# Applied Data Mining for Business Intelligence

Niels Arnth-Jensen

Kongens Lyngby 2006

# Summary

**Abstract**
*Business Intelligence* (BI) solutions have for many years been a hot topic among companies due to their optimization and decision making capabilities in business processes. The demand for yet more sophisticated and intelligent BI solutions is constantly growing due to the fact that storage capacity grows with twice the speed of processor power. This unbalanced growth relationship will over time make data processing tasks more time consuming when using traditional BI solutions.

*Data Mining* (DM) offers a variety of advanced data processing techniques that may beneficially be applied for BI purposes. This process is far from simple and often requires customization of the DM algorithm with respect to a given BI purpose. The comprehensive process of applying BI for a business problem is referred to as the *Knowledge Discovery in Databases* (KDD) process and is vital for successful DM implementations with BI in mind.

In this project the emphasis is on developing a number of advanced DM solutions with respect to desired data processing applications chosen in collaboration with the project partner, gatetrade.net. To gatetrade.net this project is meant as an eye opener to the world of advanced data processing and to all of its advantages. In the project, gatetrade.net is the primary data supplier. The data is mainly of a transactional character (order headers and lines) since gatetrade.net develops and maintains e-trade solutions.

Three different segmentation approaches (*k-Nearest Neighbours* (kNN), *Fuzzy C-Means* (FCM) and *Unsupervised Fuzzy Partitioning - Optimal Number of Clusters* (UFP-ONC)) have been implemented and evaluated in the pursuit of finding a good clustering algorithm with a high, consistent performance. In order to determine optimal numbers of segments in data sets, ten different cluster validity criteria have also been implemented and evaluated. To handle gatetrade.net data types a Data Formatting Framework has been developed.

Addressing the desired data processing applications is done using the capable UFP-ONC clustering algorithm (supported by the ten cluster validity criteria) along with a number of custom developed algorithms and methods. For future gatetrade.net interest a draft for a complete BI framework using some or all of the developed data processing algorithms is suggested.

**Keywords:** Business Intelligence, Data Mining, Knowledge Discovery in Databases, par-

tition clustering algorithms, kNN, FCM, UFP-ONC, classification, cluster validity criteria.

# Resumé

*Business Intelligence* (BI) løsninger har igennem mange år været et populært emne blandt firmaer på grund af deres evne til at optimere og træffe beslutninger i forretningsprocesser. Efterspørgslen på mere avancerede BI løsninger er voksende på grund af det faktum, at udviklingen af lagringskapacitet vokser med dobbelt hast af udviklingen af processorkraft. Dette ubalancerede vækstforhold bevirker, at det i fremtiden vil tage længere tid at behandle data ved brug af traditionelle BI løsninger.

*Data Mining* (DM) tilbyder en bred vifte af avancerede databehandlingsteknikker, som med fordel kan anvendes til BI formål. Denne proces er ikke simpel og kræver ofte tilpasning af DM algoritmen med hensyn til et givent BI formål. Den omfattende proces, at anvende BI i forretningshenseender, kaldes *Knowledge Discovery in Databases* (KDD) og er vital for succesrig implementering af DM i BI løsninger.

I dette projekt er vægten lagt på at udvikle en række avancerede DM løsninger med hensyn til ønskede databehandlingsanvendelser, som er udvalgt i samarbejde med projektpartneren, gatetrade.net. Projektet skal for gatetrade.net's vedkommende få selskabets øjne op for avanceret databehandling og dets fordele. I projektet er gatetrade.net den primære dataleverandør. Dataene er hovedsageligt af en transaktionskarakter (ordrehoveder/-linier), da gatetrade.net udvikler og vedligeholder e-trade løsninger.

Tre forskellige segmenteringsfremgangsmåder (*k-Nearest Neighbours* (kNN), *Fuzzy C-Means* (FCM) and *Unsupervised Fuzzy Partitioning - Optimal Number of Clusters* (UFP-ONC)) er blevet implementeret og evalueret med henblik på at finde en god segmenteringsalgoritme med en høj, pålidelig ydelse. Til at bestemme optimale antal af segmenter i datasæt er ti forskellige segmenteringsvaliditetskriterier også blevet implementeret og evalueret. Til at håndtere gatetrade.net datatyper er et Data Formatting Framework blevet udviklet.

De ønskede databehandlingsanvendelser er imødekommet ved brug at UFP-ONC segmenteringsalgoritmen (med hjælp fra de ti segmenteringsvaliditetskriterier) samt et antal af særligt udviklede algoritmer og metoder. Til fremtidig gatetrade.net interesse er en skitse af et komplet BI framework indeholdende nogle eller alle af de udviklede databehandlingsalgoritmer foreslået.

# Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark (DTU) in partial fulfillment of the requirements for acquiring the Master of Science (M.Sc.) degree in engineering.

Thesis supervisor is Associate Professor Jan Larsen, Department of Informatics and Mathematical Modelling (IMM), DTU. Thesis co-supervisors are Christian Leth, Project Chief, gatetrade.net A/S and Allan Eskling-Hansen, Chief Financial Officer, gatetrade.net A/S. Thesis work was conducted from Apr. 2006 - Nov. 2006.

Lyngby, November 2006

Niels Arnth-Jensen (s001515)

# Contents

Chapter 1

# Introduction

Many present Business Intelligence (BI) analysis solutions are manually operated making it both time consuming and difficult for users to extract useful information from a multidimensional set of data. By applying advanced Data Mining (DM) algorithms for BI it is possible to automate this analysis process, thus making the algorithms able to extract patterns and other important information from the data set.

The process of applying DM for BI purposes (referred to as the Knowledge Discovery in Databases (KDD) process) is the main subject in this project. The data analyzed in the project is provided by gatetrade.net (profile of company found in chapter 3.1) who is keen on exploring the various advanced data processing possibilities of their data.

## 1.1 Project focus

Due to the large number of analysis methods DM offers, it is necessary to narrow the scope on a project of this kind. A list (made in collaboration with gatetrade.net) of desired data processing applications is found in table 3.1. The list reflects goals that time wise are realistic to accomplish within the given project period. Thus, the project's focus/goal is to develop advanced data processing algorithms that are able to fulfill the requirements of the desired applications.

## 1.2 Roadmap

Chapter 2 describes the basic concepts of BI, DM, KDD and presents examples of various BI solutions. It further comments on different data types and structures.

Chapter 3 contains a profile of the company gatetrade.net and a list of their desired data processing applications. General structures of relevant gatetrade.net databases and tables of extracted data attributes are also presented in this chapter.

Chapter 4 elaborates on three different clustering algorithms (kNN, FCM and UFP-ONC), shows how they are implemented and evaluates their individual performance with respect to test data sets. In the last part of the chapter, ten various cluster validity criteria for finding the optimal number of subgroups in a dataset are presented and individually tested on test data sets.

Chapter 5 describes how the UFP-ONC algorithm and custom developed algorithms are used to process gatetrade.net's data with respect to their desired application.

Chapter 6 comments on future perspectives regarding the algorithms discussed in this project. It outlines the structure of a complete BI framework able to support one or more of the developed solutions from chapter 5.

Chapter 7 contains the conclusions and sums up the work done in the project.

Chapter 8 contains all references used in the project.

A number of appendices follow chapter 8.

Chapter 2

# Business Intelligence

The term Business Intelligence (BI) is according to [1] originally popularized by Howard Dresner in 1989 and it describes "a set of concepts and methods to improve business decision-making by using fact-based support systems" [1]. In [2] BI is referred to as "a process for increasing the competitive advantage of a business by intelligent use of available data in decision making." Both quotations provide a good general understanding of the BI concept and make it pretty clear why BI is so popular among a large group of modern companies. This group includes business segments such as banking, insurance, trade, software development, intelligence services to name a few.

## 2.1 Why Business Intelligence?

Business Intelligence has become increasingly popular over the years and is currently a hot topic among many companies around the world. BI is often by companies considered to be a tool for tuning their way of doing business by guiding their decision making business-wise. In this way, the individual company can make more profitable decisions based on intelligent analysis of their data depots. The main reason for using BI among companies is probably to increase profitability. Why use data depots for storage only, when important and profitable market knowledge can be extracted from them using BI?

From a technical perspective making profit is not the only reason for using BI. Maintaining system and structure in large multidimensional data depots has always been an important task along with being able to analyze the contents of the depots. This task will in the future become even more challenging because of the evolution of storage devices and processor power. According to [3], processor power follows Moore's law and doubles each 18 months. On the other hand the capacity of storage devices quadruples in the same amount of time, thus making it increasingly difficult and time consuming to perform traditional analysis on the large data depots in the future. Therefore, intelligent analysis of data is becoming increasingly necessary over time as well as research in this field.

## 2.2 Applied Business Intelligence

A huge variety of BI solutions and techniques are currently available. Some of them are listed below [1].

**AQL (Associative Query Logic)** - Analytical data processing tool that compared to OLAP is less time consuming and more machine driven.

**Scorecarding, Dashboarding and Information visualization** - Scorecarding is a method that allows managers to get a broad view of the performance of a business while Dashboarding/Information Visualization deal with visual representation of abstract data.

**Business Performance Management** A tool for analyzing the current state of a business and for improving future strategies.

**DM (Data mining)** - Numerous methods for automatically searching large amounts of data for patterns and other interesting relations.

**Data warehouses** - Logical collections of information with structures that favor efficient data analysis (such as OLAP).

**DSS (Decision Support Systems)** - Machine driven system that aids the decision making process in a business.

**Document warehouses** - Instead of informing the business what things have happened (like the data warehouse does) the document warehouse is able to state why things have happened.

**EIS (Executive Information Systems)** - These systems are often considered as a specialized form of DSS with the purpose of facilitating the information and decision making needs of senior executives.

**MIS (Management Information Systems)** - A machine driven system for processing data and providing analysis reports for decision making and planning. In order to retrieve data the system has access to all communication channels in a business.

**GIS (Geographic Information Systems)** - A computer system for working with geographical data (e.g. satellite images) with editing, analyzing and displaying functionality.

**OLAP (Online Analytical Processing)** - OLAP is a tool for doing quick analytical processing of multidimensional data by running queries against structured OLAP cubes that is build from a set of data sources.

**Text mining** - This task is generally referred to as the process of extracting interesting and nontrivial information/knowledge from unstructured text

As shown in the list, BI can be applied in many interesting ways with one important thing in common - they all aid the user in the process of analyzing extensive quantities of information. However, the BI complexity of the individual solution varies a lot and it is possible to distinguish the solutions in terms of how automatic and intelligent they are. To generalize, BI solutions can be divided into two groups of analysis types.

**Query-Reporting-Analysis** - This type of analysis is often query based and is normally used for determining ”What happened?” in a business over a given period of time. Because queries are used the user already knows what kind of information to search for. Additionally, BI solutions of this kind are generally operated manually and are therefore time consuming.

**Intelligent Analysis (Data Mining)** - While the Query-Reporting-Analysis is able to provide answers for questions of the ”What happened?” kind, Data Mining utilizes clever algorithms for a much deeper and intelligent analysis of data. BI solutions using Data Mining techniques are then capable of handling ”What will happen?” and ”How/why did this happen?” matters. All this is done in a semi- or full-automatic process saving both time and resources.

This is exemplified by comparing two different cases of BI, OLAP and Data Mining.

As described earlier, OLAP is used manually and the user has to know what to look for (analytic queries of dimensional nature). The OLAP cubes make it easy to slice/dice the multiple data dimensions in order to investigate a certain data relation. However, this can be a difficult and time consuming task when working with large amounts of data with high dimensionality - similar to finding a needle in a haystack. Finally, OLAP provides the user with a low level data analysis able to handle ”What has happened?” queries.

Compared to OLAP, Data Mining operates very differently and offers a much more powerful and deep data analysis. The user does not have to locate the interesting patters/relations manually. Instead, the Data Mining algorithms will ”mine” multidimensional data intelligently in a semi-/full automatic process and extract interesting findings. Further, Data Mining can be used in a wide range of complex scenarios - often of the ”What will happen?” or ”How/Why did this happen?” character (see section 2.4).

The example demonstrates that the term Business Intelligence covers different types of data analysis methods/tools regardless of their level of intelligence (the depth of the data analysis) and automation. A rule of thumb states that the depth of a data analysis method is proportional to its complexity - this is perhaps the main reason for ”low-level” Business Intelligence to be so widespread.

## 2.3 Knowledge Discovery in Databases

Another popular term in the world of intelligent data processing is Knowledge Discovery in Databases (KDD). Fayyad [4] defines KDD as ”the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data”. Understanding the difference between Knowledge Discovery in Databases and Business Intelligence (and Data Mining) is important for this project and should therefore be elaborated on.

The terms Knowledge Discovery in Databases and Data Mining are often believed to have the same meaning. However, this is not the fact! While Data Mining is the name of a group of intelligent BI methods the term KDD describes the entire process of extracting information from a data warehouse. Moreover, the Data Mining task is part of the KDD process and according to Fayyad [4] the KDD process can be divided into the following steps (once the wanted goals of the process has been decided on).

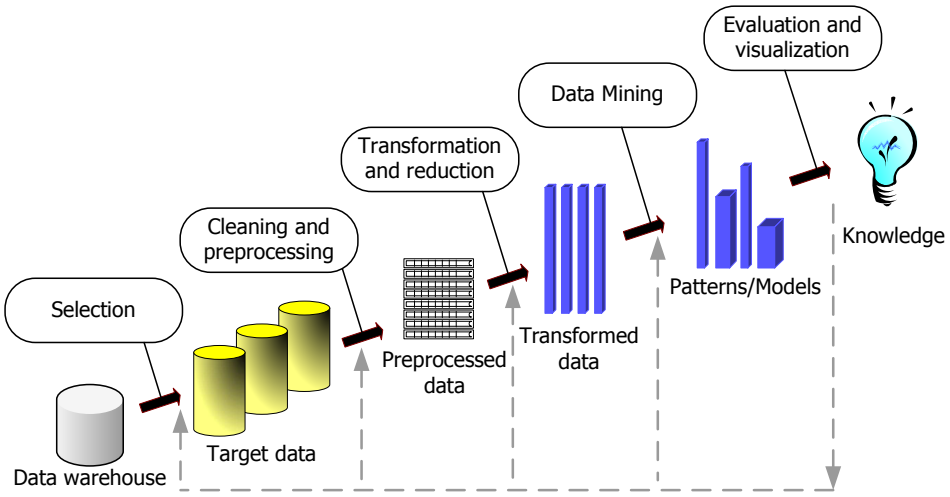


**Figure 2.1:** Fayyad's Knowledge Discovery in Databases process.

1. **Selecting target data from a data warehouse** - A data warehouse often contains many databases which each contain large amounts of data. To save resources only relevant target data should be selected from the data warehouse.

2. **Cleaning and preprocessing the target data** - The raw data is often in an unwanted format and may contain noise and missing data fields. Strategies for handling these factors should be decided on.

3. **Transformation and reduction of the preprocessed data** - In this step, useful features to represent the data depending on the goal in a given task should be found. Further, dimensionality reduction/transformation can reduce the effective number a variables in consideration.

4. **Applying Data Mining to the transformed data** - Once the data has been transformed, a proper Data Mining technique should be applied in order to intelligently process the data for patterns and other information.

5. **Evaluation/visualization of Data Mining results** - The results of the Data Mining step are not always easy to interpret. Using visualization in the evaluation process can therefore be of great advantage.

All of the steps in the KDD process are essential to ensure useful models/patterns are extracted from a given data set. Solely applying Data Mining methods to data sets regardless of the other KDD steps often results in discovery of misleading models/patterns and is therefore a risky activity. As shown in figure 2.1, the KDD process is iteratively involving numerous steps with many decisions made by the user. Finally, if useful knowledge is extracted in the KDD process this should be implemented in the respective company's business model in order to optimize important factors such as turnover and profit.

## 2.4 Data Mining

The most challenging step of the Knowledge Discovery in Databases process (figure 2.1) is probably performing the actual Data Mining. As mentioned earlier, Data Mining is the task of extracting patterns and other interesting relations from large volumes of data. This nontrivial task is accomplished by the use of complex algorithms in a (semi-)automatic manner.

Before applying the Data Mining, it is crucial that the data has been properly reduced and transformed to avoid unnecessary complications. The preparation of the data also depends on what kind of Data Mining is wanted. Because several Data Mining tasks exist it is important to decide on which kind of task to use when defining the goals and purposes of the Knowledge Discovery in Databases process. The most important Data Mining tasks are described in the following sections.

### 2.4.1 Data Mining Tasks

#### Classification

Classification is supposedly the most popular Data Mining tasks considering its broad application domain. Its main purpose is to classify one or more data samples that may consist of few or many features (dimensions). The latter case makes the classification task more complex due to the large number of dimensions.

The actual number of classes is not always given or obvious in a classification task. Therefore, it is possible to distinguish between supervised and unsupervised classification. For supervised classification the number of classes is known along with the properties of each class. Neither of these is given in unsupervised classification which makes this task the more challenging one of the two.

The list below further exemplifies the use of the classification task.

1. Is a given credit card transaction fraudulent?
2. What type of subscription should be offered a given customer?
3. What type of structure does a specific protein have?
4. Is this customer likely to buy a bicycle?
5. Why is my system failing?

#### Estimation

Estimation is somewhat similar to classification algorithm-wise. However, estimation does not deal with determining a class for a particular data sample. Instead, it tries to predict a certain measure for a given data sample.

The list below further exemplifies the use of the estimation task.

1. What is the turnover of a company going to be?
2. What is the density of a given fluid?
3. When will a pregnant woman give birth?
4. For how long will this product work before failing?
5. How much is a specific project going to cost?

### Segmentation

Segmentation basically deals with the task of grouping a given data set into a few main groups (clusters). The task of describing a large multidimensional data set (say customers) will therefore benefit from the use of segmentation. Moreover, many algorithm types can be used in segmentation systems.

The list below further exemplifies the use of the segmentation task.

1. How can a given buyer/supplier group be differentiated?
2. Which types of ground does a given satellite image contain?
3. Is a specific transaction an outlier?
4. Which segments is a market based on?
5. Which groups of visitors are using a given search engine?

### Forecasting

Forecasting is another important Data Mining task that is used for predicting future data values given a time series of prior data. Forecasting is a popular task often performed using simple statistical methods. However, forecasting done in the Data Mining domain uses advanced (learning) methods (e.g. *Neural Networks*, *Hidden Markov Models*) that in many cases are more accurate and informative than the standard statistical methods (e.g. moving averages).

The list below further exemplifies the use of the forecasting task.

1. What will the weather be like tomorrow?
2. Will a particular stock price rise over the next couple of days?
3. What are the inventory levels next month?
4. How many sunspots will occur next year?
5. How will the average temperature on earth evolve throughout the next 10 years?

#### Association

Association deals with task of locating events that are frequently occurring together and benefiting from this knowledge. One of the most popular examples of association is probably Amazon.com's web shop that is able to recommend related products to customers.

The list below further exemplifies the use of the association task.

1. Which products should I recommend to my customers?
2. Which services are used together?
3. Which products are highly likely to be purchased together in a supermarket?
4. Which books are highly likely to be borrowed together in a library?
5. Which dishes from a cookbook go well together?

#### Text Analysis

Another key Data Mining task is text analysis. Text analysis has several purposes and is often used for finding key terms and phrases in text bits. In this way, text analysis can convert unstructured text into useful structured data that can be further processed by other Data Mining tasks (e.g. classification, segmentation, association).

The list below further exemplifies the use of the text analysis task.

1. Which segments does a given mailbox contain?
2. How is a document classified?
3. Which subjects does a specific web page contain?
4. How is a quick overview of multiple lecture notes from a classmate gained?
5. Which terms are likely to occur together?

## 2.5 Data types

The previous section described some of the most important Data Mining tasks and their applications. However, in order to implement and use the different Data Mining tasks, knowledge about their input (data) is a necessity. The term data may seem somewhat ambiguous and in [5] data is described as "a collection of facts from which conclusions may be drawn". Further, data may be represented in many ways and because it plays a major part in this project a common notation for data is needed.

Often a given data set may be regarded as set, $\mathcal{O}$, of $N$ objects (or populations) where each object is put together by a finite set, $\mathcal{V}$, of $m$ variables. Each variable represents a property

of object $x \in \mathcal{O}$ by the value $v(x)$ (also denoted $x_v$ in a range $R_v$ of the variable $v \in \mathcal{V}$). By using this notation it is possible to express each object in $\mathcal{O}$ as an element in the direct product space

$$x = (x_v)_{v \in \mathcal{V}} \in \mathcal{U} := \prod_{v \in \mathcal{V}} R_v \tag{2.1}$$

for all given variables in $\mathcal{V}$. In this context $\mathcal{U}$ is referred to as the object space (or universe).

Besides having a common notation for handling data it is also important to know the various natures in which data may exist before doing any processing of it. Below is described some of the most common data types [7].

**Qualitative** (also categorical, nominal, modal) variables explicitly describe various properties of an object e.g. hair color $R_v = \{blonde, brown, brunette, red, etc...\}$ or Boolean values $R_v = \{0, 1\}$. Note that qualitative variables have no intrinsic ordering i.e. there is no agreed way of ordering hair colors from highest to lowest.

**Quantitative** (also real, numeric continuous) variables may describe properties such as age, profit, temperature. A $m$ dimensional data set, where it for each dimensional holds $R_v = \mathbb{R}$, has a $m$ dimensional real object space $\mathcal{U} = \mathbb{R}^m$.

**Set valued** variables have multiple attributes e.g. variables for describing movies where each variable has 3 attributes (movie title, leading actors and year).

**Ordinal** variables are similar to qualitative variables, but with a clear ordering of the variables e.g. the fuel level in a car given by 3 variables $R_v = \{low, medium, high\}$. Various quantitative variables are also regarded to be ordinal.

**Cyclic** variables are of a periodic nature e.g. the 60 minutes in an hour for which $R_v = \mathbb{Z}/(60\mathbb{Z})$. For cyclic variables standard mathematical operations such as addition and subtraction are not directly applicable - certain periodicity precautions are necessary when processing such variables.

The variable types described above are some of the most important and common data types in Data Mining and being able to differentiate these types is a crucial factor to successful processing of data. Further, it is important to tell structured data sets apart from unstructured ones. Examples of both structured and unstructured data are given below.

- Structured Data
  - Dimensional Data
    - Buyer demographics
    - Supplier demographics
    - Product properties
  - Transactional Data
    - Order headers with buyer, supplier, ship to address, etc.
    - Order lines with product id, unit price, number of items, etc.
    - Credit card purchases
- Unstructured Data

  - Textual Data
    * Descriptions of order lines
    * Buyer comments
    * A doctors notes

In the following chapters of this project, gatetrade.net data is processed with reference to the BI concepts, DM methods and data types/structures mentioned in chapter 2.

Chapter 3

# gatetrade.net company profile and project goals

It makes good sense to use Fayyad's KDD process [4] in the complex task of exploring gatetrade.net data and extracting useful information from it. Before the different steps of the KDD process are carried out in chapter 5, it is important to have clear goals and intentions of what the applied Data Mining should accomplish with respect to the gatetrade.net data. A short profile of gatetrade.net and its business areas is also relevant.

