NTNU
Norwegian University of
Science and Technology

Clustering users in an electronic business reference system

Sondre Hoff Dyvik

Master of Science in Informatics
Submission date: May 2017
Supervisor: Sobah Petersen, IDI
Co-supervisor: Andreas Landmark, Sintef

Norwegian University of Science and Technology
Department of Computer Science

Sammendrag

Når en bedrift skal ta strategiske beslutninger er det fordelaktig å vite så mye som mulig om brukerne av produktet deres. Informasjon om hvilke interessesegmenter som eksisterer muliggjør søkeoptimalisering, produktforbedring og spisset markedsføring.
Dette prosjektet hører til feltet kunnskapsfunn i databaser og omhandler oppdagelsen av klynger basert på interessene til brukerne av et elektronisk oppslagsverk for bedrifter der informasjon om brukernes preferanser er implisitt gitt i systemets logger. For å realisere dette målet gjennomføres et litteratursøk for å kartlegge de nyeste relevante metodene innen feltet. Deretter blir disse metodene brukt i eksperimenter og resultatene av disse kvalitativt analysert.
Hovedbidraget til denne oppgaven er en sammenligning av korrelasjonsmålene Spearmans rangeringskorrelasjon og frekvensvektet Pearsonkorrelasjon der målet er å finne ut hvilken som er mest skalerbar. I tillegg brukes Blondels algoritme for å utføre klyngingen av et uutforsket datasett der dataene er generert av brukere i en arbeidsituasjon. Resultatene viser at frekvensvektet pearsonkorrelasjon er det mest skalerbare alternativet og at det finnes klynger i datasettet. Videre viser resultatene at det er sesongvariasjoner i datasettet og i de identifiserte interessegruppene.

Abstract

When making strategic decisions in a business setting it is advantageous to know as much about the users of your products as possible. Information about what interest segments exist can be used for search optimization, product improvement and custom tailored marketing.
This project belongs to the field of knowledge discovery in databases and concerns the discovery of user interest clusters in an electronic business reference system using an implicit voting scheme based on the sytem's web logs. A literature review is conducted to explore recent efforts in the field, experiments are conducted to apply the theory from the literature review and a qualitative analysis is conducted on the results of the experiments.
The main contributions of this thesis are a comparison of Spearman Rank correlation and Frequency-Weighted Pearson correlation in terms of scalability and the application of Blondel's algorithm on a previously unexplored data set generated by users in a professional work setting. The results show that Frequency-Weighted Pearson correlation is the more scalable alternative, and that clusters do exist in the data set. Furthermore it is shown that there is seasonal variations in the data set and the discovered interest groups.

Preface

This thesis was written in Trondheim from August 2016 to May 2017. It was written as part of a larger collaboration project between Sticos AS and SINTEF AS. I would like to thank my supervisor Sobah Abbas Petersen for helping me with matters related to research method and formal requirements and my co-supervisor Andreas Dypvik Landmark for helping me with technical matters. I would also like to thank Sticos for accommodating me with a work desk and Stian Standahl at Sticos for being there when I needed to discuss ideas.

Sondre Hoff Dyvik
Trondheim, May 22, 2017

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

> "Without big data analytics, companies are blind and deaf, wandering out onto the Web like deer on a freeway".
> — Geoffrey Moore, *author, speaker, management expert*

It is a commonly accepted truth that informed decisions are generally better than uninformed ones. Of course, this also applies when making strategic decisions about what direction to focus on when developing a product. A company that knows their customers is better suited to fulfill their customers' needs, and consequently retain their customers for a longer period. This thesis concerns the discovery of user clusters based on web logs in an electronic business reference system. The project is done in collaboration with Sticos AS and SINTEF, and is part of a larger collaboration project aimed at improving Sticos' core business processes. Sticos AS is a company that specializes in educating businesses about Norwegian laws and regulations. The rest of this chapter will give a description of the purpose of this project, summarize a conceptual framework for the reader, account for the scientific contributions of this project and describe the applied methodology.

1.1 Goals and Research Questions

When making decisions about the development of their products, the management of Sticos rely on assumptions about their users and how they interact with their product. Today Sticos rely upon information such as their users line of work and positions when identifying user segments. However, they do not have that information about all their users, and they have experienced that users with the same job title may have different interests. In example the chief executive officer

of a small entrepreneur company may have different information needs than the chief executive officer of a large software company. The purpose and scope of this project is to apply clustering techniques to identify the natural clustering of users based on implicit relations if such clusters indeed exist. This thesis will discuss the characteristics of the discovered clusters and suggest further work, but the application of the discovered information in a business context is outside the scope of this thesis.

Chapter 4 describes a review of relevant literature on the subject, below is a list containing the different data sets used by the papers in the review as well as the papers that used them.

MovieLens: Agarwal et al. [2005]; Adomavicius and Kwon [2012]; Desrosiers and Karypis [2011]; Boratto and Carta [2014, 2015]; Boratto et al. [2009]; Costa et al. [2016]; Gao et al. [2007]; Li and Kim [2003]; Li and Murata [2012]; Sarwar et al. [2001]; Yanxiang et al. [2013]

Last.fm Costa et al. [2016]

Flickr: Zeng et al. [2012]

These three data sets all have in common that they are generated by users browsing media for entertainment. The author notes from personal experience that when browsing for entertainment media, not all of the encountered resources are actually of interest. A user may need to play a movie or song for a short period to even decide whether or not play the rest of it. In addition, several users may share a user account, and the discovered interests of one user may actually be the interests of two different users. In contrast to this, the data set in this thesis is generated by professional users in a work setting. This might mean that their use of the system is more purpose driven, and it is reasonable to assume that they are indeed interested in the resources they visit. If that is the case, it will mean an increased likelihood of discovering underlying clusters. The rest of this section lists the goals and research questions for this projects

Scientific goal To apply known machine learning techniques to a new data set generated by professional users.

Business goal To gain insight into the natural clustering of the users of Sticos' reference system, such that it may be used for product improvement and marketing purposes.

Main research question 1: How can the users of Sticos' systems be clustered based on what resources they visit?

Main research question 2: What characterizes the different user clusters found using the techniques from **main research question 1**?

1.2 Research Method

In order to produce purposeful results in both a business context and a scientific context, the research method had to be adapted to reflect these needs. The project arose from a business goal of *using machine learning techniques on Sticos' web logs for the purpose of acquiring knowledge about how their users use their systems such that it can be used to improve their products.* It was left to the author to decide upon a scientific goal in coalition with Sticos. These factors led to the need for a three-phase process:

Phase one: Exploring data set and possibilities.

Phase two: Structured literature review

Phase three: Experimental setup, experiment execution and qualitative analysis

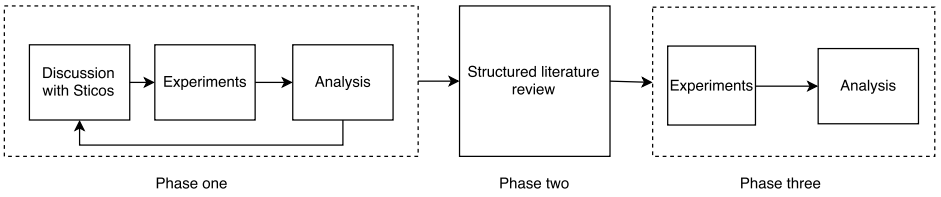


Figure 1.1: Work process

Figure 1.1 shows a graphical description of this process. As this thesis is written as part of a masters degree in artificial intelligence, the main body of content will be dedicated to phases two and three. However, phase one will be described to some extent in Chapter 3 to further elaborate on the motivation for this project and give the reader an understanding of it's conceptual framework. The end result of the first phase was the scientific goal and a more focused business goal as presented in Section 1.1. Figure 1.2 highlights the research strategies that were used during the processes depicted in figure 1.1. The business goal led to the definition of the research questions. After having created these, experiments were performed to create a conceptual framework of the problem and the challenges that were faced. Chapter 3 describes this iteration of experiments in detail. During this phase, the basic theory from Chapter 2 was applied to explore the data set's possibilities and challenges. After having applied the basic theory, a literature review was conducted to suggest a new framework for experiments that addressed the identified challenges. This framework was then used to perform experiments on the entirety of the data set and the results were qualitatively

analysed in appendix B. The results from the first experiment were not satisfactory and led to a second experiment with the data separated in time frames. Chapter 5 explains this process in detail. The experiments were performed on Sticos' web logs, and the data generation process is described in Chapter 3 and Chapter 5.

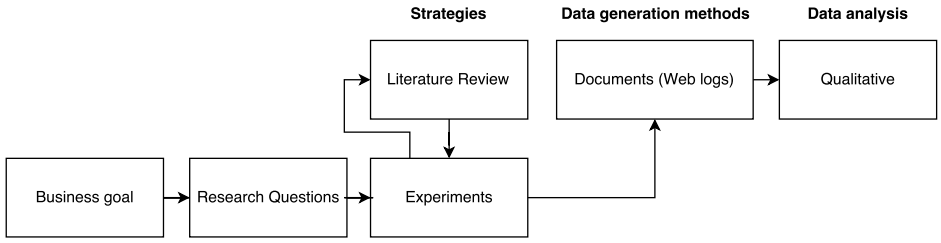


Figure 1.2: Research methodologies - figure adapted from Oates [2006]

1.3 Scientific Contributions

In addition to having value for Sticos in a business context, this thesis should add to the body of knowledge on the field of knowledge discovery in databases. The author identifies the following as the thesis' three main contributions.

Application of known techniques to a new data set generated by professional users: Known data mining techniques will be applied to a new data set generated by users in a professional setting, whereas the benchmark data sets are generated by users in a leasurely setting.

Comparison of two similarity measures: The thesis will compare the *Spearman Rank Correlation* and the *Frequency Weighted Pearson Correlation* metrics in terms of scalability.

A review of recent efforts: The thesis will present a literature review summarizing recent efforts on clustering users based on preferences for different resources.

1.4 Sticos AS

Sticos AS was established in 1983 in Trondheim. Since then, their core business has been to assist their clients in understanding and adhering to the Norwegian laws and regulations that apply to businesses of all kinds. This assistance is realized through courses, seminars, counseling, online tools and articles. One of

Sticos' most important products is Sticos Oppslag, which is an electronic business reference system hosted on their website. This reference system contains 9868 resources such as articles, documents and account plans. As the product is hosted on their website, the traffic to this system is logged, and every interaction the user has with the system can be found in these logs. Today Sticos use this data to create analysis dashboards that allow them to gain insights such as which topics are trending in a given period, or how much traffic is generated on a daily basis. These are valuable insights, but the data set has potential for unlocking hidden insights about implicit relations between users. Starting out with this project, the conditions around the project were loosely defined. The agreed upon goal was to explore the use of machine learning techniques on their web logs to gain insights about their customers. Consequently, there was still some work to be done as to defining the specifics of the project.

This project was done in close collaboration with Sticos. The author was given an office space at Sticos' offices, and weekly meetings ensured the project had value both for Sticos and from a scientific point of view.

1.5 Privacy of Participants

The data used in this project is generated by the users of Sticos' systems. As such, an explicit consent from the users is necessary to use this data for research purposes. This consent is given when the users accept the terms and conditions of Sticos' products. Below is an excerpt from said document in Norwegian and the authors translation of the same text to english.

Norwegian original: Sticos AS vil registrere informasjon om den enkelte bruker og dennes bruk av produktet som grunnlag for tilgangskontroll, **statistisk måling og analyse for å forbedre nettstedets og produktets funksjonalitet**, samt for regnskaps- og faktureringsformal. Opplysningene vil også kunne benyttes for å gi informasjon om aktuelle kurs og produkter som tilbys av Sticos AS.

English translation: Sticos Inc. will register information about the user and its use of the product for access control, **statistical measurements and analysis to improve the website and the product's functionality**, as well as for accounting and billing purposes. The information will also be used to give information about relevant courses and products provided by Sticos Inc.

By accepting these conditions, the user accepts that data about their use of the system is used for this purpose. In addition to having the consent of users, it

is important to handle data that contains sensitive information about users with care. For this project, most of the data does not contain personal information, except from the metadata source that links a user id to a position. In this source, information about email, phone number and company is also contained. Therefore, this data source is only stored on the authors workstation at Sticos' offices. Seeing as the sensitive information is not of interest to this thesis, a copy of the original data source was made, stripped of any personal data. It was this copy that served as the data source for the metadata used in this thesis. By doing this, the rights of the systems users are protected.

1.6 Thesis Structure

The structure of this thesis will be as follows:

Chapter 1 - Introduction and methodology: This chapter has discussed the motivation for the project, summarized it's conceptual framework, presented it's contributions and described the research methodology followed in this thesis.

Chapter 2 - Basic theory: This chapter will explain the project's conceptual framework further by giving an introduction into the basic theory needed to follow the work presented in this project.

Chapter 3 - Preliminary work and problem description: This chapter discusses the work that was done in order to define the scientific goal and gives the reader an introduction to the business goal of this thesis.

Chapter 4 - Structured literature review: This chapter describes the steps and results of a review on relevant literature.

Chapter 5 - Experimental setup and results: This chapter describes the experimental setup of the techniques found in chapter four and their results.

Chapter 6 - Evaluation and discussion: This chapter discusses and evaluates the results from chapter five.

Chapter 7 - Conclusion: This chapter summarizes and concludes the thesis.

Chapter 2

Background theory

The term "knowledge discovery in databases" hereby referred to as KDD was coined by Piatetsky-Shapiro in 1989 during the first workshop on KDD[1]. According to Fayad and Smyth [1996] traditional analysis was often performed by field experts sifting through data and making sense of it, serving as an interface between data and the consumers of the information contained within it. This approach was both time consuming, costly and subjective. As the amount of available data increased, it became apparent that traditional methods were no longer feasible, and a need for techniques that scaled up beyond the analysis capabilities of humans arose. The article written by Fayad and Smyth [1996] was published in 1996. Since then the ability to store and process data has only increased and the need for fast scalable techniques has increased with it.

2.1 An Overview

There are few limits to the applicability of KDD. Today it can be used to aid doctors in diagnosing illnesses, detect fraud, improve search results, aid executives in decision making processes, automate production lines or give customers discounts based on purchasing habits. In the introduction above, the words "data" and "information" were used in a manner that suggests they have different meaning. They are indeed related, but in the field of KDD an important distinction between the two is often made. If one were to view the KDD process as a black box process, data is the input to the box whereas information is the output of the box. In this setting, the data is the raw facts, the individual entries in the database. The information on the other hand, is knowledge of the underlying

[1] According to Fayad and Smyth [1996]

patterns in the data. The steps that turn data to information can be broken down into three sequential tasks commonly known as the *knowledge discovery process*.

1. Preprocessing
2. Data Mining
3. Postprocessing

The rest of this chapter will be sectioned according to these three steps, giving an introduction to all three steps, but mainly focusing on the first two.

2.1.1 Data

The quality of the information discovered at the end of the knowledge discovery process is of course dependent on the quality of the data that is used. According to [Tan et al., 2013, p.22] a data set can often be viewed as a collection of data objects, also called records, cases or entities. These data objects are represented by attributes, which according to the definition given by [Tan et al., 2013, p.23] is a property or characteristic of an object that may vary over time or from object to object. There are more than one way to differentiate between different types of attributes, but the distinction which is most beneficial for this thesis is the distinction made by [Tan et al., 2013, p.28] between discrete and continuous attributes. In this distinction, discrete attributes are described as attributes that has a finite or countably infinite set of values. Attributes such as counts, categories or boolean attributes fall into this category. Continuous attributes are attributes that can be described by floating point numbers such as weight, height or temperature. Both discrete and continuous attributes may also be asymmetric, meaning that only the attributes that are non-zero is of importance. A typical example of an entity with asymmetric attributes is a document. A document may be described by a feature vector, with each attribute indicating whether or not a given word appears in said document. If one were to treat these attributes as symmetric, the similarity between two documents may be very high simply because they share the property that there are many words which are not present in either. However, if the attributes were treated as asymmetric, only the words that are present in one or both of the documents are included in the computation, allowing for a more precise result. The reader should also be aware of the concept of *outliers* and *noise*. An *outlier* is a legitimate data object whose attribute values differ significantly from other data objects whereas the term *noise* is used to describe illegitimate data objects caused by errors such as measurement errors. Both outliers and noise often have a negative impact on clustering results.

2.2 Preprocessing

Data may not, and often does not come in a format that is ready to be fed into a data mining algorithm. The aim of the preprocessing phase is to present the data in such a way that the data mining task has optimal prerequisites for discovering meaningful insights. All of the three steps in the KDD process are important, but neglecting the preprocessing phase will increase the likelihood of achieving poor results. [Tan et al., 2013, p 45] lists the six most common approaches to this phase as follows:

- Aggregation
- Sampling
- Dimensionality reduction
- Feature subset selection
- Discretization and binarization
- Variable transformation

These approaches attempt to either reduce the size of the data, in a way that preserves the implicit relationships between data objects or to reduce the dimensionality of the data. The need for reducing the dimensionality of the data arises due to what is commonly referred to as *the curse of dimensionality*. This refers to the fact that for each attribute/feature that is added to describe the data objects, a dimension is also added to the space in which the data objects may occupy. An added dimension means the volume of the space increases exponentially with it, and without a proportional increase in data objects, the accuracy of the results will suffer greatly. When discussing the curse of dimensionality in relation to clustering, the definitions of density and the distance between points become less meaningful according to [Tan et al., 2013, p 51].

2.2.1 Aggregation

When analysing data it is often possible to look at the data at different levels of granularity. For example one could imagine a national database of temperature readings from weather stations across the country. At the highest level of granularity, the data can be looked at as separate entries on the form:

- Timestamp
- Location

- Temperature

Having the data on this format is useful for some purposes such as measuring the variance or the mean in the data from measurement to measurement. This representation however, may hide underlying patterns that can be discovered by aggregating the data. In principle one could aggregate the data across any of the recorded dimensions, but depending on the aggregate function, it may not be very meaningful. In this scenario it might be desirable to look at the measurements in terms of months, rather than by individual recording. One would then have to aggregate the temperatures using a fitting aggregate function to represent all the measurements made that month. For temperatures, a fitting aggregate function would be the average of all measurements, whereas for sales transactions the sum of sales might be better. This depends on the purpose of the aggregation, and what information one wants to keep. When aggregating the data, one could use more than one aggregate function. For instance the measurements made during one month can be represented as both the average and the variance of all recorded measurements that month. One could then reduce the data entries for one year from several hundred or thousand measurements, to two values per month. This form of preprocessing is a way of reducing the size of the data set, but a similar approach may be used to reduce dimensionality and will be discussed further in Section 2.2.3

2.2.2 Sampling

Another way of reducing the size of the data set is sampling the data, an approach which is commonly used in statistics. The objective of sampling is to reduce the size of the data set while still maintaining it's original characteristics. This subsection describes two ways of sampling data according to [Gu et al., 2000]. The simplest form of sampling is *simple random sampling*. In this approach N samples are drawn at random either with or without replacement. This approach works when the data is uniformly distributed, but when the population consists of different segment with different number of members, this approach often lacks the ability to adequately represent the smaller classes. *Stratified sampling* mitigates this issue by dividing the data into pre-specified amount of groups and drawing either the same amount of objects from each group, or an amount which is proportional to the size of the group.

2.2.3 Dimensionality Reduction

Dimensionality reduction is an approach which is particularly relevant for this thesis due to the nature of the data that is to be clustered. The aim of this step is to reduce the number of dimensions in the data by creating new attributes based on the original attributes. One way of doing this, is an approach similar to aggregation. Instead of reducing the number of data points by aggregating them together, this approach attempts to reduce the number of dimensions by grouping together dimensions before aggregating their values. This approach often requires domain knowledge in the form of an ontology, which specifies how different attributes may be grouped together. In the case of this thesis, this approach is explored by grouping together resources based on topic, and then aggregating the values for each individual resource. Other techniques in this category include *Principal Component Analysis* ([Jolliffe, 2002]) and *Singular Value Decomposition* ([Demmel, 1997]), but they will not be discussed further, as they require a thorough explanation and are not necessary to understand in order to follow the work that is presented in this thesis.

2.2.4 Feature Subset Selection

Whereas dimensionality reduction reduces the dimensionality by replacing several original attributes with fewer new attributes, feature subset selection concerns the selection of the original attributes which allow for the best segregation of the data. Some attributes may be either irrelevant or redundant and can therefore be removed. This may be achieved either by using an algorithm that automatically selects the attributes that best segregate the data points, removing variables using statistical analysis or iteratively testing the data set on a data mining algorithm, and selecting the attributes that give the best results. Decision trees is an example of an algorithm that automatically detects the variables that best separates the data using measures such as *entropy* or *GINI index*, and then selects the best attribute to split on based on this information[Buntine, 1992; Moret, 1982; Murthy, 1998]. These measures may also be used without decision trees when performing statistical analysis to determine the most important variables.

2.2.5 Discretization and Binarization

When working with continuous variables, the conceptual difference between two values may not always be proportional to their mathematical difference. In some cases it may be sufficient to divide the range of possible values an attribute can take on into a fixed number of discrete categories. In the case of this thesis an

example could be an attribute describing how frequent a user visits a given article. Instead of representing this value as a real number, it can be discretized into categories such as *infrequent*, *frequent* and *very frequent* based on thresholds defined by the data scientist. An even coarser categorization could be to reduce the set of possibles values of an attribute to two categories; either *visited* or *not visited*. In the latter case it is called binarization. Both techniques include some information loss, but help separate the data and can be of value as long as the researcher is aware of their limitations.

2.2.6 Variable Transformation

In Section 2.2.5 an example was given where an attribute describes how often a user visits a resource. If that attribute is simply the count of how many times the specific user has visited the given resource, problems arise when computing similarities between frequent and infrequent users. In example, user one visits resource A ten times a day and resource B once a day, whereas user two visits resource A twice a day and resource B once a day. Intuitively user one's preference for resource B is a tenth of his or her preference for resource A whereas user two's preference for resource B is half of his or her preference for resource A. However, as both users visit resource B once a day, their indicated preference for resource B is equal. Another example may be if similarity was to be computed between two cars, possible attributes may be weight, price and production year. Production years may vary by somewhere between one and twenty years, whereas price may vary by several thousands. A difference of twenty years is a lot more significant than a difference of 20 dollars. This is why variable transformation often is an important step in the preprocessing phase.
A number of variable transformation schemes exist, and the choice of which to use is dependent on context. Feature scaling is a type of variable transformation where the attributes are scaled down to the range 0-1. The traditional way of doing this for a given attribute in a given record is by subtracting the smallest recorded value of said attribute in the data set and dividing this on the largest recorded value minus the smallest recorded value. The attribute is scaled according to all possible recorded values of the attribute in the data set. However, if one were to scale each recorded value by all values recorded by the same user, one would achieve a number on the scale 0-1 indicating each users relative preference for a given resource. Other ways of variable transformation according to [Tan et al., 2013, p.63-64] include scaling by simple functions such as square roots, logarithms, absolute values and more complex functions such as normalization, where for each value, the mean is subtracted and the resulting number is divided by the standard deviation of the recorded values.

2.3 Data Mining

After preprocessing, the next step in the KDD process is to apply a *data mining* algorithm. As one may expect there are a countless number of data mining algorithms, and the choice of which to use depends on both the properties of the data set and the information one seeks to discover. Data is either *labeled* or *unlabeled*. A data set that contains information about what class each entry belongs to is called a *labeled* data set and the data mining task which is commonly associated with a labeled data set is *classification*. In the other case, when data is *unlabeled*, the most commonly associated task is *clustering*. Clustering and classification techniques share a number of common traits, and are in some just variations of one another. All though labels such as occupation and title may be present for some of the users in the data set this thesis is concerned with, the purpose is to discover implicit interest groups that exist independently of these labels. Consequently, it is an *unlabeled* data set and the related data mining task is *clustering*. The rest of this section will give a short introduction to classification and a thorough introduction to clustering. Other data mining techniques exist, and are widely used, but are not necessary to follow the work done in this thesis.

2.3.1 Classification

When working with classification, the data set is often divided into a training set, a test set and a validation set. How much of the data goes into each category is usually a trade-off. Generally, it is desirable to use the largest portion of the data for training, so as to achieve the most accurate model possible. However, if the training and validation sets are not of a sufficient size, they may not be representative of the true data set and give a false accuracy score. Different classification algorithms include, but are not limited to the following:

Decision trees: The model seeks to separate the data by building a tree where each parent node tests an attribute and separates the data in two or more segments based on the result of the test and each leaf node is the label of a class. At each level, the choice of attribute test is decided by choosing the test that best separates the data, measured by measures such as *GINI index* or *Entropy*[Buntine, 1992; Moret, 1982; Murthy, 1998].

Neural networks: One of the common tasks for neural networks is classification. A neural network used for classification typically consists of an *input layer*, with one neuron for each of the attributes that describe the data, a number of *hidden layers* and *output layer* with one neuron for each different class. In the most general form, each neuron in one layer is connected to every neuron in the subsequent layer, with every connection given some

weight signifying it's importance. Each cell contains an activation function such as the sigmoid or tanh, and uses the weighted sum of it's incoming connections as the input for said activation function, and fires the output of the activation function through it's outgoing channels. The network is trained by computing the error of the output and adjusting the weights of the individual connections using gradient descent.

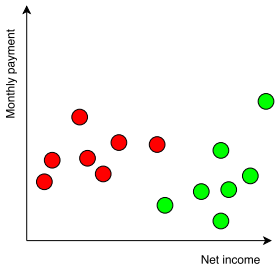
K-Nearest Neighbours: Whereas the two previous models are so called *eager learners*, meaning that they adjust their model for each received training sample and seek to generalize beyond the cases it has seen, k-nearest neighbours is a *lazy learner*. A Lazy learner simply stores each training case as it receives them and waits until prediction time before making any generalization. At the time of predictions it computes the new case's similarity to the stored cases and assigns it the class of the K neighbours which are most similar the new case. If the K nearest neighbours are of different classes, the class of the new case is decided by a majority vote. This lazy learner shares many common traits with clustering techniques and is described below.

