Data relationships and their visualization

SUMMARY OF DOCTORAL THESIS

Ostrava 2016

Martin Radvanský
Data relationships and their visualization

by

RNDr. Ing. Martin Radvanský, M.Sc.

Supervisor: Prof. RNDr. Václav Snášel, CSc.

Summary of the dissertation submitted to the Faculty of Electrical Engineering and Computer Science of VŠB - Technical University of Ostrava in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the branch of study Computer Science

Jun 2016
Data analysis stands among the main directions for research in recent years. The quantity of produced data is growing at a dizzying pace, and the results of the analysis directly affect not only the operation of companies or the State but also the life of every human being. This has had a significant impact on not only existing but also future technology. Analysed data are increasingly composed of hierarchical structures with hidden links, which are hard to detect using the classic methods of data mining. This work is focused on two primary areas of data analysis. In the first a combinations of classic methods are used for the detection of hidden links in the data. We use methods for analysis of social networks, formal concept analysis, self-organising maps and others. These methods are then used with the real data, to search for experts in co-author networks, find the performance profiles of the power plant or examine the behaviour of the authors. The second area is a visualisation of obtained results, where Sammon projection is utilised to visualise the links between data with the ability to change visualisation using the different data preprocessing method.
Contents

1 Introduction 7
  1.1 Overview ................................................. 7
  1.2 Main contributions ........................................ 8

2 Tools 9
  2.1 Data representation ......................................... 9
  2.2 Formal Concept Analysis ..................................... 9
  2.3 Basics of Graphs and Networks ............................... 9
    2.3.1 Social networks ...................................... 10
  2.4 Self Organizing Maps ....................................... 10
  2.5 Sammon Projection ......................................... 11

3 Experiments 13
  3.1 Analysis of Co–Author Network ............................... 13
    3.1.1 Data source ........................................ 13
    3.1.2 Experiment 1 – Author profiles ......................... 14
    3.1.3 Experiment 2 – Head experts identification .......... 18
    3.1.4 Chapter summary ................................... 23
  3.2 Static and dynamic network layout visualisation .......... 23
    3.2.1 Experiment 3 – Network visualisation by Sammon projection ..... 23
    3.2.2 Experiment 4 – Visualisation of network evolution ........ 29
    3.2.3 Chapter summary ................................... 33
  3.3 Solar power data analysis and prediction .................... 33
    3.3.1 Data source ........................................ 33
    3.3.2 Experiment 5 – Power profiles ......................... 35
    3.3.3 Experiment 6 – Short term prediction based on relationships between profiles ........................................ 40
    3.3.4 Chapter summary ................................... 46
  3.4 Stream data analysis using FCA .............................. 46
    3.4.1 Experiment 7 – Anomalies identification in production data ........ 46
    3.4.2 Experiment 8 – Authors publication activity trends .......... 51
    3.4.3 Chapter summary ................................... 57

4 Conclusion 59

Bibliography 61
1 Introduction

1.1 Overview

The area of data mining began on a large scale at the beginning of the 1990’s. Development of technologies for data storage has enabled us to store huge amounts of data, perhaps from all areas of human activity. This stored data itself contain hidden knowledge which has to be used. Data mining is the area that allows us to use methods such as statistics, machine learning, clustering, etc. The analysis of large amounts of data helps us to find a model that describes the data, the prediction of the future development or just search for interesting patterns in data.

Methods for data mining work mainly with data that are represented in the form of a table that contains rows of the individual observations and columns which are the corresponding values. This representation is equal to the commonly used format for example in statistical processing and takes full advantage of the benefits of store data into relational database. Data over time together with the development of information technology, are still stored in the same tabular representation, but a method of data mining are done by more sophisticated manners than before. Xindong’s article [A33], presents the so-called HACE theorem in which Big data are characterised by their natural properties and the author proposes a processing model from the data mining view.

Increasing the complexity of the process in which data are generated, meant that the data have inherently mutually intertwined internal ties and classical data mining methods are no longer able to deliver them the desired results. Over time, there have been several views on Big data. One of the widely used which describes them is the 5 V’s model [A14]: Volume, Velocity, Variety, Veracity and Value. Work with Big data; it is necessary to be focused on all of these characteristics to obtain the required results.

Many of the experiments used in this work convert classic object attribute data to the network, and with it then further works. Formal Concept Analysis is applied mainly as a clustering method on data extracted from the networks, as in the case of the search profiles of authors or experts. For the visualisation of networks, preprocessing methods are used based on weights or distances in the network that allow us to effectively visualise relationships within it. All presented experiments were published and in the corresponding articles, the description can be found in greater detail.
1.2 Main contributions

The primary goal of this dissertation is to improve analysis of complex, often high-dimensional data or networks by using data relationships as an alternative way to the widely used scenarios. This work utilises and combines ideas from pattern recognition, machine learning, graph theory, social network analysis and information theory to build an extension of cluster analysis based on data relationships.

The specific contributions of this dissertation are as follows:

- Development of a framework for a description of experts, in a co-author network, based on the combination of Formal concept analysis (FCA) and social network analysis. The evaluation demonstrates how FCA method and properties of concept lattice can help us better understand network structure. (Experiments 1 and 2)

- Proposal of an easily modifiable static and dynamic visualisation of networks based on Sammon projection and distances or weights between vertices in the network. (Experiments 3 and 4)

- Development of a method for analysis and short-term prediction, based on relationships in data originating from a small photovoltaic power station. The evaluation shows how relationship based methods could be used for short time prediction without using any extra information such as weather data or solar irradiation, etc. (Experiments 5 and 6)

- Development of a method and basic notions for working with stream data using Formal concept analysis and relationships between processed stream data and size of a concept lattice. (Experiments 7 and 8)
2 Tools

This chapter describes the methods used in this thesis. Further details of these methods can be found in the referenced documents. The authors’ extensions of the used methods will be described in the corresponding experiment. Every tool mentioned in this section was applied in a particular experiment as is noted in the text.

2.1 Data representation

The main data representations used for analysis in this thesis are in the form of a data table. This form is a generally usable type of data and is primarily used as visualisation of the stored data. This form can be formally called object–attribute data table [A1]. Object–attribute representation of data is used in all presented experiments.

2.2 Formal Concept Analysis

In this chapter are formally summarised the basic notions of formal concept analysis and provides the fundamental definitions and theorems of FCA, as established by Wille and Ganter. It is assumed that the reader has some basic knowledge of algebra. All definitions in this chapter can also be found in [A11]. As the extension of the Wille and Ganter approach (using binary context) of FCA, there exist fuzzy versions of FCA. Methods used in this dissertation are focused on the classic approach to FCA with binary context.

FCA is a method for information and knowledge analysis, data representation and management, clustering, etc. It is applied lattice theory that applies Galois connections within binary relationships in order to represent and analyse various forms of information. FCA was introduced by Rudolf Wille in the early 1980s [A32], his work extends Bernard Ganter in 1999 [A10], and published mathematical foundations and algorithms. More recent publications of these founders can be found in ([A9], [A8]). Carpineto and Romano summarized in papers ([A5], [A6] and [A7]) both the mathematical and computer scientist’s (with a focus on information retrieval) perspective of the FCA. A good overview of the recent state was also written by Uta Priss [A22].

2.3 Basics of Graphs and Networks

In this dissertation, methods of Social Networks Analysis (SNA) are used for extracting some interesting relations from network data. In experiments in Section 3.1 methods of SNA are used as the main method for preprocessing network data and the results are used by FCA to get experts from a network of authors and later visualised by Sammon projection in Section 3.2.
Social networks are usually modelled by graphs, therefore, the next few sections will briefly describe some basic notions from the graph theory as is described in [A4], [A31] and [A27].

2.3.1 Social networks

From a historical point of view, literature about social network models is rooted in social sciences. In the sixties, Travers et al. [A25] analysed characteristics of real social networks, conducting several social experiments, and in conclusion, proposed the well known “small-world” model. Kleinberg [A15], analysed this model from an algorithmic perspective, providing necessary algorithms for computing metrics on graphs representing social networks, the so-called social graphs.

Another important concept, introduced in work [A34] by Zachary, is the community structure. The author analysed a small social community of a karate club, defining a model which describes the clustering of social networks via cuts and fissions in sub–groups.

One of the first model networks is the so-called Erdös-Rényi model and employs random graphs in order to represent real networks. Watts et al. [A30], [A29] worked with a one–parameter model that interpolates between an ordered finite dimensional lattice and a random graph. They empirically found that real–world social networks are well connected and have a short average path length similar to random graphs, but they also have very large clustering coefficients, a feature which is not reflected by random graph models. Barabási [A2], [A20] and [A3] introduced different models that can be applied to friendship networks, collaboration networks, the World Wide Web, etc. He proves that they all share similar properties so it is possible to use the same approaches for their analysis.

The basic description of social networks written in the next sections is used from [A28], [A24] and [A3], where topics of social networks are described in greater detail. Static and dynamic visualisation of co–authors network using modified Sammon projection is studied in Section 3.2.

2.4 Self Organizing Maps

Self-organizing maps (SOM) were proposed by Tuevo Kohonen in [A16] and described thoroughly in [A17] and are one of the distinguished unsupervised artificial neural network models. They provide cluster analysing by producing a mapping of high–dimensional input space into usually 2–dimensional output space while it preserves the topological relationship between input data points as faithfully as possible. A SOM is formed of neurons located on a regular, usually 1 or 2–dimensional grid. Neurons are connected to adjacent neurons by a neighbourhood relation dictating the structure of the map. In the 2–dimension case, the neurons of the map can be arranged either on a rectangular or a hexagonal lattice.
The property of topology preserving has the capability to generalise. It means that a network can recognise or characterise inputs it has never encountered before. A new input is assimilated with the map unit it is mapped to. The SOMs are often used in the area of exploratory data analysis of an unknown data source for obtaining a first generalised view of the explored data and pattern recognition.

2.5 Sammon Projection

One of the methods for projecting a dataset from high–dimensional space to a space with lower dimensionality is the method introduced by Sammon [A23] in 1969. This method is based on preserving inner–point distances. This goal is achieved by minimizing the error criteria that penalizes the difference in distance between points in the high–dimensional original space and the projected low–dimensional space. We are primarily interested in projections into two and three dimensional space because obtained projection can be easily explored and evaluated by humans. A modified version of this algorithm is used for static and dynamic visualisation of network in Section 3.2.
3 Experiments

This chapter contains experiments which are focused on the analysis and visualisation of social networks, prediction and stream data. All of these experiments were published as papers in conferences. Each particular section discusses these topics: motivation, the state of the art, methodology, used algorithms and extensions of used methods and finally the conclusion.