## 3.1 gatetrade.net Profile

gatetrade.net is a 100% Danish owned company that is leading in its field of business in Denmark with more than 1500 customers [6]. gatetrade.net specializes in e-trade solutions which basically make e-trading easier and cheaper for buyers and suppliers with typical commodities such as different office supplies and services. The e-trading is taking place on multiple systems [6] and two of these are of interest in the KDD analysis.

**gatetrade.net - Marketplace** : The Marketplace system offers a complete basic e-trade solution to companies of all sizes and the ability to fully integrate with prior commerce systems should that be the case. Marketplace is solely web based which makes it easy to generate and handle invoices. By further offering updated supplier catalogues and administration of suppliers price and framework agreements, Marketplace constitutes a cost efficient solution. In order to use Marketplace every buyer and supplier must pay a periodic subscription fee which amount depends on the buyer/supplier's size/usage. Moreover, suppliers must additionally pay a given percentage of their turnover.

**gatetrade.net - eProcurement** : Besides offering a complete basic e-trade solution the eProcurement system also supports full integration with most accounting systems. Moreover, eProcurement is fully integrated with Marketplace allowing the buyer to do

trading through Marketplace and benefit from its advantages. eProcurement thereby provides a broader and more complete e-trade solution compared to Marketplace.

## 3.2 Goals of the KDD process

Fayyad's [4] KDD process requires predefined goals/objectives before any of its steps can be carried out. A list of prioritized goals for BI solutions has therefore been made in collaboration with gatetrade.net. Time-wise, the list should reflect goals that are realistic to complete within the period during which the project is done. This limitation is necessary due to the countless possibilities/applications of Data Mining and other Business Intelligence methods as mentioned earlier on. Table 3.1 shows the list of prioritized goals of accomplishments for the data processing task.

1. Examine transactions that are not made through the Marketplace and the possible reasons for this.
2. Analysis of buyers' use of suppliers in Marketplace. Examine if the number of suppliers of a given buyer can be reduced in order to obtain lower prices for the buyer.
3. Analysis/profiling of monthly buyer/supplier development for Marketplace/eProcurement.
4. Analysis of lost suppliers in Marketplace.
5. Examine in Marketplace if trade is canalized through top buyers in a given company leaving a number of inactive buyers.
6. Examine the possibility of recommending relevant products to buyers in Marketplace.

**Table 3.1:** List of goals of accomplishments for the data processing task.

## 3.3 Extraction of proper data from gatetrade.net data depot

Locating relevant data from the gatetrade.net data depot should be done with respect to the list of data processing goals. The gatetrade.net data depot holds several databases for both the Marketplace and the eProcurement system. Marketplace data is available from the 15 month period, January 2005 - March 2006 while eProcurement data is available from the 7 month period, September 2005 - March 2006. The structures of the main databases available for both Marketplace and eProcurement are exclusively of a transactional character as order information are continuously being stored in these. Each new order is for both systems stored as an order header (containing various details of buyer, buyer's company, supplier, etc.) and a number of order lines (one line per different product in the order). It should be stressed that although the eProcurement system names its contents "invoice" these are in this project referred to as orders similar to the orders in the Marketplace system. For each order this project operates with three main players - the buyer in the order, the buyer's company and finally the supplier of the product(s) in the order.

The following sections illustrate from which databases data for this project was gathered along with a short description of the data. All gatetrade.net data were extracted in Excel format directly from the data depots.

### 3.3.1 Marketplace databases

Data from the Marketplace system was collected from two primary databases in the gatetrade.net data depot. The first database contains order headers (`POM_ORDER_HEADERS`) while the second stores order lines (`POM_ORDER_LINES`). The common attribute `ORDER_NUMBER` binds headers and lines to specific orders. Moreover, `POM_ORDER_HEADERS` numbered 150.000+ order headers and `POM_ORDER_LINES` numbered 400.000+ order lines for the given time period. Database relations are illustrated in figure 3.1.

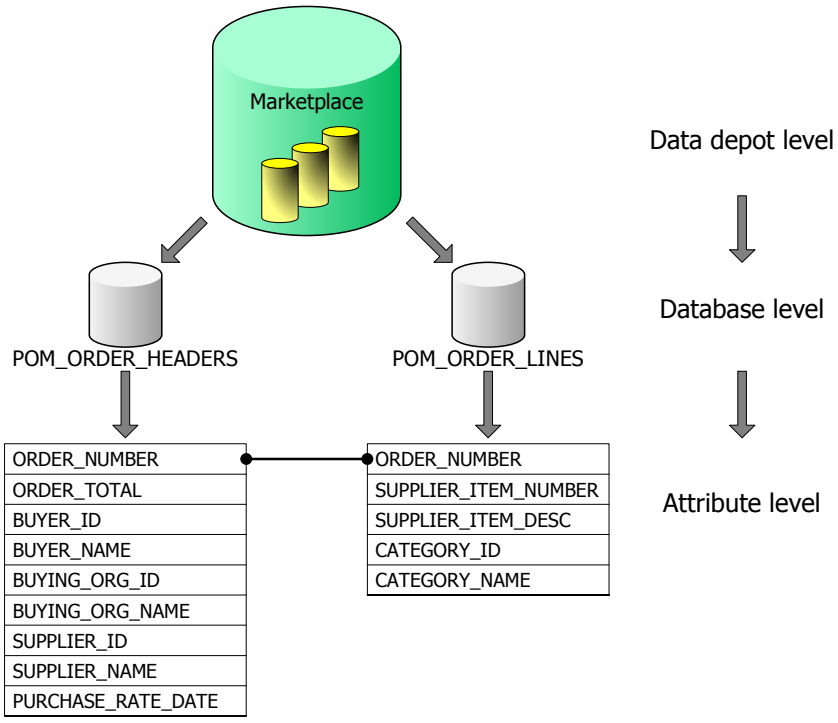


**Figure 3.1:** Overview of extracted Marketplace databases and attributes.

Attribute descriptions of databases POM_ORDER_HEADERS and POM_ORDER_LINES are found in table 3.2 and 3.3 respectively.

| Attribute | Description |
|---|---|
| ORDER_NUMBER | ID number of order |
| ORDER_TOTAL | Total of order |
| BUYER_ID | ID of buyer |
| BUYER_NAME | Name of buyer |
| BUYER_ORG_ID | ID of buyer's company |
| BUYER_ORG_NAME | Name of buyer's company |
| SUPPLIER_ID | ID of supplier |
| SUPPLIER_NAME | Name of supplier |
| PURCHASE_RATE_DATE | Date of order |

**Table 3.2:** POM_ORDER_HEADERS attribute descriptions.

| Attribute | Description |
|---|---|
| ORDER_NUMBER | ID number of order |
| SUPPLIER_ITEM_NUMBER | Supplier's ID of a given product |
| SUPPLIER_ITEM_DESC | Supplier's description of a given product |
| CATEGORY_ID | ID of product category |
| CATEGORY_NAME | Name of product category |

**Table 3.3:** POM_ORDER_LINES attribute descriptions.

### 3.3.2 eProcurement databases

Similar to Marketplace, orders from the eProcurement system are collected to two main databases in the gatetrade.net data depot. The first database contains order headers (`eProc_INVOICE`) while the second stores order lines (`eProc_INVOICELINE`). The common attribute `pkid`/`invoice_id` binds headers and lines to specific orders. Further on, information on eProcument buyers, companies and suppliers are provided by three smaller databases (`eProc_USER`, `eProc_COMPANY` and `eProc_SUPPLIERS`). Moreover, `eProc_INVOICE` numbered 65.000+ order headers and `eProc_INVOICELINE` numbered 350.000+ order lines for the given time period. All database relations are illustrated in figure 3.2.

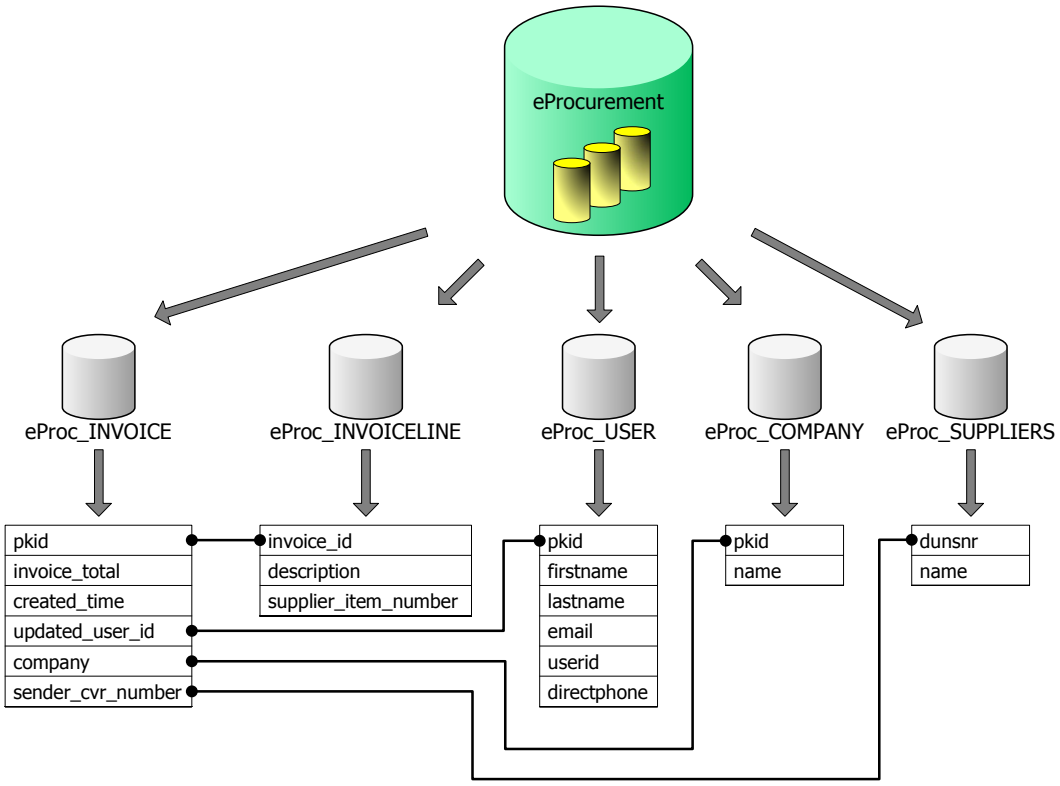


**Figure 3.2:** Overview of extracted Marketplace databases and attributes.

Attribute descriptions of databases `eProc_INVOICE` and `eProc_INVOICELINE` are found in table 3.4 and 3.5 respectively.

| Attribute | Description |
|---|---|
| pkid | ID of order |
| invoice_total | Total of order |
| created_time | Date of order |
| updated_user_id | ID of buyer |
| company | ID of company |
| sender_cvr_number | CVR number of supplier |

**Table 3.4:** `eProc_INVOICE` attribute descriptions.

| Attribute | Description |
|---|---|
| invoice_id | ID of order |
| description | Buyer's description of a given product |
| supplier_item_number | Supplier's ID of a given product |

**Table 3.5:** `eProc_INVOICELINE` attribute descriptions.

Attribute descriptions of databases (`eProc_USER`, `eProc_COMPANY` and `eProc_SUPPLIERS`) are found in Appendix A.

Chapter 5 elaborates on further formatting and processing of Marketplace and eProcurement data with respect to the desired tasks in table 3.1.

Chapter 4

# Relevant Data Mining approaches to processing gatetrade.net data

In this chapter, relevant Data Mining approaches to processing gatetrade.net data are discussed and elaborated on. Chapter 2 briefly mentioned some of the most common Data Mining tasks and a couple of their main utilizations. Besides serving as examples the utilizations possibilities may give a hint of which Data Mining tasks are of interest for the work in the project.

To further determine which Data Mining tasks are relevant to this project the list of prioritized goals of data processing accomplishments in table 3.1 has to be taken into consideration. Each item in this list represents a demand for a Business Intelligence solution regardless of its sophistication level. Moreover, a low level Business Intelligence solution may in some cases be sufficient.

The list in table 3.1 contains a relative large variety of goals. The goal of item 1 on the list is of a more abstract nature compared to the goals of the other items. That said, the goals of item 2-6 are potential realizable using Data Mining tasks such as segmentation, classification, association and text analysis. Generally, goals 3, 4 and 5 on the list have a common demand for segmentation of Marketplace buyers, buyer's companies and suppliers on a total basis (for all 15 months in the period) and on a monthly (development) basis.

Because of the strong demand for the segmentation Data Mining task, the rest of this chapter focuses on various segmentation approaches along with a mutual evaluation of them.

## 4.1 Segmentation approaches

Segmentation (clustering) of a set $\mathcal{O}$ of $N$ objects with $m$ attributes each is a popular task in the field of signal processing which is why numerous methods and algorithms for solv-

ing this particular task already exist. Even though all of these algorithms are able to do segmentation, each algorithm has its advantages and drawbacks (e.g. level of complexity, clustering ability, etc.). Choosing a proper algorithm for a given application therefore requires a broad knowledge of the properties of the different clustering algorithms and the data under consideration.

According to [7] there are two main classes of clustering algorithms - namely *hierarchical* and *partitioning* clustering algorithms.

***Hierarchical*** clustering algorithms make use of an $N$ level hierarchy in the search for good clusterings. There are two standard ways of building the $N$ level hierarchy - a top-down and a bottom-up approach. Top-down approaches start from a coarse clustering (1 cluster of $N$ elements) and add an additional cluster for each new level in the hierarchy resulting in a full $N$ level hierarchy with a bottom layer that consists of $N$ 1-element clusters. Oppositely, bottom-up approaches start from a fine clustering and end with coarse clustering. Optimal numbers of clusters are located using various cluster validation criteria. Methods using the top-down/bottom-up approaches are called *divisive/agglomerative* respectively and their constructed hierarchies are typically illustrated graphically in a so called *dendrogram*.

***Partitioning*** clustering algorithms generally try to locate the best possible clustering for $K$ clusters. Each of the $N$ elements in the set, $\mathcal{O}$, is individually assigned to one of $K$ initial clusters prototypes. By iteratively modifying the cluster prototypes (and thereby the cluster assignments of the elements) with respect to a given objective function more optimal clusterings appear. It is moreover possible to find the optimal number of clusters with respect to cluster validity criteria by iterating over a specified range of clusters.

For a more detailed overview of the different clustering algorithm classes/subclasses a graphical tree is available in [7] in figure 6 on page 340.

Both of the two main clustering algorithm classes are able to do segmentation of a set, $\mathcal{O}$, of $N$ objects. However, their clustering ability and level of robustness are very different which [7] also states. Hierarchical clustering algorithms provide good clusterings only for very special cases of data sets, but prove inaccurate when processing data sets of multi-dimensional, multivariate classes. For these more demanding data sets robust partitioning clustering algorithms are more suitable which is why they for most segmentation purposes are considered more relevant. Moreover, hierarchical clustering methods may be used for initial clusterings in partitioning clustering algorithms.

In order to choose suitable segmentation approaches an assumption of the nature of the gatetrade.net data has to be made. In this project the gatetrade.net data is regarded as normal (*Gaussian*) distributed classes having probability density functions [8] of the form

$$p(x) = \frac{1}{(2\pi\sigma^2)} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.1}$$

where $\mu$ and $\sigma^2$ are called the *mean* and *variance* respectively, while $\sigma$ (the square root of the variance) is called the *standard deviation*. The gatetrade.net data may further be

expressed in $m$ dimensions. For $m$ dimensions the multivariate normal probability density function [8] can be written

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{4.2}$$

where the mean $\boldsymbol{\mu}$ is a $m$ dimensional vector, $\boldsymbol{\Sigma}$ is a $(m \times m)$ *covariance matrix* and $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$.

Based on the assumption of the gatetrade.net data this project will focus on various partitioning clustering algorithms since a good clustering of the gatetrade.net data is wanted.

For the segmentation of the gatetrade.net data three various partitioning clustering algorithms have been chosen, *k-Nearest Neighbours* (kNN), *Fuzzy C-Means* (FCM) and *Unsupervised Fuzzy Partition - Optimal Number of Classes* (UFP-ONC). The three selected algorithms represent three rather different segmentation approaches with respect to

**Algorithm complexity** - The complexity level of an algorithm depends mainly on its structure and clustering capabilities. A large number of calculations and iterations make the algorithm slow and cpu hungry which in some cases is undesired. For most clustering algorithms the complexity/speed versus clustering accuracy trade-off applies.

**Soft versus hard clustering** - The cluster membership of $N$ elements in a set, $\mathcal{O}$, is in principle defined in two ways - either as being soft or hard. For soft (also called *fuzzy*) clusterings it holds that each element is allowed to be a member of more than one cluster (resulting in degrees of membership). For hard (also called *crisp*) clusterings the opposite holds - each element can only be member of one of the $K$ clusters.

***Similarity/dissimilarity* indices** - In order to form cluster memberships of $N$ elements in a set, $\mathcal{O}$, a similarity/dissimilarity index is needed depending on the type of data. The purpose of the index is to be able to measure similarity/dissimilarity between an element and a cluster. Dissimilarity indices are much more intuitive to use when processing data of a quantitative character. Dissimilarity indices are realizable via *distance functions* (e.g. in the geometric sense through an Euclidian distance measure).

The following three sections elaborate on the three chosen partitioning cluster algorithms and their individual characteristics. It should be mentioned that the algorithms before processing data set $\boldsymbol{X}$ normalize it by subtracting the global mean of $\boldsymbol{X}$ followed by a dimension-wise division of the respective standard deviations.

### 4.1.1 k-Nearest Neighbours

In [9] Böhm and Krebs propose the *k-nearest neighbour* similarity join (index) and compare this to two well known similarity joins, the *distance range* join and the *k-closest pair* join (also known as the *k-distance* join). Before going into details about the $k$-nearest neighbour join and the clustering algorithm using this join operation it is necessary to describe the

two other similarity joins. It is further noticeable that all three similarity joins operate with respect to Euclidian distance measures.

The *distance range join*, $\mathcal{R} \underset{\varepsilon}{\bowtie} \mathcal{S}$, of two multidimensional sets, $\mathcal{R}$ and $\mathcal{S}$, is the set of pairs for which it holds that the distance between the objects does not exceed the specified parameter $\varepsilon$ as stated by the definition

$$\mathcal{R} \underset{\varepsilon}{\bowtie} \mathcal{S} := \left\{ (\boldsymbol{r}_i, \boldsymbol{s}_j) \in \mathcal{R} \times \mathcal{S} : d^2 (\boldsymbol{r}_i, \boldsymbol{s}_j) \leq \varepsilon \right\} \tag{4.3}$$

The distance range join operation, $\mathcal{R} \underset{\varepsilon}{\bowtie} \mathcal{S}$, is depicted in figure 4.1(a).

The *k-closest pair join*, $\mathcal{R} \underset{k-CP}{\bowtie} \mathcal{S}$, of two multidimensional sets, $\mathcal{R}$ and $\mathcal{S}$, retrieves the $k$ pairs of $\mathcal{R} \times \mathcal{S}$ having minimum distance. For the $k$-closest pair join the following condition holds

$$\forall (\boldsymbol{r}, \boldsymbol{s}) \in \mathcal{R} \underset{k-CP}{\bowtie} \mathcal{S}, \forall (\boldsymbol{r}', \boldsymbol{s}') \in \mathcal{R} \times \mathcal{S} \backslash \mathcal{R} \underset{k-CP}{\bowtie} \mathcal{S} : d^2 (\boldsymbol{r}, \boldsymbol{s}) < d^2 (\boldsymbol{r}', \boldsymbol{s}') \tag{4.4}$$

The $k$-closest pair join operation , $\mathcal{R} \underset{k-CP}{\bowtie} \mathcal{S}$, is depicted in figure 4.1(b).

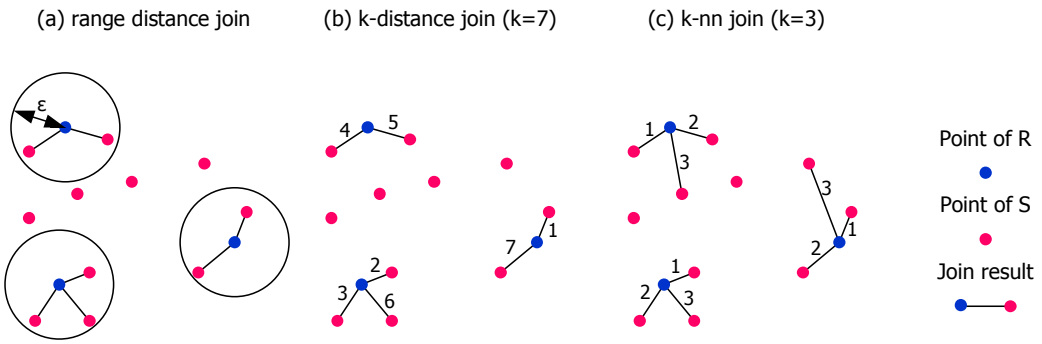


**Figure 4.1:** Difference between the three join operations.

The reason for Böhm and Krebs [9] to propose the $k$-nearest neighbour join was primarily because of the inconvenient functionality of the two mentioned similarity joins. The distance range join has the disadvantage of a result set scope which is difficult to control. The $k$-closest pair join overcomes this inconvenience by controlling the result set size with the $k$ parameter. Unfortunately, the consideration of the $k$ best pairs of two sets is only required in very few applications. Much more common are applications (such as classification/clustering) in which each object in a set must be combined with the $k$ nearest objects in given set. This operation corresponds exactly to the one of the $k$-nearest neighbour join.

The *k-nearest neighbour join*, $\mathcal{R} \underset{k-nn}{\ltimes} \mathcal{S}$, of two multidimensional sets, $\mathcal{R}$ and $\mathcal{S}$, is the set of pairs for which it holds that each object in set $\mathcal{R}$ is paired with the $k$ closest objects in set $\mathcal{S}$. For the $k$-nearest neighbour join the following condition holds

$$\forall (\boldsymbol{r}, \boldsymbol{s}) \in \mathcal{R} \underset{k-nn}{\ltimes} \mathcal{S}, \forall (\boldsymbol{r}, \boldsymbol{s}') \in \mathcal{R} \times \mathcal{S} \backslash \mathcal{R} \underset{k-nn}{\ltimes} \mathcal{S} : d^2 (\boldsymbol{r}, \boldsymbol{s}) < d^2 (\boldsymbol{r}, \boldsymbol{s}') \tag{4.5}$$

Similar to the previous two join operations, the $\mathcal{R} \underset{k-nn}{\ltimes} \mathcal{S}$ join operation is illustrated in figure 4.1(c).

Additionally, it is also possible to express the $k$-nearest neighbour join as an extended `SQL` query as shown in table 4.1.

```
SELECT * FROM R,
  ( SELECT * FROM S
    ORDER BY || R.obj - S.obj ||
    STOP AFTER k )
```

**Table 4.1:** The KNN-join expressed in `SQL` syntax.

#### k-Nearest Neighbours clustering algorithm

The $k$-nearest neighbour join operation does not by itself constitute a complete clustering algorithm - in [9] it is proposed to serve the purpose of being an important database primitive for supporting Data Mining and similarity searches. However, [10] describes a clustering algorithm using $k$-nearest neighbour operation.