K-Nearest Neighbours

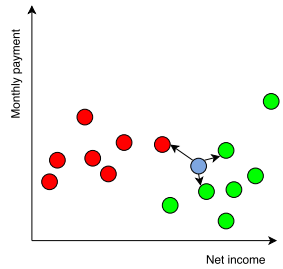
K-Nearest Neighbours, hereby referred to as kNN, is perhaps best described using an example the author remembers being used in a lecture given February 11th 2016 by Kerstin Bach in the course TDT 4173 Machine Learning at the Norwegian University of Science and Technology. This example is an oversimplification of a banks task of identifying potential bad loans. In this example a loan is either *good*(green) or *bad*(red) and represented by the attributes *monthly payments* and *net income*. Figure 2.1 shows an example of how the kNN process works. In figure 2.1b, the new case is represented by the color blue. During this phase, it's distance to all the other data points is computed, and a majority vote decides its classification[Cover and Hart, 1967]. The distance can be computed with several different distance measures, and in this thesis it will become apparent that the choice of distance measure is paramount to achieving good results. A commonly used distance function is the *euclidean distance measure*. Letting q_i denote the Ith attribute of instance q and p_i denote the Ith attribute of instance p, the euclidean distance is defined by:

$$\sqrt{\sum_i (q_i - p_i)^2} \tag{2.1}$$

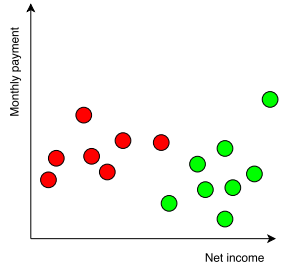
In figure 2.1 k has been set to 3, but this parameter should be chosen by optimization. If k is too large, instances that are far away may affect the result badly, and if k is too small, outliers may affect the result. Another variation of



(a) Training cases of good(green) and bad(red) loans based on the banks experience.



(b) Distance is computed to all of the training cases, and the three closest neighbours are chosen.



(c) The new instance is assigned the class of it's nearest neighbours by majority vote. Although present in this figure, the new case is not stored with it's predicted value as a training case

Figure 2.1: Process of kNN classification. Figure recreated by memory from figure used in lecture. Originally from Watson [2016]

kNN makes it more robust by weighting each neighbours vote by it's similarity, meaning that closer neighbours affect the result stronger than neighbours that are further away.

2.3.2 Prototype Based Clustering

Moving from classification to clustering, prototype based clustering will be explained by describing the algorithm *k-means*, which although similar in name to the kNN algorithm, differs in method and purpose. Recalling Section 2.3.1, kNN is a classification algorithm, whereas k-means is a clustering algorithm, seeking to find clusters of similar data entities. However, what they do have in common is their use of the euclidean distance measure described in equation 2.1 when calculating the distance between points. The algorithm, which is also referred to by the name Lloyd's algorithm, has it's roots in an algorithm proposed by Stuart P. Lloyd for a problem related to signal processing and it was Edward W. Forgy who proposed using it for clustering of data objects in the sense it is used today in 1965 according to Bock [2008]. Although old, it is still widely used today, perhaps owing it's success to it's ease of use and efficiency. The principle of k-means is to divide the data set into k clusters, where the members of each cluster are closer to the *centroid* in their cluster than to any other cluster centroids. The use of centroids is what gives prototype based clustering it's name. A centroid is defined as the average of the attribute vectors describing each of the members belonging to it's cluster. a centroid can be thought of as a *prototype* of the members of it's cluster. Algorithm 1 shows the basic implementation of k-means.

Algorithm 1 Basic k-means algorithm

Input: data set, K
Initiate K centroids randomly
repeat
  for all data objects **do**
    1. compute euclidean distance to all K centroids
    2. assign data object to cluster belonging to closest centroid
  end for
  for all centroids **do** compute new values of centroid attributes to be the average of all cluster members
  end for
until centroids don't change

On the negative side, k-means is vulnerable to noise and outliers, as all data points will be assigned to some cluster, and will affect the computation of the new

centroid for it's belonging cluster, resulting in prototypes that are suboptimal. Secondly, it can only discover "well formed" globular clusters that are similar in size [Tan et al., 2013, p 510]. It is also highly dependent on initialization of the k centroids, and different initializations may often lead to different end results. Lastly it requires k to be selected *a priori*, requiring either optimization efforts or domain knowledge.

2.3.3 Density Based Clustering

The aim of density based clustering is to find clusters of densely connected data objects while at the same time marking outliers as objects not belonging to any of the clusters. This section will describe the *DBSCAN* algorithm proposed by [Ester et al., 1996]. Although not as old as the k-means, it has become an algorithm which is widely used today. Whereas noise and outliers may gravely impact the end result of k-means, DBCSAN is much less vulnerable to those factors, as the clusters are based on densely connected neighbours. With k-means all points will be assigned to one of the clusters, whereas with DBSCAN only the points within a certain proximity to a cluster will be assigned a cluster. The consequence is that the rest will be labeled as noise. Another important aspect of the DBSCAN algorithm is that it takes as input a *distance matrix*, a matrix with precomputed distances between data points. The advantage of this is that the algorithm is independent of the similarity function that is chosen, meaning that the similarity function may be tailored for each purpose. The DBSCAN algorithm is described algorithm 2, which requires the following definitions to understand:

MinPts: Mininum number of data objects that must be present in a neighbourhood for it to be considered dense.

Eps: The radius of a neighbourhood.

Neighbourhood: The area within a radius of *Eps* to a data object,

Dense neighbourhood: A neighbourhood with more than *MinPts* data objects within it.

Core point: A data object with more than *MinPts* in it's neighbourhood.

Border point: A data object which is not itself a *core point*, but has a *core point* in it's *neighbourhood*.

Noise point: All data objects that are neither core points nor border points.

Algorithm 2 DBSCAN algorithm

Input: Proximity matrix, MinPts, Eps
for all rows in proximitiy matrix **do**
    decide if corresponding data object is a *core point*
end for
for all rows in proximitiy matrix **do**
    decide if corresponding data object is a *border point*
end for
discard all noise points
connect all core points that are within *Eps* of each other together
assign a cluster label to each group of connected core points
assign each border point to one of the clusters of their *core point* neighbours

Although arguably not as simple to implement as the k-means, DBSCAN has several advantages over k-means. As described in Section 2.3.2, k-means requires the number of clusters to discover k as an input and the user of the algorithm must either run the algorithm several times in search for the optimal value for k, or possess domain knowledge of how many clusters that exist. The parameters *Eps* and *MinPts* can be chosen by a heuristic where each point in the data set is mapped to a value representing the given point's distance to it's kth closest neighbour and sorting the data according to this value. By plotting these values in a graph, one may upon visual inspection note that there will appear "valleys" in the graph where the degree of decline in distance suddenly changes. Setting the value of *Eps* to the corresponding distance at this point ensures all points with a shorter distance to is kth closest neighbour will be part of some cluster. The value of *MinPts* is then set to the value of k. In their paper [Ester et al., 1996] noted that for values of *MinPts* larger than 4, the distances did not vary significantly for two-dimensional data sets, and they suggest using 4 as the standard value for *MinPts*.

The other advantage DBSCAN has over k-means is it's ability to discover clusters of arbitrary shapes. Figure 2.2 shows a comparison between DBSCAN and k-means with $k = 2$ on three different data sets. Inspecting the figure one can see that the clusters discovered by k-means share the property that they are vulnerable to the choice of k and that densely connected data objects may be put in different clusters. DBSCAN on the other hand, is able to discover clusters of odd sizes as long as the points within the clusters are densely connected, and may thus prove to be more suitable for this thesis than k-means.

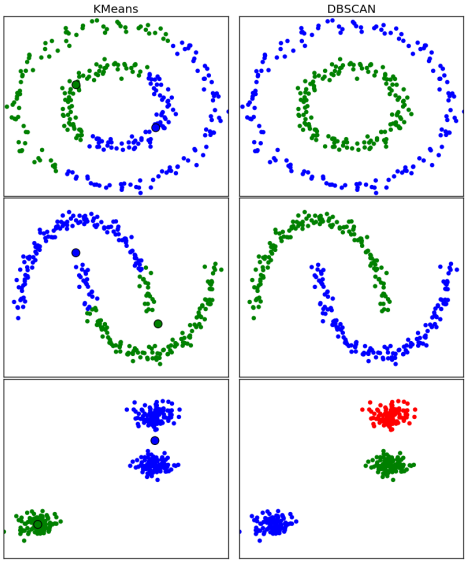


Figure 2.2: Comparison of k-means and DBSCAN generated by a modification of example code [2]using the open source framework scikit-learn by [Pedregosa et al., 2011]

2.3.4 Hierarchical Agglomerative Clustering

Whereas k-means and DBSCAN are only able to find non overlapping clusters, Hierarchical agglomerative clustering, hereby referred to as *HAC*, is able to discover clusters nested in a hierchical structure. In other words, it is able to find clusters at different levels of granularity, which in some domains may coincide with meaningful taxonomies. Similar to DBSCAN, HAC takes as input an $N \times N$ matrix of similarities between the data points, a *proximity matrix*. As such, HAC shares DBSCAN's advantage of being able to work with any similarity function. The results are of course dependent on the accuracy of the similarity function, but the similarity function may be tailored to the specific problem domain.

[2]Example code downloaded February 22nd, 2017 from: `http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/cluster/plot_cluster_comparison.html`

Algorithm 3 Hierachical agglomerative clustering

1: **Input:** Proximity matrix
2: let all data objects in the matrix be their own clusters
3: **repeat**
4: join the two closest clusters
5: compute new similarity matrix
6: **until** all clusters are joined in a root cluster

Algorithm 3 shows the HAC algorithm. The algorithm in itself is not very complicated, but the challenge lies in computing similarity between two clusters in the fifth line of the algorithm. Initially, when all the clusters contain only one object, the matter is as simple as searching the proximity matrix for the two closest objects and then joining them, but then the need for a method to compute distance between clusters arises. Again, the algorithm will work with any similarity measure, but the results will of course be dependent on a good similarity measure. Tan et al. [2013] lists the following variations of HAC:

MIN / Single link: Similarity is computed based on the two most similar objects in the two clusters. This measure can handle non-eliptical clusters, but is sensitive to noise and outliers.

MAX/ Complete link: Similarity is computed based on the two least similar objects in the two clusters.

Group average: Similarity is computed based on the average distance between all members of the first cluster to all members of the second cluster.

Other approaches: Another approach, first presented by Ward [1963] is to merge the two clusters that gives the best results according to some objective function. Examples of such functions are presented in Section 2.4.1.

All of the described variations involve a substantial amount of look-ups in the proximity matrix. For each of the objects in one cluster the distance to every object in the other cluster must be found in the proximity matrix. Consequently, this method is quite computationally expensive, with a complexity of $O(N^3)$. This however, can be reduced to $O(N^2 log(N))$ with efficient memory structures[Tan et al., 2013, p 518].

2.4 Postprocessing

The final step of the KDD process varies by the chosen type of data mining algorithm. As this thesis main concern is that of clustering, the methods presented in this section will be those that are relevant to that task.

2.4.1 Cluster Validation

An integral part of the postprocessing step in the KDD process is cluster validation. This step is done to ensure that the patterns discovered are not simply patterns in random data. K-means will in every case separate the data in k clusters, HAC will always partition the data in some way or another and the results of DBSCAN must be analysed to discover whether the results are in fact meaningful. Although it is often said that analysing cluster validity is often more of an art than a science, this section will present techniques that are commonly used to analyse cluster results. According to Jain and Dubes [1988] cluster validity measures can be divided into the following three categories:

External indexes: Performance is measured by comparing clusters with a priori information. An example of such a measure would be to analyse how well cluster structures match with labels that already exist. For this project it may be relevant to connect information about the different users occupation to the clusters and see whether one cluster contains a high fraction of a few occupations or equal fractions of many occupations.

Internal indexes: Performance is measured only by inspecting the data itself without external factors. One example of this is measuring how similar objects in one clusters are versus how dissimilar the same objects are to the objects of other clusters.

Relative criteria: Two different cluster structures are analysed to see which fit the data best.

In other words there are several different ways to measure cluster validity, and the choice of which measure to use is dependent on the a priori information available as well as the characteristics of the data set and chosen data mining algorithm. The rest of Section 2.4.1 will be dedicated to describing the similarity measures that are applicable in this thesis.

Sum of squared errors : For prototype based clustering techniques such as k-means a common measure for cluster validity is sum of squared errors,hereby referred to as SSE, which is in fact the underlying objective function of the k-means algorithm [Tan et al., 2013, p 500]. Recalling Section 2.3.2, the centroid is computed as the average of all of it's cluster members. The effect of this re-computation is that the sum of squared errors is minimized. Letting K be the number of clusters, C_i be the data objects in the ith cluster and c_i be the centroid of cluster C_i and $dist(c_i, x)$ be the distance between data object x and it's centroid c_i, Tan et al. [2013] defines SSE for

a given cluster as in equation 2.2:

$$\sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2 \tag{2.2}$$

As mentioned in 2.3.2 the value of k can be chosen either by a priori knowledge or via optimization. In the latter case the clustering is performed for a number of different values for k and the average SSE for all the clusters is plotted. Looking at the graph, the scientist may identify "valleys" where the graph declines less than for previous valleys. The corresponding values for k at these locations are often chosen.

Silhouette coefficient: Whereas the aforementioned SSE validity measure is a measure of *cohesion*, indicating how close the members within a cluster are to each other. The silhouette coefficient originally proposed by Rousseeuw [1987] is a measure of both *cohesion* and *separation* which indicates how well clusters are separated from each other. Let a_i be the average distance of object i to it's cluster members. For each of the clusters i is not a member of, compute i's average distance to it's members. Let b_i be the average distance to the members to which i's average distance is smallest. The silhouette coefficient s_i of object i is then defined by equation 2.3:

$$s_i = \frac{b_i - a_i}{max(a_i, bi)} \tag{2.3}$$

From the definition one can see that the possible values range from -1 to 1. [Rousseeuw, 1987] further states that a value close to one implies that the data object is closer to the members of it's own cluster than to members of any other cluster, and it can be said to have been put in the "right" cluster. A value close to zero indicates that the difference between i's distance to it's own cluster members and i's distance to the members of the closest cluster is small, and the clustering is considered intermediate. In the case where the value is closer to -1, i is closer to members of the other cluster than to it's own, and the clustering is considered bad. The silhouette coefficient of a given clustering of the data may then be achieved by computing the average of the silhouette coefficient of all the data objects. This measure can be used to in the same way as SSE to decide the number of centroids for a prototype based clustering, only this time by identifying the configuration with the highest value for the silhouette coefficient. However, this measure is not dependent on there being a centroid to compute the distance to, and is applicable to other clustering techniques as well.

Visual inspection: Moving away from the mathematical measures of similarity, another way of inspecting the cluster validity is by visualizing the proximity

matrix. [Tan et al., 2013] suggests ordering the proximity matrix according to labels and plotting it. Well separated clusters should then be block diagonal whereas poorly separated clusters should be more noisy. This approach is expensive when applied to large data sets, but sampling is a valid method of reducing complexity with this method.

Correlation: The researcher can create an incidence matrix I for the N data objects on the form $N \times N$ with element $I_{i,j}$ set to 1 if object i and j are in the same cluster, and 0 if not. A validity index may be achieved by measuring how well the incidence matrix I correlates with the proximity matrix. [Tan et al., 2013] argues that this incidence matrix represents the *ideal* similarity matrix as members of the same cluster should be as equal as possible and a value of 1 signifies that they are exactly equal. This measure however is not a good measure for density based clusters as the incidence and similarity matrices are symmetric and correlation is only computed for the entries above or below the diagonal.

2.4.2 Separating Good Results from Bad Results

After having computed one of the aforementioned measures the data scientist is left with a real value depicting how good the given clustering is. This number alone is of little help to the data scientist as a value which is good for a data set with a given set of characteristics may be bad for a data set with different characteristics. This is why the presented methods are often used to compare different clusterings to determine which clustering is best. The introduction to this section stated that cluster validation is often more of an art than an exact science. This is because clusters may make sense mathematically, but if they do not make sense in the real world they are useless. To determine whether clusters make sense in the real world one has to evaluate the clusters against ground truth data and qualitatively decide whether the clustering is meaningful. This is often done by deciding upon a meaningful way to represent each cluster either by adding metadata, or by using original data to describe each cluster by a number of features representing the data objects contained within. Each cluster is then qualitatively analysed and determined to be either meaningful or not using domain knowledge.

2.5 Web Usage Mining

A reader with a background in data mining may wonder why this thesis has not yet discussed the domain of *web usage mining*, which indeed is concerned with the discovery of usage patterns based on web logs. The above mentioned data

mining techniques are often used in the later phases of the web usage mining process, and in some ways web usage mining can be said to be concerned with the necessary preprocessing of web logs before the aforementioned techniques can be applied. Chapter 3 will describe why they are not necessary for this thesis. However, examples of common web usage mining tasks are listed below for the convenience of the reader:

User identification Web pages that don't require users to be logged in don't have a user id to associate with a user's web log entry. Instead, techniques must be applied to separate users from each other, often by using a combination of ip-adress, web browser information and timestamps.

Session identification For some purposes it is also necessary to separate a user's different sessions from each other. This is among other things useful for examining which pages are frequently visited together, and is often done based on timestamps. A session is often considered to be over once the user has remained inactive for a duration longer than a given threshold.

Path completion Not all of the visits to a resource are logged because in some cases the resource may be cached and can be served to the users without the request reaching the server. In this case it is necessary to use path completion techniques to augment the session information stored in the web logs. One way of doing this is by analysing the field containing referrer id, which stores the id of the page the user visited before arriving at the current page.

Chapter 3

Preliminary Work and Problem Description

As was described in Section 1.2 this project arose from a business goal of using *using machine learning techniques on Sticos' web logs for the purpose of acquiring knowledge about how their users use their systems such that it can be used to improve their products.* The purpose of this chapter is to give the reader an understanding of the problem this thesis attempts to solve as well as the processes that led to it's definition.

3.1 The Data

The end of chapter 2 described common web usage mining tasks and stated that they were not necessary in this project. This is because users have to log in before accessing Sticos' products. In other words each user is associated with a unique user id, and the task of user identification is already done. The task of session identification is not necessary as it is the given user's total traffic to the different resources which is of interest. For the same reasons, path completion is not necessary as a user is likely to visit a resource more than once if it is of interest, and later visits will ensure that the visit is recorded in the web logs. As the web logs in question span two years of usage, the missing entries due to cache hits may be neglected. How the relevant information has been extracted from the web logs is discussed in depth in the rest of this section.

Table 3.1: Structure of web logs

Record
User Id
Date and Time
Product
Client IP
Client DNS
Url
Event description
Data

3.1.1 Raw Web Logs

Since the beginning of April 2015 Sticos has extensively logged their users interaction with their products. Every server request generated by their users is recorded along with metadata identifying the user and the context within which the request is being sent. As the need for more information emerged, the system was updated to log the new information. Thus the structure of the web logs may vary over time, and the newer entries are in some cases more informative than the older ones.

After initial discussions with Sticos about ambitions and ideas, the author was given access to a database dump of the web logs from September 29th. 2016. This database dump contained around 34 million entries of server interactions spanning all of Sticos' products. The recorded fields are detailed in table 3.1. The content of the fields are self explanatory, except from the field Data, which contains a JSON like object that differs in content. In most cases, this object contains information about referrer URL, type of event, timestamp and content of the site visited. It may also contain metadata such as keywords for a subject or number of records matching the search criterion.

3.1.2 Extracting the Relevant Information

As mentioned in the introduction the product which is of interest in this project is the product called Sticos Oppslag, a reference system that allows users to search for articles about business related topics. Consequently, not all of the entries in the web logs are of interest. More specifically only the logs with the number two in the field "Produkt" in table 3.1 belong to Sticos Oppslag. The amount of data matching this criterion is roughly 32.6 million records.

At this point the data set has been reduced to all the server requests generated

from user interactions with Sticos Oppslag. There is still information in these logs that is not relevant for the purposes of this project, such as searches and site navigation. The next criterion chosen to further restrict the amount of data was on the data contained within the URL field, as seen in table 3.1. This field contains the URL the user requested from the server. From the URL, it was possible to extract information about the visited site. In particular there are three different events of interests in this context.

1. Whether a user visited one of the 2832 articles
2. Whether a user visited one of the 7030 documents
3. Whether a user visited one of the 6 account plans.

This information was extracted by writing a script in the Python programming language using the pyodbc[1] module, allowing for connection with a Microsoft SQL server from Python. The script executed a query specifying that all returned entries should:

- belong to Sticos Oppslag
- not be generated by a user id belonging to Sticos
- have a url that indicates the user visited either a subject, a document or an account plan.
- not have a url indicating it was a search or test event

The results from the database query were iterated through and a method was written to extract the id and type of resource the user visited from the URL field. The results were then written to a normalized database with the fields indicated in table 3.2. The rationale for this was that a normalized database would significantly speed up database queries as it was indexed on date and only contained entries that were of interest for this project.

Table 3.2: Structure of normalized database

Events
Timestamp
TypeId
UserId
ResourceId

[1] `https://github.com/mkleehammer/pyodbc`, Last accessed 2017-04-27

Subject to topic area
Subject_ID
Topic Area_id

(a) Subject to topic area

Topic area
Topic Area_id
Topic Name

(b) Topic area

Figure 3.1: Structure of tables mapping subject to topic area

3.1.3 Metadata

In addition to the Raw Web logs, a crucial data source for this projects is Sticos' own metadata. In a database separate from that which contains the web logs, Sticos keep tables of metadata. These tables contain data that; when coupled with the raw data can provide further insights about the natural clustering of users and subjects. This subsection lists all the database tables that have relevance to this project.

Topic Area

There exists a database that connects 2812 of the 2832 subjects to one or several of 11 broad topic areas. Figure 3.1 shows the structure of the tables necessary to map the subjects to their respective topic areas. Both tables contain other fields as well, but they are not of interest to this project. Using the python programming language, a dictionary was created to store the mapping between subjects and topic areas. This data was read and saved to a dictionary for later use.

Position Held

Prior to the beginning of this project, Sticos gave their users a questionnaire with questions about the following:

- Areas of interest
- Occupation/Position
- Line of work
- Personnel responsibilities
- Authorisations

The answers were stored in a csv file with 15895 entries. Each entry contained a user id and a JSON data field. Of these 15895 records, 6229 were non-empty. The data was extracted by using the JSON module for Python and saved in a persistent dictionary with the pickle modules of Python.

Category
Category Id
Title
Parent

(a) Category

Subject to category
Subject Id
Category ID

(b) Subject to category

Figure 3.2: Structure of tables mapping subject to category

Category

The last source of metadata that was used in this project was a database mapping each subject to one or several of 179 categories. These categories form a hierarchichal structure, and there are a total of 139 root categories. The tables in figure 3.2 show the structure of the two data tables needed to map a subject to a category.

3.2 Initial Experiments

As a way of understanding the possibilities and limitations of the data as well as a means to understand Sticos' information needs, different techniques from chapter 2 were applied to Sticos' web logs. The rest of this section is dedicated to describing those efforts and the knowledge gained by the individual experiments. Although a substantial amount of time was spent on these experiments, they will not be described meticulously as they are not a part of this thesis' scientific contribution, but rather a step in the process of defining the specifics of this project

3.2.1 Applying the Basic Theory

The first experiment that was performed was DBSCAN and k-means with the euclidean distance function described in equation 2.1. Each user was represented by a feature vector of 9868 elements, each index in the feature vector represents one of the resources, and the value at that position represents the number of times the user visited that resource. It became apparent that the clustering had to be performed on a subset of the 27738 users, as the users were represented by such a large number of features and the number of computations necessary to cluster them was too large. The clustering was performed on a subset of 1000 users for a range of different parameter values. The results were unsatisfactory, probably due to the curse of dimensionality described in Section 2.2. Another flaw of the experiment was that no feature scaling was done, so the experiment

was performed again, but this time each element in the feature vector corresponded to the percentage of total traffic the individual user had to the resource at the corresponding index. Although the addition of a feature scaling step was more correct with regards to what the literature suggests, the results were still unsatisfactory.

3.2.2 Aggregating by Metadata

The first experiments did not produce satisfactory results, and the literature suggests this is due to the curse of dimensionality. The next attempt included the use of the metadata presented in figure 3.1. Each user was represented by a feature vector of eleven elements, one element for each of the identified topic areas with each value in the vector corresponding to the users percentage of total traffic to articles belonging to the topic area at the given position. Again the results were unsatisfactory. This time the number of dimensions were 11 and the curse of dimensionality should be less severe. Discussion of the results with Sticos suggested that the categories were to broad to capture the variations along the different dimensions with which the original data may vary.

The other aggregation approach that was attempted was to aggregate the data according to the root categories in figure 3.2. The rationale for this was that it was more detailed than the topic area aggregation, but with fewer dimensions than with no aggregation at all. Again the results were not satisfactory. Recalling the definition of the curse of dimensionality the reader may remember that for each dimension that is added to describe an object, a dimension is added in which the data objects may be spread out. Without an exponential increase in data objects, the data will be to scarce to be clustered in a meaningful way using traditional distance functions. Although the data set is rather big, it is several orders of magnitude to small to occupy the space defined by the 139 root categories. Later discussions with Sticos also suggested that the resources for which a category was defined mostly belonged to a certain field of interest. This would have rendered it an unsuitable way to aggregate as it would exclude articles from other interest fields.