3.1 Analysis of Co–Author Network

One type of collaboration network is the co–author network. In this type of network are rough links between the author of a paper, book, or journal article. We can identify which authors work together on particular papers, what is the authors publication activity, how often they collaborate with different authors, etc. Data used for our experiments were extracted from the DBLP computer science bibliography which is an on-line reference for bibliographic information on major computer science publications and data are freely available to the public.

3.1.1 Data source

This section provides some basic notions about the dataset used in our experiments. The Digital Bibliography & Library Project (DBLP) is one of the best-known collections of electronic resources which can be accessed over the Internet. This project was founded in 1993 and contains, among other things, more than 3.2 million publications, published by more than 1.6 million authors. These papers come from the field of computer science and were published in different journals, conferences or workshops and monographs.

For our following experiments, we downloaded the DBLP data set in XML format and it has been pre–processed for further usage. First of all, we selected journal volumes and conferences held by IEEE, ACM and Springer. For every record, we identified the month and year of the publication. In the next step, we extracted all authors having at least one published paper in the selected period (starting from the year 2000 up to the year 2010 to get the most complete data set) and we obtained 11,355 authors. Because there is no information about keywords in the DBLP records, finally we extracted keywords and phrases from paper titles. The approach of extraction was based on Faceted DBLP (which is a set of frequently used keywords in titles). For each author, we obtained a list of months with used keywords.

The next section contains the state of the art for the same class of problems, which we want to solve by our proposed method (expert identification and author profiles).
3.1.2 Experiment 1 – Author profiles

Co–author networks are studied using a different point of view. Some research papers are focused on finding an author or group of authors who is an expert in a particular area, analysis of used language in papers, the importance of authors in the network, evaluation of the importance of the paper, etc. In this experiment, we used an idea for finding author profile. More details about this experiment can be found in the published papers [C42], [C37] and [C38].

**Author profile** in this experiment is defined as a set of characteristic keywords which he used repeatedly and frequently, during the observed period. It can be seen as a reliable link between a set of keywords and a particular author.

**Motivations**

In case of analysis of co–author networks, we are interested in author characteristic keywords and we would like to group authors with the same profile, within a group of experts in a particular area. This can be interesting information for people who want to find a particular research paper and who need to quickly find as many papers and authors as possible. This is not as easy a task as it seems to be because the authors change their interest during the time and widely used methods take into consideration only some short selected period. In this experiment we would like to answer the following questions:

- Can we find the profile of authors?
- Is it possible to group authors by these profiles?
- What is the evolution of authors profiles during the time?

**Methodology**

For getting answers to the questions in the motivation section, we need to choose the right methodology and select appropriate methods to achieve the required goals. To do this, Formal concept analysis (see Section 2.2) was selected as the main method. It is not very common to use FCA as a method for this analysis, but there exist several advantages for using it.

- FCA is a method which can work directly with object attribute table, containing relations between authors and keywords.
- By the basis of FCA, the results (formal concepts) can catch in the lattice structure, cases when an author changes the keywords during the time.
- Concept stability help us filter large lattice to get interesting intents (authors profiles) with a choice of filtering threshold.

A block diagram of the process for extracting author profiles is displayed in Figure 1.
Preprocessing

This phase of the experiment was very important. We have to pre-process data to satisfy our requirements as defined in the profile. We wanted to work with keywords which were used by authors repeatedly and frequently. To achieve this goal we worked with author weighted keywords. To calculate the weight (importance) of keywords used in the titles of author’s papers we have applied a forgetting function (a detailed description of the function can be found in [A19]).

This function is based on the simple hypothesis inspired by nature. In our case, we assign a default weight for the word used for the first time. If this keyword is used regularly and frequently, then the weight gradually increases. If it is not used, the weight is gradually reduced (the word is forgotten). The forgetting allows us to naturally reduce noise in the data. This method marginally reduced the size of data which are necessary to handle. We obtained 1,735 authors and 525 keywords (at the end of the selected period 2010) for weights higher than the threshold.

In this part of the experiments, we obtained some interesting results. Eleven yearly cumulated datasets of authors and their topics were filtered. At the end of the year 2010, we obtained 1735 authors who published with more than two keywords in their papers. Filtering removed keywords which were not used repeatedly and therefore many authors were removed due to this filtering.

The next step of preprocessing was to create binary formal context, which represents authors of papers in rows and columns corresponding to the keywords used in a particular paper in the DBLP. The value of an intersection between row and column was set to “1” when the author used a keyword, or “0” otherwise. The density of the binary context for year 2010 was about 4.9%.

Computing set of all concepts and filtering it using stability index

The FCA gave us a tool for finding profiles of authors, based on the keywords they use in papers. For the context created in the previous step, we have computed a concept lattice. Number of concepts is displayed in Figure 2, for year 2010, 2130 concepts in lattice were computed. Every concept in concept lattice contains in its extent set of authors and in the intent set of keywords used by authors in titles of published papers.
We are interested in non–trivial concepts where author profiles are the intents of these interesting concepts. In order to decide which concepts are interesting for us, we used the concept stability index. This method helped us reduce the size of concept lattice, and intents of particular concepts create profiles of authors. For finding the correct level of stability index threshold, we have calculated all concepts with the particular level of stability.

**Profiles extraction and its evolution during time**

Choosing the threshold of stability index has been done according to the basic meaning of stability. Stability 0.5 gives us information that there exist one–half of subsets of all possible subsets of authors in the concept that have a special property. Removing subset authors from concept extent (set of authors) does not cause a change in the intent (set of keywords) of concept. A higher value of concept stability makes the concept more confident. We have used concept stability for pruning concept lattices. After applying the pruning lattice by concept stability, we obtained a relatively small number of concepts which can be easily explored and concepts have a high number of keywords in their intents.

Although the reduction by the stability index 0.5 of the number of concepts is not very high (about 22%) we can extract author profiles with up to eight keywords. It helps us describe authors using a higher number of characteristics. According to the selected level of stability, the most interesting concepts for creating author’s profiles were concepts with more than one attribute. Author’s profiles are concepts that contain particular keywords in their intents.
Hierarchical structure of concept lattice and stability index provide us with the necessary information to create author profiles. We select the formal concepts with stability index higher than 0.5 and for particular keywords we should extract from lattice structure, a list of keywords ordered by the stability index. This list of keywords creates the authors profile. To find authors with this profile we combine all extents of concepts used for it. A higher stability index has concepts with just a single keyword in intent and it describes the most frequently used keywords in topics of papers.

Examples of concepts and their profiles are shown in the following tables (1, 2).

<table>
<thead>
<tr>
<th>ID</th>
<th>Stability</th>
<th>Extent</th>
<th>Intent</th>
</tr>
</thead>
<tbody>
<tr>
<td>704</td>
<td>0.8125</td>
<td>Vijay V. Vazirani, Steve Benford, Tak-Wai Chan, Larry Korba, Tatsuhiro Yonekura</td>
<td>games</td>
</tr>
<tr>
<td>15</td>
<td>0.5</td>
<td>Tak-Wai Chan</td>
<td>computing, games, case study, learning, curriculum</td>
</tr>
<tr>
<td>80</td>
<td>0.5</td>
<td>Vijay V. Vazirani</td>
<td>algorithms, games</td>
</tr>
<tr>
<td>303</td>
<td>0.5</td>
<td>Larry Korba</td>
<td>agents, games</td>
</tr>
<tr>
<td>340</td>
<td>0.5</td>
<td>Steve Benford</td>
<td>design, games, virtual worlds</td>
</tr>
<tr>
<td>705</td>
<td>0.5</td>
<td>Tatsuhiro Yonekura</td>
<td>games, education</td>
</tr>
</tbody>
</table>

Table 1: Concepts with keyword “games” and stability index ≥ 0.5 for year 2005

For each author, we can create a profile based on concepts in Table 1. Author profiles containing keywords ordered by the stability index are displayed in Table 2.

<table>
<thead>
<tr>
<th>Author</th>
<th>Author profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tak-Wai Chan</td>
<td>games (0.8125), case study (0.5), computing (0.5), curriculum (0.5), learning (0.5)</td>
</tr>
<tr>
<td>Vijay V. Vazirani</td>
<td>games (0.8125), algorithms (0.5)</td>
</tr>
<tr>
<td>Larry Korba</td>
<td>games (0.8125), agents (0.5)</td>
</tr>
<tr>
<td>Steve Benford</td>
<td>games (0.8125), design (0.5), virtual worlds (0.5)</td>
</tr>
<tr>
<td>Tatsuhiro Yonekura</td>
<td>games (0.8125), education (0.5)</td>
</tr>
</tbody>
</table>

Table 2: Author profiles for a particular area of research (keyword “games”)

Figure 3, shows the hierarchical structure of profile, where the root node is a keyword with the highest stability index and numbers in brackets indicate the number of authors who used this keyword. Grouping authors by their profiles is possible using the hierarchical structure of keywords and in case of using one keyword with the highest stability index the result is a
When we increase the number of keywords in the profile, the group of authors assigned to the profile will shrink but we obtain a more precise area of research.

How the profiles change over time, depends on the preprocessing phase, where we remove keywords which were not used regularly and repeatedly. The main keywords, which are used by authors are extended by new ones and the profiles grow rapidly to cover new challenges and methods. This should be seen as the seasonal component where authors used a combination of popular keywords to publish their works. The evolution in the profile for root keyword “systems” is depicted in figures (4 and 5).

### 3.1.3 Experiment 2 – Head experts identification

One of the most important subjects for mining from social networks is expert finding. The main aim of this experiment is to identify the person with high experience in a particular area.
Results of our approach were published in the papers [C42] and [C36], where the used method is described in greater detail.

**Motivations**

Expert, in the widely used meaning, is a person with relevant expertise or experience on a given topic. During our experiments with finding experts, we have identified a special kind of expert who was different from others. We have defined these experts as **Head experts**.

The **Head expert** is a person who is building a team of other experts. He is in a position such as a manager in a company. He can give ideas for other people to start some research, or focus them on a particular field of research. This person is identified as the expert in many (often different) areas. His advantage is a wide knowledge of different areas and the ability to manage people. Including this head expert into a working team increases the probability of achieving the team goals.

In our experiment, we have tried to design a method for identification of this particular type of expert in co-author networks.

**Data source**

The data source in this experiment is the DBLP and its main characteristics are described in Section 3.1.1 but we worked with a snapshot of the DBLP database from December 2011. Unlike the previous experiment, we worked only with authors and relations between them.

**Methodology**

To achieve our goal we have used a combination of two methods. Firstly, we have used a method of network analysis – extracting communities from co-author networks. Secondly, Formal concept analysis (see Section 2.2) was used as a method for clustering extracted communities. The whole process of head expert identification and extraction is depicted in Figure 6. Our proposed method is different from commonly used approaches for expert mining, in particular, by the used tools and their combinations.