The k-Nearest Neighbours (kNN) clustering algorithm [10] (see Algorithm 1) does not rely on initial cluster prototypes when processing a data set $\boldsymbol{X}$ with $N$ objects of $m$ attributes each. Necessary initial conditions are $k$ (number of nearest neighbours)and $K$ (number of clusters) - a standard value for $k$ is $round\,(n/K - 1)$. The first cluster center, $\boldsymbol{V}_1$, is found by taking the mean value of $\boldsymbol{y}_1$ and its k-nearest neighbours where $\boldsymbol{y}_1$ is the object in $\boldsymbol{X}$ that is furthest away from the global mean ($\bar{\boldsymbol{V}}$) of the $N$ objects in $\boldsymbol{X}$. $\boldsymbol{y}_1$ and its k-nearest neighbours are deleted from $\boldsymbol{X}$ since they are now members of the first cluster center $\boldsymbol{V_1}$. The second cluster center, $\boldsymbol{V}_2$, is found by taking the mean value of $\boldsymbol{y}_2$ and its k-nearest neighbours where $\boldsymbol{y}_2$ is the object among the remaining objects in $\boldsymbol{X}$ that is furthest away from $\boldsymbol{V_1}$. This procedure is repeated until until the $K$ cluster centers have been located. If any objects remain in $\boldsymbol{X}$ these are assigned nearest cluster centers followed by an update of all $K$ cluster centers.

### 4.1.2 Fuzzy C-Means

The Fuzzy C-Means (FCM) clustering algorithm [11] is based on minimization of the objective function in (4.6) with respect to a fuzzy $K$ partition ($\boldsymbol{U}$) of the data set ($\boldsymbol{X}$ containing $N$ objects of $m$ dimensions each) and to a set of $K$ cluster prototypes ($\boldsymbol{V}$).

$$J_{q,FCM} (\boldsymbol{U}, \boldsymbol{V}) = \sum_{j=1}^{N} \sum_{i=1}^{K} (u_{ij})^q \, d^2 (\boldsymbol{X}_j, \boldsymbol{V}_i), \qquad K \leq N \tag{4.6}$$

In (4.6) $\boldsymbol{X}_j$ is the $j$th feature vector in $\boldsymbol{X}$, $\boldsymbol{V}_i$ is the $m$ dimensional centroid of the $i$th cluster,

**Algorithm 1** The k-Nearest Neighbours (kNN) clustering algorithm.

```
Input: X (N × m)

Define K, k
Normalize X
i = 1
Locate element y_i with greatest distance to V̄
V_i is the mean of y_i and its k nearest neighbours
Assign membership of y_i and its k nearest neighbours to V_i
Delete y_i and its k nearest neighbours from X
for i = 2 to K do
    Locate element y_i with greatest distance to V_{i−1}
    V_i is the mean of y_i and its k nearest neighbours
    Assign membership of y_i and its k nearest neighbours to V_i
    Delete y_i and its k nearest neighbours from X
    Increment i
end for
Assign membership of remaining elements in X to nearest cluster
Update clusters centers
```

$\mathrm{d}^2(\boldsymbol{X}_j, \boldsymbol{V}_i)$ is the Euclidian distance between $\boldsymbol{X}_j$ and $\boldsymbol{V}_i$, $u_{ij}$ is the degree of membership of $\boldsymbol{X}_j$ in the $i$th cluster. Moreover, $N$ is the number of data points and $K$ is the number of clusters. The parameter $q$ is the weighting exponent for $u_{ij}$ controlling the "fuzzyness" of the resulting clusters ($q$ is any real number greater than 1).

For the degree of membership, $u_{ij}$, the conditions (4.7), (4.8) and (4.9) hold.

$$u_{ij} \in \{0, 1\} \quad \forall i, j \tag{4.7}$$

$$\sum_{i=1}^{K} u_{ij} = 1 \quad \forall j \tag{4.8}$$

$$0 < \sum_{j=1}^{n} u_{ij} < n \quad \forall i \tag{4.9}$$

In [11] the fuzzy partitioning is carried out though an iterative optimization of (4.6). In each iteration (4.10) (cluster memberships) and (4.12) (updated cluster centroids) are calculated until the objective function (4.6) has converged. Prior to the iteration loops initial cluster prototypes, $\boldsymbol{V}_0$, are found either by drawing $K$ random points from $\boldsymbol{X}$ or by drawing $K$ random points from within the standard deviation, $\sigma$, of $\boldsymbol{X}$.

In the FCM clustering algorithm $\boldsymbol{U}$ is the $(N \times K)$ matrix holding all degrees of membership, $u_{ij}$, calculated in (4.10).

$$u_{ij} = \frac{\left(\frac{1}{d^2(\boldsymbol{X}_j, \boldsymbol{V}_i)}\right)^{\frac{1}{q-1}}}{\sum_{k=1}^{K}\left(\frac{1}{d^2(\boldsymbol{X}_j, \boldsymbol{V}_k)}\right)^{\frac{1}{q-1}}} \tag{4.10}$$

The Euclidian distance $d^2(\boldsymbol{X}_j, \boldsymbol{V}_k)$ measure is calculated as in (4.11) where $\boldsymbol{I}$ is a $(m \times m)$ dimensional identity matrix.

$$d^2(\boldsymbol{X}_j, \boldsymbol{V}_i) = (\boldsymbol{X}_j - \boldsymbol{V}_i)^{\mathrm{T}} \boldsymbol{I} (\boldsymbol{X}_j - \boldsymbol{V}_i) \tag{4.11}$$

The updated cluster centroids, $\hat{\boldsymbol{V}}_i$, are found according to (4.12).

$$\hat{\boldsymbol{V}}_i = \frac{\sum_{j=1}^{N}(u_{ij})^q \boldsymbol{X}_j}{\sum_{j=1}^{N}(u_{ij})^q} \tag{4.12}$$

Algorithm 2 sums up the Fuzzy C-Means algorithm.

**Algorithm 2** The Fuzzy C-Means (FCM) clustering algorithm.

1: Input: $\boldsymbol{X}$ $(N \times m)$
2:
3: Define $K$, $q$
4: Draw $K$ initial cluster centroids, $\boldsymbol{V}_0$, within the standard deviation, $\sigma$, of $\boldsymbol{X}$
5: Normalize $\boldsymbol{X}$
6: $i = 1$
7: **while** $J$ has not converged **do**
8: Update each $u_{ij}$ element in membership matrix $\boldsymbol{U}$ using (4.10)
9: Update cluster centroids, $\boldsymbol{V}$, using (4.12)
10: Calculate $i$th value of $J$ using (4.6)
11: Increment $i$
12: **end while**

### 4.1.3 Unsupervised Fuzzy Partition - Optimal Number of Classes

The Unsupervised Fuzzy Partition - Optimal Number of Classes (UFP-ONC) algorithm [12] is able to do unsupervised fuzzy partitioning of a data set $\boldsymbol{X}$ containing $N$ objects of $m$ dimensions each. By using two *segmentation validity criteria* (*fuzzy hypervolumen* and *partition density* - see 4.3.1 and 4.3.2) the UFP-ONC algorithm in [12] is able to determine an optimal number of clusters (segments) in a given data set - more information on validity criteria and their purpose in 4.3. Further, the UFP-ONC algorithm require a basic knowledge of Bayes' Theorem [13] which is defined as in (4.13).

$$P\left(\boldsymbol{V}_i|\boldsymbol{X}_j\right) = \frac{P\left(\boldsymbol{X}_j|\boldsymbol{V}_i\right)P\left(\boldsymbol{V}_i\right)}{P\left(\boldsymbol{X}_j\right)} \tag{4.13}$$

In (4.13) $P\left(\boldsymbol{V}_i|\boldsymbol{X}_j\right)$ is the *posterior* (the probability of selection the $i$th cluster given the $j$th feature vector, $\boldsymbol{X}_j$), $P\left(\boldsymbol{X}_j|\boldsymbol{V}_i\right)$ is the *likelihood* (the probability of selection the $j$th feature vector, $\boldsymbol{X}_j$, given the the $i$th cluster), $P\left(\boldsymbol{V}_i\right)$ is the *prior* (the probability of selection the $i$th cluster), $P\left(\boldsymbol{X}_j\right)$ is the *evidence* (the probability of the $j$th feature vector, $\boldsymbol{X}_j$). The denominator is a normalization factor so that (4.14) holds.

$$\sum_{i=1}^{K} P\left(\boldsymbol{V}_i|X_j\right) = 1 \tag{4.14}$$

Note that a slightly different notation of the posterior and prior is used in the further description of the UFP-ONC algorithm.

The UFP-ONC algorithm works pretty similarly to the FCM algorithm in terms of structure (calculate membership matrix $\boldsymbol{U}$, calculate cluster centers $\boldsymbol{V}$ with respect to $\boldsymbol{U}$, repeat). One important difference though, is the distance measure used in the UFP-ONC algorithm. The distance measure used in the FCM algorithm (and the kNN algorithm for that matter) disregards the important cluster properties shape/density by using an identity matrix in (4.11). These properties are considered in the distance measure in the UFP-ONC algorithm resulting in hyperellipsoidal clusters with variable densities. This "exponential" distance measure, $d_e^2\left(\boldsymbol{X}_j, \boldsymbol{V}_i\right)$, is based on *maximum likelihood maximization* [13] and is defined as in (4.16). The distance measure $d_e^2\left(\boldsymbol{X}_j, \boldsymbol{V}_i\right)$ is used in the calculation of $h\left(i|\boldsymbol{X}_j\right)$ which is the posterior probability and calculated as in (4.15).

$$h\left(i|\boldsymbol{X}_j\right) = \frac{\frac{1}{d_e^2\left(\boldsymbol{X}_j, \boldsymbol{V}_i\right)}}{\sum\limits_{k=1}^{K} \frac{1}{d_e^2\left(\boldsymbol{X}_j, \boldsymbol{V}_k\right)}} \tag{4.15}$$

$$d_e^2\left(\boldsymbol{X}_j, \boldsymbol{V}_i\right) = \frac{\left|\boldsymbol{F}_i\right|^{\frac{1}{2}}}{P_i} \exp\left(\frac{\left(\boldsymbol{X}_j - \boldsymbol{V}_i\right)^{\mathrm{T}} \boldsymbol{F}_i^{-1} \left(\boldsymbol{X}_j - \boldsymbol{V}_i\right)}{2}\right) \tag{4.16}$$

In (4.16) $\left(\boldsymbol{X}_j - \boldsymbol{V}_i\right)^{\mathrm{T}} \boldsymbol{F}_i^{-1} \left(\boldsymbol{X}_j - \boldsymbol{V}_i\right)$ is the *Mahalanobis* distance, $|\boldsymbol{F}_i|$ is the determinant of the $(m \times m)$ covariance matrix (defined in (4.18)) and $P_i$ is the prior probability of selection the $i$th cluster and is defined as in (4.17).

$$P_i = \frac{1}{N} \sum_{j=1}^{N} h\left(i|\boldsymbol{X}_j\right) \tag{4.17}$$

$$\boldsymbol{F}_i = \frac{\sum\limits_{j=1}^{N} h\left(i|\boldsymbol{X}_j\right)\left(\boldsymbol{X}_j - \boldsymbol{V}_i\right)\left(\boldsymbol{X}_j - \boldsymbol{V}_i\right)^{\mathrm{T}}}{\sum\limits_{j=1}^{N} h\left(i|\boldsymbol{X}_j\right)} \tag{4.18}$$

Comparison of (4.15) and (4.10) shows that $h(i|\boldsymbol{X}_j)$ is similar to $u_{ij}$ when the weighting exponent is 2 ($q = 2$). Replacing (4.10) with (4.15) in the Fuzzy C-Means algorithm results in a fuzzy modification of the maximum likelihood estimation (*FMLE*). This fuzzy maximum likelihood estimation is the essence of the partitioning abilities of the UFP-ONC algorithm. Compared to the structure of the FCM algorithm, the FMLE iterations require calculating (4.17) and (4.18) before repeating each iteration. Moreover, an objective function for the FMLE algorithm is defined in (4.19).

$$J_{q,FMLE}(\boldsymbol{U}, \boldsymbol{V}) = \sum_{j=1}^{N} \sum_{i=1}^{K} (u_{ij})^q \, d_e^2(\boldsymbol{X}_j, \boldsymbol{V}_i), \qquad K \leq N \tag{4.19}$$

As Algorithm 3 shows, the UFP-ONC algorithm consists of two main layers - the first layer runs the FCM algorithm while the second layer does the fuzzy maximum likelihood estimation (FMLE). The reason for this structure of the UFP-ONC algorithm is the "exponential" distance measure (4.16) used in the FMLE algorithm. This "exponential" distance function only seeks a partition optimum within a narrow local region causing the FMLE algorithm to become unstable if not initiated from "descent" cluster prototypes. Therefore, the UFP-ONC algorithm needs the FCM algorithm in its first layer to generate descent cluster centroids that are used by the FMLE algorithm in its second layer, thus enabling the UFP-ONC algorithm to consistently obtain good partitions.

Algorithm 3 sums up the Unsupervised Fuzzy Partition - Optimal Number of Classes algorithm. The $K$ parameter states the maximum number of clusters in the range for which the UFC-ONC algorithm should search for an optimal number of clusters.

## 4.2 Evaluation of segmentation approaches

All of the clustering algorithms mentioned in the previous sections are able to perform segmentation of a given data set $\boldsymbol{X}$. However, in order to chose a consistent and capable clustering algorithm for processing gatetrade.net data a thorough evaluation of the algorithms is necessary.

Besides the three present clustering algorithms a fourth algorithm is introduced by adding one iteration of the FCM algorithm to the kNN algorithm, thus making it fuzzy [10]. This fourth algorithm will be referred to as the *Fuzzy k-Nearest Neighbour* clustering algorithm (FkNN). All algorithms have been implemented as described in this project using Matlab.

The four algorithms are evaluated on their ability to segment/classify four different test data sets only knowing the number of classes of the respective sets. A short description of the four test data sets follows.

**Fisher** - The Fisher iris data is a popular data set for evaluating clustering algorithms (also used in [12]). The data set consists of 150 4-dimensional data objects each describing petal/sepal lengths of 3 different iris flowers (50 data objects of each). Two of the three iris data distributions overlap which poses the classification challenge of this data set.

**Test1** - The Test1 data set is a 3-dimensional constructed data set which purpose is to test the algorithms' ability to classify data objects in separated Gaussian distributions with

**Algorithm 3** Unsupervised Fuzzy Partition - Optimal Number of Classes (UFP-ONC) clustering algorithm.

Input: $\boldsymbol{X}$ $(N \times m)$
Define $K$, $q = 2$
**for** $c = 2$ to $K$ **do**
  Draw $c$ initial cluster centroids, $\boldsymbol{V}_0$, within the standard deviation, $\sigma$, of $\boldsymbol{X}$
  Normalize $\boldsymbol{X}$
  $i = 1$
  **while** $J$ (4.6) has not converged (**FCM** layer) **do**
    Update each $u_{ij}$ element in membership matrix $\boldsymbol{U}$ using (4.10)
    Update cluster centroids, $\boldsymbol{V}$, using (4.12)
    Calculate $i$th value of $J$ using (4.6)
    Increment $i$
  **end while**
  Calculate clusters priors (4.17) and covariances (4.18) based on FCM cluster centroids
  $i = 1$
  **while** $J$ (4.19) has not converged (**FMLE** layer) **do**
    Update each $u_{ij}$ element in membership matrix $\boldsymbol{U}$ using (4.15)
    Update cluster centroids, $\boldsymbol{V}$, using (4.12)
    Calculate clusters priors (4.17) and covariances (4.18)
    Calculate $i$th value of $J$ using (4.19)
    Increment $i$
  **end while**
  Calculate cluster validity criteria for the $c$ cluster partitioning
**end for**
The optimal number of clusters are found with respect to the cluster validity criteria

widely different number of members each. Test1 data set contains 2200 data objects with 4 distributions of 300, 900, 600 and 400 members each.

**Test2** - The Test2 data set is a 3-dimensional constructed data set which purpose is to test the algorithms' ability to classify data objects in overlapping Gaussian distributions. Test2 data set contains 2200 data objects with 4 distributions of 300, 900, 600 and 400 members each.

**Test3** - The Test3 data set is a 3-dimensional constructed data set which purpose is to test the algorithms' ability to classify data objects in overlapping Gaussian distributions with various shapes/densities. Test3 data set contains 2200 data objects with 4 distributions of 300, 900, 600 and 400 members each.

As described, the four data sets pose widely different data set scenarios of various classification difficulty levels. The four algorithms are evaluated on average values of classification error/speed based on 20 runs for each scenario. Table 4.2 shows the results of the evaluation runs. Figures of the true classification and the classifications of the four algorithms of the four data sets are found in Appendix B.

| Data set \ Algorithm | **kNN** | **FkNN** | **FCM** | **UFP-ONC** |
|---|---|---|---|---|
| **Fisher iris data** | | | | |
| Number of misclassifications | 33.0 | 35.0 | 24.27 | 2.48 |
| Error (%) | 22.0 | 23.33 | 16.18 | 1.65 |
| Algorithm runtime (s) | 0.0396 | 0.0514 | 0.0645 | 0.1092 |
| **Test set 1** | | | | |
| Number of misclassifications | 152.0 | 6 | 3.13 | 1.3 |
| Error (%) | 6.91 | 2.73 | 0.14 | 0.06 |
| Algorithm runtime (s) | 0.1437 | 0.2948 | 0.3406 | 0.4932 |
| **Test set 2** | | | | |
| Number of misclassifications | 715 | 689 | 277.2 | 171.3 |
| Error (%) | 32.50 | 31.32 | 12.6 | 7.79 |
| Algorithm runtime (s) | 0.1406 | 0.3203 | 0.4813 | 2.3208 |
| **Test set 3** | | | | |
| Number of misclassifications | 524 | 352 | 679.3 | 144.4 |
| Error (%) | 23.82 | 16.0 | 30.88 | 6.56 |
| Algorithm runtime (s) | 0.1443 | 0.3240 | 0.3750 | 0.9854 |

**Table 4.2:** Evaluation of the four clustering algorithms.

### 4.2.1 Conclusion of the cluster algorithms evaluation

The evaluation runs of the four clustering algorithms make it possible to distinguish one algorithm from the others in terms of classification error/speed. Table 4.2 shows one general tendency though - the classic tradeoff between classification performance and speed. The UFP-ONC algorithm is an example of this tradeoff. It is a sophisticated multilayered algorithm with a high level of classification performance (due to the FMLE layer), but has a high runtime. The high runtime is mainly because of the two layers (FCM and FMLE) of the UFP-ONC algorithm that both need to converge. Figure 4.2 shows the convergence of both layers of the UFP-ONC algorithm.

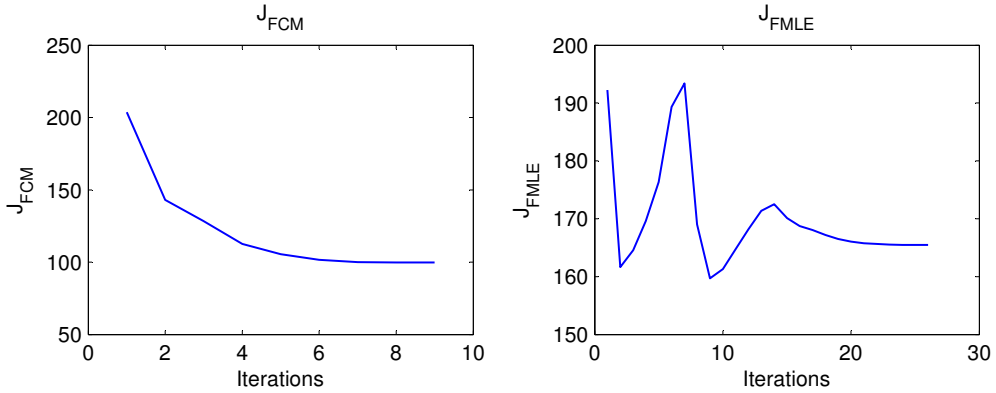


**Figure 4.2:** Convergence of the UFP-ONC algorithm for the Fisher data set.

As mentioned in section 4.1.3 the FMLE layer of the UFP-ONC algorithm requires a good initial partitioning because it searches for an optimal (maximum likelihood) clustering within

a narrow local region. This is supported by the fluttering objective function, $J_{q,FMLE}$, in figure 4.2. In contrast to $J_{q,FMLE}$ the FCM objective function , $J_{q,FCM}$, converges steadily.

| Algorithm | **Disadvantages** | **Advantages** |
|---|---|---|
| **kNN** | High error rate | Low complexity<br>High speed<br>High consistency |
| **FkNN** | Relative high error rate | Medium speed<br>High consistency |
| **FCM** | Reliant on fair initial cluster prototypes | Medium error rate |
| **UFP-ONC** | Low speed<br>Relative high complexity<br>Reliant on fair initial cluster prototypes | Low error rate |

**Table 4.3:** General advantages and disadvantages of the four clustering algorithms.

Moreover, the advantages and the disadvantages of the four different clustering algorithms are summed up in table 4.3.

Table 4.3 makes it clear that the main advantages of the kNN/FkNN algorithms are their high speed and consistent performance (due to their strict way of locating new cluster centroids). Even though these two algorithms are fast at reaching the same classification result for each run, their error rate is dissatisfying compared to the ones of the iterative FCM and UFP-ONC algorithms. Generally, the kNN algorithm benefits from the fuzzy modification (FkNN) in terms of classification performance.

When speed is disregarded, the UFP-ONC clustering algorithm is far superior to the other three algorithms. Although the FCM algorithm shares the UFP-ONC algorithm's iterative structure, its distance measure does not meet the demand of doing a good partitioning of a data set of distributions of various shapes/densities. This makes the UFP-ONC clustering/segmentation algorithm the favourite algorithm choice for the processing of the gatetrade.net data.

An interesting study (although out of the current project's scope) would be to combine the UFP-ONC and the FkNN algorithm in the search for a faster and more consistent version of the UFP-ONC algorithm. This potential algorithm should use the FkNN algorithm to consistently generate good initial cluster centroids for the UFP-ONC algorithm, thus making the FCM layer of the UFP-ONC algorithm unnecessary.

## 4.3 Segmentation validity criteria

In the evaluation section of the four clustering algorithms' classification ability, the number of clusters/segments was known in advance for each of the four different data sets. This is not the case for the gatetrade.net data for which the number of subgroups, the density of the subgroups and their distributions are all unknown. So in order to do proper unsupervised segmentation/classification of the gatetrade.net data, a number of segmentation validity criteria must be taking into consideration.

A segmentation validity criterion is a measure used to relatively judge the number of subgroups in a data set with respect to a given criterion (e.g. clear separation between the clusters, minimal volume of the clusters, maximal number of data points in the vicinity of cluster centroids). The criteria are used when running a clustering algorithm over a specified range of numbers of clusters. By measuring each criterion for each number of clusters in the range it is possible to generate validity graphs for each of the criteria. Local minima/maxima on the validity graphs suggest an optimal (or suitable) number of clusters depending on the type of validity criterion.

For the task of segmenting Marketplace/eProcurement data, 10 various segmentation validity criteria have been chosen. The following sections briefly comment/define each of the 10 criteria.

