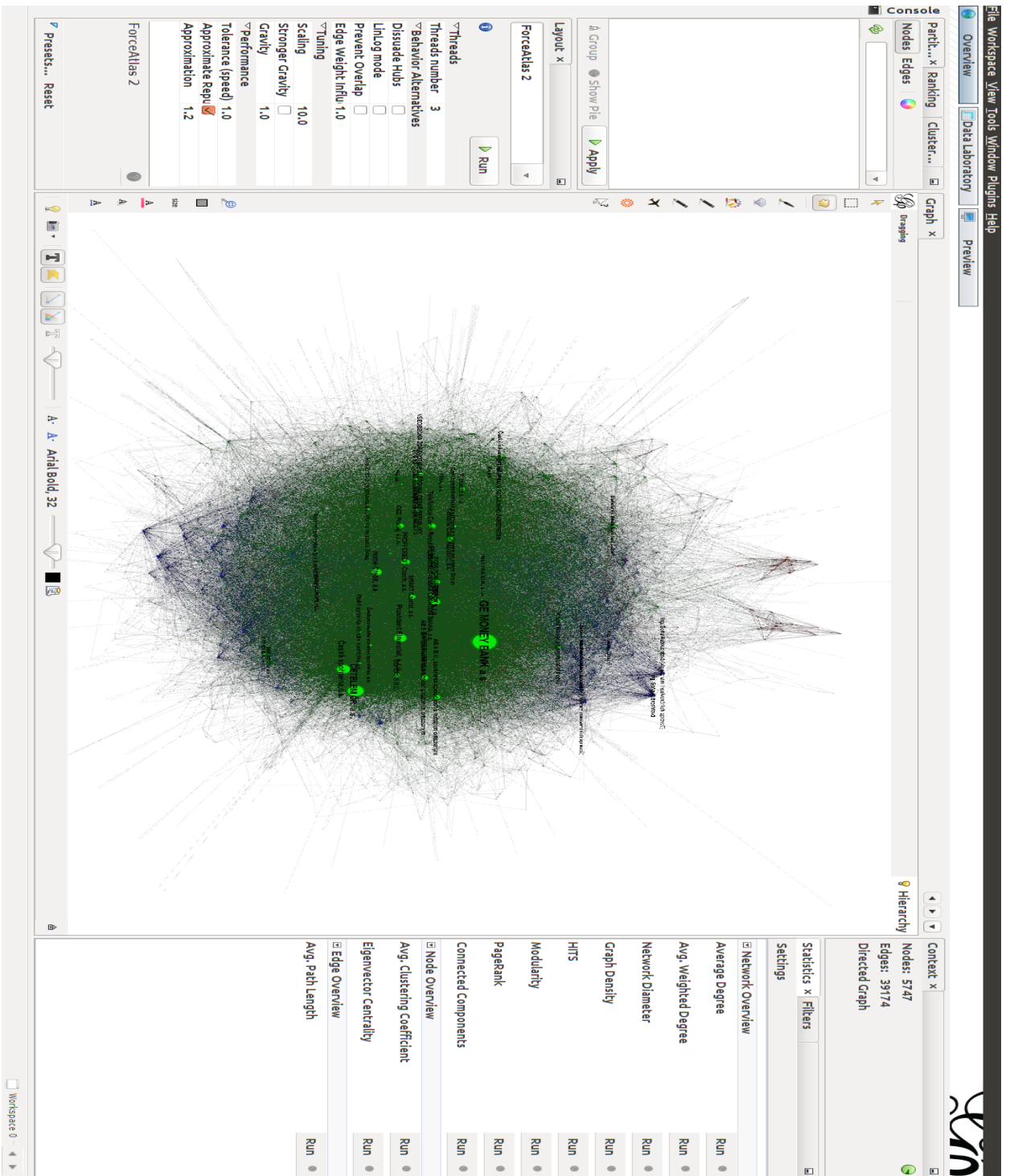


Figure 50: Gephi workbench with loaded network A.

Appendix D: Contents of the enclosed CD

An optical disk is enclosed to the printed version of this thesis with the following folder structure:

- *data* – data sets used for evaluating all models in this work
 - *gephi* – the definition of network A from Section 5.4.3 for Gephi [50].
 - *spmf* – datasets used for training model B and D by SPMF[34] (Section 5.3.3).
 - *Weka* – datasets used for training model B and C by Weka[29] (Section 5.3.2)
- *src* – source codes of all applications and scripts used in this work
 - *ipredictor* – IPredictor web application
 - *scrapers* – Insolvency Register scrapers
- *install* – compiled executables of the applications used in this work
 - *data_extraction* – scripts used for extracting data from the database
 - *db* – SQL scripts for database initialization
 - *ipredictor* – IPredictor web application
 - *scrapers* - Insolvency Register scrapers
- *thesis.pdf*
- *thesis.odt*