Advantages of our approach and selection of the used tools include:

- In the preprocessing phase, a forgetting curve is used to remove outliers (authors who published only occasionally), they will not be Head experts.
• Local communities extraction is based on vertex dependency in the network. This approach helps us find communities in a better way. We do not need to work with the whole network but just with some of the surrounding of interesting vertices. This method is similar to other egocentric approaches.

• Clustering by FCA is very well applicable to a relatively small network and clustering is more exact by the mathematical foundations of a method than other methods (K-mean, SOM, etc.).

• Due fact, that the Head expert can exist in multiple communities simultaneously, we can use the natural property of concept lattice (hierarchical order of concepts) to find them.

Data preprocessing
From the DBLP database snapshot in December 2011, we have created a weighted network, where weight includes relation to other authors. After that, we have performed a network denoising based on a forgetting curve similar to the experiment in the previous Section 3.1.2. We have obtained a sparse weighted network with 57,747 vertices (authors) and 1,787,356 edges (co-authorship). Although this network is very big, in our approach it is not necessary to work with the whole network, but we are interested in communities. From this network we have obtained, by the algorithm of local community detection, 26,326 communities. The largest community contains 36 authors, and the smallest contains two authors. At the end of this phase, we obtained 26,326 communities containing 57,747 authors. The next step is the clustering of communities over their authors.

Clustering
The main data structure for computing clusters with the help of FCA, is formal context. By examination of the communities, we can build context, which contained 26,326 communities (objects) and 57,747 authors (attributes), with density 0.497%. Although there exists methods for computing concept lattice from this huge context, working with lattice could be very time-consuming. For working with large concept lattice, we have to choose a different approach.

We utilised an alternative way of how to compute a concept lattice. First of all, we examined the obtained context. Despite the context size being very large, some authors and communities can be identified there, which can be seen as subcontext of the main context. These subcontexts are independent of each other. Therefore, we can compute each of them separately and we obtain sublattices. In case we need whole concept lattice, we can combine these sublattices.

How the number of communities is related to the subcontext is depicted in Figure 7.

From the source context, we extracted 13,956 independent subcontexts. The major part (9,095) contains only a single community. Therefore, they are not usable for computing a
concept lattice and do not satisfy our requirements for Head experts (must be a member of several communities). The size of the biggest subcontext is 102 communities and 154 authors. Computing concept lattice for the context of mentioned size is a very quick task. For our analysis, we have selected all subcontexts containing ten or more communities (total 192). Selection of subcontext with less than ten communities meant that authors were in communities with very similar areas of research and therefore do not meet our criteria for the Head experts.

### Selecting head experts

During our experiments, we have computed 192 subcontexts and we obtained all concept lattices. The final step of Head experts extraction process is browsing conceptual lattice, starting from unit concept, in a top to bottom direction. We were looking for coatoms of concept lattice. These concepts have the special property – they are in the root of sublattice and in their intent is a minimal number of authors. These authors are founding members of several communities, therefore, they satisfied requirements for Head experts. In our approach, we always omitted zero elements of subcontext, because it does not contain useful information for us. The unit element is used only in the case, where it contains some authors in intent. These authors are forming members of all communities. From subcontext containing a list of communities and the names of the author we created a binary formal context and computed a set of all concepts and concept lattice, unit and zero concepts were omitted.

Most of the concepts contain in the intents exactly authors contained in the communities computed in the first step of our algorithm. The rest of the concepts are formed by the FCA
algorithm as intersections of communities and authors. All of these concepts are interesting for us, but we need to detect Head experts. They are by the meaning of this paper, authors contained in the concepts that have just a single (or more) author in their intents and they are coatoms of concept lattice. These concepts are depicted in Figure 8 as black nodes. Circle–shaped nodes are concepts in which intents are exactly equal to authors in the community. A diamond–shaped node is a concept which is the intersection of other concepts.

In this example, three Head experts were identified for ten communities subcontext. They are contained in concepts $c(1)$ and $c(3)$ namely they are Jose Luis Calvo–Rolle, Ramon Ferreiro Garcia and Emilio Chorchado. Although we know the Head expert by the observation of concept lattice it is possible to order these concepts (authors) by their significance $\text{sig}$. It can be done by summing of the concepts which are equal to the particular community and omitted concepts which have been created as an intersection of authors by FCA and dividing it by a number of concepts (Equation 1).

$$\text{sig}(c_h) = \frac{|\text{sublattice}(c_h)|}{|\text{communities in subcontext}|}$$

(1)

Where $|\text{sublattice}(c_h)|$ is number of all concepts in sublattice, concept $c_h$ is the unit node of sublattice for particular subcontext. A higher value of significance yields an author more
confident as Head expert in a particular subcontext.

3.1.4 Chapter summary

In this chapter, an experiment working with co-author network was presented. We used an approach which combined several methods. We performed preprocessing of network data to reduce the size of working data. Formal concept analysis was used to find author profiles which helped us to describe the authors by their keywords and groups of authors by their research activity. Another way of how to use the author profile seems to be to use it to describe the evolution of research areas. We should identify the seasonal part of profiles and authors who change their preferences by the attractiveness of particular topics. Identification of Head experts was based on a combination of local network community extraction and using properties of concept lattice. This method gave us the possibility to identify the founder(s) of local communities in a social network.

3.2 Static and dynamic network layout visualisation

Visualisation is an important part of Network Analysis. It helps us find features of the network that are not easily identifiable. Although there exists several methods during our exploration of co-author networks we found it useful to visualise some parts of the network in more flexible ways using modified Sammon projection. In the following experiments, we use an approach to the visualisation of weighted networks based on Sammon projection (described in Section 2.5). We propose several methods for construction of the input for Sammon projection and observe the effect of the particular methods on the final layout. Results are illustrated using several networks in the 2D layout. The presented experiments use the well-known Karate club network and weighted co-author networks based on the DBLP database. The experiments were published in conference proceedings [C39], [C40] and in the journal [B35].

The static visualisation of network gave us a good overview of network structure, but some networks like co-author networks are a very dynamic environment. Our extension of static visualisation should catch some particular aspects of the author’s behaviour. This is studied in the experiment in Section 3.2.2. The experiment was published in conference proceedings CASoN 2013 [C41].

3.2.1 Experiment 3 – Network visualisation by Sammon projection

Motivations

Visualisation of networks in different ways together with the possibility to compare results is a big advantage for the understanding of relationships in networks. In this experiment, we
would like to develop a method for network visualisation without changing a layout algorithm, which could give us results in a comparable form and will be easy to apply to different kinds of network data.

**Methodology**

In order to achieve the desired objectives, we have to select a method for layouting of the network. It is necessary to select a method in which network layout can be easily modified by preprocessing of data. While we were working with Sammon projection (Chapter 2.5) we found it to be a good method for our approach. The advantage is that this method works with matrix of distances and dimension reduction based on it. This method minimises the error arising from the projection into a preselected dimension. In case of reduction to the 2D or 3D space we can directly visualise results as network and by the modifying of the input matrix we can obtain different visualisation. To get useful results it is necessary to prepare input distance matrix in a way that satisfies our visualisation requirements. Figure 9, depicts a process diagram of network visualisation.

Sammon projection as described in Chapter 2.5 is used on distance matrix and results are visualised in 2D/3D by our developed application. Preprocessing data – create distance matrix is the main part of visualisation process. In the following paragraphs, our approach based on vertex distance and vertex dependency is described.

**Preprocessing – building distance matrices**

In our approach, we work with the weighted undirected network. The weight is assigned to both vertices and edges of the network. We further assume that the weight of the vertex is always greater or equal to the weight of the incident edge, but the method is valid for the unweighted networks as well. The result of the projection does not guarantee the maintaining of the exact distance, but estimates the layout of the objects within the lower dimensional space. Therefore we cannot easily predict how the distances between vertices after projection will correspond to the real proportions. The projection is mainly based on the structure of the network and on the weights of vertices and edges.

We have two options of how to prepare distance matrix for projection:
1. Arranging the vertices of the network in the high-dimensional space and then calculate the distance between the vertices (we use Euclidean distance in our experiments).

2. Working directly with vertex similarity used in the role of distance between vertices.

**Vertices arranged by weights (WSPC)**

This method is based on arranging vertices and we then create a WSPC matrix for all of them. We arrange vertices as follows:

1. Dimension of the space is equal to the number of vertices.
2. Each particular axis of the coordinate system represents one particular vertex.
3. As distance from the beginning of the axis, vertex weight is used.
4. This axis contains the coordinates of the remaining vertices (weights of the edges incident to this vertex)

Let’s consider a network with $n$ vertices and particular vertices $V_i$ and $V_j$. The $i$-th coordinate of the vertex $V_i$ is the weight of the vertex $V_i$. The $j$-th coordinate of the vertex $V_i$ is the weight of the edge $(V_i, V_j)$ and $i$-th coordinate of the vertex $V_j$ is again the weight of the edge $(V_i, V_j)$.

All vertices of the network are represented as points in the hyper-rectangle of the dimension $n$ having one vertex in the beginning of the coordinate system. Sides of the hyper-rectangle are weights of the vertices. The largest theoretically possible distance between the vertices of the network is the length of the longest diagonal of this hyper-rectangle.

For all obtained points we calculate the Euclidean distance and store it within the symmetric distance matrix $A$ of dimension $n$. Particular elements $A_{ji} = A_{ij}$ of the matrix represent the distance between vertices $V_i$ and $V_j$.

Transformation of the network into the space of dimension $n$ (with axes $x_1, ..., x_n$) may be interpreted as follows:

1. Axis $x_i$ represents a direction in which the vertex $V_i$ increases/decreases its weight.
2. The coordinate of a different vertex $V_j$ on the axis $x_i$ represents a similarity between vertices $V_j$ and $V_i$.

In order to use this WSPC matrix for Sammon projection it is necessary to compute the distance matrix by using Euclidean distance.
Weight based vertex distance (WDST)

In this paragraph is introduced a method for creating the WDST matrix. Distance matrix $A$ for $n$ vertices has order $n$. Matrix element $A_{ij}$ represents the distance between vertices $V_i$ and $V_j$. This matrix is symmetric. In order to calculate the distance between any two vertices in the network, we have to calculate the largest possible distance. In this method, we set this value with respect to the vertex having the largest weight $W_{\text{max}}$ is the weight of the vertex with largest weight, $W(V_i, V_j)$ is the weight of the edge between vertices $V_i$ and $V_j$. Distance matrix is calculated as follows:

1. if $i = j$, $A_{ij} = 0$
2. else $A_{ij} = A_{ji} = W_{\text{max}} - W(V_i, V_j)$

All vertices are represented as points inside the hyperball of dimension $n$ and diameter $W_{\text{max}}$. The main diagonal of this matrix contains zeros, which do not have any effect during the calculation of the projection. By these properties, the matrix $A$ may be used directly as an input for the Sammon projection.