### 4.3.1 Fuzzy hypervolume criterion

The Fuzzy hybervolume criterion [12], $\upsilon_{FHV}$, expresses an accumulated volume measure for all clusters in a given partitioning. The criterion is defined as in (4.20).

$$\upsilon_{FHV} = \sum_{i=1}^{K} |\boldsymbol{F}_i|^{\frac{1}{2}} \tag{4.20}$$

The Fuzzy hybervolume criterion suggests an optimal segmentation for local minima on its validity graph.

### 4.3.2 Partition density criterion

The Partition density criterion [12], $\upsilon_{PD}$, expresses an accumulated density measure of the central members of all clusters in a given partitioning. The criterion is defined as in (4.21).

$$\upsilon_{PD} = \frac{S}{\upsilon_{FHV}} \tag{4.21}$$

In (4.21) $S$ is the sum of central members (4.22).

$$S = \sum_{i=1}^{K} \sum_{j=1}^{N} u_{ij}, \quad \forall \boldsymbol{X}_j \in \left\{ \boldsymbol{X}_j : (\boldsymbol{X}_j - \boldsymbol{V}_i)^{\mathrm{T}} \boldsymbol{F}_i^{-1} (\boldsymbol{X}_j - \boldsymbol{V}_i) < 1 \right\} \tag{4.22}$$

The Partition density criterion suggests an optimal segmentation for local maxima on its validity graph.

### 4.3.3 Calinski and Harabasz criterion

The Calinski and Harabasz criterion [14,15], $\upsilon_{CH}$, expresses a compactness ration with respect to $\boldsymbol{B}$ (inter-cluster scatter matrix) and $\boldsymbol{W}$ (intra-cluster scatter matrix) for all clusters in a given partitioning. The criterion is defined as in (4.23).

$$\upsilon_{CH} = \frac{\frac{trace(\boldsymbol{B})}{K-1}}{\frac{trace(\boldsymbol{W})}{N-K}},\ where\ \boldsymbol{B} = (\boldsymbol{UV})^{\mathrm{T}}(\boldsymbol{UV})\ and\ \boldsymbol{W} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{B} \tag{4.23}$$

The Calinski and Harabasz criterion suggests an optimal segmentation for local maxima on its validity graph.

### 4.3.4 Fuzziness performance criterion

The Fuzziness performance criterion [16], $\upsilon_{FPI}$, expresses the degree of fuzziness in a given partitioning. The criterion is defined as in (4.24).

$$\upsilon_{FPI} = 1 - \frac{KH-1}{K-1}, \qquad where \quad H = \frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N}(u_{ij})^2 \tag{4.24}$$

The fuzziness performance criterion suggests an optimal segmentation for local minima on its validity graph.

### 4.3.5 Fukuyama and Sugeno criterion

The Fukuyama and Sugeno criterion [17], $\upsilon_{FS}$, expresses the difference of two terms, $J_q$ (combination of the fuzziness degree in the membership matrix and the geometrical compactness of the data with respect to the cluster centroids (objective function)) and $K_q$ (combination of the fuzziness degree for a given cluster centroid and its distance to the global mean ($\bar{\boldsymbol{V}}$) of the data set). The criterion is defined as in (4.24).

$$\upsilon_{FS} = \sum_{i=1}^{K}\sum_{j=1}^{N}(u_{ij})^q\left(d^2\left(\boldsymbol{X}_j, \boldsymbol{V}_i\right) - d^2\left(\boldsymbol{V}_i, \bar{\boldsymbol{V}}\right)\right) = J_q - K_q \tag{4.25}$$

$$K_q = \sum_{i=1}^{K}\sum_{j=1}^{N}(u_{ij})^q\, d^2\left(\boldsymbol{V}_i, \bar{\boldsymbol{V}}\right) \tag{4.26}$$

The Fukuyama and Sugeno criterion suggests an optimal segmentation for local minima on its validity graph.

### 4.3.6 Normalized classification entropy criterion

The normalized classification criterion [16], $\upsilon_{NCE}$, expresses the degree of disorganization in a given partitioning. The criterion is defined as in (4.27).

$$\upsilon_{NCE} = \frac{Y}{\log K}, \qquad where \quad Y = \frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N} u_{ij} \times \log\left(u_{ij}\right) \tag{4.27}$$

The normalized classification entropy criterion suggests an optimal segmentation for local minima on its validity graph.

### 4.3.7 Partition coefficient criterion

The normalized classification criterion [18], $\upsilon_{PC}$, expresses the degree of fuzziness in a given partitioning without considering the data itself. The criterion is defined as in (4.28).

$$\upsilon_{PC} = \frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N} u_{ij}^2 \tag{4.28}$$

The partition coefficient criterion suggests an optimal segmentation for local maxima on its validity graph.

### 4.3.8 Partition entropy criterion

The partition entropy criterion [19], $\upsilon_{PE}$, expresses the degree of fuzziness entropy in a given partitioning without considering the data itself. The criterion is defined as in (4.29).

$$\upsilon_{PE} = -\frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N} u_{ij} \times \log\left(u_{ij}\right) \tag{4.29}$$

The partition entropy criterion suggests an optimal segmentation for local minima on its validity graph.

### 4.3.9 Proportion exponent criterion

The proportion exponent criterion [20], $\upsilon_{PEX}$, expresses another fuzziness measure in a given partitioning also without considering the data itself. Compared to the partition coefficient and the partition entropy criteria the proportion exponent criterion is better at detecting structural variations when the membership matrix grows fuzzier. The criterion is defined as in (4.30).

$$\upsilon_{PEX} = -\sum_{j=1}^{N} \ln \left[ \sum_{n=1}^{\lfloor u_j^{-1} \rfloor} (-1)^{n+1} \binom{K}{n} (1 - n \cdot u_j)^{K-1} \right] \tag{4.30}$$

In (4.30) $u_j$ is defined as in (4.31)

$$u_j = \max_{1 \le i \le K} \{u_{ij}\} \tag{4.31}$$

The proportion exponent criterion suggests an optimal segmentation for local maxima on its validity graph.

### 4.3.10 Xie and Beni criterion

The Xie and Beni criterion [21,22], $\upsilon_{XB}$, expresses a ratio of fuzzy compactness (numerator) and the separation of the two closest cluster centroids (denominator) in a given partitioning. The criterion is defined as in (4.32).

$$\upsilon_{XB} = \frac{\sum_{i=1}^{K} \sum_{j=1}^{N} u_{ij}^{q} d^2 (\boldsymbol{X}_j, \boldsymbol{V}_i)}{N \left( \min_{i \neq j} d^2 (\boldsymbol{V}_i, \boldsymbol{V}_j) \right)} \tag{4.32}$$

The Xie and Beni criterion suggests an optimal segmentation (compact (low numerator) and separated (high denominator) clusters) for local minima on its validity graph.

## 4.4 Evaluation of segmentation validity criteria

Basic evaluation and test of the ten mentioned segmentation validity criteria is necessary in order to find out which criteria to trust when dealing with more complex data sets (overlapping distributions of various shapes/densities).

For this evaluation purpose the Test1, Test2, Test3 data sets are used (all containing 4 distributions). The ten cluster validity criteria are implemented into the UFP-ONC algorithm and are all computed after both layers of the algorithm have converged. Moreover, the UFP-ONC algorithm is for each data set run 20 times starting at 2 clusters and stopping at 7 clusters for each run. Figure 4.3, 4.4 and 4.5 show the resulting validity graphs of the respective criteria.

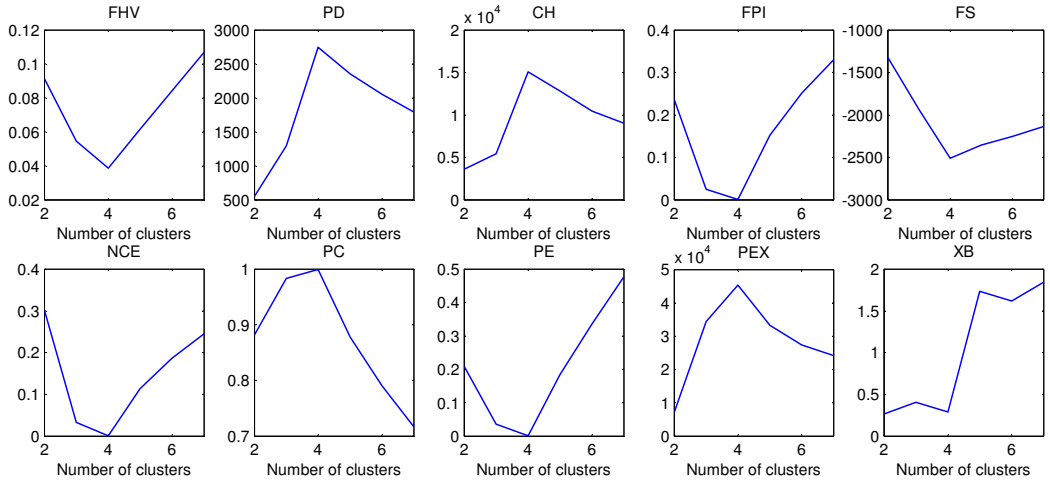


**Figure 4.3:** Optimal number of clusters in Test1 data set using the 10 cluster validity criteria.

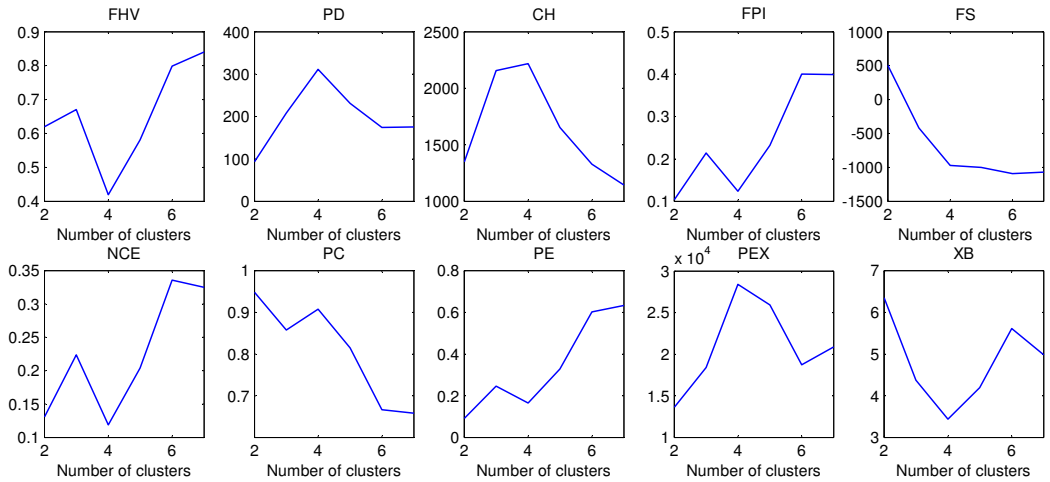


**Figure 4.4:** Optimal number of clusters in Test2 data set using the 10 cluster validity criteria.

In figure 4.3 all ten cluster validity criteria are seen to be able to suggest the correct number of clusters in the Test1 data set - all the criteria have their respective optimal condition, local minima/maxima, at 4 clusters.

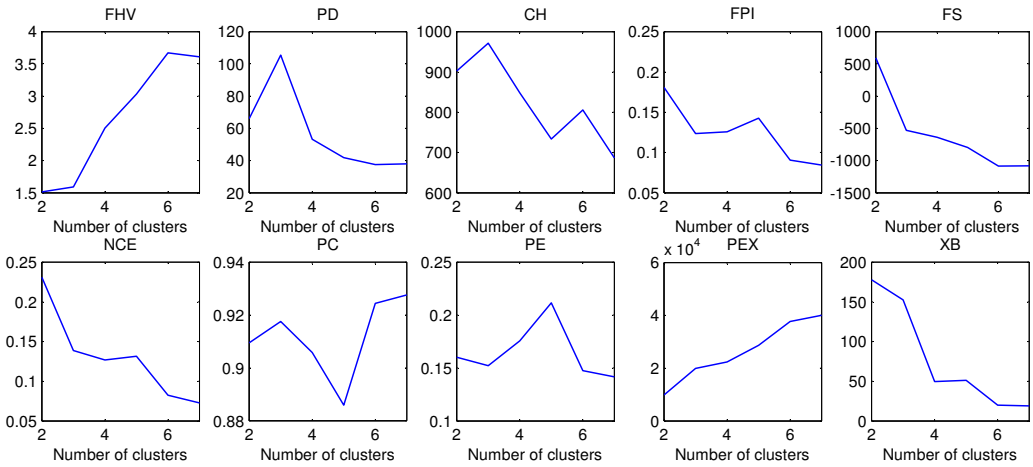


**Figure 4.5:** Optimal number of clusters in Test3 data set using the 10 cluster validity criteria.

In figure 4.4 nine of the ten cluster validity criteria are seen to be able to suggest the correct number of clusters in the Test2 data set that is more complex than the Test1 data set. Although close, the Fukuyama and Sugeno (FS) validity criterion does not produce a local minima for 4 clusters.

In figure 4.5 only two of the ten cluster validity criteria are seen to be able to suggest the correct number of clusters in the Test3 data set that again is more complex than the Test2 data set. The normalized classification entropy (NCE) and the Xie and Beni (XB) validity criteria are seen to suggest local minima for 4 clusters. It should further be noticed that the remaining criteria produce various different guesses.

From the validity criteria evaluation it may be concluded that all of the ten cluster validity criteria work/perform independently. The only two criteria, successfully suggesting the correct number of clusters for all three test data sets, were the normalized classification entropy (NCE) and the Xie and Beni (XB) validity criteria. These two criteria should be given extra weight when determining (unsupervised) the number of segments/clusters in the gatetrade.net data sets.

Chapter 5

# Applied Data Mining

The preceding chapters have theoretically explained the basic concepts of Business Intelligence and have covered Data Mining methods in more detail along with the Knowledge Discovery in Databases process. The task of applying Data Mining methods to the gatetrade.net data remains and is described in the following chapters.

## 5.1 Deploying the KDD process

It makes good sense to use Fayyad's KDD process (chapter 2.3) in the complex task of exploring gatetrade.net data and extracting useful information from it. Before the different steps of the KDD process are carried out it is important to have clear goals/intentions of what the applied Data Mining should accomplish with respect to the gatetrade.net data. These goals are in this project each of the six desired data processing applications in table 3.1. Each of the six goals can be individually processed using the five steps in the KDD process, but since all of the goals rely on the same data source, it makes good sense to combine the six individual KKD processes for step 1 and 2 (selecting, cleaning/preprocessing the data). Step 3, 4 and 5 of the KDD process is deployed individually for the six desired data processing applications.

## 5.2 Preprocessing gatetrade.net data

As mentioned in chapter 3, gatetrade.net data attributes were extracted directly from the data depots using Excel format. In order to import the extracted data to Matlab, further formatting had to be carried out (e.g. formatting dates of type dd-mm-yyyy to ymmdd, formatting Danish letters (æ, ø and å), comma/period formatting).

When formatting large amounts of data, a custom-made tool for the purpose is needed. For this project a *Data Formatting Framework* was developed in `C#` using Visual Studio 2005. The framework consists of four general classes (`pom_orderhead`, `pom_orderline`, `eproc_invoicehead` and `eproc_invoiceline`) that each contains formatting methods for the different attributes. A class diagram of the Data Formatting Framework is found in Appendix C.

## 5.3 Profiling of buyers and suppliers in Marketplace/e-Procurement

### 5.3.1 Selecting proper data attributes

The task of profiling buyers and suppliers in Market/eProcument requires the general transactional data, which are found in the `POM_ORDER_HEADERS` and the `eProc_INVOICE` databases.

In the `POM_ORDER_HEADERS` database the following attributes are of interest for this particular task.

- `ORDER_TOTAL`
- `BUYER_ID`
- `BUYER_NAME`
- `BUYING_ORG_ID`
- `BUYING_ORG_NAME`
- `SUPPLIER_ID`
- `SUPPLIER_NAME`
- `PURCHASE_RATE_DATE`

In the `eProc_INVOICE` database the following attributes are of interest for this particular task.

- `invoice_total`
- `created_time`
- `updated_user_id`
- `company`
- `sender_cvr_number`

All of the attributes are formatted using the Data Formatting Framework described in section 5.2.

### 5.3.2 Transformation and reduction of the preprocessed data

The character of the formatted Marketplace/eProcurement data attributes is entirely transactional. This data structure is inconvenient for further processing and should be transformed into a dimensional data structure.

For both Marketplace and eProcurement a buyer and a supplier dimensional data set is made containing $N$ rows (number of buyers/suppliers for the respective system) and $m$ main attributes. The four main attributes per buyer/supplier are: "Number of orders", "Totals summation", "Order total average" and "Number of orders per day".

The resulting four data sets are named,
1. `pom_orderhead_buyer_stats`
2. `pom_orderhead_supplier_stats`
3. `eproc_invoice_buyer_stats`
4. `eproc_invoice_supplier_stats`

For both Marketplace/eProcurement data sets a smaller number (considering the total number of buyers/suppliers) of top buyers/suppliers exist. gatetrade.net has a comprehensive knowledge of these top users which is why they are considered to be outliers in the data sets and are filtered away. Figure 5.1 shows a histogram of Marketplace buyer outliers with respect to the attribute "Totals summation" (order totals summarized).

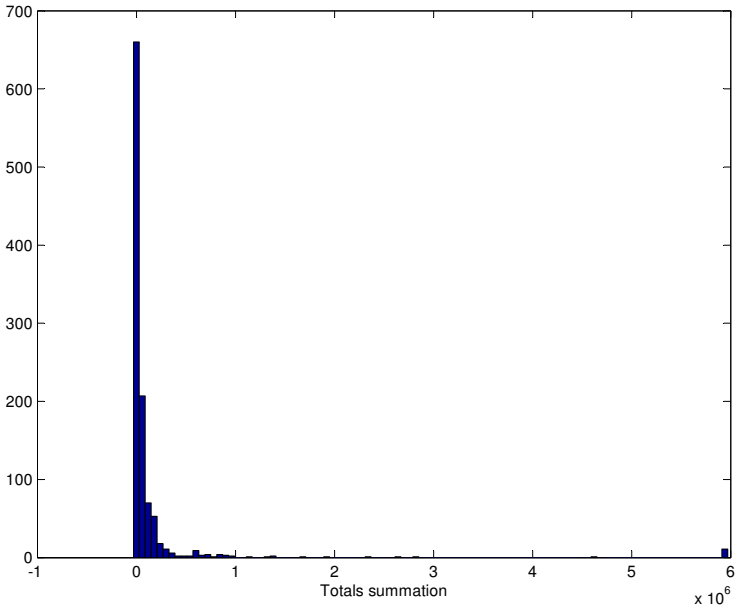


**Figure 5.1:** Marketplace buyer histogram of the attribute, "Totals summation", in DKK.

### 5.3.3 Applying Data Mining to the transformed data

The UFP-ONC clustering algorithm is used to do a segmentation analysis of the four filtered data sets (buyers and suppliers in Marketplace/eProcurement) of dimensional structure.

Optimal number of clusters are chosen with respect to the ten cluster validity criteria from chapter 4.

For each of the data sets, the four main attributes are used as input for the UFP-ONC algorithm and are listed as below.

1. ”Number of orders”
2. ”Totals summation”
3. ”Order total average”
4. ”Number of orders per day”

Even though some of the attributes (attribute 1, 2 and 3) depend on each other, they still make sense to use in the segmentation process due to the following facts.

- Users having large ”Order total average” values may have small ”Totals summation” values or small ”Number of orders” values.

- Users having large ”Number of orders” values may have small ”Totals summation values” or small ”Order total average” values.

- Users having large ”Totals summation” values may have small ”Number of orders” values or small ”Order total average” values.

The UFP-ONC clustering algorithm was run 20 times for each data set with number of clusters ranging from 2 to 15 clusters.

### 5.3.4 Evaluation of Data Mining results

The segmentation of the four data sets was carried out as explained in the previous section. Due to the large number of segmentation results, this chapter will mainly comment on the Marketplace buyer segmentations results. Figures containing cluster validity graphs and segmentation results of the other three data sets are found in Appendix D.

Figure 5.2 shows the the cluster validity graph of the UFP-ONC algorithm for the Marketplace buyer data set. For the most cases, the local extrema of the ten cluster validity criteria (figure 5.3) suggest an optimal number of clusters for six clusters.

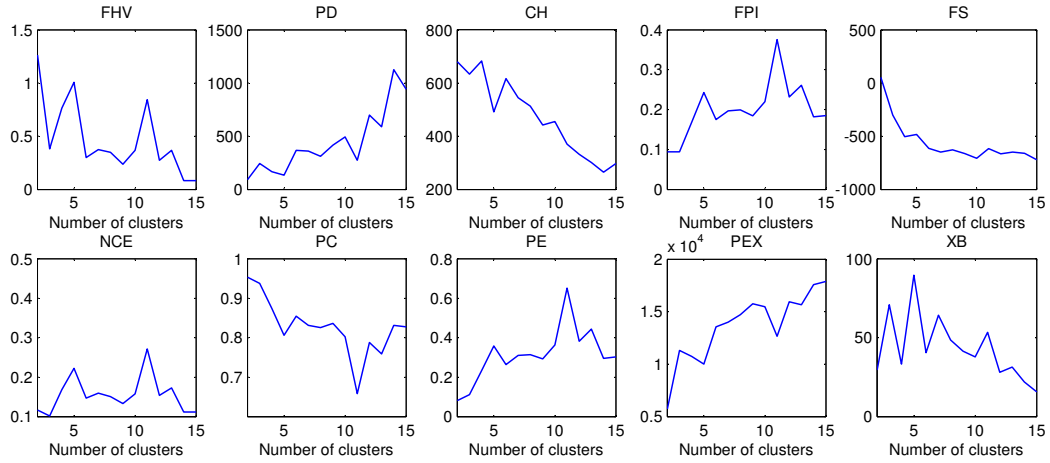


**Figure 5.2:** Optimal number of clusters in Marketplace buyer data.

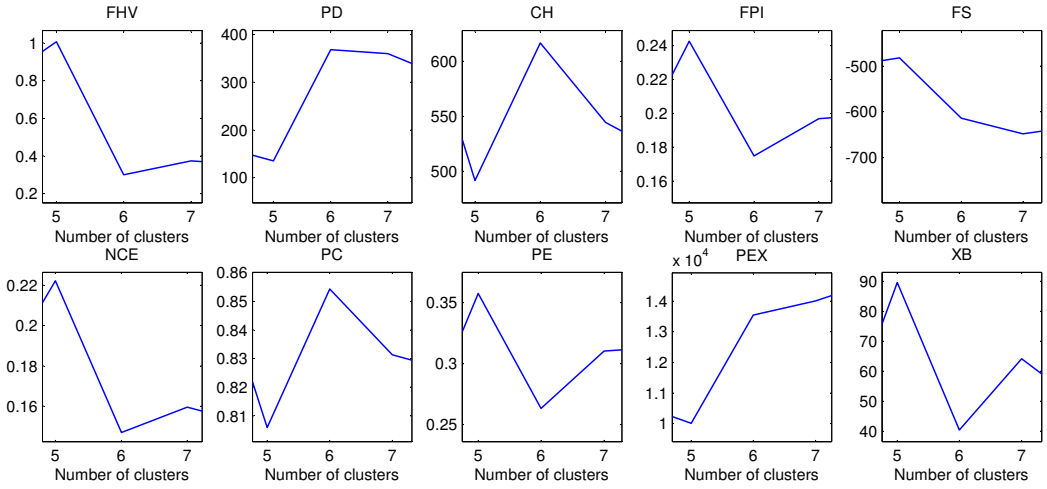


**Figure 5.3:** Local extrema near the optimal number of clusters in Marketplace buyer data.

Figure 5.4 shows the six different segments of the Marketplace buyer data set. The centroids of the six segments along with membership links are shown in figure 5.5 (a similar graph using different attributes is shown in figure 5.6).