3.2.3 Information Gained

After having performed the aforementioned experiments the author had learned that one of the main challenges of this project is understanding how to compute similarity between users in a way that captures different variations along the different dimensions, as well as taking into account that users may have different frequencies of use. The other lesson learned was that the data was of such a large quantity that the issue of scalability had to be addressed. With these two

information needs in mind, the author performed a structured literature review. Chapter 4 describes this review in detail.

Chapter 4

Structured Literature Review

Seeing as the efforts based on the basic theory provided poor results, a literature review was necessary so that a new architecture could be proposed. As written in Chapter 2 the term "knowledge discovery in databases" was coined in 1989. In other words, this particular field of research has been around for the better part of three decades, and there is a vast amount of literature on the subject. To guide the search for relevant literature, and to perform it in such a way that the results can be recreated, the literature review in this thesis was performed according to an adaptation of the guidelines of Kofod-Petersen [2014]. As the review is not in itself the purpose of this thesis, it is not as rigorously structured as in Kofod-Petersen [2014], but it includes the elements which the author deems most important. The following five steps were selected from Kofod-Petersen [2014]:

Step 1 Identify research questions

Step 2 Identification of research

Step 3 Selection of primary studies

Step 4 Study quality assessment

Step 5 Data summary

4.1 Research Questions

As previously stated, the review was performed after the author had attempted to apply the basic theory on the data set. Consequently, the author was at the

time of writing somewhat familiar with the problem that was to be solved, and why the efforts from applying the basic theory failed. The research questions defined for this review were formulated to provide an answer for this thesis' main research question 1: *How can the users of Sticos' systems be clustered based on what resources they visit?* The information gathered was used to propose the next iteration of experiments, and their results were used to answer this thesis' main research question 2: *What characterizes the different user clusters found using the techniques from main research question 1?* The research questions that were formulated to address this information need are listed below:

Research question 1 How can similarity be computed between two users based on what resources they visit?

Research question 2 What clustering techniques have previously been used to discover clusters based on user interests?

Research question 3 Which of the techniques from **RQ2** can be transferred to the problem presented in this thesis with regard to scalability?

4.2 Identification of Research

After having decided the research questions aimed at providing the information that was needed to drive this project further, the next step of the review was to decide which articles to consider. The goal of this step was to find all literature relevant to the research questions and to exclude literature that was clearly not relevant.

4.2.1 Selection of Databases

The first sub-task of this step was to select which databases to include in the search. When deciding which databases to include, there is a trade-off between making sure that one retrieves as much relevant literature as possible, and limiting the amount of articles considered to a feasible number. Consequently one should choose databases that are likely to contain the most relevant articles. Kofod-Petersen [2014] lists seven databases that are relevant to the field of computer science. To limit the amount of considered articles, this review includes the following three databases:

- ACM digital library [1]
- IEEE Xplore [2]

[1] `http://dl.acm.org/`
[2] `http://ieeexplore.ieee.org/Xplore/home.jsp`

- SpringerLink [3]

On their website [4] University of Oslo Library lists the relevant databases to computer science. These three databases were chosen because they were the first three searchable databases to appear in this list.

4.2.2 Search Phase

The second sub-task of this step was defining the search string that was to be used across the selected databases listed in table 4.2.1. This choice imposed the second restriction on which articles to consider. As with the selection of databases, there is a trade-off when defining the search string. A too narrow search string may result in relevant articles being left out, whereas a too broad search string may return an infeasibly high number of articles. To mitigate this risk factor, this review defined a rather broad search string and relied on the individual database's sorting algorithms to select the top 300 articles sorted by relevance. The intended effect was to make sure that no relevant articles were overlooked and to reduce the amount of articles selected for the next phase to a manageable amount.
The chosen databases allowed for boolean search strings and wildcards. A wildcard is a character that can be placed anywhere in a word to signify that as long as the preceding and/or trailing characters match, any combination of characters is valid at that position. In example the words "caterpillar", "category" , "cat" and "catamaran" would all match with the search phrase "cat*". Here "*" is the wildcard character.
Boolean search strings allow the user to group semantically similar keywords together and search for articles matching any combination of at least one keyword from each group. Table 4.1 lists the groups of semantically similar keywords that served as the basis for the boolean search string in this review. The driving force behind these groups of keywords were the research questions defined in 4.1, representing the information needed to drive this project further.

Group one from table 4.1 represents words that are commonly used in data mining to express the discovery and analysis of patterns and/or clusters in data sets. Group two contains the keywords that are commonly used in research articles to signify clusters in the setting that is relevant for this thesis. The "*" character at the end of the words "segment" and "cluster" means that any postfix to these keywords is desirable. In example the words "segmenting" or "clusters" would be valid matches to these phrases. Group three imposes the restriction that the search results contain the word "users" or the word "user". Group four further restricts the search results to those that concern the users' preferences or

[3] `https://link.springer.com/`

[4] http://www.ub.uio.no/english/subjects/informatics-mathematics/informatics/databases/, downloaded Feb 01, 2017

Table 4.1: Boolean keyword groups

Group 1	**Group 2**	**Group 3**	**Group 4**
discover*	groups	users	interest
mining	grouping	user	interests
analys*	segment*		rating
	cluster*		preference
	classif*		

rating. Group one and two together imply that the results should concern the field of data mining, and within this field, the sub-field of clustering data points into separate groups. Group three and four represent the specific information need that has become apparent in the previous efforts of this project as detailed in Chapter 3.

When creating the boolean search string the words in the same semantic group are put in parenthesis delimited by the logical "OR" operator $\vee$ and the groups are combined with the logical "AND" operator $\wedge$. Equation 4.1 is the resulting logical expression when combining the keywords from 4.1 with logical operators. The logical characters $\vee$ and $\wedge$ are not valid input characters to the search databases and have to be replaced with the words "OR" and "AND". When included in a boolean search string the words are used as logical operators and does not mean that articles matcing the english words "and" and "or" will be returned. With this in mind, equation 4.2 represents the final boolean search string that will be used across all the databases.

$$\begin{aligned}(discover* \vee mining \vee analys*) \wedge \\ (groups \vee grouping \vee segment* \vee cluster* \vee classif) \wedge \\ (users \vee user) \wedge \\ (interest \vee interests \vee rating \vee preference)\end{aligned} \tag{4.1}$$

$$\begin{aligned}(\text{"discover*" OR "mining" OR "analys*"}) \; AND \\ (\text{"groups" OR "grouping" OR "segment*" OR "cluster*" OR "classif*"}) AND \\ (\text{"users" OR "user"}) AND \\ (\text{"interest" OR "interests" OR "rating" OR "preference"})\end{aligned} \tag{4.2}$$

4.2.3 Additional Restrictions

Both the choice of databases to consider and the choice of search string impose restrictions on the content of the returned articles. In some cases additional restrictions may be required when relevant. For this review, the date of publishing is relevant when it comes to the matter of scalability from research question 3. Due to storage and processing capabilities, the amount of information available in modern databases is higher than ever. For these reasons a restriction is imposed on the publishing year of the returned articles. In addition to matching the search string, the articles have to be published after the year 2012. To focus on that which is state of the art, this review also imposes the restriction that only research articles will be included, excluding book chapters and video segments. SpringerLink allows for the user to sort articles by content type, whereas for the rest of the databases this has to be done manually and will be a part of the inclusion criteria as detailed in Section 4.3.

4.2.4 Results

Following the directions specified above, a search was performed in the aforementioned databases. The resulting number of hits per database is listed in table 4.2. In the case of ACM digital library the search was first performed on the boolean search string, and then refined to only include publications after 2012 by using a slider on the results page. Making sure that the results were sorted by relevance, the results were exported to a format importable by reference management program EndNote [5]. This produced a file containing the first 1000 hits of the database, which in turn was fed through a python script that wrote the first 300 references to a new file which in turn was imported to EndNote, discarding four duplicates. In the case of IEEE Xplore, the boolean search string was pasted into the advanced search option named command search. 4545 articles were returned before filtering out articles published before 2012 and hits that were not in the categories "Conference Publications" and "Journals & Magazines). Applying these refinements reduced the number of articles to 2134. The first 300 articles as sorted by relevance were then exported to EndNote. In the case of SpringerLink the initial search on the boolean search string returned 741 639 hits. Refining the search to exclude anything but articles reduced the number of hits to 460 771. Furthermore the search was refined to only include articles in the discipline of computer science and the subdiscipline of artificial intelligence, which again was reduced to the subdiscipline of "Database Management & Information Retrieval" which reduced the number of search results to 2 947. Finally, imposing the restriction that articles published before 2012 should be excluded led to a final

[5] `http://endnote.com/` last visited May 4th 2017.

result of 1 768 hits. SpringerLink however does not allow for exporting several articles to EndNote. As a consequence of this, the first 300 hits were scrolled through, and only those that met the inclusion criteria described in Section4.3 were exported to EndNote.

Table 4.2: Search results by database

Database	**Hits**
ACM digital library	2492
IEEE Xplore	2134
SpringerLink	1768

4.3 Selection of Primary Studies

The selection of primary studies can be broken down into three successive steps, each more comprehensive than the other. The first phase looks solely on the titles of the articles and views them in light of some inclusion criteria and discard any articles obviously not satisfying any of the inclusion criteria. The second step goes more in depth and looks at the abstracts and discards articles not fitting the inclusion criteria from step one. The last step consists of reading the full articles and rating them according to some quality criterion and discards those that do not meet a certain threshold.

4.3.1 Initial Screening

Table 4.3 lists the inclusion criteria for the initial screening. In this phase all articles with titles implying that the article does not meet any of the inclusion criteria are discarded. After having screened trough the article titles, removing those that obviously do not satisfy any of the criteria; 147 articles remain.

Table 4.3: Initial screening: Inclusion criteria

IC1	The data set must consist of a user's preference for different resources
IC2	The presented work should be applicable to this project
IC3	The paper should discuss similarity between users
IC4	The paper should discuss unsupervised algorithms
IC5	The paper should describe in detail the approach taken

4.3.2 Final Screening

Moving on with the 147 articles from initial screening, each article's abstract is evaluated and either discarded or included depending on whether it satisfies at least one of the inclusion criteria from table 4.3. Should the abstract not give a concise answer to whether or not the given article meets the criteria, the article's full text is skimmed through to give a better foundation for deciding whether it should be included. After completion of this step, 27 articles remained. Upon reading the full text of the remaining articles, 18 articles were discarded after not meeting certain inclusion criteria. Table A.1 lists the discarded articles and the criteria they failed to meet. In addition to this, one more article was discarded as it was highly similar to another, and written by the same author.

4.3.3 Augmenting the Primary Literature

While reading through the articles during the final screening, it became apparent that it would be wise to widen the search for primary literature as the remaining eight articles presented fairly similar solutions. Thus, the review would benefit from looking at more primary literature and explore a wider array of approaches. This approach deviates from the guidelines of Kofod-Petersen [2014], but the author deems it necessary to provide a solid foundation for the rest of the thesis. As the eight remaining articles met the inclusion criteria and were relevant to this project, it is probable that the references cited in these articles may also be relevant to this project. Consequently, their references were screened based on their titles as in Section 4.3.1. This time the previous requirements were relaxed to include articles published before 2012 as well as relevant chapters from books. 19 articles were added based on title. A full text read through resulted in the exclusion of seven articles.

4.4 Quality Screening

This is the last step that is performed while selecting articles. During this phase each of the articles are thoroughly read through and evaluated against some quality criteria. Kofod-Petersen [2014] suggests a list of ten criteria that may be used for this purpose, and it is these criteria that are used in this thesis. The criteria are listed below.

1. **QC 1** Is there a clear statement of the aim of the research?
2. **QC 2** Is the study put into context of other studies and research?
3. **QC 3** Are system or algorithmic design decisions justified?

4. **QC 4** Is the data set reproducible?
5. **QC 5** Is the study algorithm reproducible?
6. **QC 6** Is the experimental procedure thoroughly explained an reproducible?
7. **QC 7** Is it clearly stated in the study which other algorithms the study's algorithm(s) have been compared with?
8. **QC 8** Are the performance metrics used in the study explained and justified?
9. **QC 9** Are the test results thoroughly analysed?
10. **QC 10** Does the test evidence support the presented findings?

The articles are read through and for each criteria given a score of either 0, 0.5 or 1 depending on how well it satisfies the given criteria. In the cases where the article is either a survey or a chapter in a book these criteria are not applicable. Instead they were rated on the number of times they have been cited in the database where they were found. For an article to be included they had to have a rating above 7, or more than 20 citations if the quality criteria are not applicable. Table A.2 in appendix A1 lists the articles that were discarded, whereas table A.3 in appendix A1 lists the articles that met the inclusion criteria.

4.5 Data Summary

After having been through the steps outlined in this section it has become clear that the problem that has to be solved for this thesis is very similar to a problem which is often encountered in a sub-domain of recommender systems called collaborative filtering. The rest of this chapter will summarise the relevant techniques found in the papers listed in table A.3.

4.5.1 Recommender Systems

According to Adomavicius and Kwon [2012]; Adomavicius and Tuzhilin [2005]; Agarwal et al. [2005] recommender systems are usually classified into three categories based on their approach to recommendation: content-based, collaborative, and hybrid approaches. However Adomavicius and Kwon [2012] also argue that recommender systems can be classified based on the nature of their algorithmic techniques as either model-based techniques or heuristic techniques. In machine learning one often classifies techniques as either lazy or eager. Recalling the distinction made in Section2.3.1, a lazy learner simply stores training cases and then

uses all the training cases when making predictions. An eager learner however tries with each new training case to approximate a model that can generalize beyond the cases that it has seen so that it can base it's prediction on this model when it encounters an unseen test case. An argument can be made that heuristic based approaches is a kind of lazy learner as they according to Adomavicius and Kwon [2012] typically calculate recommendations based directly on a users previous activities such as transactional data or rating values. Model-based techniques on the other hand could be called eager learners as they typically use previous cases as a foundation for building a model that is used for making predictions. This review will use the first categorization and within each category summarize the relevant literature.

4.5.2 Content-based Filtering

The overall objective of content-based filtering is to recommend items that are similar in content to items that have previously been rated high. In this approach each individual item is described by a feature vector containing the item's describing attributes. In the case of movies, this feature vector can include genre, length, lead actors, production year, etc. According to Adomavicius and Tuzhilin [2005] the main focus of research in this field has been on textual items, and these techniques have roots back to information retrieval. Consequently the measures used for document similarity in information retrieval can be used for similarity in content-based filtering. Following this approach, a document feature vector is created by first removing very common words such as {and, or, the} and other words that are likely to be in most documents, and then use the remaining words as keywords with some weight attached to them. Adomavicius and Tuzhilin [2005]; Desrosiers and Karypis [2011] describe the *term frequency/inverse document frequency*($TF-IDF$) measure, which has its roots in information retrieval and is widely used today. The calculation consists of three parts; first the *term frequency* calculation which calculates the relative frequency of a term, then the *inverse document frequency* part which punishes terms that are found in many of the documents and then the multiplication of the two results. The IDF part of the formula reflects the idea that words that are found in many of the documents serve little purpose in describing the contents of any particular document. Equations 4.3, 4.4 and 4.5 shows each step in the calculation of this measure as it was written in Adomavicius and Tuzhilin [2005, p. 3]. Here $f_{i,j}$ is the number of times keyword k_i appears in document d_j. N is the number of documents in the collection and n_i is the number of documents keyword k_i appears in.

$$TF_{i,j} = \frac{f_{i,j}}{max_z f_{z,j}} \tag{4.3}$$

$$IDF_i = log\frac{N}{n_i} \tag{4.4}$$

$$w_{i,j} = TF_{i,j} \times IDF_i \tag{4.5}$$

Having computed $w_{i,j}$ for all documents d_j and keywords k_i, the documents may be represented by their term vectors, and the similarity between the two documents may be computed as the cosine angle between their term vectors. Equation 4.6 from Adomavicius and Tuzhilin [2005] shows this formula in its generic form and equation 4.8 shows the same equation when it is used to compute similarity between two rating vectors. Here $\overrightarrow{w_c}$ is the term vector of some document c and $\overrightarrow{w_s}$ the term vector of some document s.

$$cos(\overrightarrow{w_c}, \overrightarrow{w_s}) = \frac{\overrightarrow{w_s} \cdot \overrightarrow{w_d}}{\|\overrightarrow{w_s}\|_2 \times \|\overrightarrow{w_d}\|_2} \tag{4.6}$$

4.5.3 Collaborative Filtering

Whereas content-based filtering attempts to find content that is similar to content that a user has appreciated before, collaborative filtering attempts to find users that are similar and then use the ratings of these similar users to suggest new content. One of the advantages of collaborative filtering systems is that they according to Desrosiers and Karypis [2011] can recommend items with very different content, as long as similar users have shown interest for these items. There are many variations within collaborative filtering, some are more relevant for this thesis than others, and will be presented in more depth than those that are less relevant. The two major challenges in this domain is according to Agarwal et al. [2005] scalability to high dimensional data and data sparsity. Listed below is the notation used by Adomavicius and Kwon [2012]. If any paper uses another notation, it will be translated to that of Adomavicius and Kwon [2012] for the readers convenience.

- U denotes the set of users in the system
- I denotes the set of all items that can be recommended to users
- $R : U \times I \rightarrow Rating$ represents the preference of each user $u \in U$ for item $i \in I$
- $R(u, i)$ represents the rating that user u gave to item i and $R*(u, i)$ represents the rating predicted by the system for item i by user u
- $sim(u, u')$ represents the similarity between user u and another user $u' \in U$

- $\overline{R}(u)$ represents the average of all ratings made by user u
- $I(u, u')$ represents all items rated by both u and u'
- $N(u)$ represents the set of the N nearest neighbours of u

Neighbourhood-based Collaborative Filtering

A technique that was encountered often in the primary literature is the heuristic technique called neighbourhood-based collaborative filtering. This technique attempts to calculate similarity between users based on their previous activity and then uses this information to recommend items that similar users have expressed appreciation for. One way of doing this, as shown by Adomavicius and Kwon [2012] is to use the Cosine Similarity metric shown in equation 4.8 and then calculate $R*(u, i)$ as the adjusted weighted sum of all known ratings $R(u', i)$ where $u' \in N(u)$ as in equation 4.7.

$$R*(u,i) = \overline{R}(u) + \frac{\sum_{u' \in N(u)} sim(u,u') \cdot (R(u',i) - \overline{R}(u'))}{\sum_{u' \in N(u)} \mid sim(u,u') \mid} \tag{4.7}$$

Adomavicius and Kwon [2012] also list several other ways to predict item rating once the similarity is computed, and the main body of their contribution is a technique for ranking items in a way that attempts to improve diversity in recommendations. However, as the task of ranking items is not of relevance to this thesis, it will not be discussed further.

Cosine Similarity

The cosine similarity measure is often used in information retrieval to calculate the similarity between two vectors where each attribute represents the frequency with which a word occurs in a document. The nature of documents and the amount of words one can possibly use in any given document suggests that these may be very sparse. This translates well to the problem in this thesis as it is probable that there exists several articles which some users have never visited. According to Tan et al. [2013, p.75] the cosine similarity function is one of the most commonly used measures of document similarity. Equation 4.8 shows the metric in the notation used by Adomavicius and Kwon [2012]. This equation can also be found in Adomavicius and Tuzhilin [2005]; Yanxiang et al. [2013]

$$sim(u,u') = \frac{\sum_{i \in I(u,u')} R(u,i) \cdot R(u',i)}{\sqrt{\sum_{i \in I(u,u')} R(u,i)^2} \sqrt{\sum_{i \in I(u,u')} R(u',i)^2}} \tag{4.8}$$

According to Desrosiers and Karypis [2011] a problem with the cosine similarity measure(Eq. 4.8) is that it does not consider the difference in the mean and variances of the ratings made by different users. The Pearson correlation metric (eq. 4.9) however removes these components when computing the similarities.

Pearson Correlation

The Pearson correlation metric was the similarity measure that was described by the majority of the articles discussing similarity measures. It can be found in Adomavicius and Tuzhilin [2005]; Desrosiers and Karypis [2011]; Gao et al. [2007]. As previously stated, the Pearson correlation metric removes the mean and variance from each user's rating before computing similarities. Equation 4.9 shows this measure translated from the notation of Adomavicius and Tuzhilin [2005] to the notation defined in Section4.5.3.

$$sim(u,u') = \frac{\sum_{i \in I(u,'u)} (R(u,i) - \overline{R}(u))(R(u',i) - \overline{R}(u'))}{\sqrt{\sum_{i \in I(u,'u)} (R(u,i) - \overline{R}(u))^2 \sum_{i \in I(u,'u)} (R(u',i) - \overline{R}(u')^2}} \tag{4.9}$$

This measure was also used by Li and Kim [2003]; Sarwar et al. [2001] to compute item similarity.

Spearman Rank Correlation

The Spearman rank correlation measure, hereby referred to as SRC, was described by Desrosiers and Karypis [2011], but was not mentioned in any of the other papers. This measure however, avoids the problem of variance in the different users' personal rating scales. The formula is the exact same as the Pearson correlation (eq 4.9), but for each user, the items are ranked from 1 to N based on their relative rating. Similarity is then computed based on each items relative ranking rather than rating. Letting $k(u,i)$ denote the rating rank of item i for users u and $\overline{k}(u)$ denote the average rank of items rated by u, the spearman rank correlation is defined as follows:

$$SRC(u,u') = \frac{\sum_{i \in I(u,'u)} (k(u,i) - \overline{k}(u))(k(u',i) - \overline{k}(u'))}{\sqrt{\sum_{i \in I(u,'u)} (k(u,i) - \overline{k}(u))^2 \sum_{i \in I(u,'u)} (k(u',i) - \overline{k}(u')^2}} \tag{4.10}$$

The disadvantage of this method is that it is computationally more costly than the Pearson Correlation metric(eq 4.9) and that it would have to deal with a large

number of tied ratings when the rating scale is not large enough. In the case of this thesis however, the number of items might be small enough to be sorted by rank in a reasonable time. If one were to rate items according to how often a user has visited them, the probability of many ties is small. Desrosiers and Karypis [2011] compared the spearman rank correlation with the Pearson correlation on a dataset called *MovieLens*[6] and found that predictions made using the Spearman rank correlation were .26 % better than the predictions made using the Pearson correlation when using the 5 nearest neighbours to predict the rating of an unseen item.

Frequency-Weighted Pearson Correlation

The last similarity measure explored in this review is the Frequency-Weighted Pearson Correlation hereby referred to as FWPC described in Desrosiers and Karypis [2011]. This is a variant of the Pearson Correlation metric that incorporates the *Inverse User Frequency* modification described further down in this subsection. Following the notation in the subsection about inverse user frequency, the correlation measure is described as follows:

$$\lambda_i = log\frac{N}{n_i}$$

$$sim(u,u') = \frac{\sum_{i \in I(u,'u)} \lambda_i (R(u,i) - \overline{R}(u))(R(u',i) - \overline{R}(u'))}{\sqrt{\sum_{i \in I(u,'u)} \lambda_i (R(u,i) - \overline{R}(u))^2 \sum_{i \in I(u,'u)} \lambda_i (R(u',i) - \overline{R}(u')^2}} \quad (4.11)$$

Modifications of the Neighbourhood-based Approach

Cosine similarity and Pearson Correlation use the set of items rated by both users when computing similarity. This may result in poor accuracy when similarity is computed between users whose set of articles that both users have rated is small. Adomavicius and Tuzhilin [2005] refers to methods such as *default voting* and *inverse user frequency* explored by Breese et al. [1998] to mitigate this issue.

In the case of **default voting**, Breese et al. [1998] argue that instead of taking the intersection between the respective users' set of rated items, it is possible to instead use the union of the set of items at least one of the users rated, inserting a default value for the unobserved items. Breese et al. [1998] further argue that

[6] http://www.grouplens.org/

in applications with implicit voting schemes, as is the case for this thesis, an observed vote is typically an indication of a positive preference for the item. A default value of zero will then indicate that the user has not visited the resource and therefore is not interested in it.

Inverse user frequency draws upon the same principle as in the IDF part of the $TF - IDF$ formula in Section4.5.2. For the purposes of this project, it may be useful to apply this term, as items that many users have visited may be of interest to all users, and serve little purpose in discriminating users when the goal is to cluster users based on what they are interested in. This could be incorporated by factoring in $log\frac{N}{n_i}$ when computing the weights for each attribute of the feature vector in either formula 4.8 or 4.9. In this case N would be the total number of users and n_i would be the number of users who have voted on resource i.

Significance weighting as described by Desrosiers and Karypis [2011] tries to mitigate the same drawback of collaborative filtering as *default voting* when predicting ratings based on similar users. This is done by punishing votes by users whose similarity was computed based on a common set of ratings less than some predefined threshold.

- Let w denote the user similarity and w' denote the adjusted user similarity
- Let N be the number of items both users rated and γ denote a predefined threshold

The vote is then adjusted by computing equation 4.12.

$$w' = \frac{min\{|N|, \gamma\}}{\gamma} \times w \tag{4.12}$$

Default voting and significance weighting complement each other. Whereas the prerequisite for default voting is that one of the users have rated items which the other has not, significance weighting weighs down similarities computed when both users have rated the same few items.

4.5.4 Clustering Techniques

The algorithm that was found to be most common in the papers reviewed in this thesis is the k-means clustering approach as described in Chapter 2. Some variant of this approach was used by Boratto and Carta [2014, 2015], and according to Amatriain et al. [2011] it is by far the most used clustering algorithm in recommender systems today. However, this algorithm as it was described in

Chapter 2, is vulnerable to the curse of dimensionality. According to Müller et al. [2009]; Amatriain et al. [2011]; meaningful clusters cannot be detected in high dimensional data sets as distances are increasingly similar when dimensionality increases. This explains why the previous efforts have failed; the number of dimensions were to high. The following section describes the different clustering approaches explored by the papers included in this review.