Dependency measuring

In this section, a different distance measuring approach based on dependency is shown. We understand the dependency as a local unsymmetrical feature of a pair of vertices, which have at most a distance equal to 2. More distant vertices are considered as being independent, therefore, the dependency value is zero.

Let $E(x)$ be the set of all edges adjacent to the vertex $x$. $\text{Adj}(x, y)$ is the set of all edges between the vertex $x$ and any of the neighbours of the vertex $y$ ($\text{Adj}(x, y) \subseteq E(x)$). $W(e)$ is the weight of an edge $e$ and $W(v_1, v_2)$ is the weight of an edge between vertices $v_1$ and $v_2$. We assign a value $W(v_1, v_2) = 0$, if there does not exist such an edge.

Let $x$ be not an isolated vertex of the network. The dependency $D(x, y)$ (was introduced in [A18]) of the vertex $x$ on a vertex $y$ is defined as follows:

$$D(x, y) = \frac{W(x, y) + \sum_{e_i \in \text{Adj}(x, y)} W(e_i) \cdot R(e_i)}{\sum_{e_i \in E(x)} W(e_i)},$$

(2)

$$R(e_i) = \frac{W(y, v_i)}{W(e_i) + W(y, v_i)},$$

(3)

where $R(e_i)$ is the coefficient of dependency of the vertex $x$ on the vertex $y$ via the common neighbour $v_i$, therefore $v_i \in e_i$.

The dependency describes a relation of one vertex on another vertex from the point of view of their surroundings. Presented equations infer $D(x, y) \in (0; 1)$, meaning that dependency...
if equal to zero captures the state when vertices \( x \) and \( y \) have no common edge or neighbour. The full dependency \( D(x, y) = 1 \) describes a situation where vertex \( x \) has only one common edge with vertex \( y \). The dependency \( D(x, y) > 0 \) occurs in the case where there exists an edge between vertices \( x \) and \( y \) or they share at least one common neighbour.

**Vertices arranged by dependency (DSPC)**

This method is based on the WSPC approach presented above, but the difference is that the maximum value for all axes of the coordinate system is set up to 1. It corresponds to the dependency of a vertex on itself which is also defined as 1. The coordinates of remaining vertices are plotted on each axis. The exact value is calculated as a dependency on the corresponding vertex.

All vertices of the network are therefore represented as points in the unit hypercube of dimension \( n \) having one vertex in the origin of the coordinate system. The largest theoretically possible distance between vertices of the network is \( \sqrt{n} \). Euclidean distance matrix \( A \) of an order \( n \) which serves as an input for the Sammon projection was calculated for all points.

In this paragraph transformation of the network into the space of dimension \( n \), was introduced, with axes \( (x_1, \ldots, x_n) \). Axis \( x_i \) represents the vertex \( V_i \) which has a value 1 on this axis and the coordinate of a different vertex \( V_j \) on the axis \( x_i \) represents a similarity between vertices \( V_j \) and \( V_i \).

**Dependency based vertex distances (DDST)**

This method is similar to the WDST method, where the main difference is that the maximal largest distance between two points is equal to 1. This corresponds to totally independent vertices, which means that the vertices have no common neighbour. Although dependency between vertices \( V_i \) and \( V_j \) is unsymmetrical, the distance matrix \( A \) for the Sammon projection should be symmetrical. If \( D(V_i, V_j) \) is the dependency of a vertex \( V_i \) on the vertex \( V_j \), then elements of the distance matrix are calculated by the following steps:

1. if \( i = j \), then \( A_{ij} = 0 \)
2. else \( A_{ij} = A_{ji} = 1 - \text{MAX}(D(V_i, V_j), D(V_j, V_i)) \)

All vertices are therefore represented as points inside the unit hyper–ball of dimension \( n \). The main diagonal of the matrix is filled by zeros and it does not have any influence on the projection. Constructed matrix \( A \) therefore, should be directly used as input for the Sammon projection.
Evaluation of methods

Figures 10, 11, 12, 13 depict the visualisation of this network using the Sammon projection for all four methods. By the observation of figures, methods WSPC and DSPC layout the vertices with a large weight near the surface of the hypercube, respectively hyper–ball and vertices with lower weight are placed inside. The used methods maintain projection of the community structures between vertices of approximately the same weight. In case that edges are included in the visualisation we can easily inspect interconnection of particular groups to distant vertices. Results obtained by the WDST and DDST methods are similar to the force–directed algorithms, but allow us more possibilities of how to change visualisation. As with other methods, the community structures are preserved after the visualisation.

![WSPC projection of network](image1)

![WDST projection of network](image2)

![DSPC projection of network](image3)

![DDST projection of network](image4)

The following example uses our methods for visualisation of co–author networks using data from the Forcoa.NET online system. We worked with weighted network based on the
DBLP database. Weights of authors increase when they publish regularly and frequently otherwise weight is decreasing. Processed network contains a snapshot of publication activity of authors from December 2011. The network contains 264,130 authors as vertices and 652,536 links between authors as edges, where all weights are greater than zero. The maximal weight of the vertex is 538.406 (John E. Hopcroft), the maximal weight of an edge is 427.459 (the link between Andrzej Ehrenfeucht and Grzegorz Rozenberg). The weight for a new author was set to 12 (interpretation of weight is the number of months required to remove an author from the network). The network describes co-authors of Václav Snášel and contains 88 vertices. It is sub-network reduced to the depth three where vertices and edges with weight lower than 12 have been omitted. The obtained layout shows the different purpose of the methods. The layout based on the DSPC method moves important vertices to the border of the circle. From the point of view of this method the important vertices satisfy conditions that the vertex has a large or higher than average weight and it is connected to many other vertices in the network. For co-authorship network we may understand the importance of vertices as leadership. Authors on the border of the network with many links to their surroundings state the publication direction for the nearby groups of vertices (visible communities). Vertices having large weight placed inside of the network can be understood as authors that have been (probably) only invited to the cooperation and do not represent key players in this network. The main difference in our approach is that all vertices are enclosed within the hyper-ball of diameter 1 (or circle after the 2D projection).

3.2.2 Experiment 4 – Visualisation of network evolution

Motivations
Explorational network analysis depends on network visualisation and it is not an easy task due to the complexity of networks. There exists one, maybe harder, task in the visualisation and it is the visualisation of network dynamics. Evolution of network and communities during time can help us understand social mechanisms behind the network. Visualisation of the dynamics is not an easy task because there are several issues that have to be solved for correct visualisation. Some of these issues are related to the data source where we often have network data in some regular date or time interval and visualisation of it shows major changes in network structure and layout. For co-authors network studied by us we obtain data within a month period. A useful method for visualisation of an evolution of communities in co-authors network was required for a deep analysis of it such as predictions of what will be the main direction of research or how the focus of researchers changes over time.
Methodology

For snapshot network visualisation we have used Sammon projection which gave us visualisations which were easy to observe and modify. It was described in the previous experiment in Section 3.2.1. In this experiment we work with a co-author network around Václav Snášel. We have prepared 120 files (we worked with a 10 year period), containing weighted monthly calculated networks of his co-authors. Although it seems to be an easy task to visualise network dynamics simply by the visualisation of these files in the time order, there are several issues that need to be solved.

- Using Sammon projection for visualisation did not return the right results in case of visualising the sequence of network states.
- Network nodes can be added or removed between consecutive network states.
- There is a problem to easily observe the network because it is rapidly changing between two consequential states.

By our examination this method is useless for human exploration of network evolution. It could be caused by the following:

- One of the properties of Sammon projection algorithm is initial randomizing of projected vertices into low dimensional space during the starting phase of the algorithm. Setting the generator of the random number to the same initial state for each projected data set does not bring much better results.
- Another possible reason for these marginal changes in visualised network is merging or splitting communities.

After several experiments, we obtain very good results in visualisation network dynamics using the steps depicted in Figure 14.

As was mentioned above we were working in this experiment with co-author network of Václav Snášel in a period from January 2000 to December 2011. Network data was extracted from the system Forcoa.NET and in the preprocessing phase we used methods described in the
At the end of the preprocessing phase, we obtained 132 files containing distance matrices between vertices (authors) in multi-dimensional space. The number of co-authors in the data set were 148 and in distance matrix, we set distance between authors who do not publish in a particular month to maximum possible distance, in the presented experiment it was set to value one.

**Anchor layout**

The main idea behind our proposed method is based on “anchor layout”. As the anchor layout we understand the layout of co-authors network at the end of the observed period. In this time point the network contains the major part of authors and by the observation of several co-authors networks we found there the most important co-authors. By these assumptions the network snapshot in the last month should give us a hint of the initial setting for Sammon projection. Obtained positions of the data points from the layout were used as an initialization position for computing layout in all the previous periods. By this step, we offered Sammon’s algorithm some kind of fixed point in the space – anchor layout. This helped us stabilise the changes in the layout between consecutive layouts.

**Changes in Sammon’s algorithm**

For the correct visualisation of network dynamics, we have made some changes to the original Sammon projection. The usual implementation of Sammon’s algorithm initializes the data points in the space using random values. We have used random initialization only for computing the first layout (anchor layout). The anchor layout is computed from the last network state in the set of network states used for visualisation network dynamics. In all other computational steps, we have used this anchor layout as an initialization position of the data point in the space. This step helps us prevent visualisation to the major changes caused by the computation steps.

To prevent big changes in the positions of projected vertices we modified the update part of the steepest descent algorithm described in Section 2.5. We have made changes in the way of modifying the updating coefficient $\alpha$ and replaced its suggested static value to dynamically changing. We carried out several experiments, but the best results were obtained by the exponential decreasing value of this coefficient in the next Equation (4). This modification prevents the minimization algorithm from making rapid changes in the projection and the final visualisation is much more stable from the view of the position of important authors.

$$\alpha = \lambda^{\frac{cIter}{MIter}}$$

Value $\lambda$ is set to a small value, we obtained good results for values between 0.01 and 0.001. $cIter$ is current iteration and $MIter$ is maximum number of iterations. We set the maximum
number of iterations to 5000. This value was selected experimentally and needs to be evaluated
for each used dataset.

Using this Equation (4) causes exponential decreasing of the updating coefficient in the
range \( (1, \lambda) \). By this modification, we obtained a smaller network energy value than in the
original algorithm. The higher \( \lambda \) value was used for computing just anchor layout, where we
need to get the layout to make rapid changes during minimization of network energy. In the
other cases, we have used small fixed \( \lambda = 0.001 \) coefficient and 5000 iterations. The decreasing
update coefficient \( \lambda \) is used because we want to reduce major changes in the layout, that is not
possible with a fixed value of it.