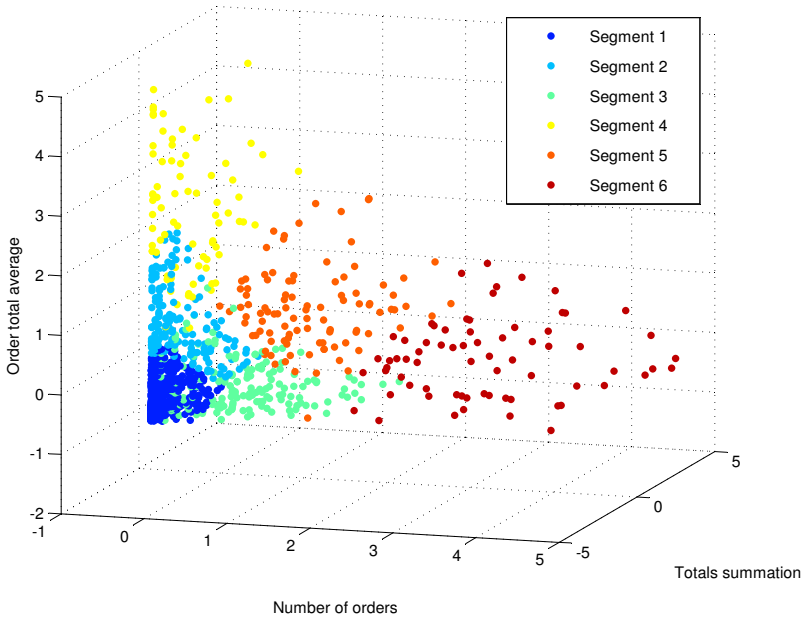


**Figure 5.4:** Segmentation of Marketplace buyer data.

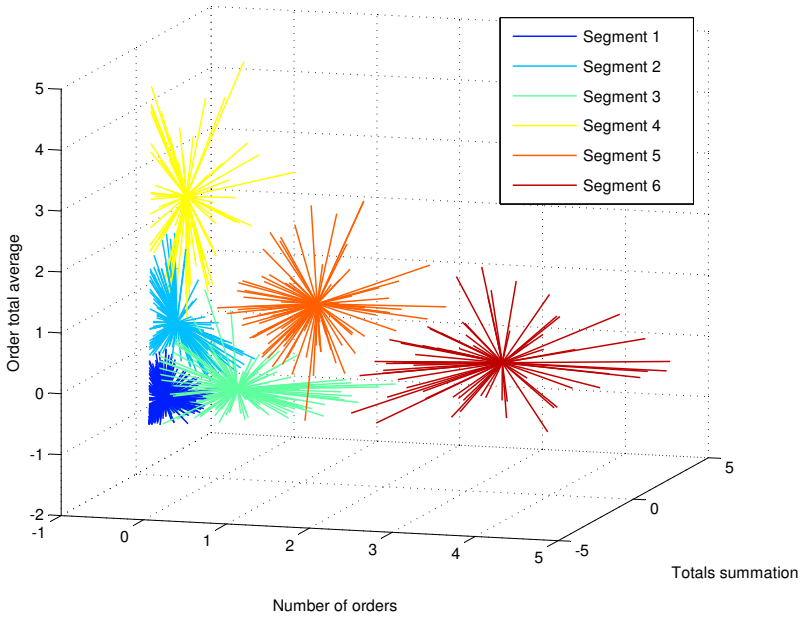


**Figure 5.5:** Marketplace buyer data linked to their respective cluster centroids.

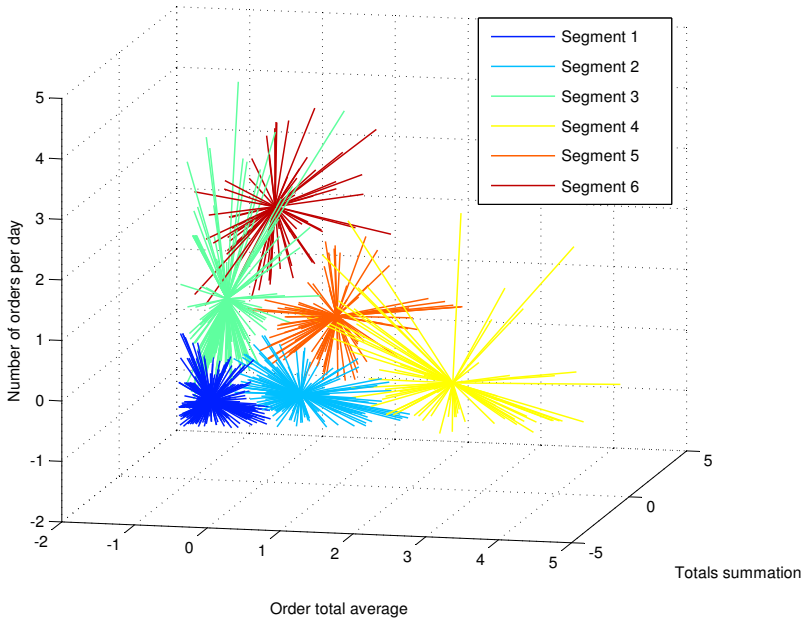


**Figure 5.6:** Marketplace buyer data linked to their respective cluster centroids.

The segmentation of the Marketplace buyer data set defines six clear buyer segments. Each of the segments represents a certain buyer profile with respect to the four used attributes. Table 5.1 shows the six different centroids of the segments.

| Centroid\Attribute | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 5.8768 | 4,432.7 | 671.41 | 0.041605 |
| 2 | 9.3807 | 20,169 | 2,535 | 0.052066 |
| 3 | 30.89 | 26,760 | 918.59 | 0.17747 |
| 4 | 9.4275 | 50,460 | 5,686.6 | 0.069083 |
| 5 | 48.853 | 133,010 | 2,811.7 | 0.13242 |
| 6 | 113.27 | 154,050 | 1,389.9 | 0.27319 |

**Table 5.1:** The six cluster centroids of the Marketplace buyer segmentation

To get a better overview of the six buyer types, table 5.2 lists each attribute using high, medium or low values.

| Centroid\Attribute | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | low | low | low | low |
| 2 | low | medium | medium | low |
| 3 | medium | medium | low | high |
| 4 | low | medium | high | medium |
| 5 | medium | high | high | medium |
| 6 | high | high | medium | high |

**Table 5.2:** The six cluster centroids of the Marketplace buyer segmentation

Table 5.2 shows that the centroids of the six clusters are all very different with respect to

the four dimensions. The centroids are ranked in terms of buyer performance. Members of segment 1 rarely use the Marketplace system and when they do, the placed order is of a small amount. In contrast to this, members of segment 6 place orders of large amounts on a regular basis.

Similar analysis is possible for eProcurement buyers/suppliers and Marketplace suppliers.

The segmentation analysis, carried out for profiling buyers and suppliers in Marketplace/e-Procurement, has many purposes and applications. Some of them are listed below.

1. The segmentations analysis of buyers and suppliers in Marketplace/eProcurement is great for profiling current users given a number of main profiles, thus getting deeper knowledge about the habits of the individual user.
2. The segments found by the UFP-ONC algorithm can be used to profile new buyers and suppliers in order to compare their performance with the ones of current users.
3. The segmentation results may also come in handy for special cases. Say gatetrade.net has the option of inviting 100 of their Marketplace buyers on a seminar designed to help the buyers use the Marketplace system and make them more familiar with the system. The segmentation results clearly suggests buyers of segment 1 and 2 as potential participants since they rarely use the Marketplace system and therefore have the need for such a training.

The UFP-ONC algorithm may also be used to profile the buyers and suppliers in terms of their monthly development. This process is described in sections 5.4 and 5.5.

## 5.4 Examine if trade is canalized through top buyers in Marketplace

### 5.4.1 Selecting proper data attributes

The task of examining if trade is canalized through top buyers in Marketplace (leaving a number of inactive buyers) requires the general transactional data, which are found in the `POM_ORDER_HEADERS` database.

In the `POM_ORDER_HEADERS` database the following attributes are of interest for this particular task.

- `ORDER_TOTAL`
- `BUYER_ID`
- `BUYER_NAME`
- `BUYING_ORG_ID`
- `BUYING_ORG_NAME`
- `SUPPLIER_ID`
- `SUPPLIER_NAME`
- `PURCHASE_RATE_DATE`

All of the attributes are formatted using the Data Formatting Framework described in section 5.2.

### 5.4.2 Transformation and reduction of the preprocessed data

The character of the formatted Marketplace data attributes is entirely transactional. This data structure is inconvenient for further processing and should be transformed into a dimensional data structure.

A dimensional buyer data set for the monthly progress of the four main attributes (”Number of orders”, ”Totals summation”, ”Order total average” and ”Number of orders per day”) is needed. This data set should besides the monthly progress also contain a monthly average of the four main attributes. The data set is called, `pom_orderhead_buyer_stats_monthly`.

The dimensional data set, `pom_orderhead_buyer_stats`, of total buyer statistics for all 15 months (mentioned in section 5.3.2) is also used for this particular task.

### 5.4.3 Applying Data Mining to the transformed data

In order to examine if trade in Marketplace is canalized through top buyers, it makes good sense to search for inactive or lost buyers in the different companies. A segmentation of the monthly progress of the buyers is therefore needed. This segmentation is carried out for the "Totals summation" attribute since the individual buyer progress of this attribute indicate their present state.

The UFP-ONC algorithm (supported by the ten cluster validity criteria) was used for this segmentation using the average progress of the "Totals summation" attribute for the four last months of the data set period (December 2005 - March 2006). The clustering algorithm was run 20 times with number of clusters ranging from 2 to 15 clusters.

### 5.4.4 Evaluation of Data Mining results

The segmentation of the average monthly progress data set was carried out as explained in the previous section. Figure 5.7 shows the the cluster validity graph of the UFP-ONC algorithm for the monthly progress of the "Totals summation" attribute. For the most cases, the local extrema of the ten cluster validity criteria (figure 5.8) suggest an optimal number of clusters for five clusters.

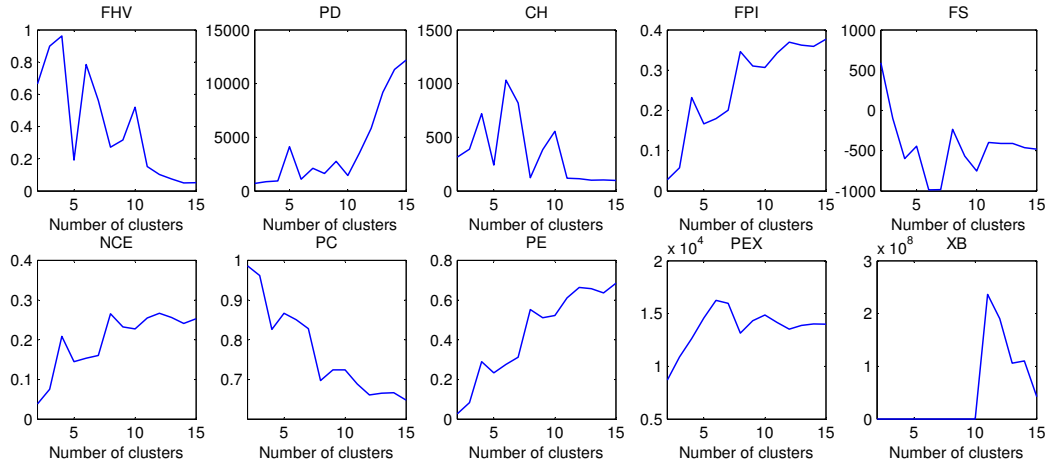


**Figure 5.7:** Optimal number of clusters for the last four months of Marketplace buyer data.

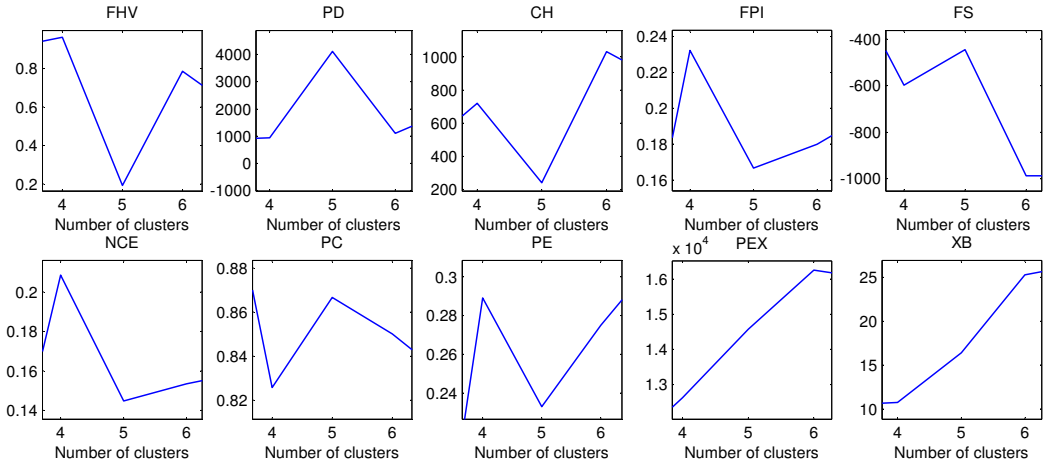


**Figure 5.8:** Local extrema near the optimal number of clusters.

Figure 5.9 shows the five resulting clusters centroids (progress trends in this case) of the progress of the "Totals summation" attribute. Because each dimension represents a month, it does not make sense to plot the centroids in a dimensional graph. Instead, each centroid is presented in its own plot as a progress over the four last months. Since all rows in the monthly progress data set have been divided with the max value of the individual rows, the y-axis in figure 5.9 ranges from 0 to 1.

Each of the five progress segments in figure 5.9 represents different buyer progress trends.

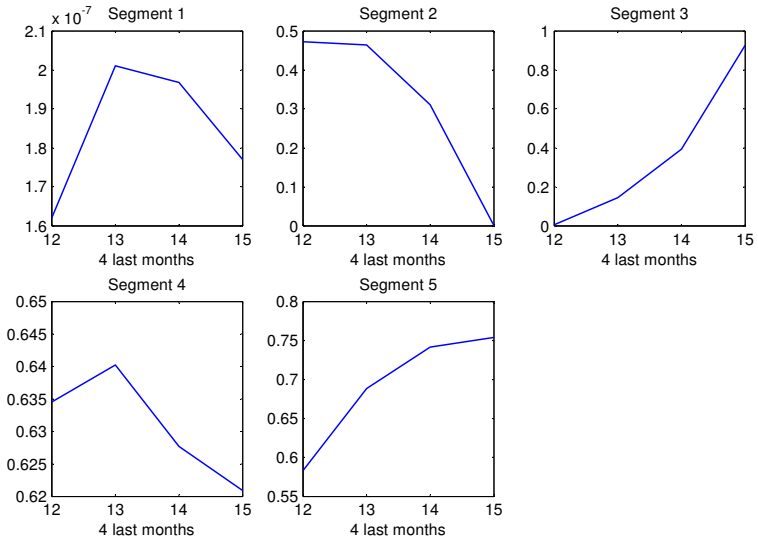


**Figure 5.9:** The five cluster centroids of monthly Marketplace buyer progress.

The trends of segment 3, 4 and 5 are flat (with a high offset) or positive, thus members of these segments have a healthy progress for the given period. On the other hand, the trends of segment 1 and 2 are flat (no offset) or negative, thus indicating lost buyers and buyers with unhealthy progresses respectively.

The members of segment 1 and 2 in figure 5.9 are therefore declared lost or inactive. By comparing the statistical data (in the `pom_orderhead_buyer_stats` data set) of these inactive buyers with other buyers in their respective companies, a list of companies containing inactive buyers (canalized trading) can be produced.

The potential in this type of analysis is being able to spot inactive buyers in a given company in advance and take precautions if needed.

## 5.5 Analysis of lost suppliers in Marketplace

### 5.5.1 Selecting proper data attributes

The task of analyzing lost suppliers in Marketplace requires the general transactional data, which are found in the `POM_ORDER_HEADERS` database along with order line data found in `POM_ORDER_LINES` database.

In the `POM_ORDER_HEADERS` database the following attributes are of interest for this particular task.

- `ORDER_TOTAL`
- `BUYER_ID`
- `BUYER_NAME`
- `BUYING_ORG_ID`
- `BUYING_ORG_NAME`
- `SUPPLIER_ID`
- `SUPPLIER_NAME`
- `PURCHASE_RATE_DATE`

In the `POM_ORDER_LINES` database the following attribute is of interest for this particular task.

- `SUPPLIER_ITEM_NUMBER`

All of the attributes are formatted using the Data Formatting Framework described in section 5.2.

### 5.5.2 Transformation and reduction of the preprocessed data

The character of the formatted Marketplace data attributes is entirely transactional. This data structure is inconvenient for further processing and should be transformed into a dimensional data structure.

A dimensional supplier data set for the monthly progress of the five main attributes (”Number of orders”, ”Totals summation”, ”Order total average” and ”Number of orders per day”, ”Number of products”) is needed. This data set should besides the monthly progress also contain a monthly average of the five main attributes.

The data set is called, `pom_orderhead_supplier_stats_monthly`.

The dimensional data set, `pom_orderhead_supplier_stats`, of total supplier statistics for all 15 months (mentioned in section 5.3.2) is also used for this particular task.

### 5.5.3 Applying Data Mining to the transformed data

In order to analyze lost suppliers in Marketplace, it makes good sense to study the ”Number of products” attribute in the search of a connection to inactivity or loss of the supplier. A segmentation of the monthly progress of the suppliers is therefore needed. This segmentation is carried out for the ”Number of products” attribute.

The UFP-ONC algorithm (supported by the ten cluster validity criteria) was used for this segmentation using the average progress of the ”Number of products” attribute for the four last months of individual supplier activity (a supplier is defined lost if being inactive for more than two months). The clustering algorithm was run 20 times with number of clusters ranging from 2 to 15 clusters.

### 5.5.4 Evaluation of Data Mining results

The segmentation of the average monthly progress data set was carried out as explained in the previous section. Figure 5.10 shows the the cluster validity graph of the UFP-ONC algorithm for the monthly progress of the "Number of products" attribute. Four of the ten cluster validity criteria's local extrema (figure 5.11) suggest an optimal number of clusters for six clusters.

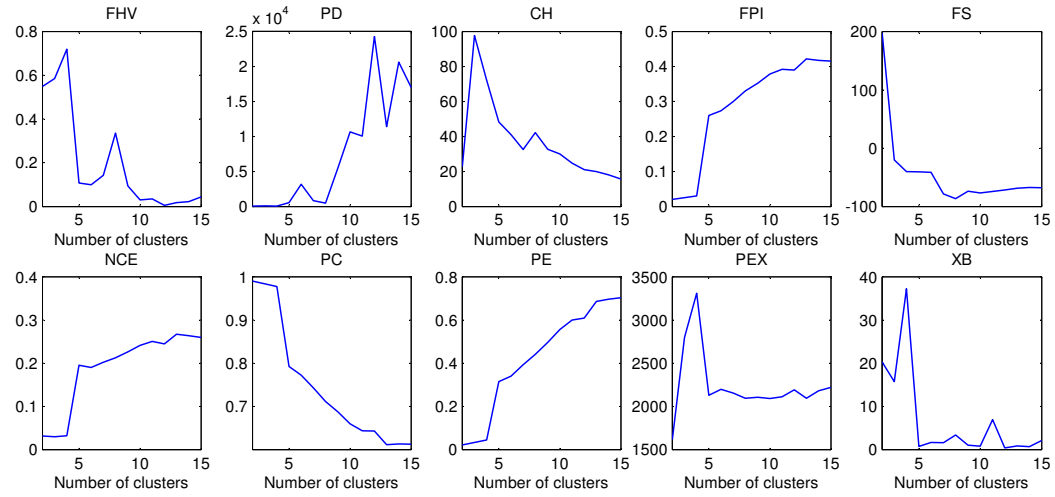


**Figure 5.10:** Optimal number of clusters for the last four months of Marketplace supplier data.

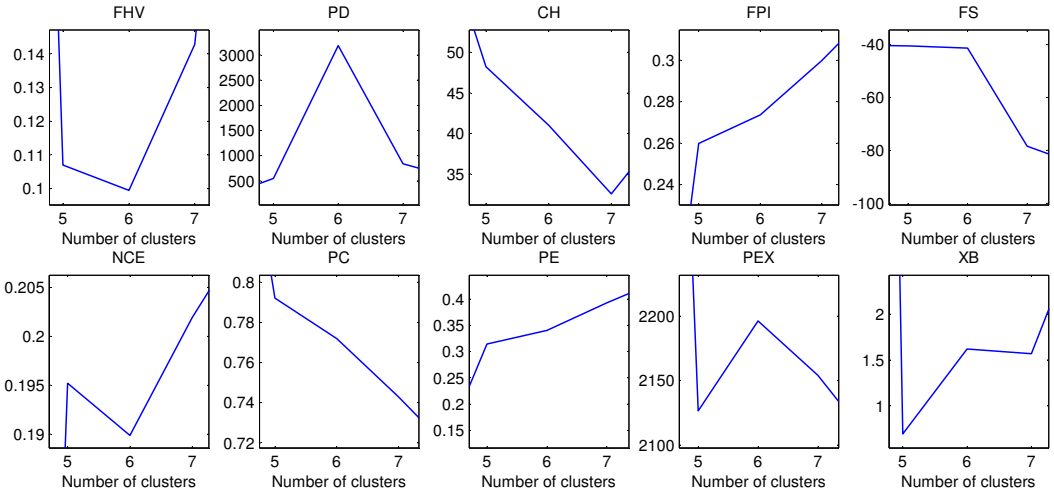


**Figure 5.11:** Local extrema near the optimal number of clusters.

The six cluster centroids in figure 5.12 show different supplier progress trends in the Marketplace system with respect to the "Number of products" attribute. Segment 1 is a flat zero trend (no offset) suggesting no supplier activity (hence a lost supplier). Segment 2 shows a negative trend (suppliers with dropping activity) while segment 3 shows a positive trend (new suppliers). Segment 4, 5 and 6 all show steady supplier activity (with different levels of offset). Since all rows in the monthly progress data set have been divided with the max value of the individual rows, the y-axis in figure 5.12 ranges from 0 to 1.