K-medoids Clustering

An algorithm which is closely related to the k-means algorithm described in Chapter 2 is the k-medoids clustering algorithm described by Costa et al. [2016]. As described in Chapter 2 the traditional k-means algorithm uses the euclidean distance measure to assign each data point to its closest *centroid*, which is the mean of all the members in the cluster it represents. This measure lies at the core of its functionality, and it is this measure that is vulnerable to the curse of dimensionality. Another known drawback of traditional k-means clustering is its vulnerability to outliers. Each point in the dataset must be part of a cluster. Consequently, outliers may severely affect the mean of the cluster members, and thus the attributes of the *centroid* representing that cluster. Instead of representing each cluster by a centroid, whose features is the mean of all the data points in its cluster, this approach uses *medoids* which Costa et al. [2016] defines as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. For this reason, it is less sensitive to outliers. Another advantage of this method is that it can take any distance matrix as input, whereas the traditional k-means clustering algorithm requires an $n \times m$ matrix of n users and m attributes. This property makes the algorithm more versatile as the distances can be precomputed using the similarity measure that best fits the data. Algorithm 4 shows this technique as it was described in Costa et al. [2016].

Algorithm 4 K-medoid clustering algorithm

Input: $n \times n$ matrix of precomputed distances.
1: Initalization randomly select k of the n data points as medoids
2: Associate each data point to the closest medoid
3:
for each medoid m **do**
    for each non medoid data point o **do**
        Swap m and o and compute total cost of configuration
    end for
end for
4: Select the configuration with the lowest cost
Repeate steps 2-4 until there is no change in medoids

Costa et al. [2016] argue that the advantage of this implementation is that it is scalable and it's convergence is proved regardless of the dissimilarity measure. The technique was applied to a data set with 10 000 users and 72 000 items in Costa et al. [2016] and should therefore be scalable enough for the purposes of this paper.

Blondel's Algorithm

In Boratto et al. [2009] each user is represented by a node in a graph where each edge is weighted by the similarity between the users it connects. This similarity could be computed with any of the similarity metrics defined in this review. Boratto et al. [2009] then use an algorithm for detecting communities in large networks presented by Blondel et al. [2008]. The paper does not name the algorithm, but in this thesis it will be referred to as Blondel's algorithm.
The objective of the algorithm is to maximize an objective function known as modularity. The modularity of a given partition is a scalar value between -1 and 1 that measures the density of links inside clusters compared to the links between the clusters. Blondel et al. [2008] define modularity as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{4.13}$$

Let A_{ij} be the weight of the edge between node i and node j. $k_i = \sum_j A_{ij}$ is the sum of the weights attached to node i, c_i is the community to which node i is assigned, the δ function $= 1$ if $c_i = c_j$ and 0 otherwise and $m = \frac{1}{2} \sum_{ij} A_{ij}$. According to Blondel et al. [2008] exact modularity optimization is a computationally hard problem. The proposed method is a method which approximates this optimization while also providing a hierarchical community structure for the network, allowing for different resolutions of community detection. The algorithm is divided into two phases that are repeated iteratively.

Step 1: Assign a different cluster to each node in the network so that there are as many clusters as there are nodes in the network. For each node i consider the neighbors j of i and calculate the gain in modularity that would be the result of moving i to the community of j. Place i in the community of the neighbour j that would provide the maximum gain in modularity. This step is then applied repeatedly and sequentially until no further improvement can be achieved by moving one node. The gain in modularity ΔQ from

moving an isolated node i to cluster C is computed as in equation 4.14

$$\begin{aligned}\Delta Q = &\left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] \\ &- \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]\end{aligned} \tag{4.14}$$

Here $\sum_{in}$ is the sum of weights of the links within cluster C, $\sum_{tot}$ is the sum of weights of the links incident to nodes in C, k_i is the sum of weights of links from i to nodes in C and m is the sum of the weights of all links in the network. A similar equation is used to calculate the change in modularity when moving a node from one cluster to another.

Step 2: When no further improvements in modularity can be made by moving any one node between clusters, a new network is built. In this network the nodes are the communities found in the previous phase. The weight of links between two clusters is computed as the sum of the weight of the links between nodes in the two corresponding communities. In this phase, links between nodes in the same cluster lead to self loops.

End condition: The two steps are repeated iteratively, with the output of step one used as the input for step two and vice versa until there is no change in modularity.

In Blondel et al. [2008] the algorithm was tested on several data sets and compared to other competing algorithms in the same field. The results showed that the algorithm achieved both better modularity and running time than it's competitors. On a network with 118 million nodes and 1 billion links, the algorithm had a running time of 152 minutes, whereas the competing algorithms had a running time of over 24 hours. On a network with 70 thousand nodes and 351 thousand links it had a running time of 1 second, indicating that it should easily handle the scale of the data in this thesis.

4.5.5 Hybrid Approaches

An effective way of overcoming the drawbacks of either content-based or collaborative filtering recommender systems is to combine the two. Adomavicius and Tuzhilin [2005] lists the following ways to combine these approaches:

- Implementing the two approaches separately and then combining their predictions.
- Incorporating characteristics of one of the approaches into the other.
- Constructing a model that uses both content-based and collaborative characteristics.

These hybrid approaches however, are not relevant to this thesis as they deal with improving rating accuracy in recommender systems after the groups have been created, and exploring this further will not be helpful when answering the research questions defined in Section 4.1.

4.5.6 Subspace Clustering

A second approach to recommender systems, which in method differs quite significantly from that which has been presented earlier in this review is subspace clustering. Müller et al. [2009] describes several variations of subspace clustering, but none in sufficient detail for the author to be able to reproduce them. They are therefore not included in this review. Agarwal et al. [2005] however, describes a reproducible approach, which is described below. This approach is according to Agarwal et al. [2005] an approach which is able to find low dimensional clusters in very high dimensional data. In their paper, they argue that collaborative filtering and content-based approaches are computationally complex and that they are vulnerable to sparse data. The presented domain may at first seem quite similar to that which is presented in this thesis. The addressed problem is that of recommending research papers to researchers with similar interests. However they argue that the number of users in their system is significantly lower than the number of research papers, and it is this property that they exploit in their proposed solution. In the case of this thesis however, there are nearly three times as many users as there are dimensions. Thus, the proposed method may not translate well to the domain in this thesis, but it is included because of how well it deals with data sparsity.

A problem which is likely to be present in many recommender systems is that for any given user; the fraction of content that has been visited or rated is significantly smaller than the fraction of content that is unvisited. To represent the data as an $M \times N$ matrix of ratings, would likely mean a very sparse matrix. In their paper Agarwal et al. [2005] address this issue by representing each user's feature vector as a list of indexes of the articles that the user has visited rather than a row of binary values indicating whether the user has visited the article that corresponds to the given index. This approach leads to reduced memory and time consumption as there is no need to store the zeros and no need to perform

unnecessary computations. Algorithm 5 shows a pseudocode interpretation of the algorithm presented by Agarwal et al. [2005]. Each row represents a user and contains the indexes of articles visited by the user. An example of this can be seen below:

- Row 1: 1,2,3,4
- Row 2: 2,3,5
- Row 3: 2
- Row 4: 1,3,4

In the form presented in algorithm 5 the subspace clustering algorithm does not account for overlapping subspaces. This can be mitigated by grouping together subspaces that differ from each other less than some predefined measure depending on the application.

Algorithm 5 Pseudocode interpretation of subspace clustering algorithm suggested by Agarwal et al. [2005].

Input: *minimum_density* ▷ Minimum size of clusters to keep
Input: Rows of data
Output: Set of subspaces with more cluster members than *minimum_density*
G = hash table ▷ Initiate global hash table
for row_i from row_0 to row_n **do**
  temp = hash table ▷ Initiate local hash table each iteration
  for *sub_row* from row_i to row_n **do**
    intersect ← intersection between *subrow* and row_i ▷ Articles visited by both users
    if *intersect* is already a key in *temp* **then**
      add index of *sub_row* to *temp*[*intersect*] ▷ Add cluster member
    else
      temp[*intersect*] ← indexes of row_i and *sub_row* ▷ Keep track of cluster members
    end if
  end for
  for *key* in *temp* keys **do** ▷ Loop through subsets
    if *key* not in G keys **then** G[*key*] ← value of *temp*[*key*]
    end if ▷ Adds all new subsets to global hash table
  end for
end for
delete all entries in G with fewer members than *minimum_density*
sort G by length of keys in descending order
for *key* in sorted G **do**
  for *subsequent_key* in sorted G **do**
    if *subsequent_key* is subset of *key* **then**
      delete G[*subsequent_key*] ▷ delete subset if it is subsumed by another subset
    end if
  end for
end for
return G

4.6 Summary

Seeing as the driving force for this literature review was the research questions defined in Section 4.1, it is natural to summarize this review in terms of these research questions.

How can similarity be computed between two users based on what resources they visit?

The literature showed in total four ways of computing similarity between users based on what resources they visit:

- Cosine Similarity
- Pearson Correlation
- Spearman Rank Correlation
- Frequency-Weighted Pearson Correlation

The cosine similarity metric had the disadvantage that it does not account for the mean and variance in the ratings of the individual users. The Pearson correlation metric removes these components before computing similarities between users, and is the metric that was most frequently used in the reviewed literature. *Spearman Rank Correlation* is a variant of the Pearson correlation metric that looks at each user's relative ranking of items instead of rating. This metric proved to give slightly better results when compared to *Pearson Correlation* and may be suitable for this thesis. The *Frequency-Weighted Pearson Correlation* metric is another variation of the *Pearson Correlation metric* that accounts for the fact that some items may be preferred by many users, and punishes these items when computing similarity. The data set in this thesis contain articles that are of relevance to most of the users, and may benefit from disregarding these articles when computing similarities.
In addition to these metrics, modifications such as **default voting** as described by Breese et al. [1998] and **default voting** as described by Desrosiers and Karypis [2011] have been used to increase accuracy when similarity is computed based on users whose set of co-rated items is relatively small.

What clustering techniques have previously been used to discover clusters based on user interests?

Three clustering techniques have been discussed in this review. The so called Blondel's algorithm, the k-medoids clustering algorithm and the subspace clustering algorithm described in algorithm 5. To answer **research question 3**

whether the presented techniques could be used for this thesis with regard to scalability; all of the presented clustering techniques have been used on data sets larger than the one in this thesis. However, the subspace clustering algorithm only accounts for whether or not a user has visited the article, disregarding important information such as ranking. The evidence presented in the literature suggests that Blondel's algorithm is more scalable than the k-medoids clustering algorithm as the experimental results of Blondel et al. [2008] included larger data sets than Costa et al. [2016].

Which of the techniques from RQ2 can be transferred to the problem presented in this thesis with regard to scalability?

Seeing as Sticos would like to be able to use the work presented in this thesis in the future it seems a good option to select the more scalable alternative for clustering algorithm. As such the algorithm described by Blondel et al. [2008] will be used to cluster the data in the next phase of this project. Furthermore both *Spearman Rank Correlation* and *Frequency-Weighted Pearson Correlation* will be used as similarity metrics in the next phase incorporating both *default voting* and *significance weighting*. Default voting will be used because the voting scheme in this thesis an *implicit one* based on the observed activity rather than an *explicit one* based on explicitly expressed ratings. As Breese et al. [1998] argued, in implicit voting schemes an observed vote is typically an indication of positive preference for a resource, whereas an unobserved vote is an indication that the user is not interested in the resource. Finally, significance weighting will be used to weigh down similarities computed when the number of co-rated items is small.

Chapter 5

Experimental Setup and Results

Chapter 4 gave a review of the state of the art techniques that have been used to solve similar problems to the problem this thesis attempts solve. This chapter will describe how these techniques are applied to the specified problem as well as discussing the achieved results. The first part of the chapter will discuss the application of both the Spearman Rank Correlation metric (Equation 4.10) and the Frequency-Weighted Pearson Correlation metric (Equation: 4.11) with Blondel's algorithm (section 4.5.4) from Chapter 4 on a randomly selected sample of 10 000 of Sticos' users, where the similarity is computed based on all the data in the web logs. The second part will discuss the application of the Frequency-Weighted Pearson Correlation metric to the data separated into months.

5.1 Overview Of the First Experiment

The activities in Chapter 3 were performed during the autumn of 2016, whereas the efforts of this chapter took place in the spring of 2017. The copy of the database that was originally given to the author was made in September 2016 so the first step was to make a new copy of the database and then retake the steps from Chapter 3 by reading all the data into a normalized database with the fields indicated in table 3.2. Figure 5.1 gives a graphical overview of the experimental process of the first experiment. Each subprocess will be described in detail in it's own subsection below.

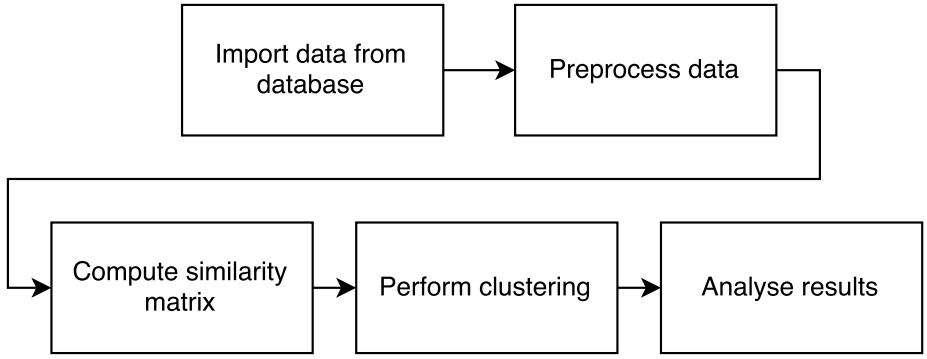


Figure 5.1: Process overview of first experiment

5.1.1 Importing Data from the Database

Having retaken the steps from Chapter 3, with the data stored in an database with the fields indicated in 3.2, the process of importing the data was simply a matter of moving the data from the database to another means of storage which is more easily accessed in the Python programming language. This was achieved by running an SQL query against the database to retrieve all entries contained within it. A Python dictionary object indexed on user id was created, and the rows from the database were added to their corresponding users, with the end result being a dictionary object connecting user ids to a chronological list of events corresponding to said users' traffic logs. Recalling Section2.2, a common preprocessing task is that of aggregating the data across different dimensions. Seeing as the similarity measures explored in this thesis require a vector of ratings, aggregation is a logical means of preprocessing the data in this thesis. For each user, the list of chronological events was iterated through and the number of visits to each resource were aggregated such that each user was presented by a dictionary of resource ids with a corresponding number signifying total hits to said resource. Figure 5.2 shows how one user might be represented. In this representation the key tuple is a unique resource identificator, with the first entry stating which of the following types the resource belongs to :

1. An article
2. A document
3. An account plan

The second entry in the tuple refers to the resource's unique identifier within the type it belongs to. The entries in the dictionary specify the user's total traffic to

the corresponding resource.

$\{(3,1):10,(1,7657):32,(1,4560):9,(2,2341):14\}$

Figure 5.2: Example of user representation

5.1.2 Preprocessing

FWPC and SRC both have the strength that they remove the variance and the mean from each user's vectors when computing similarity. However, this thesis will in addition to aggregation, also use feature scaling to further preprocess the data to adapt it to be as similar to the data sets that were used in the literature. Both FWPC and SRC normalize ratings before computing similarities, but information is lost since the ratings are normalized relative to the elements used in the similarity computation, namely the co-rated items. Feature scaling the attributes before the similarity computation retains the information of how good a given attribute is relative to the item the user visits most often.
At this point the data can either be scaled so that all elements in the feature vector sum up to one, so that each entry represents a percentage of the user's total traffic or by dividing all elements in the vector by the largest element in the vector. As described in Section2.2.6 one normally scales the attribute relative to all recordings of the same attribute in the database, but in this thesis each attribute will be scaled relative to other attributes from the same user. The rationale for this is that similarity is computed based on an implicit voting scheme in this thesis, whereas it is computed based on an explicit voting scheme in the reviewed literature. Scaling attributes relative to other attributes from the same user allows for the expression of each user's preference for the given article relative to his or her frequency of use. In other words, it better matches with explicit voting schemes, as preference for individual resources are given on a fixed scale. More formally this is computed as

$$\frac{R(U) - \max_{i \in I} R(U)_i}{\max_{i \in I} R(U)_i - \min_{i \in I} R(U)_i} \tag{5.1}$$

Here $\max_{i \in I} R(U)_i$ is highest value in the rating vector, $R(u)$ is the current element and $\min_{i \in I} R(U)_i$ is the lowest possible rating - 0.
To justify the choice of feature scaling scheme, an example is due. In this example similarity is to be computed between the following two users:

User 1: $(3,1):100,(1,7657):100,(1,4560):100,(2,2341):14$

User 2: $(3,1):100,(1,7657):100$

Using the first form of feature scaling gives the following representation:

User 1: $(3,1) : 0.32, (1,7657) : 0.32, (1,4560) : 0.32, (2,2341) : 0.04$

User 2: $(3,1) : 0.5, (1,7657) : 0.5$

User 2 only visits two subjects, and prefers them equally, thus each resource is given the value 0.5. User 1 however, prefers the first three resources equally much, in addition to visiting the fourth resource once in a while. In both cases, resource (3,1) and (1,7657) have the highest possible preference value, but this is not reflected by the scaled value because user 1 also has a significant amount of traffic elsewhere. In some cases it may be relevant to include the information that user 1 also has a lot of traffic to other resources, but for the purposes of this project, the aim of feature scaling is to capture each user's relative preference for the different resources, which the second feature scaling scheme indeed does. The second feature scaling scheme yields the following representation.

User 1: $(3,1) : 1, (1,7657) : 1, (1,4560) : 1, (2,2341) : 0.14$

User 2: $(3,1) : 1, (1,7657) : 1$

The author notes however, that values computed by the first scheme incorporates the information that user 1 also prefers other resources whereas the values computed by the second scheme does not. This information does indeed have value, but this information will be incorporated by using default voting as described in Section4.5.3 when computing similarity between users. Whereas the literature in Chapter 4 mostly concerns applications with explicit rating schemes, the rating scheme in this thesis is an implicit one. In explicit voting schemes an unrated resource is assumed to not yet have been visited, and thus contains little information. As [Breese et al., 1998] argues however, in implicit voting schemes an unvisited resource may suggest that the user is not interested in said resource. This is the rationale for incorporating default voting into the similarity computation in this thesis. After feature scaling users 1 and 2 are represented as stated above, but at the time of similarity computation they will be represented as follows:

User 1: $(3,1) : 1, (1,7657) : 1, (1,4560) : 1, (2,2341) : 0.14$

User 2: $(3,1) : 1, (1,7657) : 1, (1,4560) : 0, (2,2341) : 0$

The last preprocessing step is that of *sampling*. A metaphor comes to mind when determining the necessary number of computations to compute a similarity matrix. The metaphor is the classical math problem named *The Handshake Problem* in which students are tasked with finding the total number of handshakes necessary for all N people in a room to have shook hands exactly once. Every time a new person is introduced to the room, N handshakes need to take place.

In mathematical terms the number of handshakes can be computed as $n(n-1)/2$. This holds for the computation of similarity matrices well, but instead of handshakes, a number of heavy set and vector computations have to be performed to arrive at a similarity score. The worst case being the case where all users have visited all 9868 resources, requiring the computation of similarity between two vectors containing 9868 elements for each of the similarity computations between all pairs of users. Needless to say the problem does not scale well and the use of sampling will be necessary. Running the computation program for different selections of users suggest that a selection of 10 000 users should yield a computation time of around three hours. This amounts to 49 995 000 pair-wise similarity computations.
The sampling methods described in Section 2.2.2 are *stratified sampling* and *simple random sampling*. Seeing as the author has no previous knowledge of the underlying clusters, the sampling scheme that is chosen for this project is simple random sampling. To ensure that the data is as information rich as possible the percentiles for both total traffic and number of distinct resources visited per user was computed for the entire data set, and the 10 000 users were sampled from the set of users with total traffic and number of distinct resources visited above the 70th percentile. This amounts to at least 20 resources and 100 visits.

5.1.3 Computing the Connection Matrix

The steps taken in the preprocessing phase were aimed at preparing the data so that the similarity computations had the best prerequisites for arriving at meaningful results. In this experiment, two different similarity matrices are to be computed. One for the FWPC metric and one for the SRC metric.
As it is possible that many users only share a small subset of corated items the experiment will also incorporate the significance weighting described in Section4.5.3. The threshold is set to 10, weighing down similarities computed based on less than 10 common resources. Algorithm 6 was performed for each user, and their similarity was inserted into the similarity matrix When computing FWPC similarities, an additional step of computing each resource's *inverse user frequency* as described in Section4.5.3 was added at the beginning. The similarity function in *step 3* of algorithm 6 was a function the author wrote as a modification of the "pearsonr" function in the open source module "scipy.stats"([Jones et al., 2001]) in which the vector of inverse user frequencies for the corresponding resources is incorporated into the computation. The total computation time for the FWPC similarity matrix computation was approximately 3 hours and 3 minutes.
The computation of the SRC similarities was performed exactly as in algorithm 6, where the similarity function in *step 3* was the "spearmanr" function from

Algorithm 6 How computation of similarity between two users is done

Input: Two dictionary objects representing each user's relative preference for different resources
Step 1: Create a set of all the keys contained in either of the users dictionaries to find items rated by either of the users
Step 2: Create an empty feature vector for each both users
for key in set of keys **do**
    for Both users **do**
        if key in users set of keys **then**
            Add corresponding value to vector
        else
            Add the value zero to vector
        end if
    end for
end for
Step 3: Input data vectors into similarity function
Step 4: Factor in significance weight

the "scipy.stats"([Jones et al., 2001]). The total computation time for the SRC similarity matrix computation was approximately 5 hours and 33 minutes.

5.1.4 Clustering the Data

Having overcome the time-consuming task of generating the connection matrices, the remaining task is that of clustering them. As discussed in the literature review in Chapter 4 the clustering algorithm that will be used in this experiment is Blondel's algorithm described in algorithm 5. Two python modules were used to perform this task. A python implementation of Blondel's algorithm by Aynaud [2012] called "python-louvain" and a graph structure module by Hagberg et al. [2008] called "NetworkX". The NetworkX module was used to convert the connection matrix to a format that the "python-louvain" module could operate on and the clusters were generated by using the "best_partition" function of the "python-louvain" module which returned a list of labels corresponding to the indices of the input data. This list of labels was then input to a function that matches the list of labels with user ids, and sorts the user ids into a dictionary object, indexed by cluster label. The FWPC clusters were computed in 2 minutes and 53 seconds. However, the computer's 8GB of memory was insufficient to compute the clusters based on the SRC connection matrix. After having upgraded the computer's memory to a total of 24 GB, the computation of the SRC connections took 24 minutes and 46 seconds.

5.2 Discussing the Results

This section will provide a short summary of the results and their impact on the direction of this project. The reader is referred to appendix B for a more thorough analysis of the results of experiment one. In the analysis of the results, the feature scaled values described in Section5.1.2 will be referred to as the user's relative preference or relative rating. Table 5.1 shows how the FWPC and the SRC metric divided the users in their respective clusters. As shown in appendix B the clusters discovered in this experiment poorly matches Sticos' idea of their user segments. The confidence in these clusters and their business value is further weakened when looking at the ground truth data as it is presented in appendix B. A suggested explanation for these poor results is that over time, Sticos' users become more and more similar. This explanation necessitates a new experiment, in which the data is analysed in smaller time frames. It is this experiment that will be the main focus of the results discussion later in this thesis. In the discussion

Table 5.1: User distribution in clusters based on FWPC and SRC

	FWPC	**SRC**
Noise	3	1
Cluster 1	6647	4787
Cluster 2	1479	5122
Cluster 3	1682	90
Cluster 4	189	

about the SRC measure in Section4.5.3 the literature suggested that this measure often is slow as it necessitates the sorting of both users' feature vectors before computing similarity. The author noted that this might not be a problem in this thesis as the number of items likely to be in the feature vector was small. Comparing the computation time for the similarity matrices based on FWPC and SRC, FWPC finished the computation in 3 hours and 3 minutes whereas SRC took roughly 5 hours and 30 minutes proving the author's initial assertion wrong. In addition to being more costly when computing the similarity matrices, the clustering computation was more memory intensive than FWPC, likely because the SRC metric produced more similar users, so more edges needed to be stored in memory. Lastly, the results in table 5.1 may suggest that FWPC is stricter when computing similarities, separating the users into smaller clusters. For these reasons, only the FWPC similarity measure will be used in the second experiment.

5.3 Overview of the Second Experiment

Apart from a few technicalities, the second experiment is a repetition of the first experiment, but with the data separated in smaller time frames. Possible time frames include quarter years, months, weeks or days. Looking at the web logs, it seems that the average activity levels for days and weeks will make for sparse feature vectors and possibly as a consequence of this; poor results. This leaves months and quarter years as candidates. Seeing as months are of a higher granularity than quarter years, and more scalable since less data is generated in it's timespan, it is the preferred time frame for this experiment.
It goes without saying that there is less traffic in a month than two years, so the significance weighting parameter is adjusted down to 4 such that similarities between users with less than 4 corated items are weighed down. Figure 5.3 shows the process overview for the second experiment. Comparing it to figure 5.1 one will find that the process overviews are similar except that the similarity computations and cluster computations are repeated for each month.