**Layout interpolation**

Although we obtained network layout with relatively small differences between consecutive
snapshots, we still have to solve the problem caused by merging and splitting of communities.
We used linear interpolation between the consecutive network layouts. We have defined a
threshold \( th \) value for computing a variable number \( n \) of the interpolation layouts. This number
\( n \) depends on the maximal distance between equal vertices in the consecutive layouts and
\( th \) and is computed by Equation (5). Value \( d(V_i, V_j) \) is Euclidean distance between the same
vertices that are contained in both layouts \( L_s \) and \( L_d \).

\[
\begin{align*}
n = \frac{\text{MAX}(d(V_i, V_j)|\forall V_i \in L_s \land V_j \in L_d \Rightarrow V_i = V_j)}{th}
\end{align*}
\]

(5)

Where \( L_s \) is source layout on the network, \( L_d \) is consecutive destination layout, \( V_i \) and \( V_j \)
are the same vertices in both layouts. For computing interpolation in 3D space, we used parametric
equations for a line in 3D space. In case of starting point \( A(x_0, y_0, z_0) \) from source layout and
the same point \( A'(x', y', z') \) in the destination (next month) layout, we computed parametric
equations for the line between these points in the 3D space.

\[
\begin{align*}
\vec{u}(a, b, c) &= (x' - x_0, y' - y_0, z' - z_0) \\
x &= x_0 + at \\
y &= y_0 + bt \\
z &= z_0 + ct
\end{align*}
\]

(6)

(7)

Parameter \( t \) in Equation 7 is used for interpolation. Its value is in the range \( (0, 1) \), where value
0 and 1 compute the position of the start and end point in the source and destination layout.
Interpolation was computed only for vertices that were further than the selected threshold.

The following eight figures (15 to 22) display results of interpolation for Václav Snášel co–
author networks between the period 11/2011 and 12/2011. Blue coloured vertices are vertices
where interpolation was not used and their position was not affected by it. The position of
vertices with yellow colour was affected by interpolation. The red circle focuses on one part of the network with major changes between source and destination layout.

### 3.2.3 Chapter summary

In this chapter, different methods of static and dynamic method network visualisation were discussed. All introduced methods utilised Sammon projection as an alternative to the classic force–based approaches. The advantages of using this method are that input data for Sammon projection is distance matrix and the method of their construction could have a large impact on the final layout and visualisation. In the first experiment, we have offered four methods for constructing this distance matrix based on weights and dependency. It helps us maintain different visualisation to obtain an easily observable static network. The second experiment offers a method for visualisation of network dynamic. Introduced method is based on the idea of anchor layout and modified minimization method of Sammon projection to maintain rapid changes in consecutive network states. The dynamic of the network was visualised by interpolation with a variable number of interpolated network states between source and destination network layout.

### 3.3 Solar power data analysis and prediction

Intelligent and efficient houses have been at the forefront of interest to scientists for several years. The basis for such a solution is an alternative source of energy; often it is a small solar or wind power station. These energy sources are dependent on natural conditions that are different in each location. Analysis of the behaviour of these resources during the year, and the creation and obtaining a model of it is the cornerstone of all related technologies in smart houses. The experiments presented in this chapter are focused on creating a model of the behaviour of a small photovoltaic power station and prediction of its performance. We will utilise relationships between extracted profiles of generated power in the collected data. Introduced experiments have been published in conference proceedings [C43] and [C44].

#### 3.3.1 Data source

Data have been collected from small photovoltaic power station since 2009 up to present day. During experiments, hourly cumulated values of power generated from the power station, are used. We focused on a period from 9/26/2011 to 6/5/2013 and we generally worked with 608 vectors of dimension 24 (working hours of the PVPS are different during the annual period, therefore we worked with vectors of fixed size 24).
3.3.2 Experiment 5 – Power profiles

Motivations
Each owner of a small PVPS is trying to maximise consumption of power generated from it and avoid the overflow of energy to the distribution network. This is possible by the optimal using of electrical appliances in the house. Everybody can imagine that generated power from PVPS in winter is different than in summer but what is a difference between spring and autumn, is there any difference or not? To make a correct plan for the use of appliances with high wattage (oven, washing machine, etc.), it is necessary to create a classification of days by generated power and make a prediction for overall power. This experiment tries to address these requirements by the creation of power profiles based on information which is available to the owner of PVPS but without any particular device such as a weather station.

Methodology
Data received from the inverter are very volatile, there are no two days the same in view of the generated power, but there can be easily seen the similar shapes of function, of generated power during a day. The absolute value of generated power is different for every day, but if we will work with normalised values of power during a day, then it is possible to identify power profiles.

As power profile we understand typical day patterns of power generated by a PVPS independent of the absolute value of generated power. We require that a power profile satisfies criteria that RMSE, between it and covering records of day power, is minimal. An example of power production during a particular day for each month in the year is shown in Figure 23, where changes in the absolute value of generated power are captured.
Our proposed method is based on several techniques. We used network algorithm as a method for removing noise from the data, Sammon projection as the tool for dimension reduction and finally we cluster data by self-organising maps. Extraction of power profiles should be described by the process containing before mentioned four steps (Figure 24).

1. **Preprocessing and denoising data**

We are working with measurement data produced by the inverter and it is no surprise, that the data contains errors. Some are caused by the inverter itself, sometimes it is a problem with solar panels that are covered by snow. Before we use the denoising method, measurement hour power data were normalised according to total daily power in each record. To overcome the mentioned problems in data we used a method based on network algorithm. It helps us remove noise (outliers) from data by setting an appropriate level of threshold. Vector space of measurement data can be easily converted to a undirected weighted network. In our case, the vertices of the network are individual days. The weight of edges between any two vertices (days) corresponds to the similarity of vectors that they represent (similar days). For the calculation similarity, we used the widely known cosine similarity (Equation (8)).

\[
\text{cosine similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}
\]  

The result of data set conversion and denoising is a weighted undirected network, where \( A \) and \( B \) are daily vectors of hourly cumulated power from PVPS.

For visualisation purposes, we used Sammon projection, which, however, requires the calculated distance between all vertices. The accepted input of this is a distance matrix. To obtain this, we used a method based on the calculation of local dependency (see Section 3.2.1, method DDST). The used dependency takes into account the surroundings of each vertex into depth 2 and by the threshold (value of the dependency) we can influence a number of vertices in the network and therefore, the size of the distance matrix. In order to calculate the distance between any two vertices in the network, we set the largest possible distance between points to one. This corresponds to wholly independent vertices (the vertices have no common neighbour).
2. Dimension reduction and visualisation

For visualisation using Sammon projection, we worked with symmetrical distance matrix $A$ for $n$ vertices with order $n$. Each element $A_{ij}$ represents the distance between vertices $V_i$ and $V_j$. Sammon projection is used in the profile extraction process in two different ways.

- Firstly, it helps us visualise source data set for exploration analysis. This kind of analysis should point us in which direction the further analysis is to go if there exist some interesting places in 3D space, and where is the high density of points.

- Secondly, Sammon projection is a non–linear projection method to map a high dimensional space onto a space of lower dimensionality, usually 2D or 3D for visualisation. In the case of projection onto high dimensional space ($>3$), but lower dimension than original data, we can think of it as just a method for reduction dimensionality of data.

In the second step of the process of identifying power profiles, we made dimension reduction. Although one can expect that there are similar days in the year where the output power of the PVPS has the same profile according to the maximal day power, we used Sammon projection to visualise if there are some clusters in the data. Firstly, this method helped us visually check the structure of data. Secondly, we used it for dimension reduction of data.

3. Clustering by self organizing maps

Sammon projection is a good method for visualising data in 2D or 3D space, but it does not have the property of generalisation, therefore we chose self–organising maps (SOM, described in Section 2.4) as the main method for final clustering. In the last step of our process for extraction of power profiles data from Sammons projection were used as input for SOM. From the result of this, we extracted top $n$ most activated neurons (highest number of covered input vectors) and we connected it with original data for visualising power profiles. The power profile curve was calculated as an average value of all records in the cluster.

Solar power profiles extraction

Evaluation of introduced process of solar power profiles extraction is based on real data. The data source is described in Section 3.3.1 and contains 608 vectors of dimension 24. For a better understanding of data, we used different colours for the particular period of the year. The months of December, January and February were coloured brown, green colour was used for March, April and May, yellow for June, July and August and finally September, October and November were coloured orange. These four groups of months divide the year into parts, where obtained power is different in its absolute value. There does not exist a crisp border between these parts and received power from the power station is very dependent on the
weather. To make matters worse a photovoltaic power station needs the sun for getting maximal output power, but the high temperature of the photovoltaic cell reduces output power and the particular location of the PVPS and its installation may also have a big impact on the power. The results can not be generalised and are related to the particular PVPS.

Data from the inverter were transformed into the weighted undirected network where vertices are equal to individual days and weights of edges are computed by the similarity between each of the days. Reduction of the network (denoising and removing outlayers) we can manage using a different value of the threshold, where the meaning of threshold value is, that higher value is equal to the higher value of dependency between vertices. Threshold 0 do not have any effect on the size of the source network. For our experiments we were preparing several matrices with different threshold value (Table 3).

<table>
<thead>
<tr>
<th>Threshold value</th>
<th>Number of vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>608</td>
</tr>
<tr>
<td>0.90</td>
<td>568</td>
</tr>
<tr>
<td>0.95</td>
<td>551</td>
</tr>
<tr>
<td>0.99</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 3: Number of vertices (days) dependent on the threshold value

In the first part of our experiment we used SOM on all the normalised source data and we expected to obtain profiles directly from the SOM. We worked with SOM with the following settings:

- Rectangular topology 5×5 neurons. This size of SOM can produce 25 profiles which are a number near to a number of days in one month.
- Neighbourhood topology was the hexagonal size of 5.
- Learning factor was set to 0.7.
- Number of iterations was limited to 10,000 or network error \( \leq 0.001 \).

The obtained result did not meet our criteria (see Table 4), three neurons covered 53% of all power records which seems to be a very good result, but there is a very high value of RMSE. This shows the problem that SOM creates greater generalisation than we need. We were interested in neurons which covered more than 5% of all records (608).

The power profile was created as an average value of all records covered by a particular neuron. The following figures 25, 26 and 27, show power profiles extracted from covered records for three neurons in Table 4.
We have tried to overcome this generalisation problem and we focused on Sammon projection. In case that we can easily identify clusters by observation we can use SOM on the projected data. We have evaluated data using different settings of Sammon projection with combination of size of reduction of the data set. In profile extraction we focused on threshold 0.95, where outliers from data were removed and we still worked with the major part of source data set (551 records).

We used Sammon projection in the settings for dimension reduction from 24D to 3D, 5D, 8D and 12D. Next we trained SOM using these reduced vectors. We used the setting for SOM in the same way as was described in the paragraph above. Interested clusters were obtained from the top $n$ best matching units of the network. We were focused on the neurons with
more than 26 activations (i.e. 4.5%). The results of these evaluations are in Table 5. Column neurons contain only neurons which cover more than 26 records and the next column contains the largest number of covered records by neuron in SOM.