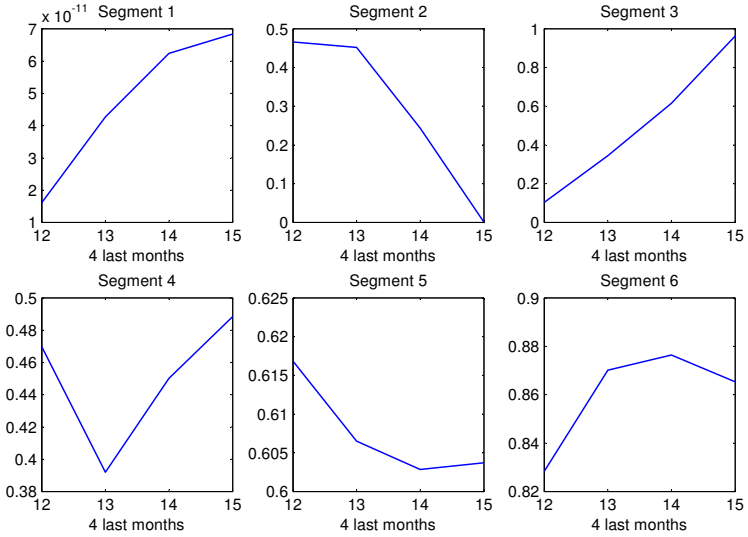


**Figure 5.12:** The six cluster centroids of monthly Marketplace supplier progress.

It is possible to show a connection between lost suppliers and the "Number of products" attribute by comparing the segmentation results with the actual lost suppliers (2 months or more of inactivity) found in the statistical supplier data set based on the entire period. When the monthly progress trend (of the "Number of products" attribute) is negative for several months, the supplier is likely to become inactive (or lost altogether).

The potential in this type of knowledge is being able to prevent suppliers from becoming inactive by tracking the "Number of products" attribute (or one or more of the four other main attributes) on a monthly basis and take the necessary action if needed.

## 5.6 Analysis of Buyers' use of suppliers in Marketplace

### 5.6.1 Selecting proper data attributes

The task of analyzing buyers' use of suppliers in Marketplace requires the general transactional data, which are found in the `POM_ORDER_HEADERS` database along with order line data found in `POM_ORDER_LINES` database.

In the `POM_ORDER_HEADERS` database the following attributes are of interest for this particular task.

- `ORDER_TOTAL`
- `BUYER_ID`
- `BUYER_NAME`
- `BUYING_ORG_ID`
- `BUYING_ORG_NAME`
- `SUPPLIER_ID`
- `SUPPLIER_NAME`
- `PURCHASE_RATE_DATE`

In the `POM_ORDER_LINES` database the following attributes are of interest for this particular task.

- `SUPPLIER_ITEM_NUMBER`
- `SUPPLIER_ITEM_DESC`
- `CATEGORY_ID`
- `CATEGORY_NAME`

All of the attributes are formatted using the Data Formatting Framework described in section 5.2.

### 5.6.2 Transformation and reduction of the preprocessed data

The character of the formatted Marketplace data attributes is entirely transactional. This data structure is inconvenient for further processing and should be transformed into a dimensional data structure.

The dimensional data set, `pom_orderhead_buyer_stats`, of total buyer statistics for all 15 months (mentioned in section 5.3.2) is used for this particular task. However, it needs to be extended using the product and the category information of the individual order lines.

Moreover, it is handy to generate a lookup table containing a list of suppliers for each category.

### 5.6.3 Applying Data Mining to the transformed data

Due to the specialized character of this task, a custom algorithm had to be developed.

An algorithm (see Algorithm 4) for possible reductions of the number of suppliers for each of the 1000+ buyers was made. The algorithm is for each buyer creating lists of total supplier usage with respect to categories. Further, the buyer turnover with respect to each supplier is added the individual lists.

For each list of supplier usage the algorithm is able to optimize (the `Optimize` method in Algorithm 4) the supplier usage by looking up (suppliers per category dictionary) alternate suppliers for each category of each supplier in the list. If all categories of one supplier can be replaced by other suppliers from the list, optimization is possible. The algorithm starts with trying to replace suppliers with a low turnover with respect to the buyer.

Since a small number of order lines are categorized, two methods have been applied to increase the number of categorized order lines:

1. Many uncategorized order lines have the same product descriptions as categorized order lines. In these cases it is therefore possible for the uncategorized order lines to inherit the categories of the categorized order lines with identical product descriptions.

2. An increased number a categorized order lines can be achieved by making a dictionary of the most common specified product terms used in the product descriptions of the order lines. Uncategorized order lines containing a term from the dictionary is thereby able to inherit the respective category of the term in the dictionary.

**Algorithm 4** Optimize buyers' use of suppliers

```
Sort orders with respect to buyer_id
for each buyer_id do
    Sort orders with respect to supplier_id
    Sort suppliers with respect to their turnover for the given buyer
    for each supplier_id do
        Sort categories in orders with respect to category_id
        for each category_id do
            Add category_id to supplier usage list for the given buyer
        end for
    end for
    Optimize(buyer's supplier usage list)
end for
```

### 5.6.4 Evaluation of Data Mining results

The developed algorithm manages to reduce (optimize) the use of suppliers for around 20% of the Marketplace buyers.

The algorithm is also able to look for possible supplier reductions over a given period of time (e.g. the last 4 months) since buyer habits may change now and then.

Using the two methods of increasing the number of categorized order lines, supplier reduction tables have been made for buyers for the last 4 months and for the entire period of 15 months.

## 5.7 Recommending products to relevant buyers in Marketplace

Association is as mentioned earlier another popular Data Mining task. Basically, association systems are used to obtain beneficial knowledge about customer business habits. E.g supermarkets can use association rules to optimize their turnover by adapting the display of products in the supermarket to fit consumer patterns (Market Basket analysis). If two products frequently end up in the same shopping basket, the supermarket may profit from this pattern by displaying the two products next to each other.

One of gatetrade.net's desired Data Mining accomplishments is to be able to recommend relevant products to buyers in the Marketplace system - a system somewhat similar to the recommendation feature in Amazon.com's web shop. Such a system relies on association rules based on prior data.

### 5.7.1 Selecting proper data attributes

The task of recommending products to relevant buyers in Marketplace requires the general transactional data, which are found in the `POM_ORDER_HEADERS` database along with order line data found in the `POM_ORDER_LINES` database.

In the `POM_ORDER_HEADERS` database the following attributes are of interest for this particular task.

- `ORDER_TOTAL`
- `BUYER_ID`
- `BUYER_NAME`
- `BUYING_ORG_ID`
- `BUYING_ORG_NAME`
- `SUPPLIER_ID`
- `SUPPLIER_NAME`
- `PURCHASE_RATE_DATE`

In the `POM_ORDER_LINES` database the following attributes are of interest for this particular task.

- `SUPPLIER_ITEM_NUMBER`
- `SUPPLIER_ITEM_DESC`
- `CATEGORY_ID`
- `CATEGORY_NAME`

All of the attributes are formatted using the Data Formatting Framework described in section 5.2.

### 5.7.2 Transformation and reduction of the preprocessed data

The character of the formatted Marketplace data attributes is entirely transactional. This data structure is inconvenient for further processing and should be transformed into a dimensional data structure.

The dimensional data set, `pom_orderhead_buyer_stats`, of total buyer statistics for all 15 months (mentioned in section 5.3.2) is used for this particular task. However, it needs to be extended using the product and the category information of the individual order lines.

### 5.7.3 Applying Data Mining to the transformed data

Multiple approaches are feasible for the task of recommending relevant products to buyers in Marketplace. Products can be recommended with respect to many different rules and parameters. Below is listed some of the various approaches.

It is possible to recommend products based on the following areas.

**Category** - Popular (frequently purchased) products of the same category type.

**Popularity** - Products with an overall high popularity (regardless their supplier).

**Supplier** - Popular products from the same supplier.

**Order statistics** - Products that often are purchased together.

Each of the approaches will result in a product recommendation that in some way relates to the product the buyer wants to order. Despite this, implementing a system using one of these approaches solely would cause unwanted drawbacks, thus resulting in less effective solutions.

The developed algorithm (Algorithm 5) is mainly used to generate specific association lists based on order line data for each product. The algorithm can be combined with some of the mentioned parameters (category, popularity and supplier) in order to form a complete recommendation system.

**Algorithm 5** Compute all possible single product associations

1: Sort order lines with respect to *order_id*
2: **for** each *order_id* **do**
3: Compute all single product associations pairs
4: Add computed associations pairs to global product association list
5: **end for**
6: Format global association list into top 10 association list for each product

### 5.7.4 Evaluation/visualization of Data Mining results

An algorithm for counting and keeping track of product associations for each of the 35000+ different products has been made. The algorithm generates a top X association list for each product - this list is applicable in the task of recommending relevant products to buyers. The algorithm also supports monthly product associations.

The following examples show what buyers tend to buy together in an order (product descriptions are in Danish).

Top 5 associations to 'inkjet patron hp c6656a no 56 sort':

1. 'inkjet patron hp c6657a color'.
2. 'inkjet patron hp 51645a sort'.
3. 'inkjet patron hp c6578a no78 color'.
4. 'inkjet patron hp c1823d no23 color'.
5. 'automatbaeger 21cl pk100'.

Top 5 associations to 'plastlommer a4 pp, 0,08mm aaben top':

1. 'plastcharteques a4 pp 0,11mm'.
2. 'plastlommecharteques a4 aab. top/side'.
3. 'blok memo impega 76x76 gul'.
4. 'impega highlighter, gul'.
5. 'impega blok a5 kvadreret toplimet'.

Top 5 associations to 'soedmaelk purepak, pakke med 1 liter':

1. 'skummetmaelk purepak, pakke med 1/4 liter'.
2. 'letmaelk purepak, pakke med 1/4 liter'.
3. 'letmaelk purepak, pakke med 1 liter'.
4. 'solsikkebreve det gode i ski., pr. stk. med 950 gram'.
5. 'soedmaelk purepak, pakke med 1/4 liter'.

Top 5 associations to 'opvaskemiddel suma star free 1 liter':

1. 'opvaskeboerste med haarblanding'.
2. 'automatbaeger 21cl pk100'.

3. 'kalkfjerner minus kalk 1 liter'.
4. 'inkjet patron hp c6656a no 56 sort'.
5. 'kaffe 500 g bki luxus'.

The association lists are suited for making a purchase system that suggests relevant products to buyers.

## 5.8 Possible reasons for transactions not made through Marketplace

### 5.8.1 Selecting proper data attributes

The task of locating possible reasons for transactions that are not made through Markplace, requires the general transactional data, which are found in the POM_ORDER_HEADERS and the eProc_INVOICE databases. Also order line data from the POM_ORDER_LINES and the eProc_INVOICELINE databases were used.

In the POM_ORDER_HEADERS database the following attributes are of interest for this particular task.

- ORDER_TOTAL
- BUYER_ID
- BUYER_NAME
- BUYING_ORG_ID
- BUYING_ORG_NAME
- SUPPLIER_ID
- SUPPLIER_NAME
- PURCHASE_RATE_DATE

In the POM_ORDER_LINES database the following attributes are of interest for this particular task.

- ORDER_NUMBER
- SUPPLIER_ITEM_NUMBER

In the eProc_INVOICE database the following attributes are of interest for this particular task.

- invoice_total
- created_time
- updated_user_id
- company
- sender_cvr_number

In the eProc_INVOICELINE database the following attributes are of interest for this particular task.

- `description`
- `supplier_item_number`

All of the attributes are formatted using the Data Formatting Framework described in section 5.2.

### 5.8.2 Transformation and reduction of the preprocessed data

The character of the formatted Marketplace and eProcurement data attributes is entirely transactional. This data structure is inconvenient for further processing and should be transformed into a dimensional data structure.

The four dimensional data sets from section 5.3.2 are used for this particular task. The four data sets are further supported by the Marketplace/eProcurement product descriptions, that are available through the respective order line data.

### 5.8.3 Applying Data Mining to the transformed data

Various algorithms and approaches were tried for solving this task.

*Latent Semantic Analysis* [22, 23] were used for grouping different order types in order to gain a better overview of the product spaces in Marketplace/eProducrement. This analysis did not produce any useful results due to the very limited product descriptions, that resulted in too sparse co-occurrence matrices. Instead, lists of the most traded products in Marketplace/eProcurement were made.

The segmentation results from section 5.3 were used to compare the different segments for buyers and suppliers in the Marketplace/eProcurement systems.

### 5.8.4 Evaluation/visualization of Data Mining results

This analysis is compared to the other types of analysis the far most abstract one, since there can be many reasons for buyers choosing eProcurement over Marketplace. However, the list below gives an impression of some probable reasons for buyers using eProcurement.

1. eProcurement offers a greater number of suppliers and by the looks of Marketplace/e-Procurement top purchased product lists, it also offers a wider range of commodities. This may be considered a relevant factor.

2. By the looks of the product lists of top purchased products, Marketplace/eProcurement deal with different product groups. Marketplace top commodities are product related (”plastcharteques”, ”blok memo”, ”port replicator”, etc.) while eProcurement top commodities tend to be service related (”bestilt bil”, ”rabat paa samtaleforbrug”, ”lyngby holte taxa”, ”servicetekniker”, etc.).

3. When comparing Marketplace/eProcurement buyer segments, the eProcurement buyer segments show a tendency of being ”broader”. This suggests that eProcurement buyers are not as ”serious” (small number of orders with small total amounts) and consistent as Marketplace buyers.

4. Purchase traditions/habits are also an important factor. Some buyers probably does not use Marketplace because they have always used eProcurement and will not risk anything by using Marketplace.

5. Finally, some buyers have perhaps not yet realized what Marketplace is and therefore have no clue of its advantages.

The list above shows some probable reasons why buyers do not use Marketplace. For future work, this analysis can be used to target the most potential eProcurement users (buyers and suppliers) and convince them to use Marketplace more often. By choosing eProcurement users in segments similar to best Marketplace user segments, only ”serious” and consistent users are targeted.

Chapter 6

# Outline of a potential Business Intelligence framework

The methods and algorithms developed in this project to process gatetradet.net data, all have the potential of being used in a full scale Business Intelligence framework with the purpose of processing gatetrade.net data.

For such a BI framework, one of the most fundamental functions is to extract needed buyer/-supplier transactional data on a regular basis (say each month) and convert it to dimensional data sets that are stored. This monthly storage process can be implemented in using a combination of `SQL` queries combined with relevant formatting methods of the Data Formatting Framework.

The process of collecting, formatting and storing gatetrade.net data constitutes the bottom layer of the BI framework. In this way the BI framework allows a number of selected algorithms to be implemented independently on top of it. These algorithms (or some of them) can automatically be executed monthly following the collection of data process.

The outlined BI framework in this chapter is an example of how such a framework may practically be structured. The system may be implemented in `C#` using the `.net` platform, thus making `SQL` data collecting/formatting convenient. Matlab algorithms can either be executed directly from the `C#` environment or be implemented using `C`, `C++` or `C#`.

Chapter 7

# Conclusion

The conclusion chapter is divided into two subsections. A section presenting the achieved results of this project and another section suggesting ideas for future work.

## 7.1 Results for this project

In the project the four different clusterings algorithms, kNN, FkNN, FCM and UFP-ONC, were implemented using Matlab as described in chapter 4. A mutual evaluation of them showed that the UFP-ONC algorithm performed superiorly (although being the slowest) with respect to four test data sets of various segmentation difficulty.

In order to determine the optimal number of clusters in a given data set, ten cluster validity criteria were implemented in Matlab as described in chater 4. To evaluate their individual performance, they were tested against three test data sets of various segmentation difficulty. The two best performing cluster validity criteria were the Normalized Classification Entropy criterion (NCE) and the Xie and Beni criterion (XB).

With respect to processing of the gatetrade.net data, a lot of work was put into formatting the different attribute types in order to ease the processing process. For this particular purpose, a Data Formatting Framework containing various helpful formatting methods was developed. Moreover, the transactional gatetrade.net data were converted into many structured, dimensional data sets.

For the most part of gatetrade.net's six desired data processing applications, the UFP-ONP clustering algorithm combined with the ten cluster validity criteria proved sufficient. This modified UFP-ONC algorithm was capable of segmenting gatetrade.net data on a total basis as well on a monthly basis for buyers and suppliers in the Marketplace/eProcurement systems.

A couple of gatetrade.net's desired data processing applications ((buyers' use of suppliers and trade canalization)) required more specialized algorithms. These specialized algorithms were also implemented in Matlab and were able to solve the desired goals satisfyingly. Due to the somewhat abstract nature of the desired task of examining transactions not made through Marketplace, no main algorithm for this purpose was made. Instead, a number of various approaches (e.g. latent semantic analysis) resulting in different conclusions were tried.

To sum up the conclusion, this project has presented various advanced (semi-)automatic Data Mining algorithms and has shown the value of applying these methods for business proposes (Business Intelligence). Further, the project has suggested an outline of a potential Business Intelligence framework based on the findings of this project.

## 7.2 Future work

Algorithm-wise, an interesting idea for further studies could be to replace the first layer (FCM) of the UFP-ONC algorithm with the fuzzy $k$-Nearest Neighbour (FkNN) algorithm in order to speed up the clustering process. Although the FkNN algorithm consistently is able to do a decent clustering of a data set, evaluation of the combined algorithm should show whether the initial centroids (generated by the FkNN algorithm) for the FMLE layer of the UFP-ONC algorithm are of a sufficient quality.

Chapter 6 suggested an outline for a potential Business Intelligence framework that has an optimal structure for implementing one or more of the mentioned Data Mining algorithms or continuously extending the framework with new algorithms. At the same time, the Business Intelligence framework is able to supply the user with basic statistical information on the buyers or suppliers, due to the dimensional structure of the stored data sets. Finally, the Business Intelligence framework has the potential of becoming a powerful, flexible tool and an essential partner in the ongoing task of analyzing and structuring large data amounts.

Chapter 8

# References


1. Wikipedia. *Business intelligence*. http://en.wikipedia.org/wiki/Business_Intelligence.

2. Center for Mathematical and Information Sciences (CMIS), CSIRO. *What is Business Intelligence?* http://www.cmis.csiro.au/bi/what-is-BI.htm.

3. Microsoft. *An Introduction to SQL Server 2005 Data Mining*. http://www.microsoft.com/technet/prodtechnol/sql/2005/intro2dm.mspx.

4. Usama Fayyad, Microsoft Research. *Data Mining and Knowledge Discovery in Databases: Implications for Scientific Databases*. Scientific and Statistical Database Management, 2-11, 1997.

5. Wikipedia. *Information*. http://en.wikipedia.org/wiki/Information.

6. gatetrade.net. Information on gatetrade.net and some of their solutions (Marketplace/eProcurement). http://www.gatetrade.net.

7. Johannes Grabmeier, Andreas Rudolph. *Techniques of Cluster Algorithms in Data Mining*. Data Mining and Knowledge Discovery, 6, 303-360, 2002.

8. Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

9. Christian Böhm, Florian Krebs. *The k-Nearest Neighbour Join: Turbo Charging the KDD Process*. Knowledge and Information Systems, 6, 728-749, 2004.

10. N. Zahid, O. Abouelala, M. Limouri, A. Essaid. *Fuzzy clustering based on K-nearest-neighbours rule*. Fuzzy Sets and Systems, 120, 239-247, 2001.

11. James C. Bezdek, Chris Coray, Robert Gunderson, James Watson. *Detection and Characterization of Cluster Substructure*. SIAM Journal on Applied Mathematics, 40, 339-357, 1981.

12. I. Gath, A. B. Geva. *Unsupervised Optimal Fuzzy Clustering.* Pattern Analysis and Machine Intelligence, 11, 773-781, 1989.

13. David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

14. Mukundan Srinivasan, Young B. Moon. *A comprehensive clustering algorithm for strategic analysis of supply chain networks.* Computers Industrial Engineering, 36, 615-633, 1999.

15. Kok Sung Won, Tapabrata Ray. *Performance of Kriging and Cokriging based Surrogate Models within the Unified Framework for Surrogate Assited Optimization.* IEEE 0-7803-851 5-2/04/.

16. Cüneyt Güler, Geoffrey D. Thyne. *Delineation of hydrochemical facies distribution in a regional groundwater system by means of fuzzy c-means clustering.* Water Resources Research, 40, 2004.

17. Nikhil R. Pal, James C. Bezdek. *On Cluster Validity for the Fuzzy c-Means Model.* Transactions on Fuzzy Systems, 3, 370-379, 1995.

18. Jiu-Lun Fan, Cheng-Mao Wu, Yuan-Liang Ma. *A modified partition coefficient.* IEEE 0-1803-5141-1/00/.

19. M. D. Alexiuk, N. J Pizzi. *Cluster validation Indices for fMRI data: Fuzzy C-Means with feature Partitions versus Cluster Merging Strategies.* IEEE 0-7803-8376-1/04/.

20. M. P. Windham. *Cluster validity for fuzzy clustering algorithms.* Fuzzy Sets and Systems, 5, 177-185, 1981.

21. Iveta Mrázová. *Intelligent Data Mining Techniques.* Tutorial for ANNIE (Artificial Neural Networks In Engineering), 2003.

22. Thomas Hofmann. *Unsupervised Learning by Probabilistic Latent Semantic Analysis.* Machine Learning, 42, 177-196, 2001.

23. Guandong Xu, Yanchun Zhang, Xiaofang Zhou. *Using Probabilistic Latent Semantic Analysis for Web Page Grouping.* IEEE 1097-8585/05.