5.3.1 Modifications

The added benefit of analysing the data month by month is that the reduced computational effort when analysing month by month removes the need for sampling. Therefore sampling is not a part of the preprocessing in the second experiment. A new step is added however, to sort the data by month. Recalling Section5.1.1 the data is organized in a chronological list for each user before it is aggregated. The data was sorted by iterating through this list, assigning each data log entry to it's corresponding month in a dictionary object indexed by month, containing the same aggregated dictionary objects as in the previous experiment. In the discussion of the results of experiment one, each cluster was represented by the relative aggregated resource ratings by it's cluster members. This representation makes it possible to sort the different resources by their aggregated ratings, and thus represent the relative preferences of the group. However, in the cases were no particular resource had a disproportionate preference value, such as in figure B.6a, it is not possible to decide whether this is due to high variance in the individual top preferences or if the users of the cluster prefer several resource equally much.
Because the feature vectors are scaled to the range 0-1 it is also possible to represent each cluster's members' preference for the various resource by the average ratings made by the cluster members. By looking at the averaged ratings instead of the scaled aggregated ratings, it is possible to determine whether there is a lot of variance in the preferences of the cluster members.

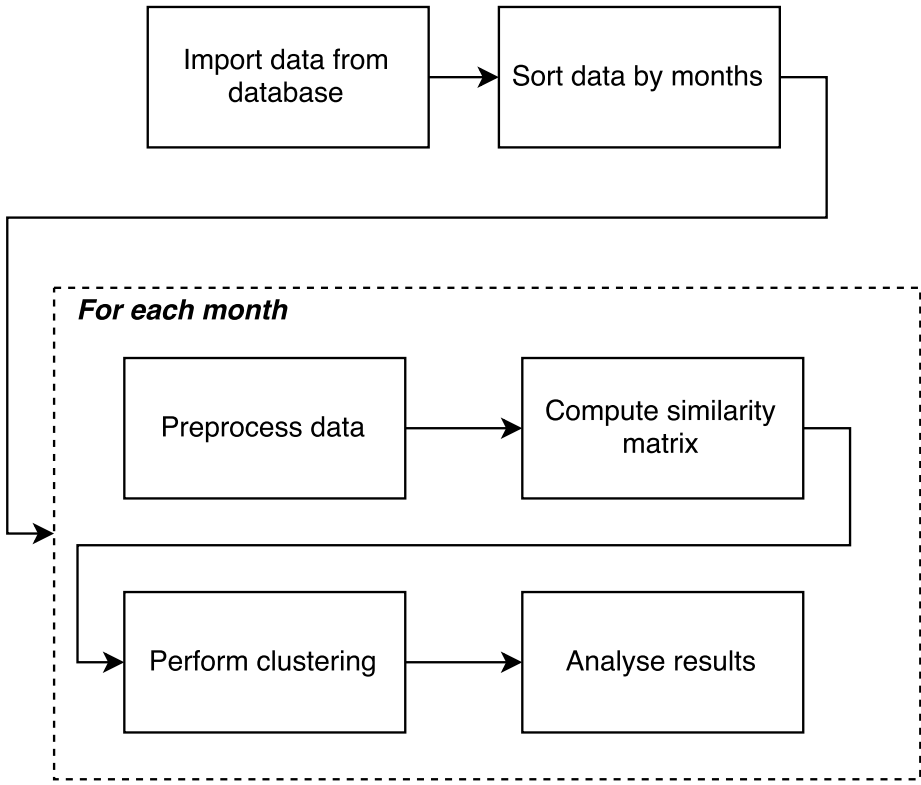


Figure 5.3: Process overview of second experiment

5.4 Results - Second Experiment

For this experiment, the results include 24 different result sets, one for each month since April 2015 to March 2017 and several clusters within each month. To limit the length of this thesis, only a subset of the results will be included. More specifically this paper will include a discussion of the results for January 2016 and June 2016 to show results for two different seasons, as well as January 2017 to show the same month over two years. All of the clusters for these three months are included in appendix C. The rest of this chapter will be dedicated to the textual description and the discussion of these clusters, but the reader is referred to appendix C for the figures describing the clusters. In the discussion of the results, a cluster is meaningful if there is a semantic relationship between individual resources and if high average rating values are present in either top

categories or top individual resources.

5.4.1 Structure of Results Analysis

As stated in Section1.2 the results of the experiment will be analysed qualitatively. To facilitate the comparison of results across different data time frames, the clusters will be analysed following a framework designed to synthesize their describing features. For each time frame the analysis will start by looking at how the users are distributed across the individual clusters and then look at each individual cluster's describing features in terms of their top average preference values, defined as the average of the normalised ratings of the individual cluster's members. By looking at these describing the author will determine whether there is a semantic relation between the top resources, and classify the cluster as meaningful if there is a semantic relation between resources and there is a presence of high average preference values. Finally, a comment will describe why the cluster was or was not classified as meaningful.

5.4.2 January 2016

When clustering the user generated traffic from January 2016 a total of 7 clusters were discovered, with the users distributed as follows:

- Cluster 1(fig. C.1): 2175 users
- Cluster 2(fig. C.2): 1732 users
- Cluster 3(fig. C.3): 2365 users
- Cluster 4(fig. C.4): 3834 users
- Cluster 5(fig. C.5): 2297 users
- Cluster 6(fig. C.6): 1497 users
- Cluster 7(fig. C.7): 565 users

Comparing this distribution with the distributions of the clusters generated in experiment one (Appendix B), the users in these clusters are more evenly distributed amongst a larger number of clusters than in the clusters of the first experiment.

Cluster 1 (fig. C.1)

- Users: 2175
- Top categories: "Satser" - 0.4, "Fildokument" - 0.28, "Reiseutgifter" - 0.15
- Top individual resource: "Bilgodtgjørelse ved bruk av egen bil på reiser i Norge": 0.34
- Semantic relation: Work related travel
- Meaningful: Yes

- Comment: Fits criteria

Cluster 2 (fig. C.2)

- Users: 1732
- Top categories: "Naturalytelser" - 0.58, "Fildokument" - 0.39
- Top individual resource: "Fri bil" - 0.61, "Firmabil - beregning av fordel" - 0.39
- Semantic relation: Payment in kind(Norwegian: Naturalytelser)
- Meaningful: Yes
- Comment: Fits criteria

Cluster 3 (fig. C.3)

- Users: 2365
- Top categories: "Gaver - ansatte og forretningsforbindelser" - 0.49
- Top individual resource: "Gaver -ansatte og styremedlemmer" - 0.41
- Semantic relation: Gifts and employee welfare
- Meaningful: Yes
- Comment: Fits criteria

Cluster 4 (fig. C.4)

- Users: 3834
- Top categories: "Fildokument" - 0.23
- Top individual resource: "Kontoplan: Sticos kontoplan for aksjeselskaper(bygger på NS 4102)" - 0.05
- Semantic relation: No apparent relation
- Meaningful: No
- Comment: The rest of the categories have an average rating of less than 0.1. Furthermore the highest rated individual resource has an average rating of less than 0.05 and there is no apparent semantic relation between the individual resources. Seeing as "Fildokument" is the category encompassing all downloadable documents from different categories, a high preference value for this category alone is not an indication of a meaningful cluster. Therefore this cluster does not seem meaningful. It may be that this cluster encompasses the users with no specific usage patterns or users that use the system infrequently.

Cluster 5 (fig. C.5)

- Users: 2297

- Top categories: "Kontoplan" - 0.89
- Top individual resource: "Kontoplan: Sticos kontoplan for aksjeselskaper(bygger på NS 4102)" - 0.93
- Semantic relation: Kontoplan
- Meaningful: Yes
- Comment: Apart from the top resources, other resources have a preference value less than 0.05, indicating that the top resource is the most important resource for the members of this group.

Cluster 6 (fig. C.6)

- Users: 1497
- Top categories: "Utgiftsgodtgjørelse" - 0.43
- Top individual resource: "Fri telefon" - 0.44
- Semantic relation: General topics
- Meaningful: Somewhat
- Comment: Although fairly high average ratings are present, there is no apparent semantic relation between top individual resources other than that they are fairly general. As such the cluster may be said to be meaningful, but not as much as the other meaningful clusters for this month.

Cluster 7 (fig. C.7)

- Users: 565
- Top categories: "Fildokument" - 0.59, "Periodiseringer og avsetninger" - 0.18
- Top individual resource: "Arbeidsgiveravgift 2015" - 0.28
- Semantic relation: Managerial topics(Salary, Accounting)
- Meaningful: Yes
- Comment: In addition to the top individual resource, the four next entries on the top list have average preference values ranging from 0.17 to 0.13 which is fairly high compared to the ratings for topics at the position in the preference lists of other clusters. This can be an indication that the many members of the cluster have a high preference for these topics.

Summary

In total five of the clusters were categorized as meaningful, one was categorized as somewhat meaningful and one was categorized as not meaningful. Seeing as the chosen clustering method will assign every member to some cluster, this last cluster may contain those members that could not be assigned to any specific cluster, in other words; the users that had no close neighbours.

5.4.3 January 2017

During the clustering of the traffic from January 2017, a total of 8 clusters were discovered and the users were distributed among them as follows:

- Cluster 1: 3138 (fig. C.8)
- Cluster 2: 6356 (fig. C.9)
- Cluster 3: 2429 (fig. C.10)
- Cluster 4: 1447 (fig. C.11)
- Cluster 5: 1006 (fig. C.12)
- Cluster 6: 2533 (fig. C.13)
- Cluster 7: 376 (fig. C.14)
- Cluster 8: 3 (fig. C.15)

Ignoring cluster 8 due to it's insignificant size, it seems as though the same number of clusters have been discovered for January 2017 as for January 2016. Below is a summary of the different clusters' characteristics.

Cluster 1 (fig. C.8)

- Users: 3138
- Top categories: "Satser" - 0.33, "Fildokument" - 0.29 and "Utgiftsgodtgjørelse" - 0.20
- Top individual resource: "Bilgodtgjørelse ved bruk av egen bil på reiser i Norge"
- Semantic relation: Work related travel
- Meaningful: Yes
- Comment: As both high average ratings are present and the individual resources are semantically related, this cluster seems meaningful.
- Similar cluster from previous year: Cluster one (fig. C.1)

Cluster 2 (fig. C.9)

- Users: 6356
- Top categories: "Fildokument" - 0.19
- Top individual resource: "Lån til personlig aksjonær" - 0.07
- Semantic relation: No apparent relation
- Meaningful: No
- Comment: Apart from having a high average rating for the category "Fildokument" there are no high preference values present and no apparent semantic relation between individual resources.
- Similar cluster from previous year: Cluster four (fig. C.4)

Cluster 3 (fig. C.10)

- Users: 249
- Top categories: "Kontoplan" - 0.89
- Top individual resource: "Kontoplan: Sticos kontoplan for aksjeselskaper(bygger på NS 4102)" - 0.94
- Semantic relation: Account plan
- Meaningful: Yes
- Comment: Apart from the top resources, other resources have a preference value less than 0.05, indicating that the top resource is the most important resource for the members of this group.
- Similar cluster from previous year: Cluster 5 (fig. C.5)

Cluster 4 (fig. C.11)

- Users: 1447
- Top categories: "Naturalytelser" - 0.59, "Fildokument" - 0.37, "Satser" - 0.17
- Top individual resource: "Fri bil": 0.64, "Firmabil - beregning av fordel" - 0.37
- Semantic relation: Payment in kind (Norwegian: Naturalytelser)
- Meaningful: Yes
- Comment: Fits criteria
- Similar cluster from previous year: Cluster 2 (fig. C.2)

Cluster 5 (fig. C.12)

- Users: 1006
- Top categories: "Kjøp av driftsmidler" - 0.32, "Utland - kjøp og innførsel" - 0.20, "Fildokument" - 0.15
- Top individual resource: "Innførsel av varer -oversikt" - 0.32, "Kjøp av fjernleverbare tjenester fra utlandet" - 0.19
- Semantic relation: International trade / import
- Meaningful: Yes
- Comment: Fits criteria for meaningful cluster, no similar clusters from previous year
- Similar cluster from previous year: None

Cluster 6 (fig. C.13)

- Users: 2533
- Top categories: "Gaver - ansatte og forretningsforbindelser" - 0.43
- Top individual resource: "Gaver - ansatte og styremedlemmer" - 0.34

- Semantic relation: Gifts and employee welfare
- Meaningful: Yes
- Comment: Fits criteria
- Similar cluster from previous year: Cluster 3 (fig. C.3)

Cluster 7 (fig. C.14)

- Users: 376
- Top categories: "Fildokument" - 0.68
- Top individual resource: "Oversikt avstemminger" - 0.17, "Arbeidsgiver-avgift" - 0.17
- Semantic relation: Managerial topics(Salary, Accounting)
- Meaningful: Yes
- Comment: In addition to the top two individual resources, the four next entries on the top list have average preference values ranging from 0.15 to 0.12 which is fairly high compared to the ratings for topics at the position in the preference lists of other clusters. This can be an indication that many members of the cluster have a similar preference for these topics.
- Similar cluster from previous year: Cluster 7 (fig. C.7)

Cluster 8 (fig. C.15) and summary

Cluster 8 had an average rating value of 1 for the indiviudal resource " Emne: Salg av ny elbil" and it's corresponding category "Salg av nytt kjøretøy" and the five next preference values were 0.11. However analysing this cluster further serves little purpose, as it only contains three users. For this period six of the discovered clusters are classified as meaningful, and six of the clusters have a similar cluster in the same time frame for the previous year.

5.4.4 June 2016

During the clustering of June 2016, a total of eleven clusters were discovered and the users were distributed among them as follows:

- Cluster 1 (fig. C.16): 6321
- Cluster 2 (fig. C.17): 2047
- Cluster 3 (fig. C.18): 1682
- Cluster 4 (fig. C.19): 3017
- Cluster 5 (fig. C.20): 889
- Cluster 6 (fig. C.21): 676
- Cluster 7 (fig. C.22): 4
- Cluster 8 (fig. C.23): 147

- Cluster 9 (fig. C.24): 303
- Cluster 10 (fig. C.25): 6
- Cluster 11 (fig. C.26): 3

Ignoring clusters 7, 10 and 11 due to their size a total of 8 clusters were discovered and their characteristics will be summarised below:

Cluster 1 (fig. C.16)

- Users: 6321
- Top categories: "Fildokument" - 0.199, "Disponering og egenkapital" - 0.12
- Top individual resource:"Fritaksmetoden" - 0.072
- Semantic relation: No apparent relation
- Meaningful: No
- Comment: No high average rating values present. No semantic relation
- Similar cluster(s) from other time frames: Cluster four (Januar 2016) C.4) and cluster two C.9)

Cluster 2 (fig. C.17)

- Users: 2047
- Top categories: "Kontoplan" - 0.907
- Top individual resource:"Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)" - 0.956
- Semantic relation: Account plan
- Meaningful: Yes
- Comment: Apart from the top resources, other resources have a preference value less than 0.05, indicating that the top resource is the most important resource for the members of this group.
- Similar cluster(s) from other time frames: Cluster 5 January 2016 (fig. C.5) and cluster 3 January 2017 (fig. C.10)

Cluster 3 (fig. C.18)

- Users: 1682
- Top categories: "Feriepenger" - 0.39
- Top individual resource:"Emne: Feriepengeavregning til ansatte med månedslønn og avtalefestet fem uker ferie" - 0.249
- Semantic relation: Vacation and salary during vacation

- Meaningful: Yes
- Comment: Fairly high rating values present, clear semantic relation. No similar clusters, seems reasonable considering time of year.
- Similar cluster(s) from other time frames: None

Cluster 4 (fig. C.19)

- Users: 3017
- Top categories: "Fildokument" - 0.24, "Satser" - 0.17
- Top individual resource:"Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge" - 0.076
- Semantic relation: Work related travel
- Meaningful: Somewhat
- Comment: Slightly high values present, some semantic relation
- Similar cluster(s) from other time frames: Some relation to Cluster 1 from January 2016(fig. C.1) and Janaury 2017 (fig. C.8)

Cluster 5 (fig. C.20)

- Users: 889
- Top categories: "Gaver - ansatte og forretningsforbindelser" - 0.61
- Top individual resource:"Emne: Gaver - ansatte og styremedlemmer " - 0.53
- Semantic relation: Gifts and employee welfare
- Meaningful: Yes
- Comment: High rating values present, clear semantic relation
- Similar cluster(s) from other time frames: Cluster 3 January 2016(fig. C.3 and Cluster 6 (fig. C.13)

Cluster 6 (fig. C.21)

- Users: 676
- Top categories:"Naturalytelser" - 0.49, "Fildokument" - 0.36
- Top individual resource:"Emne: Fri bil (firmabil) " - 0.51
- Semantic relation: Payment in kind (Norwegian: Naturalytelser)
- Meaningful: Yes
- Comment: High rating values present, clear semantic relation
- Similar cluster(s) from other time frames: Cluster 2 January 2016 (fig. C.2) Cluster 4 January 2017 (fig. C.11)

Cluster 8 (fig. C.23)

- Users: 147

- Top categories:"Regnskapsførerregelverket" - 0.72
- Top individual resource:"Emne: Oppbevaring av regnskapsmateriale" - 0.72
- Semantic relation: Accounting
- Meaningful: Yes
- Comment: High rating values present, clear semantic relation
- Similar cluster(s) from other time frames: None

Cluster 9 (fig. C.24)

- Users: 303
- Top categories:"Fildokument" - 0.695
- Top individual resource:"Fildokument: Bankavstemming" - 0.28
- Semantic relation: Accounting(Period termination)
- Meaningful: Yes
- Comment: High rating values present, clear semantic relation
- Similar cluster(s) from other time frames: None

Clusters 7, 10 & 11 and Summary

Cluster four had an average preference value of 1 for the individual resource "Emne: Utgifter til drift av båt som brukes til sosiale formål" and it's corresponding category, and a total of four members. Cluster ten had an average rating of 0.66 for the individual resource "Emne: Flere avgiftssubjekter driver virksomhet fra felles lokaler" and a slightly lower average rating for it's corresponding category, and a total of six members. Cluster eleven had an average rating value of 1 for it's top resource "Emne: Flere avgiftssubjekter driver virksomhet fra felles lokaler" and it's corresponding category, and a total of three members. These clusters do seem meaningful as all of the users must have a high preference for the top resource for the average rating to be as high as it is, but they will not be discussed further due to their size. In summary, a total of 11 clusters were discovered, with eight being significant in size. Of these eight clusters, six clusters were meaningful and four had similar clusters from previous time frames and one cluster was slightly meaningful and had some similarity to clusters from other time frames.

Chapter 6

Evaluation and Discussion

This chapter will evaluate the results presented in the Chapter 5 and discuss the overall merits and limitations of this thesis.

6.1 Evaluation

Chapter 5 presented the setup and results of two experiments. The results of the first experiment were presented in appendix B with a summary in Section 5.2. The results of the second experiment were presented in appendix C and discussed in depth in Section 5.4.

6.1.1 First Experiment

To reiterate the description of the first experiment, the data source was aggregated for all the entries contained in the web logs, the similarity was computed using both the SRC and FWPC metric for 10 000 randomly selected users. Upon analysing the clusters, no meaning could be attributed to the discovered cluster, and there seemed to be no semantic relation between the top resources for the individual resources. However, the experiment proved that in terms of scalability, SRC with a computation time of 5 hours and 3 minutes took 2 hours and 30 minutes longer than the FWPC whose computation time was 3 hours and 3 minutes. Furthermore the increased memory requirements of SRC when computing the clusters is an indication that more edges need to be stored in the similarity matrix. More edges is in turn an indication that SRC classifies more users as similar than FWPC, in other words it is not as good at discriminating users from each other. In addition, experience gained during analysis of the clusters showed

that the normalized aggregated preference values used to analyse the clusters had it's limitations, resulting in the use of average ratings in the second experiment

6.1.2 Second Experiment

As the results from the first experiment showed that FWPC was superior to SRC for the purposes of this thesis, the SRC metric was discarded in the second experiment. Recalling the description of the second experiment, it was a reiteration of the first experiment, but with the data separated in months and discussion of the results in Section 5.4 was limited to the clusters generated for three different months due to length considerations.
By analysing the results, the following can be concluded:

- For all three months, the majority of the clusters had semantically related preferences and as such, seemed to be meaningful.
- No meaning could be attributed to the largest cluster that was discovered each month. It may be that the discovered clusters that are indeed meaningful, represent groups of special interest and that the large cluster with seemingly no meaning represents a general user base with no special characterisations.
- Several clusters have similar clusters in all three timeframes. No explanation for why this is will be asserted in this thesis, but a possible explanation may be that some clusters are persistent through the seasonal variations.

The results from the second experiment suggest that the experiment was successful and that meaningful clusters have indeed been discovered. Table 6.1 shows a graphical summary of the results.

Table 6.1: Comparison of semantical description of top preferences. Colors mark similar clusters

Cluster #	January 2017	January 2016	June 2016
Cluster 1	Work related travel	Work related travel	No apparent relation
Cluster 2	No apparent relation	Payment in kind	Account plan
Cluster 3	Account plan	Gifts and employee welfare	Vacation and salary during vacation
Cluster 4	Payment in kind	No apparent relation	Work related travel
Cluster 5	International trade / import	Account plan	Gifts and employee welfare
Cluster 6	Gifts and employee welfare	General topics	Payment in kind
Cluster 7	Managerial topics	Managerial topics	n/a
Cluster 8	n/a		Accounting
Cluster 9			Accounting(Period termination)
Cluster 10			n/a
Cluster 11			n/a

6.2 Discussion

Chapter 4 described a systematic literature review that was performed in order to satisfy the information need that became apparent in Chapter 3 after applying the techniques from the basic theory described in Chapter 2.
The first question of the review was *how can similarity be computed between two users based on what resources they visit?* and the review identified four possible methods. Of these four methods, Spearman rank correlation and Frequency-weighted Pearson correlation were chosen with the modifications *significance weighting* and *default voting*.
The second question was *what clustering techniques have previously been used to discover clusters based on user interests?* and the review identified three algorithms: The subspace clustering algorithm, K-medoids and Blondel's algorithm.
The last question was *which of the techniques from **RQ2** can be transferred to the problem presented in this thesis with regard to scalability?*. The presented data suggested that Blondel's algorithm was the more scalable algorithm and it was for this reason it was chosen for the experiments.

Recalling Section 1.1 the following overview showed the data sets that had been used in the literature:

MovieLens: Agarwal et al. [2005]; Adomavicius and Kwon [2012]; Desrosiers and Karypis [2011]; Boratto and Carta [2014, 2015]; Boratto et al. [2009]; Costa et al. [2016]; Gao et al. [2007]; Li and Kim [2003]; Li and Murata [2012]; Sarwar et al. [2001]; Yanxiang et al. [2013]

Last.fm Costa et al. [2016]

Flickr: Zeng et al. [2012]

Except for Flickr, all of these data sets have in common that they rely on the user giving explicit feedback in the form of a rating for each resource. In contrast to this, the data set in this thesis is based on the implicit feedback of the users, as given by their frequencies of visits to the different resources. As such the data set explored in this thesis is somewhat dissimilar to the data sets explored by the papers identified in the review.
There is a difference in bias between implicit and explicit voting schemes. In an implicit voting scheme a user may have a certain amount of traffic to a given resource, even though the user is not interested in it. This may be because the user clicked the wrong link once, or because the user mixes up two resources and visits one in search of the other. The benefit however, is that over time, it will become apparent which resources the given user visits frequently, and thus the user's interests can be inferred. With explicit voting schemes however, the user has explicitly stated his or her preference for the different resources.

These explicit ratings however, may be biased by the user's own idea of his or her interests. In the case of multimedia the user's idea of his or her interest may be biased by a number of external factors such as cultural correctness or ideas of what is cool or not cool. In a business setting this bias may be the user's misjudgment of his or her abilities. Furthermore Desrosiers and Karypis [2011] argue that even though an explicit scale is supplied to the users, they may have their own rating scale. Some users may be reluctant to using the extremes of the scale, whereas some users are not. To summarize, an explicit voting scale contains information about what the user thinks he or she is interested in whereas an implicit voting scheme contains information of what the user actually visits, regardless of his or her interests.

The data sets explored by the reviewed literature were generated by users in a leisurely setting, whereas the data set explored in this paper was generated by users in a professional work setting. This distinction may have some implications in terms of seasonal variations and outliers.

6.3 Scientific implications

The similarity measures and the clustering algorithm were implemented exactly as they were described in the literature. In other words this paper makes no attempt at improving the discovered techniques, but has explored their applicability to the data set with which this thesis is concerned. However, as the data sets used in the reviewed literature are based on explicit feedback given on fixed scales, they have no need for feature scaling. Recalling Section 5.1.2, an additional benefit of the feature scaling performed in this thesis is that the similarity computation implicitly includes information about items rated by only one of the users as the ratings in the corated vector are scaled according to the most frequent item in the individual users' web log history. Although feature scaling has little effect when the ratings are given on a fixed scale, the idea is transferable to explicit voting schemes as well. Both Frequency-weighted Pearson Correlation and Spearman Rank Correlation normalize the corated vectors before computing correlation. If one were to instead normalize each item relative to the individual user's rating vectors, then this information would be preserved and may perhaps lead to better results. Furthermore, this normalization could be done in the preprocessing phase rather than each time similarity is computed, thus making the similarity measures more scalable. This thesis presents no evidence to suggest whether or not this will improve results as it is not the purpose of this thesis, but it is discussed such that it may be explored by others.

Furthermore, this thesis has compared Frequency-weighted Pearson Correla-

tion and Spearman Rank Correlation and shown that for the comparison of 10 000 users' traffic over two years, Spearman Rank Correlation is two and a half hour slower than Frequency-weighted Pearson Correlation, confirming the assertion made by Desrosiers and Karypis [2011], who stated that Spearman Rank Correlation is computationally expensive, while at the same time showing it's cost relative to the cost of Frequency-weighted Pearson Correlation.