<table>
<thead>
<tr>
<th>Reduction</th>
<th>Neurons</th>
<th>Largest number of records</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>9</td>
<td>44</td>
<td>56%</td>
</tr>
<tr>
<td>5D</td>
<td>8</td>
<td>38</td>
<td>44%</td>
</tr>
<tr>
<td>8D</td>
<td>8</td>
<td>66</td>
<td>54%</td>
</tr>
<tr>
<td>12D</td>
<td>10</td>
<td>66</td>
<td>68%</td>
</tr>
</tbody>
</table>

Table 5: Interesting neurons for different level of dimension reduction

For final selection of the power profiles, we chose extracted profiles with 12D dimension used in Sammon projection. This selection was caused by the criteria of percent covering of the source data records and evaluation of the rooted mean square error (RMSE) of the group of power profiles. According to our evaluation, SOM learned by 12D dimensional data have greater coverage of source records than 3D (68% for 12D and 56% for 3D). Source data of power measurement contain 608 days which is equal to one year and eight months. Covering more than 50% was our minimal required accuracy.

Evaluation of the average RMSE was used as the second criteria for verifying our choice. A lower RMSE value describes clusters more homogeneous according to computed profile from cluster records. RMSE for clusters based on Sammon 3D projection was 0.0942 and 0.0878 in the case of Sammon 12D projection. From the results in Table 5, we selected for power profiles extraction data of dimension 12. It covers 68% of input data space.

3.3.3 Experiment 6 – Short term prediction based on relationships between profiles

Motivations
Due to the annual increase in energy prices, photovoltaic power stations (PVPS) are often used as a primary source of power for smart off–grid houses. Integration of this kind of energy source is challenging because it is a source of variable generated power due to meteorological uncertainty. It is an attractive technology for people due to rapidly decreasing a cost of this technology and often it is possible to get support from the State. The big boom of smart technologies, devices for Internet of things (IoT) and smart houses in general, creates high demands on the electricity consumption. For the planning of using devices with high consumption, it is necessary to be able to predict the supply of energy. A forecast for the following several hours, or for the next day should be sufficient to satisfy requirements for this task.
Methodology

Short term power prediction we can be considered as one process consisting of two subprocesses. We were inspired by the neural networks and these processes we call learning phase and prediction phase. These are depicted graphically in figures 28 and 29. A detailed description of these processes is in the following paragraphs.

![Figure 28: Learning phase: creating a model of system](image)

![Figure 29: Short term prediction phase](image)

Learning process

The purpose of this process is to model the behaviour of the PVPS at its place of installation and under natural conditions. Although weather conditions are very variable, we assume that there are not rapid changes between consecutive days. We are working with power profiles extracted from data in the same way as were described in the previous experiment (Section 3.3.2). Source data used for learning phase were normalized by the unity–based normalization using Equation (9). Each vector component is a value between zero to one.

$$x_i' = \frac{x_i - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$ \hspace{1cm} (9)

Where $X_{\text{min}}$ ($X_{\text{max}}$) is the minimal (maximal) value of generated power in days and $x_i$ is power during a particular hour in this day. From these normalised data are, by the method for profiles extraction, obtained all profiles with the capability to describe source data with minimal RMSE value. Then for each profile, we compute its normalised power $P$ by the summation of the profile values using Equation (10).

$$P = \sum_{i=1}^{\mid X \mid} x_i$$ \hspace{1cm} (10)

Where $\mid X \mid$ is a number of elements in the source data record and $x_i$ is the particular element of it. Normalised power $P$ was used for ordering profiles by their power and also for comparison between sequences. Each of the extracted power profiles was described by its normalised...
power and a single capital letter starting from letter A and ordered by its power value. By this ordering, we get profiles with similar normalised power near to each other in created sequences.

The final step of the learning phase is to build a model of power production for the PVPS, modelled by the extracted profiles. This should be done easily by creating sequence letters which represent the profiles for all the recorded data in an observed period. For example, we can represent records of measurement for one year by the string containing 365 letters (profiles). From this long sequence, we create a list containing N-Grams (all consecutive sub-sequences length of \( N \)) extracted from it and the corresponding vector of normalised power values. The best length of N-Gram is examined by testing of prediction.

**Prediction process**

The prediction process is depicted in Figure 29. For short term prediction (we work with one day in advance), we use recorded measurements for the previous \( K \) days. Our prediction will be calculated for the one day ahead. To get the prediction, we need to go through several steps.

1. Before prediction it is necessary to process input data into a normalised form using Equation (9).

2. Normalized data sets for the previous \( K \) days are transformed into profile sequence length \( Q \). It is performed by the SOM created in the learning phase. The number of previous \( Q \) days used for prediction is selected by the result of our experiments and we suggest using four to seven days to obtain acceptable results of power prediction for the day \( K + 1 \).

3. For the next steps of our prediction process we try to find a sequence of \( K + 1 \) days with the best similarity to some sequence in the list of sequences created in the learning phase. The F-Score is a measure that combines precision and recall (Equation (11)).

\[
F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]  

\[
\text{Precision} = \frac{tp}{tp + fp}
\]  

\[
\text{Recall} = \frac{tp}{tp + fn}
\]

Where results of LCS are used as:

- **True positive (tp):** number of cases without change.

42
• False positive (fp): number of delete operations.
• False negative (fn): number of insert operations.

4. The main point of our prediction algorithm is to use weighted F–Score as a measure of similarity. We used the well–known algorithm LCS [A12] as one variant of the Edit distance as a way of quantifying how dissimilar/similar two strings are. We were focused on the basic Insert and Delete operations on the strings.

The prediction method is focused on three most important aspects:

a) We take into account the longest common subsequence in sequences of power profiles and related days (we apply a combination of LCS distance and F–Score).

b) In prediction we cover equality of generated power in specific days. (Calculation ratio between generated power in the same days).

c) Similarity between profiles is weighted by the distance of particular days from the predicted day (Further days in the power sequence have the smallest impact on the total weight than closer days to the predicted day.

Remark 3.1 These aspects should be naturally described by the following way:

Part a) defines correspondence of power profile sequences.

Parts b) and c) determines a relevancy of corresponding sequences to the generated power in the considered period.

Results obtained from LCS algorithm (numbers of insert and delete operations) were used for computing F–Score [A21] as a measure for testing the accuracy of strings (power profiles) comparison. We have modified F–Score by the weight computed from \( K \) days power profiles (Equation (14)).

\[
\text{weight} = \sum_{i=0}^{K} \Delta P_i \times i 
\]

(14)

Where \( P_i \) is a normalised power of \( i^{th} \) profile and \( K \) is the number of input profiles used for power prediction for day \( K + 1 \). Due to this equation, a significant part of the weight is dependent on the difference between power profiles in the day before the day of prediction.

Short term prediction

In this experiment, we evaluated the prediction process described in previous paragraphs. The data set (Section 3.3.1) was divided into two parts, where the first part (data set I.) contains data for the whole year 2012 and the second data set (data set II.) contains merged data for two
years 2012 and 2013. For evaluation of our prediction method, we used 167 days outside of the period contained in the learned data (Table 6).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Records number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set I. (2012)</td>
<td>357</td>
</tr>
<tr>
<td>Data set II. (2012, 2013)</td>
<td>719</td>
</tr>
<tr>
<td>Evaluation data</td>
<td>167</td>
</tr>
</tbody>
</table>

Table 6: Working data sets

In the learning phase, we used a method for profile extraction with the same settings as in the previous experiment with power profiles (Section 3.3.2). We worked with threshold value 0.9 and rectangular SOM with 25 neurons. The 25 power profiles were extracted as an average value of all records assigned to the particular neuron in SOM. These profiles have the ability to model all records of the source data set. For each of these power profiles, we calculated normalised power by the Equation (10). The profiles were ordered in descending order by size of normalised power. Then we attached to this ordered list of power profiles one capital letter – in our case, we obtained 25 letters A–Y for simple profile description.

We have replaced each record in the input data by its power profile (single letter) therefore, we obtained a string of characters – a sequence of profiles. An example of this sequence shows the first 30 days of the year 2012:

(TTXXXWTTTVUTUXVTLXXXVQTRTXLLT)

For the first data set I., we get string length of 357 letters and for second data set II. a string of length 719 letters. The final step of the learning phase was a transformation of the input profile string into a short sequence of power profiles with different lengths (we worked with the maximal length of sequence 7). Each short sequence has been taken into account just once.

At the beginning of the prediction and evaluation phase, we transformed evaluation data set into a string of profiles. For this purpose, we used SOM which was trained on the data set I. and after the learning phase, it represents a model of the year 2012. From the obtained string of profiles, we created lists of all short sequences of lengths $K = 3$ to 7 profiles. From short sequence length of $K$ we used $K - 1$ profiles as prediction input and we need to predict the profile at position $K$. By the proposed algorithm we computed profile at position $K$ (it is equal to the prediction of generated power for next day). Because we know from source data which power profile was exactly at position $K$, we can compute the difference between predicted profile and real profile which is known. We performed this evaluation for all short sequences created from the evaluation data.
Figure 30: Percent of records for error up to 40% and for different length of used sequence

For the next evaluation we were focused on a prediction error up to 40%, which is depicted in Figure 30.

From the results of the first part of prediction evaluation (for learning phase using data for the year 2012), the best result of prediction we obtained in the case of using five days length of profile sequence and we predict 6th day by our method. The difference between predicted power and real power will be less or equal to 40% in 77% of cases. How prediction is affected by using more power records in the learning phase is described in the following paragraphs.

The second part of the evaluation of our proposed prediction method tries to evaluate the same testing sequences on the model which was created from data set II, containing data for two years, 2012 and 2013, which include 719 recorded days of generated power from the PVPS. Preprocessing and setting of SOM are the same as in the previous experiment.

For prediction, we used short sequences extracted from the evaluation data and for sequence length K we used sequence length K – 1 as input data for prediction. The increasing number of days used for prediction in the evaluation phase of data set II. slightly increased the percentage of records with power prediction error up to 40%. The best result is covering 77% of all predictions for evaluation data in case of using sequences of length 7.

From the results of the evaluation we can see that using more records in the learning phase causes slightly more prediction with zero error but in general, we did not obtain much more improved precision prediction. It can be caused by the high number of sequences which are used for comparison with the predicted sequence. On the first view, the precision of prediction is not so significant but we have to take into account that our prediction is based only on the
single feature. Our prediction is based only on generated power.