Appendix A

# Attribute descriptions of three smaller eProcurement databases

Attribute descriptions of `eProc_USER`, `eProc_COMPANY` and `eProc_SUPPLIERS` databases are contained in table A.1, A.2 and A.3 respectively.

| Attribute | Description |
|---|---|
| `pkid` | ID of buyer |
| `firstname` | First name of buyer |
| `lastname` | Last name of buyer |
| `email` | Email address of buyer |
| `userid` | Alias of buyer |
| `directphone` | Direct phone number of buyer |

**Table A.1:** `eProc_USER` attribute descriptions

| Attribute | Description |
|---|---|
| `pkid` | ID of company |
| `name` | Name of company |

**Table A.2:** `eProc_COMPANY` attribute descriptions

| Attribute | Description |
|---|---|
| `dunsnr` | CVR number of supplier |
| `name` | Name of supplier |

**Table A.3:** `eProc_SUPPLIERS` attribute descriptions

Appendix B

# Classification figures of clustering algorithms evaluation

## B.1 Fisher iris data set

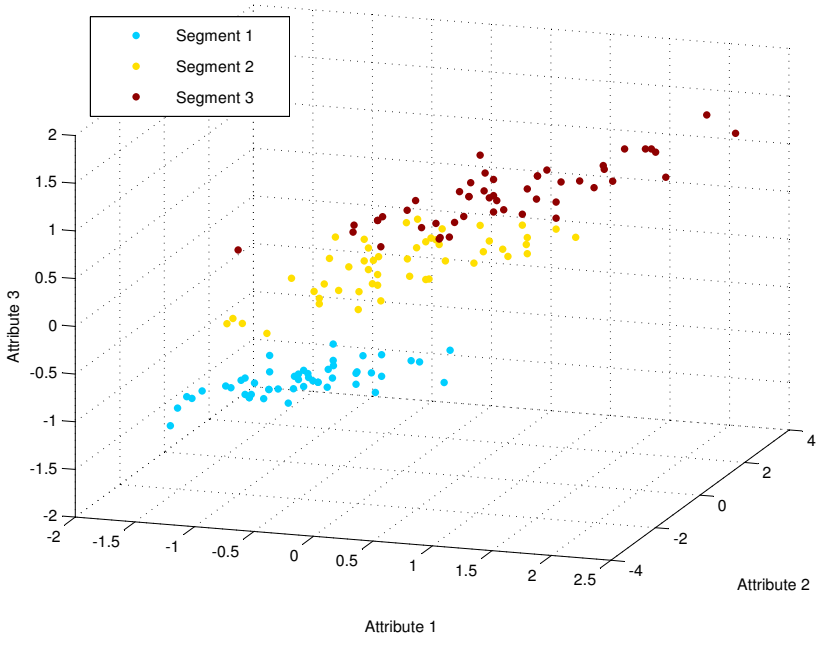


**Figure B.1:** True classification of Fisher iris data set.

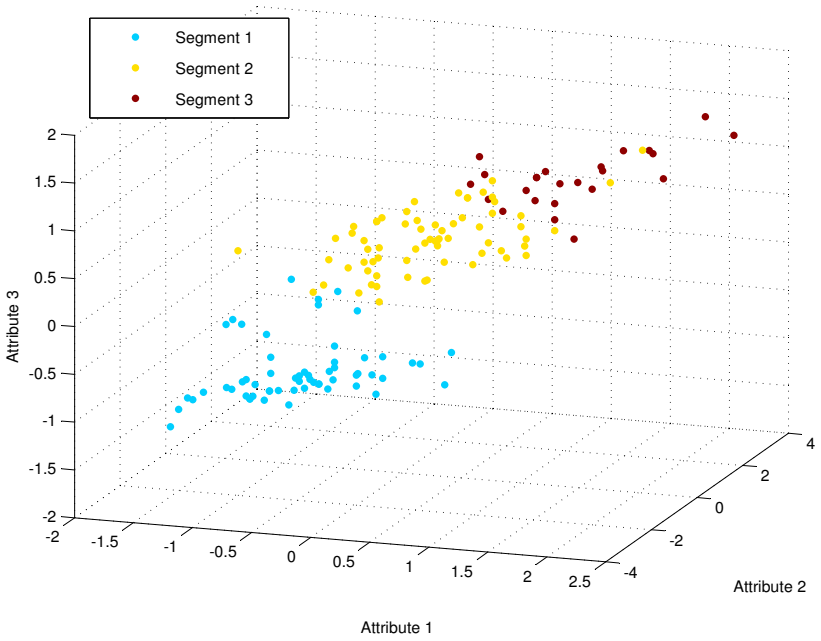


**Figure B.2:** kNN classification of Fisher iris data set.

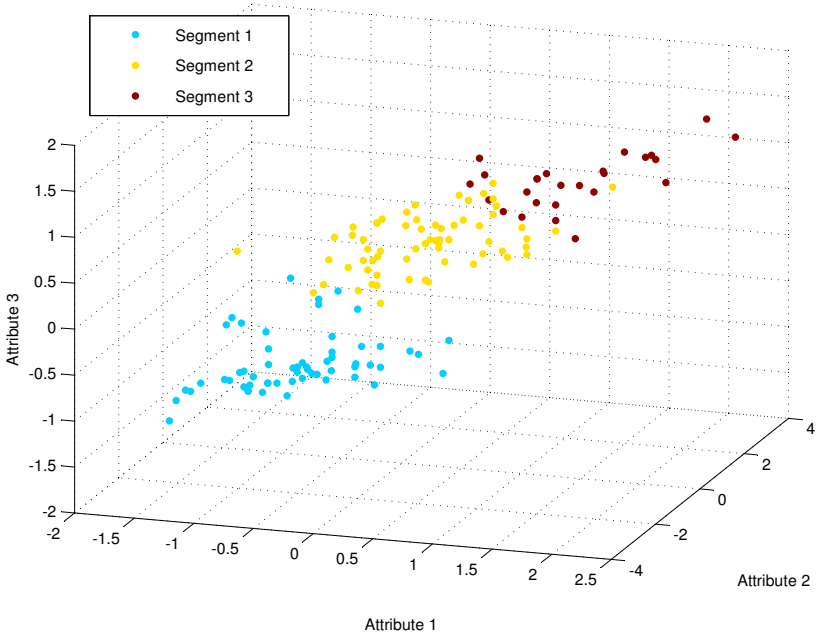


**Figure B.3:** FkNN classification of Fisher iris data set.

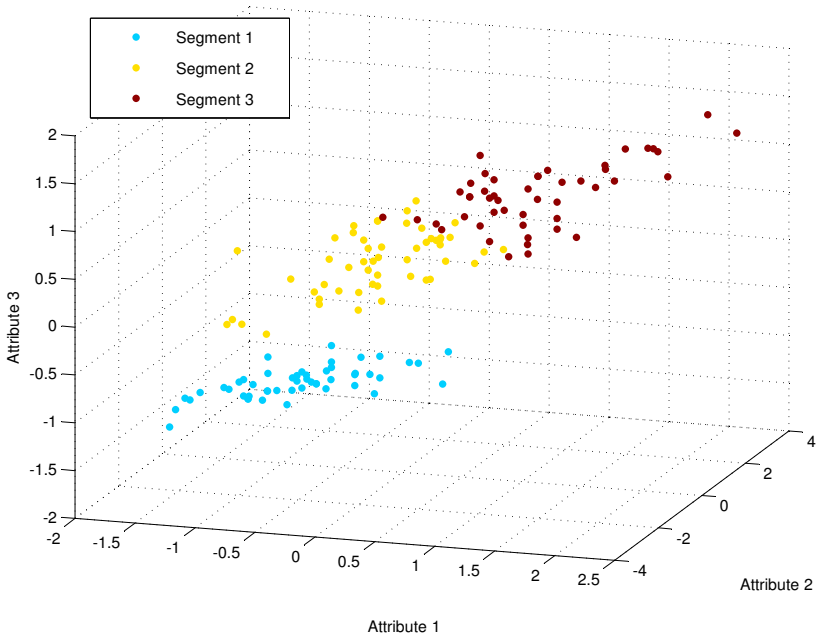


**Figure B.4:** FCM classification of Fisher iris data set.

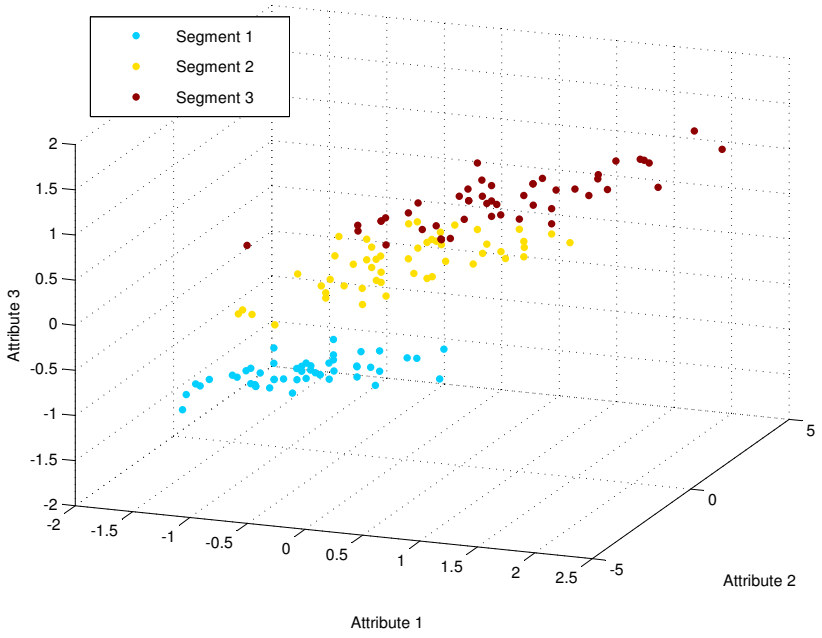


**Figure B.5:** UFP-ONC classification of Fisher iris data set.

## B.2 Test1 data set

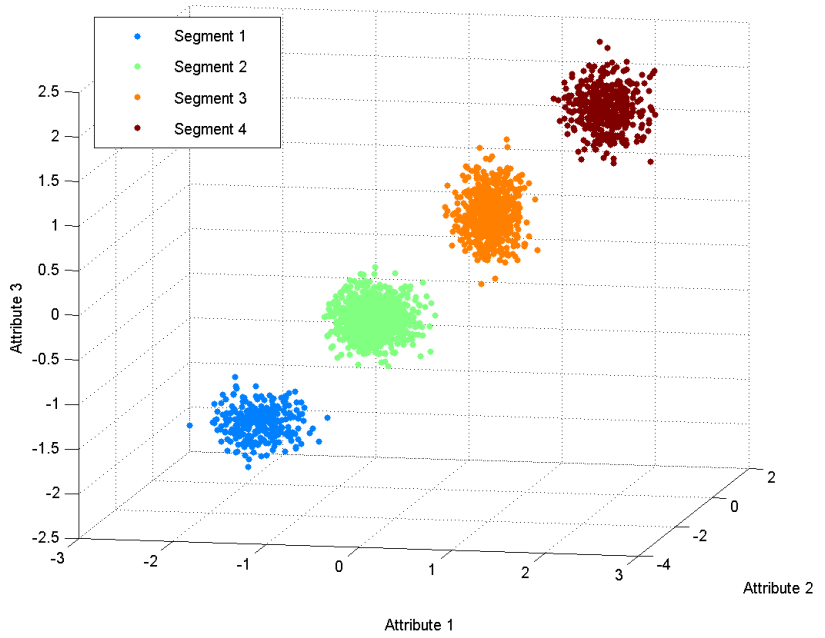


**Figure B.6:** True classification of Test1 data set.

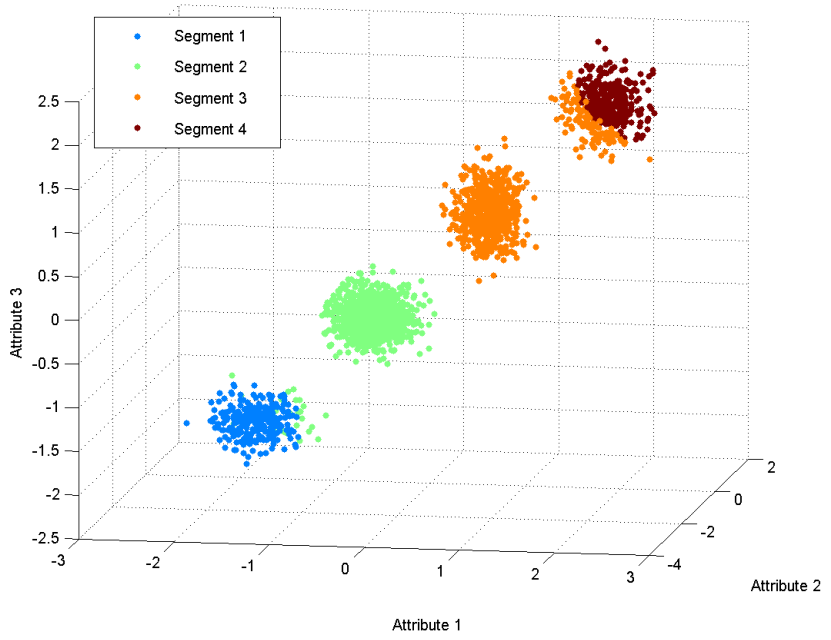


**Figure B.7:** kNN classification of Test1 data set.

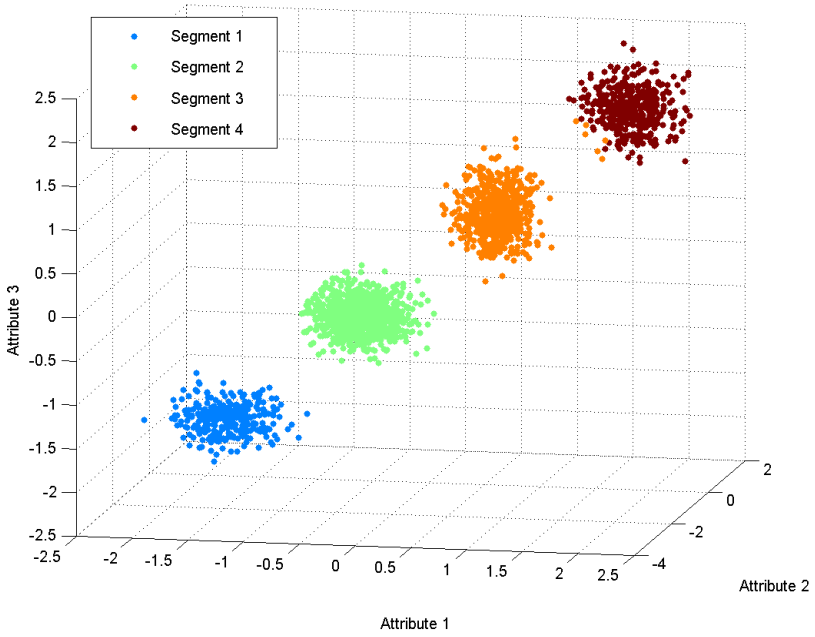


**Figure B.8:** FkNN classification of Test1 data set.

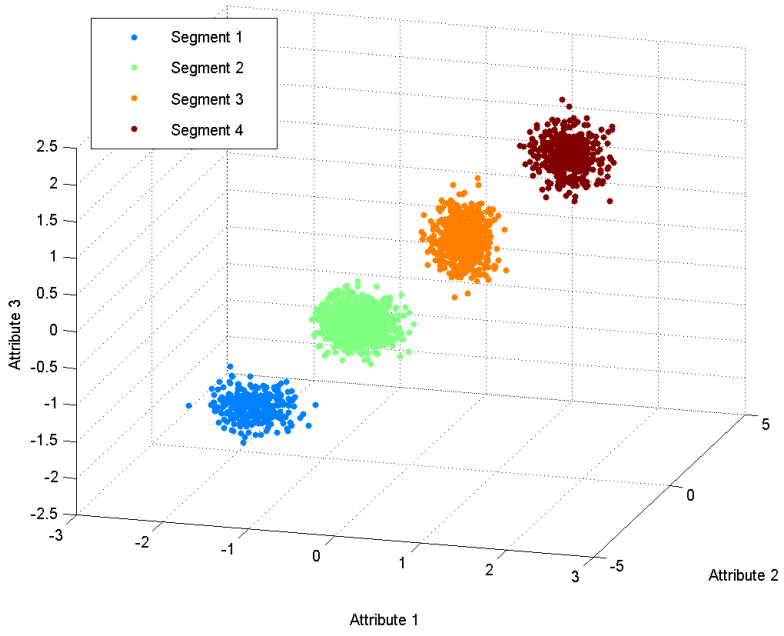


**Figure B.9:** FCM classification of Test1 data set.

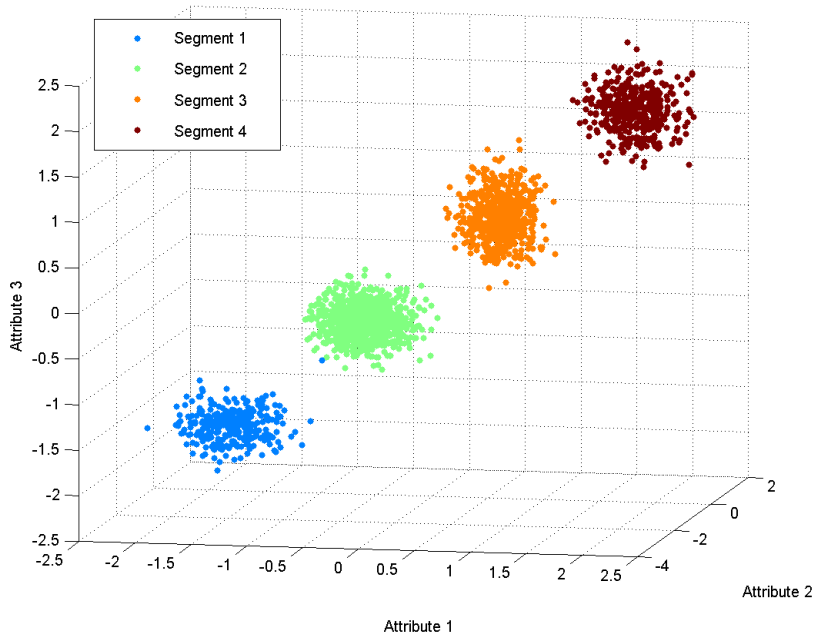


**Figure B.10:** UFP-ONC classification of Test1 data set.

## B.3 Test2 data set

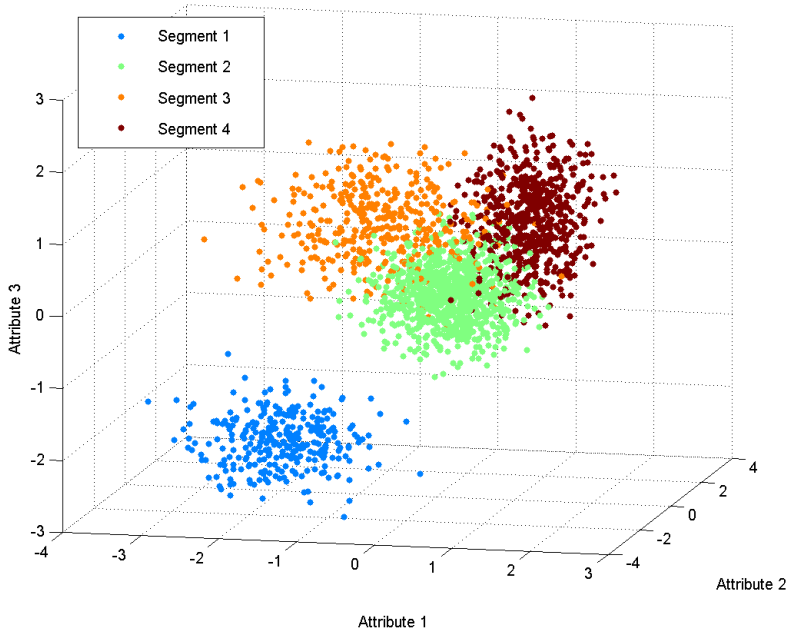