A comparison of the achieved results against the results presented in the literature cannot be done quantitatively as the results in the literature concern the accuracy of rating predictions after the clustering is done rather than the actual clustering. In addition to this, it is somewhat meaningless to compare the clustering of a data set with certain characteristics to the clustering of a data set with other characteristics as the results rely on the underlying clusters in the data. As stated in Section 2.4.2, results which are good for one data set may be bad for another.

The results of experiment one were not in line with what was suggested by the literature. A possible explanation may be that the users will become similar over a longer time span. This may be a consequence of the bias in implicit voting schemes, that over time users will visit many of the same articles - by accident or by choice or it may be specific to the seasonal variations in the explored data set. Nonetheless, separating the data in smaller time frames has it's cons and it's pros. The benefit of doing so is a reduction in the computational complexity of the similarity computations and the added possibility of finding two users in the same interest group in one time frame but in different interest groups the next, thus capturing the seasonal variations of the data set. The disadvantage however, is the risk that the time frames poorly match the underlying seasonal variations and that the time frame spans two different seasons. The price to pay for the reduction in computational cost, is an increased analysis cost. Separating the data in smaller time frames has the effect that the number of different cluster results to analyse increases with the number of time frames. If one were to only examine a subset of the time frames as in this thesis, there is an underlying risk that the chosen time frames are uninteresting and not representative for the rest of the data.

6.3.1 Generality of Proposed Solution and Other Viable Solutions

As the techniques used in this thesis are suggested by the literature it goes without saying that they are applicable to the data sets explored by the literature in which they were proposed. In general, the similarity metrics are applicable

to problems were users are represented by a vector of preferences, where each preference is indicated numerically. If the ratings are not given on fixed scales, preprocessing in the form of feature scaling or normalisation should be applied. Blondel's algorithm was only used by one of the papers, but in theory it will work on any precomputed similarity matrix.

Blondel's algorithm was chosen due to it's scalability. In theory any clustering algorithm that takes as input a precomputed similarity matrix may provide meaningful results for the data set in this thesis as long as it uses a suitable similarity matrix. From the *curse of dimensionality* as described in Section 2.2 it follows that traditional distance functions work poorly with the data in this thesis as the number of describing features is equal to the number of resources in the system. Therefore, some form of correlation metric should be used.

The previous two sections outlined the scenarios in which the applied methods are mathematically applicable. To decide when the methods will in fact yield good results however, is not as straight forward. This thesis has shown that the applied methods yielded good results for the data set with which this thesis was concerned. A logical implication of this is that it may also yield good results for similar data sets. The data set in this thesis is generated by professional users with different professions and different interests. The resources contained within the system span different subjects that are relevant to one or more of these professions. A similar scenario in which it is likely that the same methods yield good results may be the web logs of a reference system for legal documents. Attorneys have different specializations. Some may be interested in business law while others are interest in criminal law. This implies that is likely that clusters exist and that it is possible discover them with the methods used in this thesis, as long as the web logs contain the same information. A scenario in which the methods are less likely to produce good results may be the web logs of a web site for veteran car enthusiasts, where there is little difference in the interests of the different users.

Due to the sheer amount of users in the system it is unlikely that the business goal could have been realized without some form of data mining involved. A small scale alternative would be to interview users to qualitatively determine the clusters, but this approach is infeasible at the scale involved in this thesis. One could possibly achieve some knowledge about existing segmentations by interviewing a sample of the users, but this approach will not give information as to the interests about the users who were not interviewed.

6.3.2 Business Implications

The experiments in this thesis has shown that it is possible to cluster the users based on their interests as expressed by their frequencies of visits to the different resources. The discovery of clusters has several business implications and although a detailed analysis of these are outside the scope of the thesis, a brief discussion is included.
The discovery of clusters in the data set can be used for numerous business applications. It allows for specific marketing towards identified interests groups such that users that have been identified as interested in certain subjects may receive marketing concerning relevant seminars to these interest groups. The discovered interest groups may also be used for search optimization by factoring in the preferences of similar users when ranking search results or it may be used to suggest resources that similar users have indicated a preference for. If further analysis shows that a given user belongs to the same cluster in the same time frame for different years, the clusters can be used to predict information need for the same time frame subsequent years and thus give the user the desired information before the user has searched for it. If the clusters identified a group of users that were mostly interested in a resource which they may find for free online, mitigating steps can be taken to keep them as customers. However if the results had shown that clusters did not exist, and that all of the users were interested in the same topics, resources that were previously spent on marketing to different market segments could be reallocated elsewhere.

6.3.3 Scope, Limitations and Critique

This project arose from the business goal of *gaining insight into the natural clustering of the users of Sticos' reference system such that it may be used for product improvement*. From a business point of view this goal is perfectly valid, but from the scientists perspective it has the implication that the results should in fact be *good*. Common problem descriptions for masters theses are often limited to testing an hypothesis and then recording the results regardless of whether they are good or bad. A consequence of the broad problem definition is that a lot of work had to be done to narrow the focus of the thesis down sufficiently to begin working in the specific direction this project has taken. In retrospect, the author would have benefited from a more specific problem description.

As the problem definition resulted in a lot of initial work, the scope of the project had to be adjusted accordingly. As such, the scope of this thesis has been limited to the discovery of the user clusters and an analysis of the results to decide whether they are in fact meaningful. Not included in the scope however, is the

optimization of the clusters by using different implicit voting schemes. Instead of frequencies of visits, one could for example look at how much time the user spent on the individual resources. This would involve a heavier focus on web usage mining which was briefly discussed in Section 2.5. Another task which was not included in the scope was the application of the results for product improvement, however the author suggests the following as possible uses:

- The clusters may be used for search optimization by factoring in preferences of similar users when ranking search results.
- The clusters may be used to direct marketing of relevant seminars to costumers belonging to a cluster with a given characteristic.
- Seeing as the clusters are separated by month, they may be used to predict information need by looking at what cluster the user belonged to the same month previous years.

The difference between explicit feedback and implicit feedback proved to be more important than the author was aware of at the beginning of the project. If the literature review was to be performed again it would have included an additional keyword to specify that results should include the some form of the word "implicit". The author believes that the inclusion of such a term would lead to the discovery of articles focusing on implicit voting schemes, so that alternatives representations of the users' preference for the different resources may have been discovered. It may be worth investigating whether other implicit voting schemes may yield better results.
Furthermore the author notes that the parameter for significance weighting was chosen by investigating the data. Ideally this parameter should have been optimized, but it was chosen the way it was due to time constraints.

Chapter 7

Conclusion

The goals and main research questions for this thesis were defined in Section 1.1. This section will revisit them and evaluate the work done in this thesis against them. Recalling the introduction, this project arose from a business goal of *gaining insight into the natural clustering of the users of Sticos' reference system such that it may be used for product improvement* and the plan to achieve this goal was through the scientific goal of *applying known machine learning techniques to a new data set generated by professional users.* To realize this goal the thesis first applied the basic theory to learn about it's shortcomings, then performed a literature review to find how others have mitigated these shortcomings. The findings from the review were used to design an experiment, from which the lessons learned were used to design a second experiment. The second experiment was successful in discovering meaningful clusters, and having done that, the clusters can be used for a number of different purposes. In the discussion the following applications were suggested:

- The clusters may be used for search optimization by factoring in preferences of similar users when ranking search results.
- The clusters may be used to direct marketing of relevant seminars to customers belonging to a cluster with a given characteristic.
- Seeing as the clusters are separated by month, they may be used to predict information need by looking at what cluster the user belonged to the same month previous years.

In other words both the business goal and the scientific goal have been realized. In addition to the scientific goal and the business goal, the following research questions were defined:

Main research Question 1 How can the users of Sticos' systems be clustered based on what resources they visit?

Main research Question 2 What characterizes the different user clusters found using the techniques from **research question 1**?

The answer to main research question 1 was found during the systematic literature review in Chapter 4 by looking at how similar problems have been solved and it's applicability was validated through the experiments. The full answer is described in the experimental setup of the second experiment and the key aspects are listed below:

- Separating the data in smaller time frames to reduce the scale of the data.
- Normalizing the feature vectors to show relative preference for the different topics
- Incorporating significance weighting to reflect reduced confidence in similarities computed between users who share few common resources in their feature vectors.
- Incorporating default voting to signify that absence of a resource in the feature vector means that the user is not interested in it and to incorporate more information about both users in the similarity computation.
- Computing similarity using the Frequency-Weighted Pearson Correlation metric.
- Clustering the data using the algorithm proposed by Blondel et al. [2008]

The answer to the second research question is more complex than the answer to the first, and cannot fully be summarized the same way. However, the characterizations of each cluster is presented in Section 5.4 and the key aspects are listed below:

- The second experiment was successful in finding clusters based on the users' interests.
- For the three time frames that were analysed, a total of eleven different interest groups were discovered.
- The results showed that there are meaningful seasonal variations in the discovered interest clusters.
- Six of the discovered interest groups could be found in more than one time frame indicating that some interest groups are persistent through seasonal variations.

- In each of the analysed time frames there exists a large "noise cluster" containing the users that are dissimilar to the users in the meaningful clusters.

7.1 Contributions

As stated in the introduction this paper makes the following contributions to the body of knowledge in the field of knowledge discovery in databases:

Application of known techniques to a new data set generated by professional users: Known data mining techniques have been applied to a new data set generated by users in a professional setting, whereas the benchmark data sets are generated by users in a leasurely setting.

Comparison of two similarity measures: The thesis has compared the *Spearman Rank Correlation* and the *Frequency Weighted Pearson Correlation* metrics in terms of scalability.

A review of recent efforts: The thesis has presented a literature review summarizing recent efforts on clustering users based on preferences for different resources.

7.2 Future Work

The separation of the data in the second experiment into months introduced an interesting dimension to this project. The results presented in this thesis showed that there is a possibility of clusters that persist throughout the seasonal variations. Further work on this project could include analysing how clusters change over time, and determining whether some clusters are persistent through certain time periods. Another interesting dimension that should be explored is the size of the time frames. Lastly, the category of resources called "Fildokument" (English: file documents) has no metadata about what categories the different resources belong to. Further insight could be achieved by categorizing these resources. It may also be worth retaking the experiment with other variable transformation schemes such as normalisation, or by exchanging frequency of visits with time spent on the individual resources or other implicit voting techniques.

Appendices

Appendix A

Literature Review

Table A.1: Discarded articles from inclusion criteria

Author	**Title**	**Reasoning**
Agosti et al. [2012]	Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction	Violates IC1, IC2 and IC5
Algiriyage et al. [2015]	Web User Profiling using Hierarchical Clustering with Improved Similarity Measure	Violates IC1, IC2 and IC3
Amatriain [2013]	Mining Large Streams of User Data for Personalized Recommendations	Violates IC3 and Ic5
Ammari et al. [2012]	Deriving group profiles from social media to facilitate the design of simulated environments for learning	Violates IC1, IC2 and IC3
Cheng et al. [2016]	FeatureMiner: A Tool for Interactive Feature Selection	Violates IC1, IC2, IC3 and IC5
Cho et al. [2012]	Implementation of personalized recommendation system using k-means clustering of item category based on RFM	Violates IC2 and IC4
Forsati et al. [2015]	An effective Web page recommender using binary data clustering	Violates IC1 and IC2
Fuxman et al. [2012]	Enabling Direct Interest-Aware Audience Selection Ariel	Violates IC1 and IC2
Gutierrez and Poblete [2015]	Sentiment-based User Profiles in Microblogging Platforms	Violates IC1 and IC2
Jain and Jain [2014]	Categorizing Twitter Users on the basis of their interests using Hadoop/Mahout Platform	Violates IC2
Kumar [2013]	Mining user interest from webs history	Violates IC1, IC2, IC3 and IC5
Martin [2014]	Behavioral Segmentation of Pinterest Users	It is a research proposal
Quintarelli et al. [2016]	Recommending New Items to Ephemeral Groups Using Contextual User Influence	Violates IC1, IC2 and IC3
Sim et al. [2013]	A survey on enhanced subspace clustering	Violates IC2 and IC3
Xie and Wang [2016]	Web page recommendation via twofold clustering: considering user behavior and topic relation	Violates IC1, IC2 and IC3
Yu and Korkmaz [2014]	Finding the Most Evident Co-Clusters on Web Log Dataset Using Frequent Super-Sequence Mining	Violates IC1, IC2 and IC3
Zhang et al. [2015]	Application of Clustering Algorithm on TV Programmes Preference Grouping of Subscribers	Violates IC3

Table A.2: Discarded articles from quality screening

Author	**Title**	**Score**
George and Merugu [2005]	A scalable collaborative filtering framework based on co-clustering	6.5
Gao et al. [2013]	A cross cluster-based collaborative filtering method for recommendation	7
Zhang and Wang [2013]	The Application of Web Log in Collaborative Filtering Recommendation Algorithm	2.5

Table A.3: Included articles from quality screening

Author	Title	Score
Adomavicius and Kwon [2012]	Improving aggregate recommendation diversity using ranking-based techniques	9
Adomavicius and Tuzhilin [2005]	Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions	2297 citations
Agarwal et al. [2005]	Research paper recommender systems: A subspace clustering approach	10
Aggarwal and Yu [2002]	Redefining clustering for high-dimensional applications	10
Desrosiers and Karypis [2011]	A Comprehensive Survey of Neighborhood-based Recommendation Methods	46 citations
Boratto and Carta [2014]	Modeling the Preferences of a Group of Users Detected by Clustering: A Group Recommendation Case-Study	9.5
Boratto and Carta [2015]	The rating prediction task in a group recommender system that automatically detects groups: architectures, algorithms, and performance evaluation	10
Boratto et al. [2009]	Group recommendation with automatic identification of users communities	8.5
Costa et al. [2016]	Group-based Collaborative Filtering Supported by Multiple Users' Feedback to Improve Personalized Ranking	10
Deshpande and Karypis [2004]	Item-based top- N recommendation algorithms	10
Gao et al. [2007]	Personalized Service System Based on Hybrid Filtering for Digital Library	8
Yanxiang et al. [2013]	User-based clustering with top-N recommendation on Cold-Start problem	8
Zeng et al. [2012]	Context-aware social media recommendation based on potential group	8
Li and Kim [2003]	Clustering approach for hybrid recommender system	8
Li and Murata [2012]	Using Multidimensional Clustering Based Collaborative Filtering Approach Improving Recommendation Diversity	10
Müller et al. [2009]	Evaluating Clustering in Subspace Projections of High Dimensional Data	10
Sarwar et al. [2001]	Item-based collaborative filtering recommendation algorithms	10

Appendix B

Experiment One - Results

B.1 Frequency-Weighted Pearson Correlation

This section will discuss the clusters identified by Blondel's algorithm analysing data from April 2015 to March 2017 using the FWPC similarity metric. Recalling section 2.4.1, it is necessary to analyse clusters against ground truth data to ensure that the clusters are indeed meaningful. A total of four clusters were identified, and the users were distributed amongst them as follows:

- Cluster 1: 6647 users
- Cluster 2: 1479 users
- Cluster 3: 1682 users
- Cluster 4: 189 users
- Noise: 3 users

This distribution may not provide very valuable information on it's own, but discussion with Sticos' management suggests that it is unlikely that over approximately 66.5 per cent of Sticos' users belong to the same interest group. Furthermore, if the purpose of this project is to identify niché groups that Sticos can focus their marketing towards, this segmentation has little business value. To further investigate the characteristics of the different clusters, the web logs of the different cluster members were analysed. For each cluster two graphs were made, one looking at the different users' preference for any individual resource and one looking at the different users' preference for the different categories of resources. For each user in a cluster, their Scaled preference for each individual resource

was aggregated, such that each resource was represented by the aggregated value of the cluster members' preference for that particular resource. Recalling the feature scaling scheme described in section 5.1.2 each user may at most have a preference of 1.0 for a given resource, indicating how often a user visits a particular resource relative to the resource to which the user has the most visits.
Aggregating the preference values required the use of the database mapping resources to categories described in figure 3.2. It was done the same way as for individual topics, but for each individual topic, a look-up was made in the database to find the corresponding category such that values were aggregated by category level.

B.1.1 Cluster One

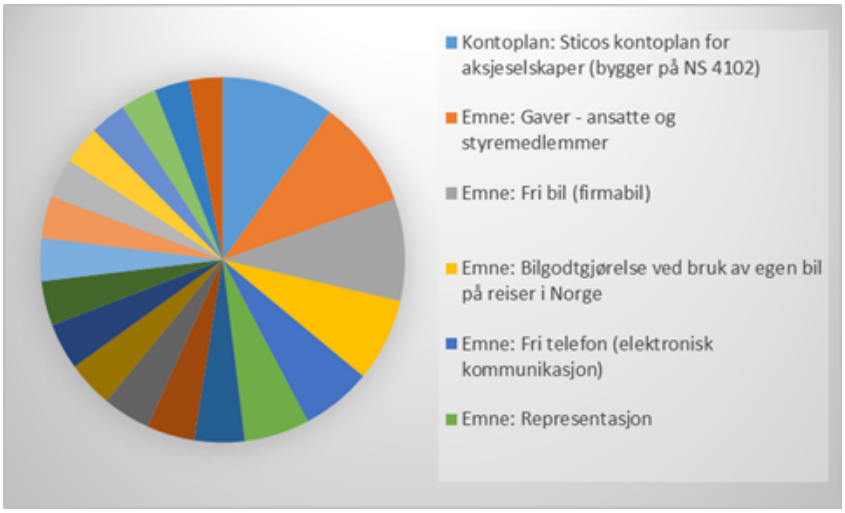
Figure B.1 shows the aggregated preference values for the users in the first cluster. One may upon inspection of figure B.1a either conclude that all of the users have roughly the same preference for the 20 most popular topics, or the more likely explanation, that there is a lot of variance in what topics the individual users prefer. Furthermore, examining the title of the topics in figure B.1a it seems that all of the resources are resources that are relevant for all users, and as such serve little purpose in discriminating users as most users have an interest in them.

B.1.2 Clusters Two and Three

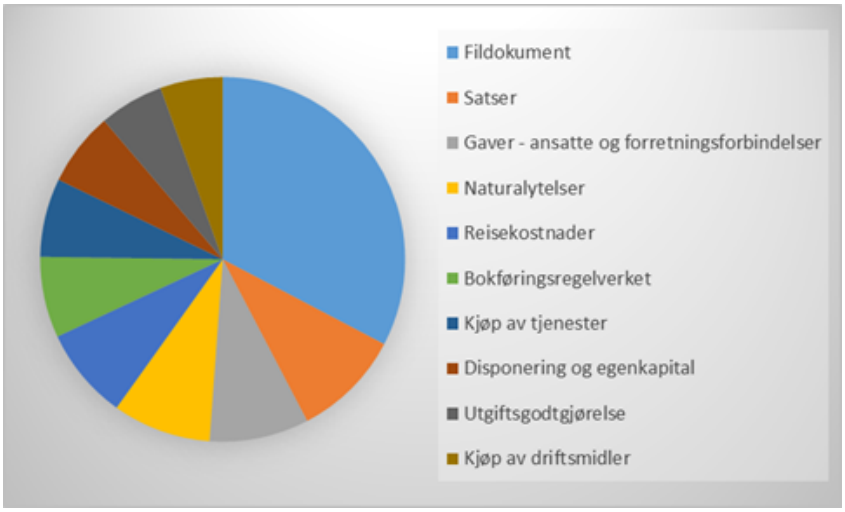
Figure B.2 and B.3 shows the metadata for cluster 2 and three respectively. One apparent characteristic is that the resource "Kontoplan: Sticos kontoplan for aksjeselskaper(bygger på NS 41029)" is more popular in these two clusters than it is in cluster one. Furthermore cluster three seems to have a higher preference for the category "fildokument" than cluster two. Having said that, it seems that the two clusters share the rest of the preferences listed, and in the same order. Therefore, little can be said about what separates the users within these clusters.

B.1.3 Cluster Four

In contrast to the other three clusters, cluster number four's top list of preferred individual resources (depicted in figure B.4) share only one resource with the other clusters. With some confidence it can be said the members of this cluster is different from the members of the other clusters. However, as with cluster number one it seems that the preferences for individual resources is fairly evenly distributed and without more data it is not possible to determine whether it is because there is a lot of variation within the cluster members' preferences or if all the members prefer many of the resource equally much.

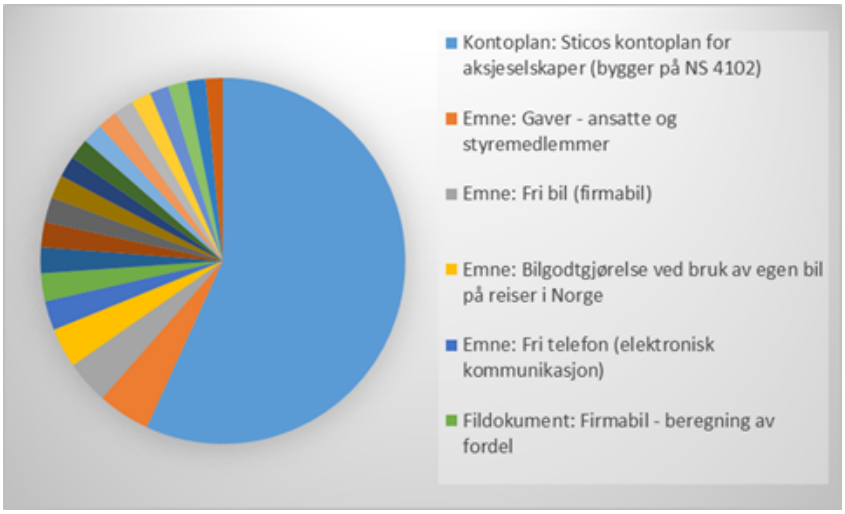


(a) Scaled preference for single resource

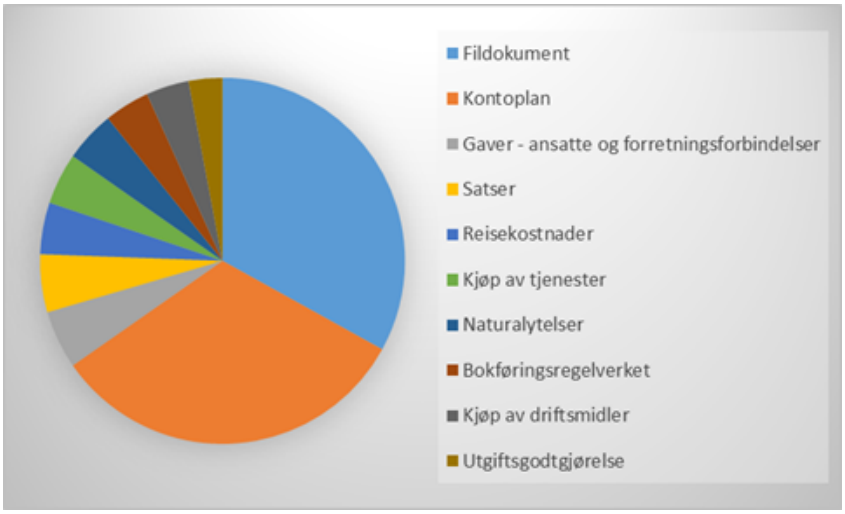


(b) Scaled preference for category

Figure B.1: FWPC Cluster 1: 6647 users

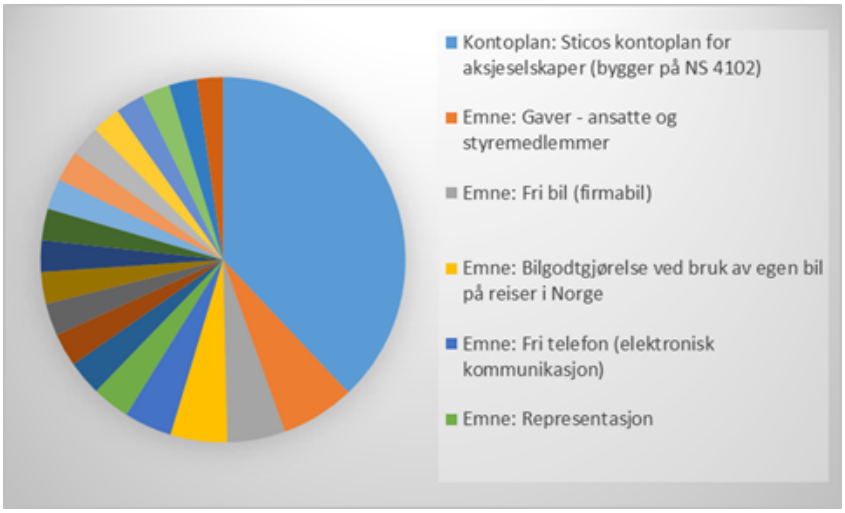


(a) Scaled preference for single resource

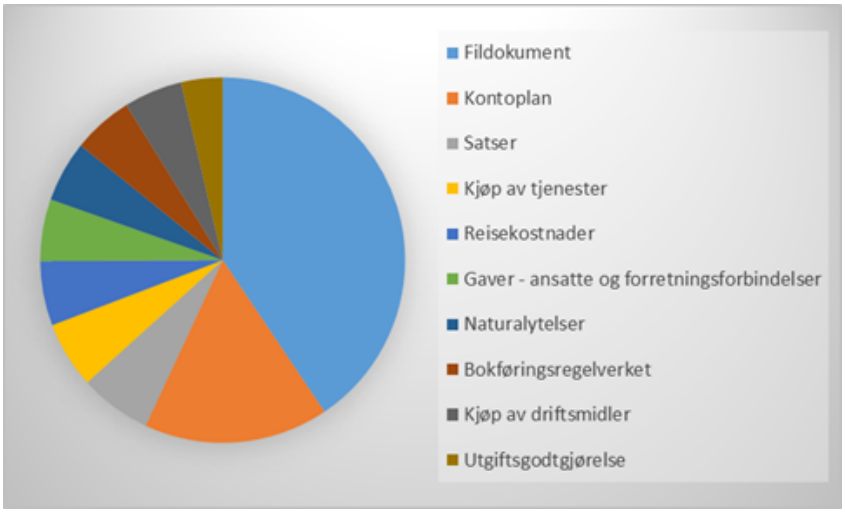


(b) Scaled preference for category

Figure B.2: FWPC Cluster 2: 1479 users

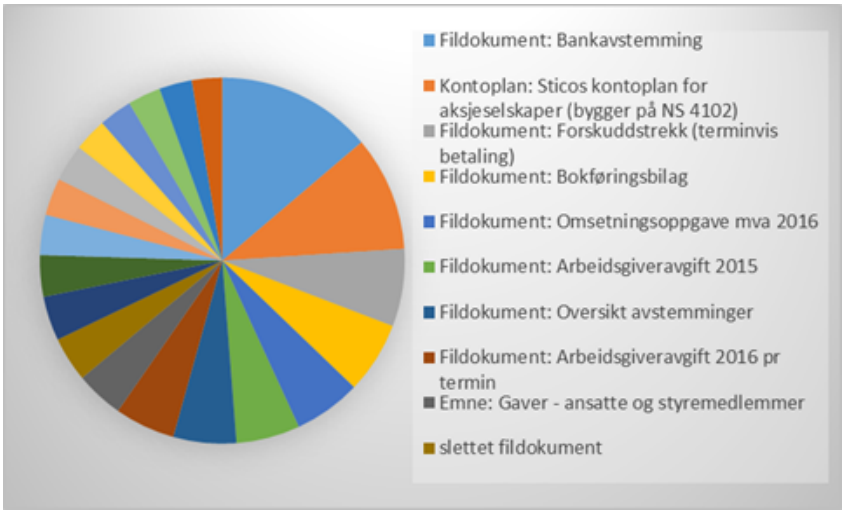


(a) Scaled preference for single resource

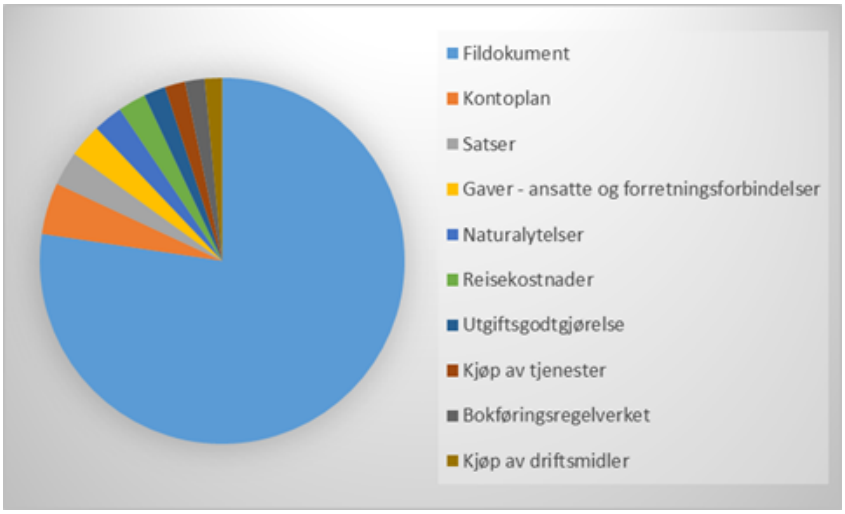


(b) Scaled preference for category

Figure B.3: FWPC Cluster 3: 1682 users



(a) Scaled preference for single resource



(b) Scaled preference for category

Figure B.4: FWPC Cluster 4: 189 users

B.2 Spearman Rank Correlation

The clusters based on the SRC metric were even coarser than the ones based on FWPC. In total three clusters were discovered, with the members distributed as follows:

- Cluster 1: 4787 users
- Cluster 2: 5122 users
- Cluster 3: 90 users
- Noise: 1

As with FWPC it seems unlikely that the majority of the users should only belong to two segments, and for marketing purposes it is of little value. The diagrams below are created the same way as they were created for the FWPC metric, and the rest of this section features a discussion of the discovered clusters.

B.2.1 Cluster One

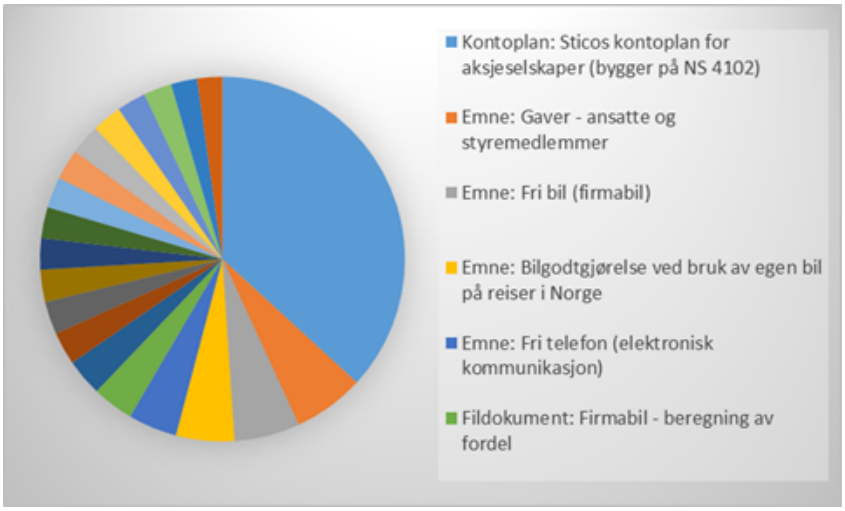
The preference diagram for cluster one depicted in figure B.5a shows that the 4787 users contained within it have a high preference for the same resource as the one that is most popular for clusters two and three based on the FWPC measure. A possible explanation for this may be that the users contained within SRC cluster one can be found in FWPC clusters two and three, and that the SRC metric was not able to separate these two clusters the way FWPC did.

B.2.2 Cluster Two

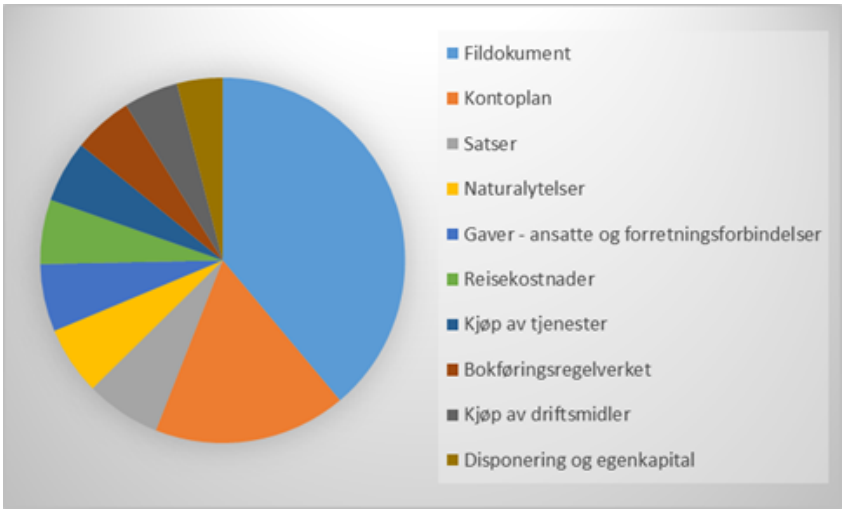
In the same way that SRC cluster one shows similar preferences to FWPC clusters two and three, SRC cluster two shares many characteristics with FWPC cluster one. The preferences are fairly evenly distributed amongst the different resources, and like FWPC clusters one and four, it could be that there is a lot of variance between the preference of the individual users or it could be that the majority of the users have an equal preference for many resources.

B.2.3 Cluster Three

Containing 90 users, cluster three is only a small fraction of the size of clusters one and two. Looking at the top list of preferences for both individual resources and categories, it is not clear what separates the members of cluster three from the members of the other clusters.

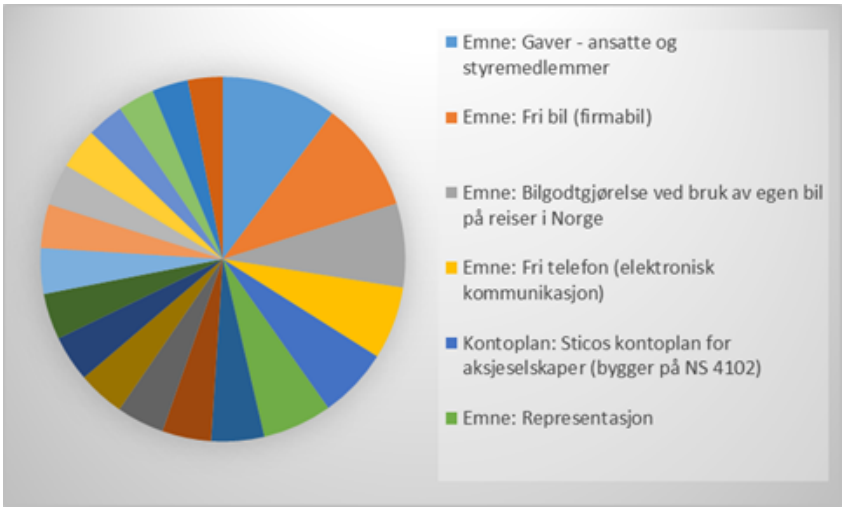


(a) Scaled preference for single resource

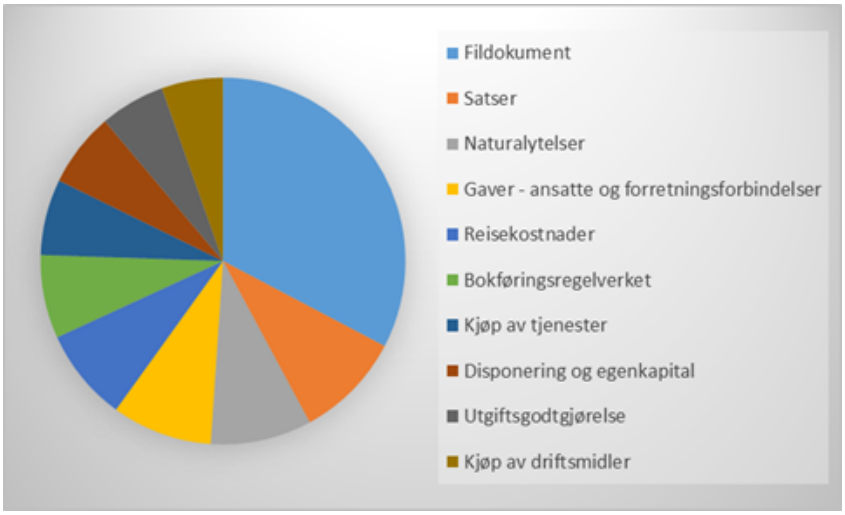


(b) Scaled preference for category

Figure B.5: SRC Cluster 1: 4787 users

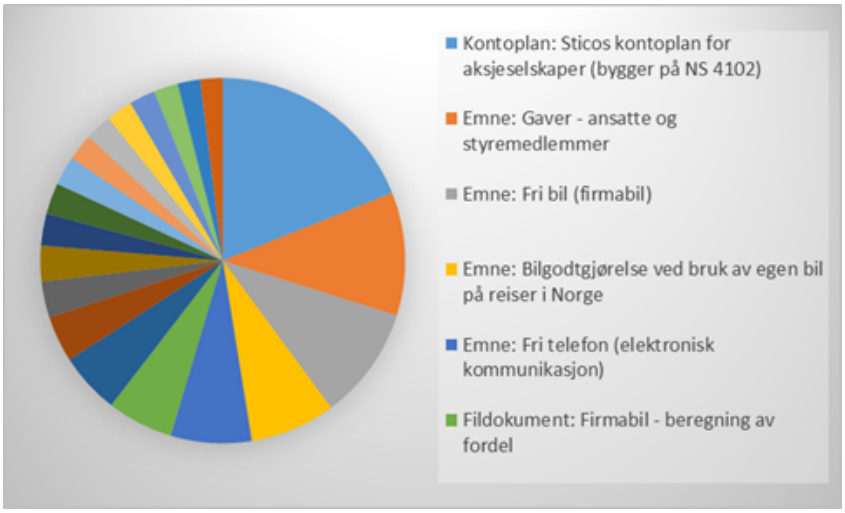


(a) Scaled preference for single resource

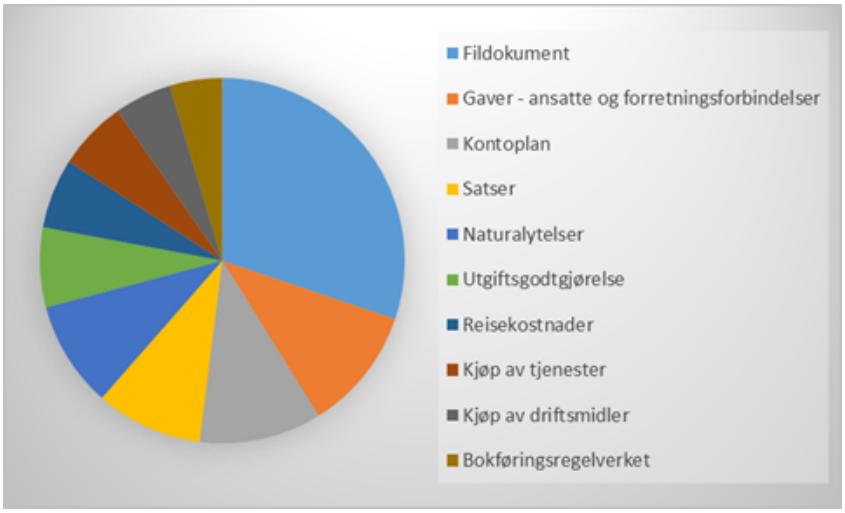


(b) Scaled preference for category

Figure B.6: SRC Cluster 2: 5122 users



(a) Scaled preference for single resource



(b) Scaled preference for category

Figure B.7: SRC Cluster 3: 90 users

B.2.4 Concluding Remarks

Neither the clusters generated by the FWPC metric nor the clusters generated by the SRC metric seem to be very distinct from one another, and as such, one may say that the discovered clusters are not very meaningful. Having discussed these results with representatives from Sticos, a possible explanation may be that the data is taken from a too large time interval. It may simply be that over time, the users become to similar to each other.
Although more insight may be gained by analysing the ground truth data from other perspectives and by adding more metadata, the presented analysis is sufficient to determine that there is a low probability that the discovered clusters are meaningful.
If the data indeed was taken from a too large time interval, it may be more meaningful to divide the data in smaller time frames and analyse them individually. Furthermore, it is highly likely that there is seasonal variance in the users interests as there are a number of yearly deadlines to which any company has to adhere to. In light of these facts, it seems like a better use of resources to conclude the first experiment and conduct a new experiment where the data is analysed over smaller time frames and the knowledge gained in this experiment is taken into consideration.

Appendix C

Experiment Two - Results

C.1 January 2016

Cluster number 1:				
Number of users in this cluster: 2175				
Category			**Individual resource**	
Satser	0,405		Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,34
Fildokument	0,288		Emne: Reiseregninger og reiseoppgjør	0,14
Reisekostnader	0,154		Fildokument: Reiseregning - kun bilgodtgjørelse	0,09
Utgiftsgodtgjørelse	0,11		Fildokument: Reiseregning inn- og utland	0,07
Naturalytelser	0,098		Fildokument: Reiseregning - kun innenlandsreiser	0,07
Kontoplan	0,072		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,07
Gaver - ansatte og forretningsforbindelser	0,07		Emne: Fri telefon (elektronisk kommunikasjon)	0,06
Lønnsutbetalinger	0,059		Emne: Diettgodtgjørelse på reiser med overnatting i Norge	0,05
Velferdstiltak	0,054		Emne: Diettgodtgjørelse på reiser uten overnatting i Norge	0,04
Kjøp av tjenester	0,051		Emne: Fri bil (firmabil)	0,04

Figure C.1: Cluster 1: January 2016

Cluster number 2:				
Number of users in this cluster: 1732				
Category			**Individual resource**	
Naturalytelser	0,585		Emne: Fri bil (firmabil)	0,61
Fildokument	0,393		Fildokument: Firmabil - beregning av fordel	0,39
Satser	0,14		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,09
Utgiftsgodtgjørelse	0,11		Emne: Fri telefon (elektronisk kommunikasjon)	0,08
Kontoplan	0,094		Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,07
Reisekostnader	0,081		Emne: Gaver - ansatte og styremedlemmer	0,05
Gaver - ansatte og forretningsforbindelser	0,08		Fildokument: Reiseregning - kun bilgodtgjørelse	0,04
Oppgaveplikt	0,05		Emne: Krav til føring og oppbevaring av kjørebok	0,04
Velferdstiltak	0,045		Emne: Skillet mellom arbeidsreise og yrkesreise	0,04
Bokføringsregelverket	0,043		Fildokument: Firmabil - bilgodtgjørelse - lønnskompensasjon	0,03

Figure C.2: Cluster 2: January 2016

Cluster number 3:				
Number of users in this cluster: 2365				
Category			**Individual resource**	
Gaver - ansatte og forretningsforbindelser	0,479		Emne: Gaver - ansatte og styremedlemmer	0,41
Fildokument	0,157		Emne: Gaver - forretningsforbindelser	0,14
Velferdstiltak	0,147		Emne: Utgifter til julegaver eller annen oppmerksomhet til ansatte	0,11
Representasjon	0,111		Emne: Utgifter til julebord o.l. arrangement i Norge og utlandet for ansatte	0,1
Satser	0,11		Emne: Representasjon	0,09
Kjøp av tjenester	0,1		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,08
Naturalytelser	0,093		Emne: Fri telefon (elektronisk kommunikasjon)	0,05
Utgiftsgodtgjørelse	0,088		Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,05
Reisekostnader	0,086		Emne: Gaver - frivillige organisasjoner	0,04
Kontoplan	0,085		Emne: Kontingenter - skattemessig fradrag	0,04

Figure C.3: Cluster 3: January 2016

Cluster number 4:				
Number of users in this cluster: 3834				
Category			**Individual resource**	
Fildokument	0,239		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,05
Oppgaveplikt	0,095		Emne: Ekstraordinært utbytte og tilleggsutbytte	0,05
Disponering og egenkapital	0,093		Emne: Tap på kundefordringer	0,04
Bokføringsregelverket	0,08		Emne: Krav til salgsdokumenter (faktura)	0,04
Satser	0,075		Emne: Lån til personlig aksjonær - skattemessig utbytte/utdeling	0,04
Naturalytelser	0,069		Emne: Avskrivninger - skattemessig (saldoreglene)	0,03
Kjøp av driftsmidler	0,058		Fildokument: Lån fra personlig skattyter - beregning av maksimal rente skjerm	0,03
Avskrivning	0,055		Emne: Sammenstillingsoppgave / Lønns- og trekkoppgaver	0,03
Kontoplan	0,054		Emne: Fri telefon (elektronisk kommunikasjon)	0,03
Transaksjoner med nærstående	0,054		Emne: Konsernbidrag - Regnskapsmessig behandling	0,03

Figure C.4: Cluster 4: January 2016

Cluster number 5:				
Number of users in this cluster: 2297				
Category			**Individual resource**	
Kontoplan	0,896		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,93
Fildokument	0,193		Emne: Gaver - ansatte og styremedlemmer	0,05
Satser	0,089		Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,05
Gaver - ansatte og forretningsforbindelser	0,083		Emne: Fri telefon (elektronisk kommunikasjon)	0,04
Naturalytelser	0,071		Emne: Fri bil (firmabil)	0,04
Utgiftsgodtgjørelse	0,065		Fildokument: Firmabil - beregning av fordel	0,04
Kjøp av tjenester	0,055		Emne: Gaver - forretningsforbindelser	0,03
Reisekostnader	0,049		Fildokument: Reiseregning - kun bilgodtgjørelse	0,03
Velferdstiltak	0,048		Emne: Utgifter til julebord o.l. arrangement i Norge og utlandet for ansatte	0,03
Kjøp av driftsmidler	0,041		Emne: Reiseregninger og reiseoppgjør	0,02

Figure C.5: Cluster 5: January 2016

Cluster number 6:				
Number of users in this cluster: 1497				
Category			**Individual resource**	
Utgiftsgodtgjørelse	0,428		Emne: Fri telefon (elektronisk kommunikasjon)	0,44
Fildokument	0,213		Emne: Obligatorisk tjenestepensjon - OTP	0,08
Satser	0,147		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,08
Naturalytelser	0,128		Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,06
Pensjon og pensjonsforsikring	0,127		Emne: Fri bil (firmabil)	0,05
Kjøp av tjenester	0,088		Emne: Gaver - ansatte og styremedlemmer	0,04
Kontoplan	0,082		Fildokument: Firmabil - beregning av fordel	0,04
Gaver - ansatte og forretningsforbindelser	0,076		Emne: Skattesatser, trygdeavgift og avgiftssatser for 2017 og historiske sats	0,03
Velferdstiltak	0,066		Emne: Pensjonsordninger	0,03
Formue	0,061		Emne: Pensjon - innskuddspensjon	0,03

Figure C.6: Cluster 6: January 2016

Cluster number 7:				
Number of users in this cluster: 565				
Category			**Individual resource**	
Fildokument	0,591		Fildokument: Arbeidsgiveravgift 2015	0,28
Periodiseringer og avsetninger	0,181		Emne: Lønn - bokføring og avstemming	0,17
Kontoplan	0,08		Fildokument: Forskuddstrekk (terminvis betaling)	0,17
Satser	0,05		Fildokument: Bankavstemming	0,15
Gaver - ansatte og forretningsforbindelser	0,049		Fildokument: Lønns- og pensjonskostnader	0,13
Utgiftsgodtgjørelse	0,04		Fildokument: Bokføringsbilag	0,09
Kjøp av tjenester	0,038		Fildokument: Omsetningsoppgave mva 2015	0,09
Naturalytelser	0,036		Fildokument: Påløpne feriepenger	0,09
Oppgaveplikt	0,027		Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,08
Bokføringsregelverket	0,027		Fildokument: Forskuddstrekk (månedlig betaling)	0,07

Figure C.7: Cluster 7: January 2016

C.2 January 2017

Cluster number 1:			
Number of users in this cluster: 3138			
Category		**Individual resource**	
Satser	0,33615224	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,29773942
Fildokument	0,29322252	Emne: Fri telefon (elektronisk kommunikasjon)	0,17397188
Utgiftsgodtgjørelse	0,20749841	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,07440601
Naturalytelser	0,10658711	Fildokument: Reiseregning - kun bilgodtgjørelse	0,07179941
Oppgaveplikt	0,08831477	Fildokument: Reiseregning - kun innenlandsreiser	0,06275865
Kontoplan	0,07893077	Fildokument: Reiseregning inn- og utland	0,05791764
Reisekostnader	0,07637771	Emne: Diettgodtgjørelse på reiser med overnatting i Norge	0,05241455
Gaver - ansatte og forretningsforbindelser	0,07376058	Emne: Fri bil (firmabil)	0,04104291
Kjøp av tjenester	0,06853777	Emne: Lønnsopplysningsplikt og arbeidsgiveravgift mv. for lag og foreninger	0,0374697
Velferdstiltak	0,05992064	Emne: Gaver - ansatte og styremedlemmer	0,03707224

Figure C.8: Cluster 1: January 2017

Cluster number 2:			
Number of users in this cluster: 6356			
Category		**Individual resource**	
Fildokument	0,19166003	Emne: Lån til personlig aksjonær - skattemessig utbytte/utdeling	0,07170547
Disponering og egenkapital	0,09772658	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,04909971
Transaksjoner med nærstående	0,08460481	Emne: Avskrivninger - skattemessig (saldoreglene)	0,04447447
Satser	0,08185244	Emne: Ekstraordinært utbytte og tilleggsutbytte	0,04263533
Bokføringsregelverket	0,07578607	Emne: Tap på kundefordringer	0,03486117
Avskrivning	0,06983148	Emne: Krav til salgsdokumenter (faktura)	0,02985825
Kjøp av driftsmidler	0,06671456	Emne: Konsernbidrag - Regnskapsmessig behandling	0,028779
Formue	0,06568803	Emne: Balanseføring av driftsmidler	0,02826006
Oppgaveplikt	0,06188655	Emne: Fritaksmetoden	0,02801573
Periodiseringer og avsetninger	0,05945746	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,02682713

Figure C.9: Cluster 2: January 2017

Cluster number 3:			
Number of users in this cluster: 2429			
Category		**Individual resource**	
Kontoplan	0,89444797	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,94032424
Fildokument	0,18568509	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,04977334
Satser	0,08913488	Emne: Gaver - ansatte og styremedlemmer	0,04319936
Gaver - ansatte og forretningsforbindelser	0,08443933	Emne: Fri telefon (elektronisk kommunikasjon)	0,0365453
Naturalytelser	0,06643152	Emne: Gaver - forretningsforbindelser	0,03624684
Kjøp av driftsmidler	0,06640171	Fildokument: Firmabil - beregning av fordel	0,03280039
Kjøp av tjenester	0,05714581	Emne: Fri bil (firmabil)	0,03073344
Utgiftsgodtgjørelse	0,05302362	Emne: Utgifter til julegaver eller annen oppmerksomhet til ansatte	0,02553218
Periodiseringer og avsetninger	0,04887726	Emne: Innførsel av varer - oversikt	0,02495127
Reisekostnader	0,04540231	Fildokument: Reiseregning - kun bilgodtgjørelse	0,02414893