3.3.4 Chapter summary

In this section, we were focused on the creation of a model small photovoltaic power station, which can help us describe the behaviour of the PVPS during the year. We have introduced a method which can describe the PVPS using the power profiles. These profiles were created by a combination of Sammon projection, as a method for dimension reduction, and SOM as the main clustering method. From the input data, we have extracted several power profiles which are able to describe this particular PVPS. In the second experiment, we were interested in short time prediction of generated power from the PVPS. We have designed a process for short time prediction based on introduced power profiles, F–Score as similarity metric and short sequences of power profiles. We have evaluated our approach using real data and the results were compared with measured values and precision of our prediction is up to 70% of short time prediction, predict power, generated by PVPS in the following day, with the error less than 40% of real power.

3.4 Stream data analysis using FCA

While working with different data sources such as the DBLP database of authors, and measurements from the inverter of a solar power station, we worked with a static data source (we take a snapshot of data for some period) and we then applied our methods for expert identification, profiles extraction and others. Some of these data sources are inherently streamed data. For example, records in the DBLP database are updated by authors publication date or measurement data are continuously created by the inverter. In some cases, there is an advantage to work with stream data and it is the main subject of the following experiments. We utilise FCA in a non–classic way to work with stream data to identify outliers in stream data which can be hazardous states of technological process or can have negative effects on the quality of products. In the second experiment, we will try to analyse authors publication trends based on FCA over stream data. Both experiments were published in the conference proceedings [C46] and [C45].

3.4.1 Experiment 7 – Anomalies identification in production data

Motivations

Data representing the production process have the character of stream data. In normal (fault–free) production, it is possible to describe the stream with relatively few patterns. In most cases the patterns are due to the values required by the standards, which must meet the technological
process (temperature, time of some actions, the chemical composition of the product and so on). Early identification of unusual states in the production process can prevent delays in production and also considerable financial losses. We would like to develop a method which can be used for modelling production or any other process which collects the measurement of quantities during the process, and is able to identify fault states from it.

**Methodology**
The stability of process in the fault–free production gives us the possibility to apply the method of Formal concept analysis to detect error states in the data. We want to design a method for anomalies detection in the stream data based on changes in the structure of conceptual lattice. Our attitude relies on the fact that although there exists a large amount of data is obtained during the production process, the size of conceptual lattice is relatively small, and therefore, it is possible to work with it in real–time. The conceptual lattice represents a model of a production process, and this model is based on historical production data. The input data stream contains measurements from the production line and it is applied to the model of the production process. The result of this activity is to identify anomalies in the incoming data and their relationship with faulty products, including disclosure of possible causes of errors by the exploration of neighbours of concepts in the concept lattice.

![Figure 31: Create model of system](image1)

![Figure 32: Anomalies detection process](image2)

Method for anomalies detection is composed of two sub–processes. These are depicted in figures 31 and 32.

**Creating a model of system**

For detection of anomalies, it is necessary to prepare a model of the system in which there are not anomalies and anomalies states identified by experts are used for creating a separate
model. It is based on historical data of the production line or any other process for which we need to create a model. In the case of the production line, it is possible to use records from the quality management system to identify when the production line was without faults and the whole system was in “stable” state. Some problems with measurements can be identified by the missed or unusual values. These records have to be removed from data which are used for creating the model. In fact, we create two concept lattice. The first one contains a model of production line without any anomalies and the second concept lattice is created only from identified anomalies from the production line. Both concept lattices are created in the same way but with different input data.

The set of historical data represents many valued context. The next step is to use conceptual scaling for creating binary context. This binary context may be redundant, in this case, we can remove some of its objects or attributes and get a formal context for which the associated concept lattice is isomorphic to the original formal context. The clarified formal context is the context where there are not empty rows and empty columns and identical rows and identical columns are merged into the only single row or the single column. More formally, if formal context \( \langle G, M, I \rangle \) is clarified then Equation (15) applies.

\[
\{g_1\}^\uparrow = \{g_2\}^\uparrow \Rightarrow g_1 = g_2, \quad \forall g_1, g_2 \in G
\]
\[
\{m_1\}^\downarrow = \{m_2\}^\uparrow \Rightarrow m_1 = m_2, \quad \forall m_1, m_2 \in M
\]

The final step of proposed process is building the model of a system. Because our approach is focused on formal concept analysis, then we use concept lattice. In this case, we use any of the available algorithms for computing concept lattice. We save it and it is later used for anomalies detection. The model of a production line is therefore captured into two concept lattices. In the next paragraph, we will discuss how concept lattice is changed by the incremental building algorithm.

**Changes in concept lattice**

In these paragraphs, we will show how concept lattice structure is changed in case of working with the incremental algorithm. For example purposes we used artificial binary formal context and we processed each row of it using the algorithm AddIntent [A26]. The rows of formal context were processed by an algorithm using the random order which is in the first column of it. All changes in the lattice were captured and analysed.

While processing, incremental lattice computation from stream data, we can, by the properties of FCA, identify two operations on the lattice.

1. **New concepts in the lattice.** In this case increased the number of concepts in the lattice and hierarchical ordering of concepts within it. How big changes are applied in the lattice depends on the current state of the lattice.

48
• Small structural changes are made in the case when a particular input row contains some attributes which were not used anytime before. In this case, we have obtained exactly one new concept and top and bottom concepts of the lattice will be updated.

• Large structural changes are caused by a combination of input attributes for currently computed row and intents of concepts in the lattice. The changes include several new concepts which are created by the splitting of concepts and concepts where there were modified extents and intents including top and bottom concepts of the lattice.

2. No structural changes of the lattice. While we are working with stream data, it is possible to get an input row exactly the same as the one which was used sometimes before. Although the FCA theory works with fixed size and clarified contexts, we have overcome this situation in a production environment. Each input row is taken as a new object in the context. Concept lattice is therefore updated by the changes extent of existing concepts. There are no concepts created or structural changes. To handle this state we extend concept structure by the counter of concept changes.

From the previous example and other experiments where we created concept lattice from a stream of data, we can see that by the analysis of structural changes of concept lattice we can obtain interesting information of input data. We can recognise three interesting changes of lattice and its structure. No changes in the structure should be interpreted as the input data were processed some time before. Creating a single new concept in the lattice suggests that the processed row arrived for the first time and it has not been processed before. Big structural changes and many updated concepts shows that currently handled row contains attributes which are made as a combination with rows processed before. Now we will refer to these two lattices as lattice of acceptable states for model of production line without anomalies and lattice of problem states which was created from identified anomalies.

Anomalies detection
Preprocessing of input stream data is the same as was used in the phase of model creation. For conceptual scaling are used the same scales as in the case of creating the model. There are three possible scenarios of online anomalies detection. Input stream record:

• Comply with any pattern of standard production (lattice of acceptable states)

• Correspond to a known problem (lattice of problem states)

• Are identified as a previously unknown anomaly. In this case, expert assistance may be required to assess the values and their impact on the production line. Possible evaluations are as follows:
- No anomalies occurred ⇒ update lattice of acceptable states (by this step we improve model).

- Anomalies occurred ⇒ lattice of problem states is updated.

**Evaluation of proposed solution**

Evaluation of our approach is done using an example that is based on measured values of temperatures in two reference rooms of a family house, outside temperature and power consumption necessary for the operation of the boiler for heating the house. The data used in this example include information on the hourly temperatures and power consumption during February 2011.

In the period between 02/01/2011 and 02/28/2011, a total of 671 vectors of hourly measured values were obtained. These vectors contain temperature in the bedroom, a temperature in the nursery, outside temperature, average power consumption.

The many-valued context was constructed from the measured values of vectors. This context contained 651 rows and five columns. For the individual measured values, there were obtained maximum and minimum values, and they were used in the conceptual scaling. All four attributes were slotted into three columns by the scaling. Finally, the binary context which contains 651 rows and 12 attributes. In the context, many duplications were found, and these were removed. The resulting clarified context contains 28 rows and 12 attributes. There are 172 concepts obtained from the context. These concepts form the model of the heating system in a given month. Visualisation of concept lattice is omitted due its size. This model is made up solely of measurements that correspond to the heating which is controlled by the controller located in the heating system.

In the source data, we identified 20 records with anomaly state. Most of them were caused by human intervention to the regulations – turn off of the heating and four were caused by an interruption of the power supply. From these records, we created the context of anomaly state which models these anomalies. Capturing of error states enables us to further analyse their causes. Therefore, there is an attribute added by an expert, which indicates a state as exceptional but not as an error. All previously mentioned structures make up a model of behaviour in the home heating system in the month of February 2011. Based on this model, we will further examine the incoming stream of measured values and then decide whether the heating system works well or not. For further analysis, each concept will be extended for additional information, particularly the number of occurrences of a given concept.

Changes in concept lattice of anomalies caused by record of measured values are depicted in Figure 33. A red node in the lattice is a newly created concept. This is a symptom of the
new situation. The measured values describe the new problem case. Green circles show newly added concepts created as the extension of the existing concepts.

Once the anomaly is confirmed by the expert, the lattice can be changed to extend the model of the system by the new fault state. In this case, it can be necessary to merge all objects within the extent into a new object representing the system status. Further analysis of input measurements of a process is the same as in the pseudo-code of the algorithm we showed earlier. During the steps of processing the lattice from atoms to the largest element of the lattice, it is possible to identify a common cause of errors. This is due to the occurrences of particular attributes along the way to the unit element in the lattice. When the process is finished, we gain a model of anomalies that have emerged. Then we can use this model for further repetitive production.

3.4.2 Experiment 8 – Authors publication activity trends

In this dissertation, several methods were introduced for analysing co–author network from the DBLP. In Section 3.1.2, we discussed creating author profiles which describes authors by their used keywords, Section 3.1.3 was focused on finding head experts in the network. These methods were mostly based on FCA and used the property of this approach to obtain the
required results. In this section, we will be extending the previous experiment which uses FCA and stream data. The proposed method was published in the conference proceedings [C45].

**Motivations**
Many authors publish in their research career, tens to hundreds of scientific papers. A closer look at the various authors over time can show us how authors work, what is their research topic, etc. Some of them have changed to several different research areas during their research lifetime. The reason can be, for example, working on various research tasks in different teams or the author is a professor who leads the university students in doctoral studies. On the other hand, we can find authors, who prefer to work on a single topic for a very long time. This topic is only occasionally combined with related issues.

In this section, we are interested in trends in the authors publication activity. We would like to know when and how often authors use their topics, how they combine topics or use new topics. To get an answer to these questions, we introduced a method for visualising publishing trends of the individual authors. As a publishing trend, we understand the course of publishing over time, whether and how often authors change their research topics defined by groups of research terms.

**Methodology**
To achieve our goal we proposed a method based on formal concept analysis and structural changes of concept lattice while source data are processed as a data stream and we use an incremental algorithm for building concept lattice. To obtain a smooth curve we utilised moving average and polynomial approximation to visualise trends of how the authors use their topics and how they change their preferences over time. The described process is depicted in Figure 34.

At the beginning of our proposed method are data which have stream character. It can be any data which can be in the form time series of records (for example measurement of generated power, the number of publications in the month, etc. within a selected period). Stream data usually are obtained from on-line systems but we are working with historical data in the same way.
Our approach is based on changes of concept lattice when we use an incremental algorithm for building it. We use the same algorithm as in the previous experiment (AddIntent [A26]). An advantage of this algorithm is that it computes set of all concepts and concept lattice structure. Next, we were working only with a number of concepts, but information of the structure of lattice should be used for deep analysis.

The mathematical foundations of FCA expected the formal context which contains only unique rows and columns and context which do not meet these criteria have to be clarified by removing all duplications. In our method, we do not want to remove duplication in context because authors are very often using the same topics in their publications. We are working with formal context as with data stream in a way that we start from the beginning of data and we process each record from it separately.

The idea behind our method is based on interpretation of the changes in lattice size. For concept lattice $C_n$ containing $|m_n|$ concepts, where $n$ is number of objects and lattice $C_{n+1}$ with $n + 1$ objects, is true that number of concepts (lattice size) $|m_{n+1}| \geq |m_n|$ and is changed in a way:

- **Number of concept was not changed**: Input record contain exactly the same attributes as some record processed before.

- **Number of concept was increased exactly by one**: That all attributes in new record are used for the first time.

- **Number of concept was increased by more than one**: Not all of attributes in the record are used for the first time. This situation may occur when some attributes used in previous records are extended by a new one. Volume of increased concepts is proportional to the number of the attributes used before. It is caused by splitting formal concepts in the lattice into several new concepts which cover intersections of attributes.

How the number of concept in concept lattice (lattice size) is changed when we incrementally process it with update from artificial contexts, was studied using bars data sets (see figures 35, 36, 37). The data set contain rows where each of these rows includes binary attributes which describe a grid with $8 \times 8$ pixels (64 attributes in the row) and shows randomly generated bars. In the following examples, we present some of the used data sets. Data sets includes 100 rows and we created 10 patterns. These patterns of bars were used randomly in the rows of the data set. How the number of concepts in the lattice is changed by combination of attributes is depicted in figures.

By results of several experiments with bar datasets and with real data from the DBLP and PVPS we recognised that it is necessary to use some other methods to obtain trend curves from stream data.
1. **Smoothing curve of cumulated lattice size**

The changes of the lattice are very variable and we have used a simple method to smooth the data. By the experimental evaluation of several data sets, we have used the method moving average (MA) for this purpose. This method is mostly used when working with a high variable time series where it is necessary to apply an algorithm for smoothing. It is widely used especially in markets analysis. Time series $y_t$ can be seen as a smooth underlying trend observed with error:

$$y_t = f(t) + \epsilon_t$$  \hspace{1cm} (16)

where $f(t)$ is a continuous function of $t$ and $\epsilon_t$ is a zero–mean error series. Smoothing (estimation) $\hat{f}(t)$ of function $f(t)$ is defined as:

$$\hat{f}(t) = \frac{1}{2k + 1} \sum_{j=-k}^{k} y_{t+j}$$  \hspace{1cm} (17)

where $t = k + 1, k + 2, ..., n - k$. 

---

Figure 35: Bars dataset containing 100 rows and 10 patterns

Figure 36: Created concepts in lattice

Figure 37: Cumulative lattice size
The basic idea behind using MA for smoothing is that observations which are nearby in time are also likely to be close in value. The average can eliminate peaks in the data, leaving a smooth trend component. The problem of using MA for forecasting or analysing of the most recent data is that MA does not allow the estimate time series value in the first \( k \) and last \( k \) periods. A large value of \( k \) causes flatter and smoother estimation of \( f(t) \). Using an odd number of observation ensures that the average is centred at the middle of the data values being averaged. A detailed description of MA can be found in the book [A13].

Figure 38, depicts moving average of length \( k = 5 \) (red coloured curve) applied on the curve of cumulated size of concept lattice with 20 patterns and 15% of noise in the context.

2. **Computing record differences from consecutive smoothed data**

   For analysis of trends in the data, we were interested in how fast concept lattice is growing. This value describes how the input of observed process is stable – there are still the same values of measurements (patterns), the size of the change between two consecutive rows (events) is proportional to the combination of attributes. We used smoothed values of size to catch the trend without noise in data.

3. **Getting trend curves by the polynomial approximation**

   For the last part of our process trend mining, we computed a polynomial approximation for the differences in lattice size. For choosing the optimal degree of polynomials we used a degree with the lowest value of RMSE. We have evaluated the degree of the polynomial
in the range from 3 to 7. The result of this final step is a polynomial of a particular degree, which can be used for characteristics of trends in the stream data. The polynomial approximation curve is depicted in Figure 39.

**Evaluation of proposed method**

Evaluation of our process of publication trends extraction is illustrated in the example where we want to extract behaviour of author publication activity. As data set, we used data extracted from the DBLP described in Section 3.1.1. We have truncated data to December 2011 to obtain the most complete data set. Then we divided the entire recorded publication period of conferences into one–month time periods. If during one month an author has published a paper, then we set keyword records corresponding to the paper title. For each author, we obtained a list of months with occurring keywords. The next step was to create a formal context for all selected authors. The size of the context was 1,134 columns and the number of rows was equal to the number of months where the author published paper. These formal contexts were used as the main dataset for our evaluation method. In our evaluation, we worked with 21 authors selected from the DBLP.

Figure 40, display example of a growing number of concepts, extracted from monthly cumulated data of author’s publications. The vertical axis in the chart is the number of concepts and the horizontal axis is the number of the publication month. The red star in the chart is
the value of the cumulated number of concepts from the start of the selected period for the particular month. Due to using formal context (author’s publication activity) in a stream way, the number of concepts rises with time. Smoothing of this chart is done by moving the average with the length of 5 (two data point before and two data point after the centre point). From a smooth curve of a number of concepts (discreet values) differences between each two consecutive months are computed. This is depicted in figure 41. Finally, the smooth curve in the charts is computed as the polynomial approximation of concept differences. The polynomial and its shape give us information of the trend of the author’s publication activity over time.

The shape of the approximation curve (author trend) should be interpreted in the following way:

- The flat part of the curve occurs when the author uses the same topics, or he started to publish in a new area and used new topics which he did not use before.
- The rising part of the curve is in case of the author is using a combination of different topics in his work. The slope of the curve depends on the topics that the author used in previously published papers.
- The falling parts of the curve should be interpreted as the change of author’s preferences so the author changes his topics. He slowly removes old topics and starts using new ones.

### 3.4.3 Chapter summary

This chapter was focused on working with stream data and the proposed methods were based on formal concept analysis. Although FCA was used in the non–classic way – input data set was processed by the incremental methods as stream data set. There were introduced methods which focus on changes in the lattice structure and its size caused by the processing data set as
a stream of data. In the first experiment was designed a solution for anomalies detection on a production line and the second experiment was focused on identification of authors publication activity trends. All methods were used on the real data and obtained results were qualitatively evaluated.
4 Conclusion

This work has been focused on data relationships and their visualisation. To fulfil our objectives, combinations of different methods were used. Preprocessing data by methods based on Social network analysis, Formal concept analysis as the method used for clustering and finally modified Sammon projection algorithm for visualisation of results. There was also developed a method for short-term prediction, based on relationships between consecutive profiles extracted from a PVPS. This work introduces a new method of how to process stream data by FCA method by analysing changes of size of concept lattice.

Contributions of this dissertation can be summarized in the following way:

- Complexity and size of data obtained from social networks or factory processes is still growing. For analysis of this kind of data, it is often necessary to use several methods. In this dissertation methods were introduced which can facilitate analysis and understanding of data. Methods were used for data preprocessing, clustering, visualisation and prediction. For one particular type of data (stream data) a new method was introduced based on Formal concept analysis and properties of the concept lattice.

- In sections 3.1.2 and 3.1.3 methods of how to analyse co-author network were introduced. We focused on a formal description of authors by their regularly and repeatedly used keywords and identification of head expert in communities of authors. We have studied relationships between keywords and authors and between authors and communities. To achieve these goals we have used Formal concept analysis and properties of coatoms in concept lattice, concept stability index and properties of concept lattices. For preprocessing of data was used the method based on forgetting curve and transformation of data source into a weighted network.

- Visualisation of co–author networks is studied in the sections 3.2.1 and 3.2.2. There was used a modification of Sammon projection, together with different ways of preprocessing input data. We have utilised distance and dependency relationships between vertices. As the result of this approach, we obtained a visualisation of the network which helped us gain a better understanding of communities structure of the network in case of explorational network analysis. The introduced method for dynamic visualisation of the network helps us to understand the behaviour of communities during the time. Their dynamics covering splitting and merging of communities without marginal changes in the visualised network is possible using the concept of anchor layout and interpolations of key states of the network.
• Analysis data from a small photovoltaic power station is studied in this thesis in sections 3.3.2 and 3.3.3. A method for classification of generated power based on normalised power and relationships between days in the observed period is introduced. For obtaining power profiles were utilised methods consisting of preprocessing and filtering input data which were converted into the undirected weighted network. To reduce dimensionality Sammon projection was used and final clustering was done by SOM. Short-term prediction of generated power based on power profiles using N–grams and F–score was studied. The developed method was able to predict generated power for the following day based only on power profiles and their sequences. There was designed a process of short-term prediction containing a learning and prediction phase. The main idea is based on relationships between power profiles in the sequences of different length.

• Main application domain for Formal concept analysis is targeted into the area of clustering data or finding dependencies between attributes. In this work, we have focused on non-traditional using of FCA - working with stream data (sections 3.4.1 and 3.4.2). We proposed a new method for analysing stream data based on relationships between structural changes of concept lattice and number of concepts in it. Evaluation of our method has been done on data from a stream of measurements and data from co–author networks. We have provided an approach to stream analysis data by FCA on artificial data sets (bars) and a basic methodology was designed for processing this kind of data.

The methods described in this work fulfil the aims of this dissertation. They work with data relationships and provide a visualisation of results. Although all the discussed methods were evaluated on real data sets, using the proposed methods in different areas will require significant further development. The application of FCA could deliver benefits by its natural supporting relationships in data but itself can not handle these data sets well. The combination of methods used together with FCA gives us a very robust tool for working with data containing relations in it. In several experiments in this work, it was shown that FCA has a big potential to work with large networks in the case the right method of preprocessing data is selected.

All methods described in this work have great potential to be extended by further research in the future. Applying Formal concept analysis on stream data seems to be an especially good direction for the future and preparing a universal methodology for handling data generated from factory processes also offers a large potential to create a useful new way for using FCA in the analysis of data.
Bibliography

References


---

**Own publications – journal papers**


---

**Own publications – conference / workshop papers**


Other papers