Figure C.10: Cluster 3: January 2017

Cluster number 4:			
Number of users in this cluster: 1447			
Category		**Individual resource**	
Naturalytelser	0,5927582	Emne: Fri bil (firmabil)	0,64224139
Fildokument	0,37826179	Fildokument: Firmabil - beregning av fordel	0,37669586
Satser	0,17189664	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,09751633
Kontoplan	0,09666995	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,0930439
Reisekostnader	0,08370052	Emne: Fri telefon (elektronisk kommunikasjon)	0,05173347
Utgiftsgodtgjørelse	0,08168736	Emne: Skillet mellom arbeidsreise og yrkesreise	0,04523644
Gaver - ansatte og forretningsforbindelser	0,07883691	Emne: Gaver - ansatte og styremedlemmer	0,0410887
Kjøp av tjenester	0,05784967	Emne: Lån til personlig aksjonær - skattemessig utbytte/utdeling	0,03815049
Oppgaveplikt	0,05469072	Emne: Gaver - forretningsforbindelser	0,03797335
Kjøp av driftsmidler	0,04741669	Emne: Krav til føring og oppbevaring av kjørebok	0,03064382

Figure C.11: Cluster 4: January 2017

Cluster number 5:			
Number of users in this cluster: 1006			
Category		**Individual resource**	
Kjøp av driftsmidler	0,32756018	Emne: Innførsel av varer - oversikt	0,32200956
Utland - kjøp og innførsel	0,20582919	Emne: Kjøp av fjernleverbare tjenester fra utlandet	0,19408209
Fildokument	0,15254303	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,09627072
Kontoplan	0,09737991	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,03358738
Kjøp av tjenester	0,0937967	Emne: Gaver - ansatte og styremedlemmer	0,02964203
Satser	0,07088403	Emne: Fri bil (firmabil)	0,02666754
Oppgaveplikt	0,06675717	Emne: Mottatt forsikringsoppgjør	0,02530705
Gaver - ansatte og forretningsforbindelser	0,06344807	Emne: Fri telefon (elektronisk kommunikasjon)	0,02508654
Utland - omsetning og eksport	0,06019626	Emne: Salg av varer til utlandet	0,02443787
Naturalytelser	0,0584493	Emne: Gaver - forretningsforbindelser	0,02368376

Figure C.12: Cluster 5: January 2017

Cluster number 6:			
Number of users in this cluster: 2533			
Category		**Individual resource**	
Gaver - ansatte og forretningsforbindelser	0,43597195	Emne: Gaver - ansatte og styremedlemmer	0,33938722
Fildokument	0,13972423	Emne: Gaver - forretningsforbindelser	0,13719174
Representasjon	0,13820562	Emne: Utgifter til julegaver eller annen oppmerksomhet til ansatte	0,13369039
Velferdstiltak	0,13702597	Emne: Representasjon	0,11924235
Kjøp av tjenester	0,11094262	Emne: Utgifter til julebord o.l. arrangement i Norge og utlandet for ansatte	0,09186185
Satser	0,10903738	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,07665391
Reisekostnader	0,09906406	Emne: Kontingenter - skattemessig fradrag	0,06843729
Naturalytelser	0,09324433	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,04993343
Utgiftsgodtgjørelse	0,08457811	Emne: Fri telefon (elektronisk kommunikasjon)	0,04190948
Kontingenter	0,0845661	Emne: Kjøp av databriller for ansatte	0,03863432

Figure C.13: Cluster 6: January 2017

Cluster number 7:			
Number of users in this cluster: 376			
Category		**Individual resource**	
Fildokument	0,67795925	Fildokument: Oversikt avstemminger	0,17244057
Kontoplan	0,06070201	Fildokument: Arbeidsgiveravgift 2016 pr termin	0,17027977
Satser	0,04645363	Fildokument: Omsetningsoppgave mva 2016	0,15778732
Periodiseringer og avsetninger	0,03683426	Fildokument: Bankavstemming	0,15702091
Oppgaveplikt	0,03067721	Fildokument: Forskuddstrekk (terminvis betaling)	0,14661143
Bokføringsregelverket	0,0284651	Fildokument: Bokføringsbilag	0,12442376
Naturalytelser	0,02774422	Fildokument: Påløpne feriepenger	0,09252511
Kjøp av tjenester	0,02760858	Fildokument: Lønns- og pensjonskostnader	0,08328295
Gaver - ansatte og forretningsforbindelser	0,02759802	Fildokument: Omsetningsoppgave mva 2016 - årsoppgave	0,0743949
Velferdstiltak	0,02557766	Fildokument: Arbeidsgiveravgift 2016 pr måned	0,06875296

Figure C.14: Cluster 7: January 2017

Cluster number 8:			
Number of users in this cluster: 3			
Category		**Individual resource**	
Salg av nytt kjøretøy	1	Emne: Salg av ny elbil	1
Salg av tjenester	0,11111111	Emne: Salg av tjenester som ledd i offentlig myndighetsutøvelse mv.	0,11111111
Gaver - ansatte og forretningsforbindelser	0,11111111	Emne: Gaver - ansatte og styremedlemmer	0,11111111
Utland - omsetning og eksport	0,11111111	Emne: Salg av fjernleverbare tjenester til utlandet	0,11111111
Velferdstiltak	0,11111111	Emne: Utgifter til bedriftshelsetjeneste og forebyggende helsetiltak	0,11111111
Utland - kjøp og innførsel	0,11111111	Emne: Kjøp av fjernleverbare tjenester fra utlandet	0,11111111
Kjøp av nytt kjøretøy	0,07407407	Emne: Kjøp av EDB-program, lisens, support fra utlandet	0,11111111
Kontoplan	0,03703704	Emne: Kjøp av ny elbil	0,07407407
Fildokument	0,03703704	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,03703704

Figure C.15: Cluster 8: January 2017

C.3 Juni 16

Cluster number 1:			
Number of users in this cluster: 6321			
Fildokument	0,199	Emne: Fritaksmetoden	0,0718
Disponering og egenkapital	0,12	Emne: Konsernbidrag - Regnskapsmessig behandling	0,0592
Årsoppgjør	0,086	Emne: Lån til personlig aksjonær - skattemessig utbytte/utdeling	0,0554
Utbytte og konsernbidrag	0,084	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,0478
Satser	0,068	Emne: Avskrivninger - skattemessig (saldoreglene)	0,0457
Bokføringsregelverket	0,067	Emne: Konsernbidrag	0,0369
Transaksjoner med nærstående	0,065	Emne: Ekstraordinært utbytte og tilleggsutbytte	0,0352
Avskrivning	0,063	Emne: Salg av driftsmidler	0,0327
Formue	0,062	Emne: Tap på kundefordringer	0,0325
Tap	0,056	Emne: Obligatorisk tjenestepensjon - OTP	0,0282

Figure C.16: Cluster 1: June 2016

Cluster number 2:			
Number of users in this cluster: 2047			
Category		**Individual resouce**	
Kontoplan	0,907	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,9563
Fildokument	0,163	Kontoplan: Sticos kontoplan for personlig næringsdrivende (årsregnskapspl	0,0248
Satser	0,066	Kontoplan: Sticos kontoplan for personlig næringsdrivende (kun bokførings	0,0213
Kjøp av tjenester	0,05	Emne: Omkostninger ved stiftelse og kapitalforhøyelse i aksjeselskap	0,0212
Naturalytelser	0,046	Emne: Fritaksmetoden	0,0205
Reisekostnader	0,046	Emne: Lån til personlig aksjonær - skattemessig utbytte/utdeling	0,0203
Feriepenger	0,043	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,0203
Bokføringsregelverket	0,038	Emne: Feriepengeavregning til ansatte med månedslønn og avtalefestet fe	0,0192
Gaver - ansatte og forretningsforbindelser	0,038	Emne: Avskrivninger - skattemessig (saldoreglene)	0,0191
Kjøp av driftsmidler	0,038	Emne: Gaver - ansatte og styremedlemmer	0,018

Figure C.17: Cluster 2: June 2016

Cluster number 3:			
Number of users in this cluster: 1682			
Category		**Individual resouce**	
Feriepenger	0,393	Emne: Feriepengeavregning til ansatte med månedslønn og avtalefestet fe	0,2486
Fildokument	0,184	Emne: Feriepengeavregning til ansatte med månedslønn og ferie etter feri	0,1692
Ferie, rettigheter og avvikling	0,148	Emne: Beregning av feriepenger for ansatte over 60 år	0,099
Satser	0,126	Emne: Ferie uten opptjente feriepenger	0,0876
Lønnsutbetalinger	0,102	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,07
Kontoplan	0,071	Emne: Satser for beregning av feriepenger	0,0632
Trekk	0,061	Emne: Tidspunkt for utbetaling av feriepenger	0,0617
Sykepenger	0,051	Emne: Ferie	0,0509
Kjøp av tjenester	0,044	Emne: Feriepengegrunnlag	0,0375
Naturalytelser	0,043	Emne: Feriepenger ved opphør av arbeidsforhold	0,0372

Figure C.18: Cluster 3: June 2016

Cluster number 4:			
Number of users in this cluster: 3017			
Category		**Individual resouce**	
Fildokument	0,241	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,076
Satser	0,175	Emne: Representasjon	0,0625
Reisekostnader	0,15	Emne: Fri telefon (elektronisk kommunikasjon)	0,0554
Utgiftsgodtgjørelse	0,105	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,0515
Kjøp av tjenester	0,103	Fildokument: Reiseregning - kun bilgodtgjørelse	0,0488
Representasjon	0,087	Emne: Reiseregninger og reiseoppgjør	0,0448
Velferdstiltak	0,082	Emne: Kjøp av fjernleverbare tjenester fra utlandet	0,044
Naturalytelser	0,064	Emne: Diettgodtgjørelse på reiser med overnatting i Norge	0,043
Kontoplan	0,064	Fildokument: Reiseregning inn- og utland	0,0396
Bokføringsregelverket	0,058	Fildokument: Reiseregning - kun innenlandsreiser	0,0383

Figure C.19: Cluster 4: June 2016

Cluster number 5:			
Number of users in this cluster: 889			
Category		**Individual resouce**	
Gaver - ansatte og forretningsforbindelser	0,609	Emne: Gaver - ansatte og styremedlemmer	0,5309
Fildokument	0,116	Emne: Utgifter til julegaver eller annen oppmerksomhet til ansatte	0,0818
Satser	0,093	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,0768
Naturalytelser	0,081	Emne: Gaver - forretningsforbindelser	0,0759
Kontoplan	0,078	Emne: Utgifter til små oppmerksomheter til ansatte	0,0511
Kjøp av tjenester	0,065	Emne: Arbeidsgivers dekning av bøter, gebyrer mv. ilagt en ansatt	0,0335
Kjøp av driftsmidler	0,061	Emne: Kontingent til arbeidsgiver- og næringsorganisasjon	0,031
Kontingenter	0,06	Emne: Leie av kaffeautomat og vanndispenser mv. for ansatte og kunder	0,0309
Utgiftsgodtgjørelse	0,06	Emne: Satser for skattefrie gaver i arbeidsforhold	0,0307
Reisekostnader	0,055	Emne: Representasjon	0,0293

Figure C.20: Cluster 5: June 2016

Cluster number 6:			
Number of users in this cluster: 676			
Category		**Individual resouce**	
Naturalytelser	0,495	Emne: Fri bil (firmabil)	0,5102
Fildokument	0,359	Fildokument: Firmabil - beregning av fordel	0,2515
Satser	0,088	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,0823
Kontoplan	0,084	Emne: Næringsbil - tilbakeføring for privat bruk	0,0494
Enkeltpersonforetak	0,06	Fildokument: Privat bruk av næringsbil	0,0444
Reisekostnader	0,058	Emne: Krav til føring og oppbevaring av kjørebok	0,0376
Feriepenger	0,057	Emne: Forskudd ved inngåelse av leasingkontrakt - leietaker	0,0306
Bokføringsregelverket	0,054	Emne: Feriepengeavregning til ansatte med månedslønn og avtalefestet fe	0,0291
Kjøp av tjenester	0,052	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,0277
Formue	0,051	Emne: Gaver - ansatte og styremedlemmer	0,0271

Figure C.21: Cluster 6: June 2016

Cluster number 7:			
Number of users in this cluster: 4			
Category		**Individual resouce**	
Velferdstiltak	1	Emne: Utgifter til drift av båt som brukes til sosiale formål	1
Fildokument	0,333	Fildokument: Næringsoppgave 1 2016 (RF-1175)	0,25
Salg av varer	0,188	Emne: Salg av brukt fritidsbåt fra bruktforhandler	0,25
Kjøp av tjenester	0,167	Emne: Utgifter ved levering av avfall/søppel	0,1667
Drift, vedlikehold og påkostning av fast eien	0,167	Emne: Kjøp av materialer til oppussing av egne/leide lokaler	0,1667
Viderefakturering	0,1	Emne: Utgifter til vedlikehold og reparasjon av bygg	0,1667
Kjøp av driftsmidler	0,083	Emne: Salg med avansemetoden	0,125
Kjøp av fast eiendom	0,083	Emne: Viderefakturering av utlegg	0,1
Oppgaveplikt	0,05	slettet fildokument	0,0833

Figure C.22: Cluster 7: June 2016

Cluster number 8:			
Number of users in this cluster: 147			
Category		**Individual resouce**	
Regnskapsførerregelverket	0,716	Emne: Oppbevaring av regnskapsmateriale	0,716
Fildokument	0,183	Emne: Oppsigelse på grunn av alder	0,0698
Bokføringsregelverket	0,154	Fildokument: Oppdragsavtale	0,058
Diskriminering	0,07	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,056
Kontoplan	0,054	Emne: Fritaksmetoden	0,0277
Registrering	0,045	Emne: Valutagevinst/-tap - investering i formuesobjekter i utlandet	0,0268
Årsoppgjør	0,044	Emne: Spesifikasjon og dokumentasjon av lønnsopplysningspliktige ytelser	0,0268
Feriepenger	0,042	Emne: Krav til salgsdokumenter (faktura)	0,0254
Formue	0,037	Emne: Krav til sikring av regnskapsmateriale	0,0243
Naturalytelser	0,036	Emne: Krav om prosjektregnskap i bygge- og anleggsvirksomhet og verftsin	0,0239

Figure C.23: Cluster 8: June 2016

Cluster number 9:			
Number of users in this cluster: 303			
Category		**Individual resouce**	
Fildokument	0,694	Fildokument: Bankavstemming	0,2785
Kontoplan	0,068	Fildokument: Bokføringsbilag	0,1863
Kjøp av driftsmidler	0,04	Fildokument: Omsetningsoppgave mva 2016	0,1668
Feriepenger	0,035	Fildokument: Oversikt avstemminger	0,1609
Reisekostnader	0,034	Fildokument: Arbeidsgiveravgift 2016 pr termin	0,1304
Kjøp av tjenester	0,032	Fildokument: Forskuddstrekk (terminvis betaling)	0,1045
Naturalytelser	0,031	Kontoplan: Sticos kontoplan for aksjeselskaper (bygger på NS 4102)	0,0666
Satser	0,03	Fildokument: Omsetningsoppgave mva 2016 (uten kontospesifikasjon)	0,0592
Bokføringsregelverket	0,026	Fildokument: Omsetningsoppgave mva 2015 (uten kontospesifikasjon)	0,0533
Pengekrav	0,023	Fildokument: Omsetningsoppgave mva 2016 - månedlig - 12 terminer i over	0,0498

Figure C.24: Cluster 9: June 2016

Cluster number 10:			
Number of users in this cluster: 6			
==========			
Oppføring av fast eiendom	0,533	Emne: Anleggsbidrag - utbygging av infrastruktur (utbygger)	0,6667
Utland - omsetning og eksport	0,5	Emne: Salg av garantireparasjoner i Norge for utenlandsk oppdragsgiver	0,5
Bokføringsregelverket	0,194	Emne: Krav om prosjektregnskap i bygge- og anleggsvirksomhet og verftsin	0,1944
Fildokument	0,139	Emne: Utgifter til tilknytningsavgift for vann og avløp til bygg	0,1111
Salg av tjenester	0,111	Emne: Salg av tjenester til offentlig vei	0,1111
Transaksjoner med nærstående	0,056	Fildokument: Timeliste	0,0833
Kjøp av tjenester	0,056	Fildokument: Timeliste for tjenesteytende næringer	0,0833
Salg av varer	0,056	Fildokument: Oppdragsavtale	0,0833
Utleie av fast eiendom	0,056	Fildokument: Styremedlems villighetserklæring	0,0833
Tap	0,056	Fildokument: Arbeidsavtale	0,0833

Figure C.25: Cluster 10: June 2016

Cluster number 11:			
Number of users in this cluster: 3			
Category		**Individual resouce**	
Næringsbegrepet	1	Emne: Flere avgiftssubjekter driver virksomhet fra felles lokaler	1
Fildokument	0,214	Emne: Gjensidige forsikring - behandling av utbytte	0,1667
Kjøp av tjenester	0,214	Fildokument: Inkassovarsel	0,1667
Andre finansinntekter	0,167	Emne: Forsinkelsesrente mellom næringsdrivende	0,1667
Renteinntekter	0,167	Emne: Renter ved forsinket betaling fra 1. januar 2017 og historiske satser	0,1667
Generalforsamling	0,143	Emne: Utgifter til presentasjon/profilering av egne varer	0,1667
Kjøp av nytt kjøretøy	0,095	Emne: Ordinær generalforsamling	0,1429
Medvirker- og solidaransvar	0,083	Emne: Kjøp av ny tilhenger	0,0952
Satser	0,048	Emne: Fellesregistrering i Merverdiavgiftsregisteret	0,0833
Styret og daglig leder	0,048	Emne: Bilgodtgjørelse ved bruk av egen bil på reiser i Norge	0,0476

Figure C.26: Cluster 11: June 2016

Bibliography

Adomavicius, G. and Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):896–911.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749.

Agarwal, N., Haque, E., Liu, H., and Parsons, L. (2005). Research paper recommender systems: A subspace clustering approach. In *Proceedings of the 6th International Conference on Advances in Web-Age Information Management*, WAIM'05, pages 475–491, Berlin, Heidelberg. Springer-Verlag.

Aggarwal, C. C. and Yu, P. S. (2002). Redefining clustering for high-dimensional applications. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):210–225.

Agosti, M., Crivellari, F., and Di Nunzio, G. M. (2012). Web log analysis: A review of a decade of studies about information acquisition, inspection and interpretation of user interaction. *Data Min. Knowl. Discov.*, 24(3):663–696.

Algiriyage, N., Jayasena, S., and Dias, G. (2015). Web user profiling using hierarchical clustering with improved similarity measure. In *2015 Moratuwa Engineering Research Conference (MERCon)*, pages 295–300.

Amatriain, X. (2013). Mining large streams of user data for personalized recommendations. *SIGKDD Explor. Newsl.*, 14(2):37–48.

Amatriain, X., Jaimes*, A., Oliver, N., and Pujol, J. M. (2011). *Data Mining Methods for Recommender Systems*, pages 39–71. Springer US, Boston, MA.

Ammari, A., Lau, L., and Dimitrova, V. (2012). Deriving group profiles from social media to facilitate the design of simulated environments for learning.

In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 198–207, New York, NY, USA. ACM.

Aynaud, T. (2012). python-louvain. Available from: `https://github.com/taynaud/python-louvain`.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Bock, H.-H. (2008). Origins and extensions of the -means algorithm in cluster analysis. *Journal Électronique d'Histoire des Probabilités et de la Statistique [electronic only]*, 4(2):Article 14, 18 p., electronic only–Article 14, 18 p., electronic only.

Boratto, L. and Carta, S. (2014). Modeling the preferences of a group of users detected by clustering: A group recommendation case-study. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, WIMS '14, pages 16:1–16:7, New York, NY, USA. ACM.

Boratto, L. and Carta, S. (2015). The rating prediction task in a group recommender system that automatically detects groups: architectures, algorithms, and performance evaluation. *Journal of Intelligent Information Systems*, 45(2):221–245.

Boratto, L., Carta, S., Chessa, A., Agelli, M., and Clemente, M. L. (2009). Group recommendation with automatic identification of users communities. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '09, pages 547–550, Washington, DC, USA. IEEE Computer Society.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Buntine, W. (1992). Learning classification trees. *Statistics and computing*, 2(2):63–73.

Cheng, K., Li, J., and Liu, H. (2016). Featureminer: A tool for interactive feature selection. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 2445–2448, New York, NY, USA. ACM.

Cho, Y. S., Moon, S. C., Noh, S. C., and Ryu, K. H. (2012). Implementation of personalized recommendation system using k-means clustering of item category based on rfm. In *2012 IEEE International Conference on Management of Innovation Technology (ICMIT)*, pages 378–383.

Costa, A. F., Manzato, M. G., and Campello, R. J. (2016). Group-based collaborative filtering supported by multiple users' feedback to improve personalized ranking. In *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web*, Webmedia '16, pages 279–286, New York, NY, USA. ACM.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Demmel, J. W. (1997). *Applied numerical linear algebra*. SIAM.

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177.

Desrosiers, C. and Karypis, G. (2011). *A Comprehensive Survey of Neighborhood-based Recommendation Methods*, pages 107–144. Springer US, Boston, MA.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

Fayad, U., P. G. and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–54.

Forsati, R., Moayedikia, A., and Shamsfard, M. (2015). An effective web page recommender using binary data clustering. *Information Retrieval Journal*, 18(3):167–214.

Fuxman, A., Kannan, A., Li, Z., and Tsaparas, P. (2012). Enabling direct interest-aware audience selection. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 575–584, New York, NY, USA. ACM.

Gao, F., Xing, C., Du, X., and Wang, S. (2007). Personalized service system based on hybrid filtering for digital library. *Tsinghua Science and Technology*, 12(1):1–8.

Gao, M., Cao, F., and Huang, J. Z. (2013). A cross cluster-based collaborative filtering method for recommendation. In *2013 IEEE International Conference on Information and Automation (ICIA)*, pages 447–452.

George, T. and Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4 pp.–.

Gu, B., Hu, F., and Liu, H. (2000). Sampling and its application in data mining: A survey. *National University of Singapore, Singapore*.

Gutierrez, F. J. and Poblete, B. (2015). Sentiment-based user profiles in microblogging platforms. In *Proceedings of the 26th ACM Conference on Hypertext 38; Social Media*, HT '15, pages 23–32, New York, NY, USA. ACM.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, E. and Jain, S. K. (2014). Categorizing twitter users on the basis of their interests using hadoop/mahout platform. In *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–5.

Jolliffe, I. (2002). *Principal component analysis*. Springer Verlag, New York.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed April 18th 2017].

Kofod-Petersen, A. (2014). How to do a structured literature review in computer science. `https://research.idi.ntnu.no/aimasters/files/SLR_HowTo.pdf`. [Online - last accessed May 20th, 2017].

Kumar, S. (2013). Mining user interests from web history. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 03*, WI-IAT '13, pages 282–283, Washington, DC, USA. IEEE Computer Society.

Li, Q. and Kim, B. M. (2003). Clustering approach for hybrid recommender system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 33–38.

Li, X. and Murata, T. (2012). Using multidimensional clustering based collaborative filtering approach improving recommendation diversity. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '12, pages 169–174, Washington, DC, USA. IEEE Computer Society.

Martin, J. M. (2014). Behavioral segmentation of pinterest users. *Proceedings of the 3rd Workshop on Data-Driven User Behavioral Modeling and Mining from Social Media*, pages 13–14.

Moret, B. M. (1982). Decision trees and diagrams. *ACM Computing Surveys (CSUR)*, 14(4):593–623.

Müller, E., Günnemann, S., Assent, I., and Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proc. VLDB Endow.*, 2(1):1270–1281.

Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389.

Oates, B. J. (2006). *Researching Information Systems and Computing*. Sage Publications Ltd.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Quintarelli, E., Rabosio, E., and Tanca, L. (2016). Recommending new items to ephemeral groups using contextual user influence. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 285–292, New York, NY, USA. ACM.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

Sim, K., Gopalkrishnan, V., Zimek, A., and Cong, G. (2013). A survey on enhanced subspace clustering. *Data Min. Knowl. Discov.*, 26(2):332–397.

Tan, P., Steinbach, M., and Kumar, V. (2013). *Introduction to Data Mining: Pearson New International Edition*. Pearson Education Limited.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Watson, I. (2016). Cs760 - instance-based learning. `https://www.cs.auckland.ac.nz/~ian/CBR/ibl.pdf`. [Online - last accessed May 19th, 2017].

Xie, X. and Wang, B. (2016). Web page recommendation via twofold clustering: considering user behavior and topic relation. *Neural Computing and Applications*, pages 1–9.

Yanxiang, L., Deke, G., Fei, C., and Honghui, C. (2013). User-based clustering with top-n recommendation on cold-start problem. In *2013 Third International Conference on Intelligent System Design and Engineering Applications*, pages 1585–1589.

Yu, X. and Korkmaz, T. (2014). Finding the most evident co-clusters on web log dataset using frequent super-sequence mining. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pages 529–536.

Zeng, C., Jia, D., Wang, J., Hong, L., Nie, W., Li, Z., and Tian, J. (2012). Context-aware social media recommendation based on potential group. In *Proceedings of the 1st International Workshop on Context Discovery and Data Mining*, ContextDD '12, pages 5:1–5:9, New York, NY, USA. ACM.

Zhang, H., Chai, J., Wang, Y., An, M., Li, B., and Shen, Q. (2015). Application of clustering algorithm on tv programmes preference grouping of subscribers. *2015 IEEE International Conference on Computer and Communications (ICCC)*, pages 40–44.

Zhang, X. and Wang, L. (2013). The application of web log in collaborative filtering recommendation algorithm. In *2013 Ninth International Conference on Computational Intelligence and Security*, pages 763–765.