**Figure B.11:** True classification of Test2 data set.

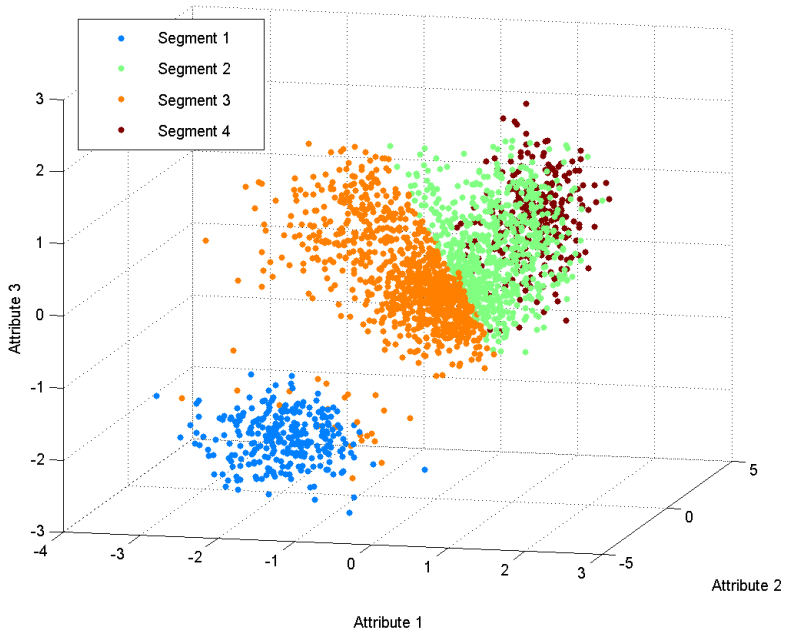


**Figure B.12:** kNN classification of Test2 data set.

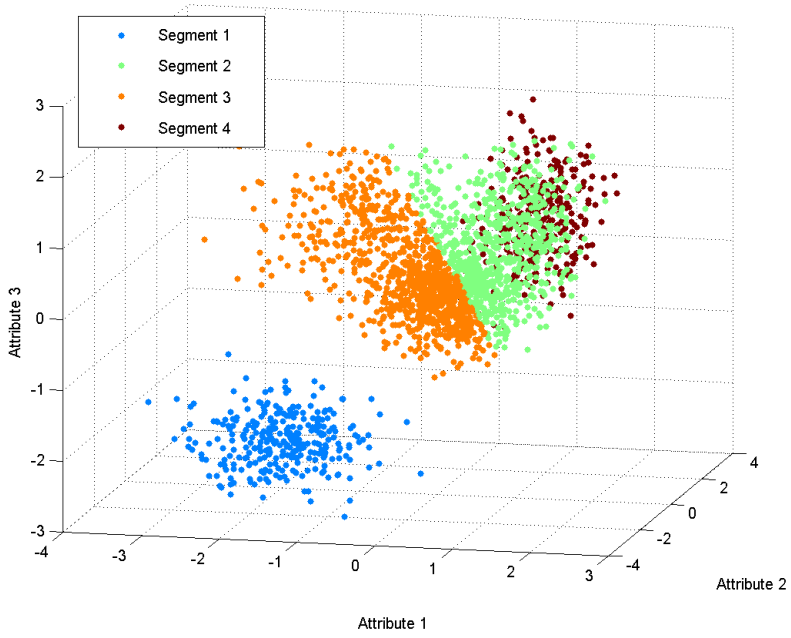


**Figure B.13:** FkNN classification of Test2 data set.

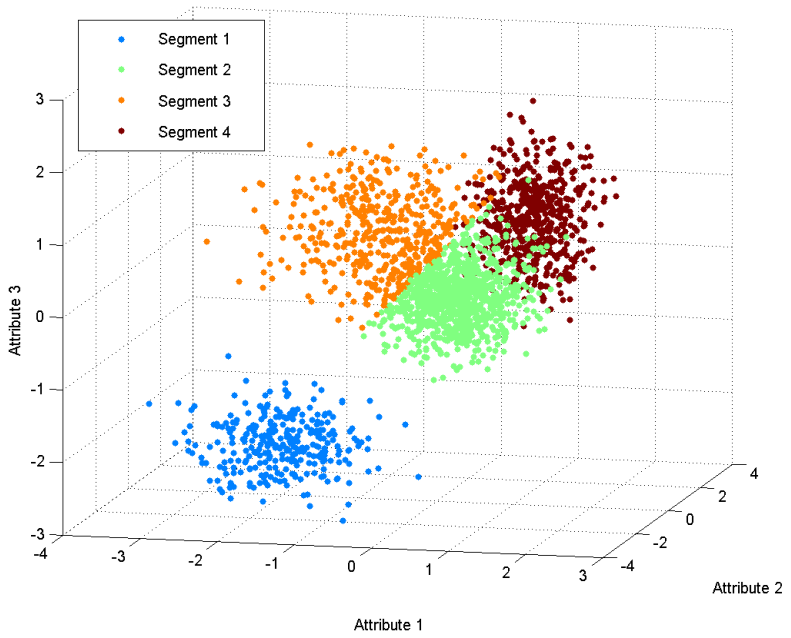


**Figure B.14:** FCM classification of Test2 data set.

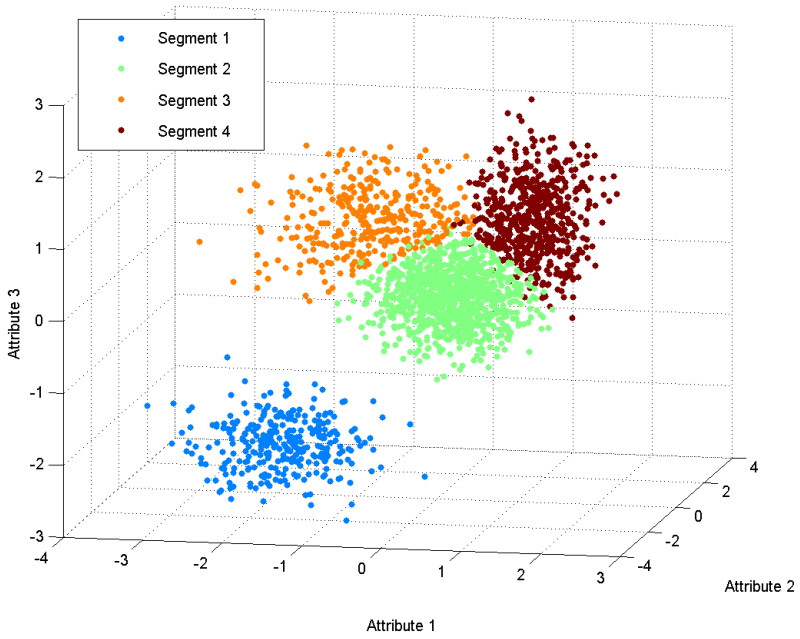


**Figure B.15:** UFP-ONC classification of Test2 data set.

## B.4 Test3 data set

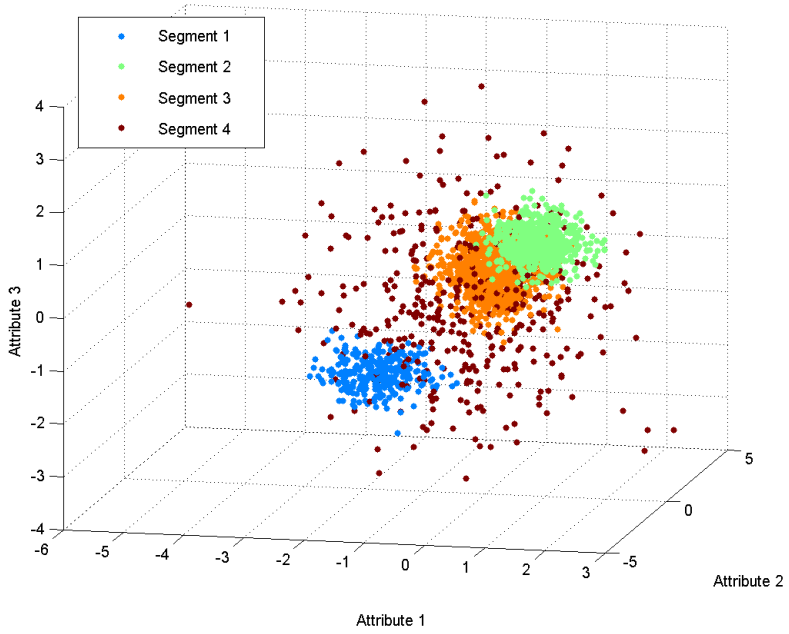


**Figure B.16:** True classification of Test3 data set.

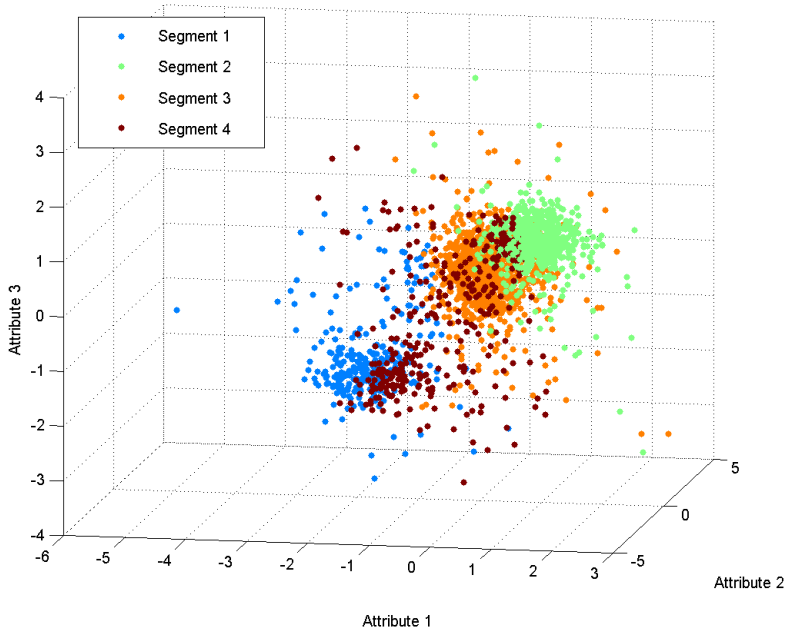


**Figure B.17:** kNN classification of Test3 data set.

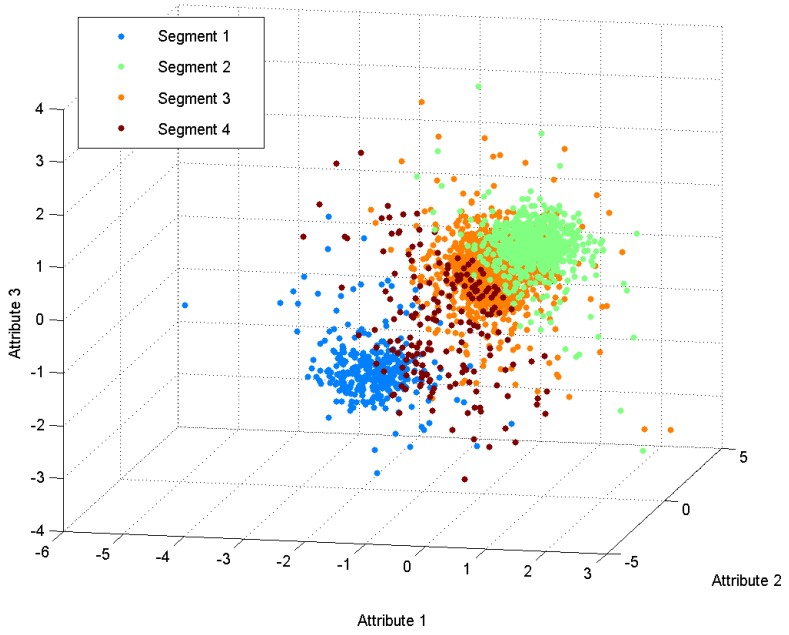


**Figure B.18:** FkNN classification of Test3 data set.

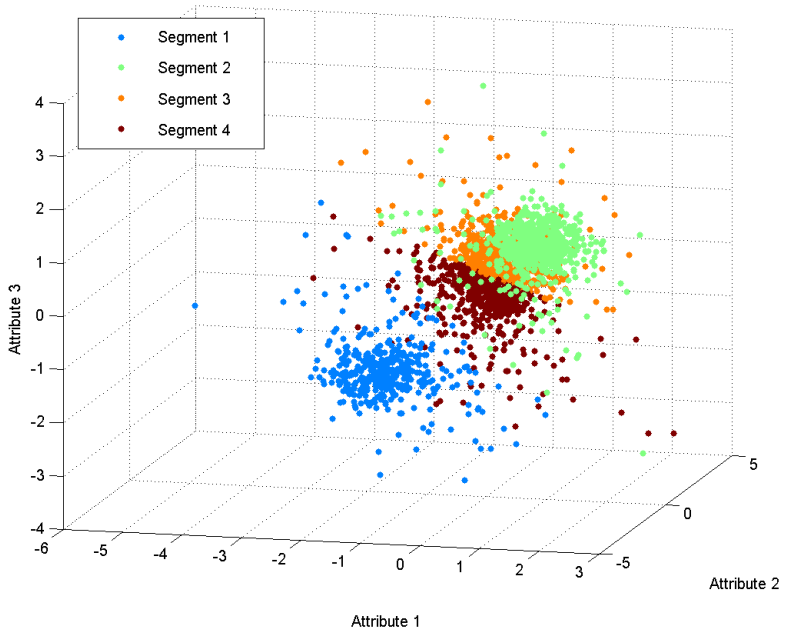


**Figure B.19:** FCM classification of Test3 data set.

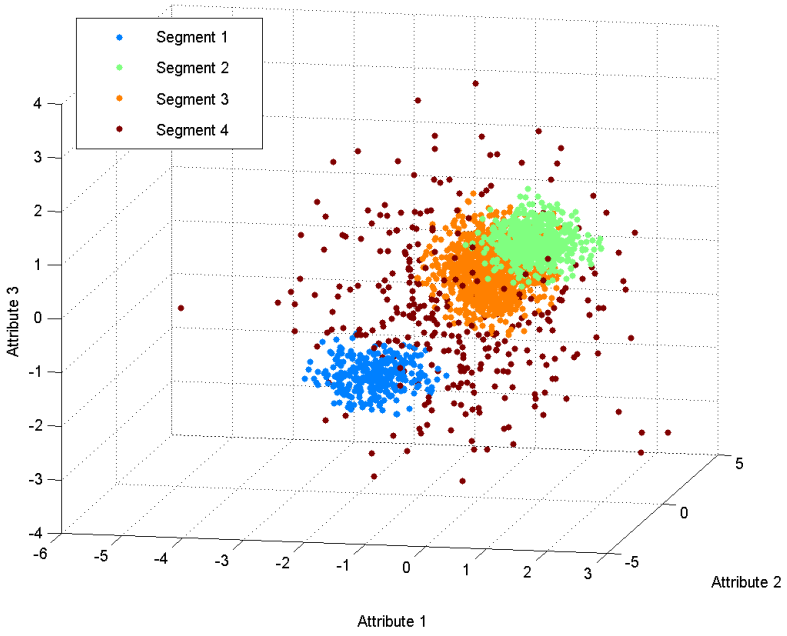


**Figure B.20:** UFP-ONC classification of Test3 data set.

Appendix C

# Data Formatting Framework class diagram

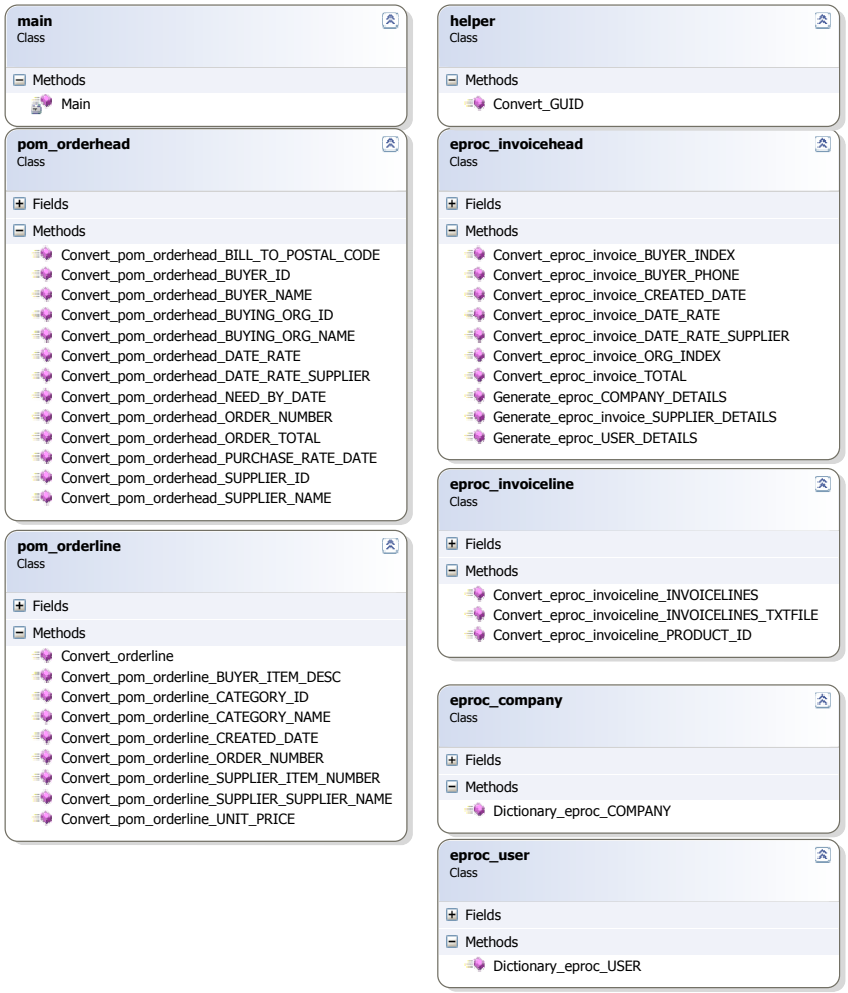


**Figure C.1:** The class diagram of the Data Formatting Framework.

Appendix D

# Segmentation results of Marketplace/eProcuremnt analysis

## D.1 Marketpalce supplier segmentation results

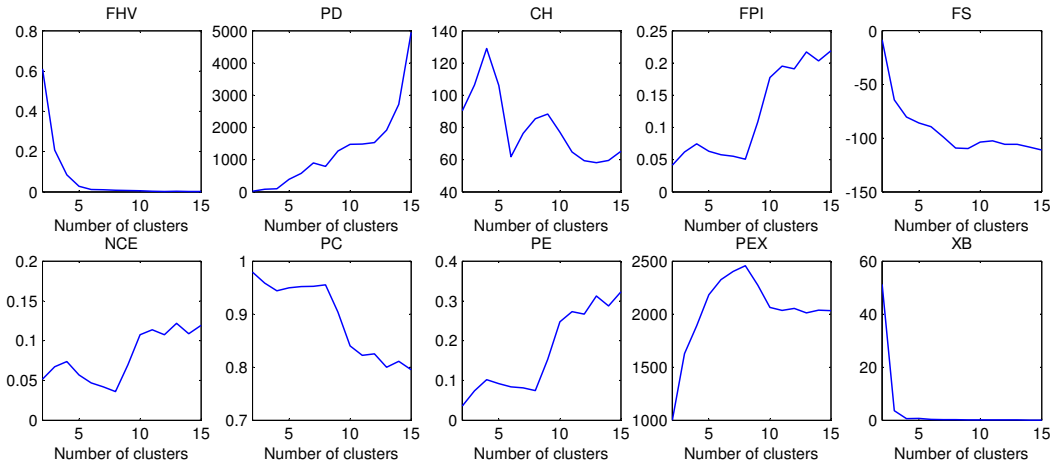


**Figure D.1:** Optimal number of clusters in Marketplace supplier data.

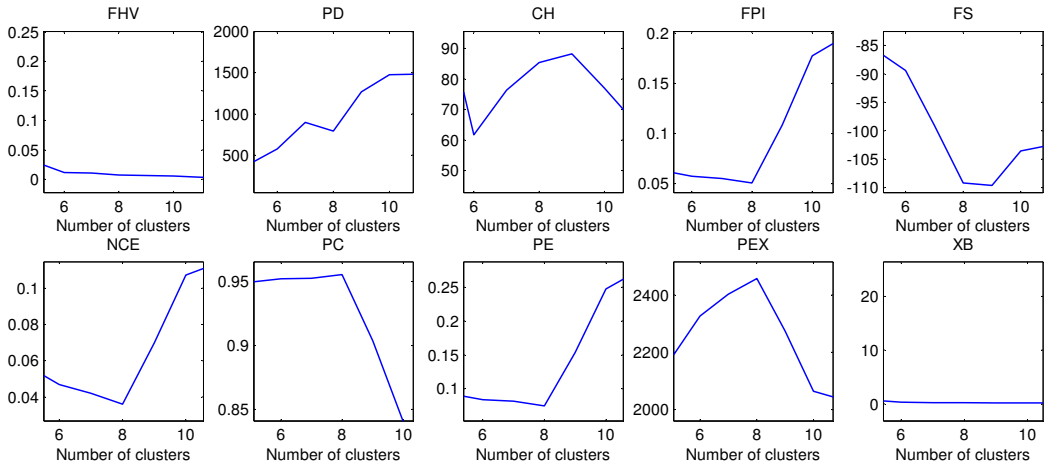


**Figure D.2:** Local extrema near the optimal number of clusters in Marketplace supplier data.

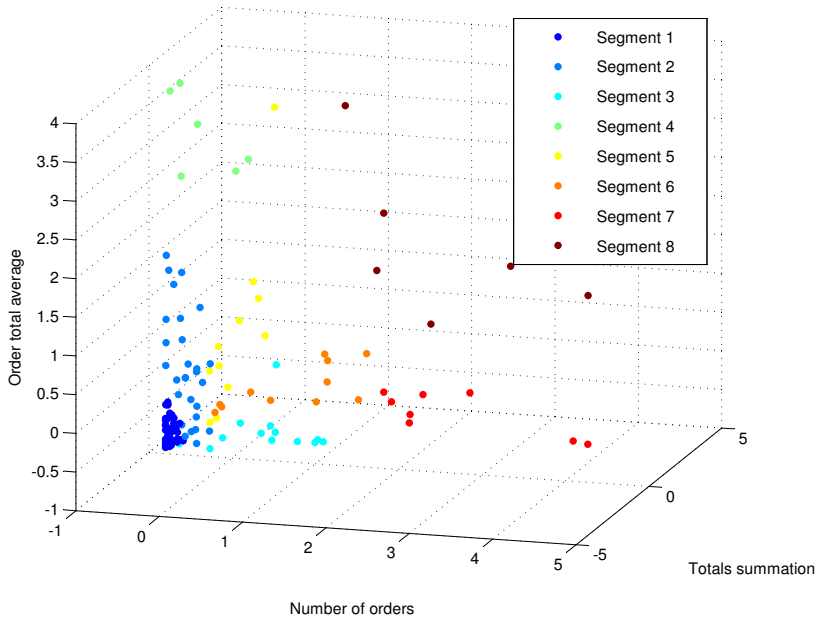


**Figure D.3:** Segmentation of Marketplace supplier data.

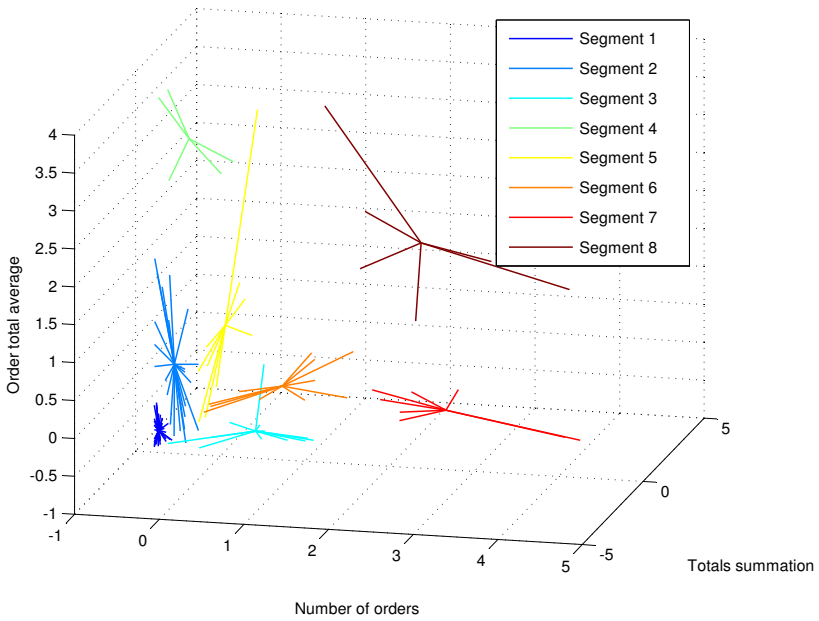


**Figure D.4:** Marketplace supplier data linked to their respective cluster centroids.

## D.2 eProcurement buyer segmentation results

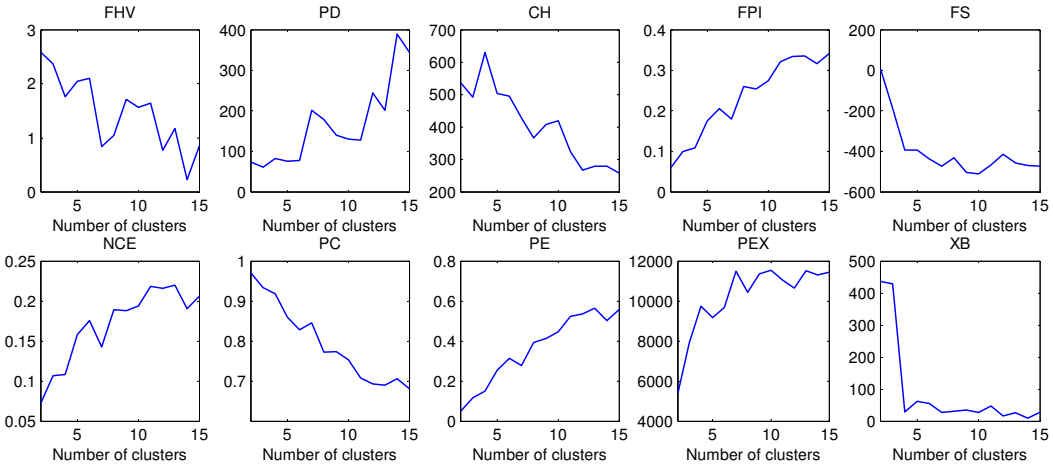


**Figure D.5:** Optimal number of clusters in eProcurement buyer data.

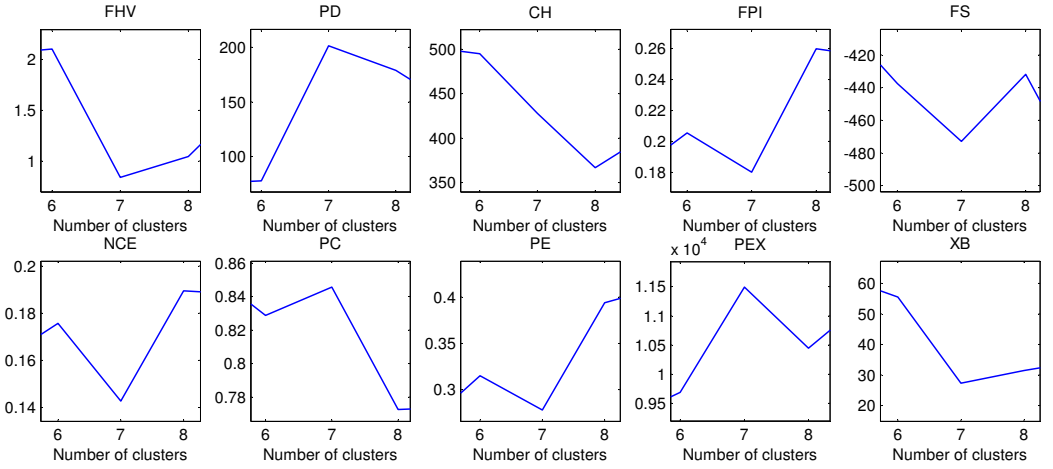


**Figure D.6:** Local extrema near the optimal number of clusters in eProcurement buyer data.

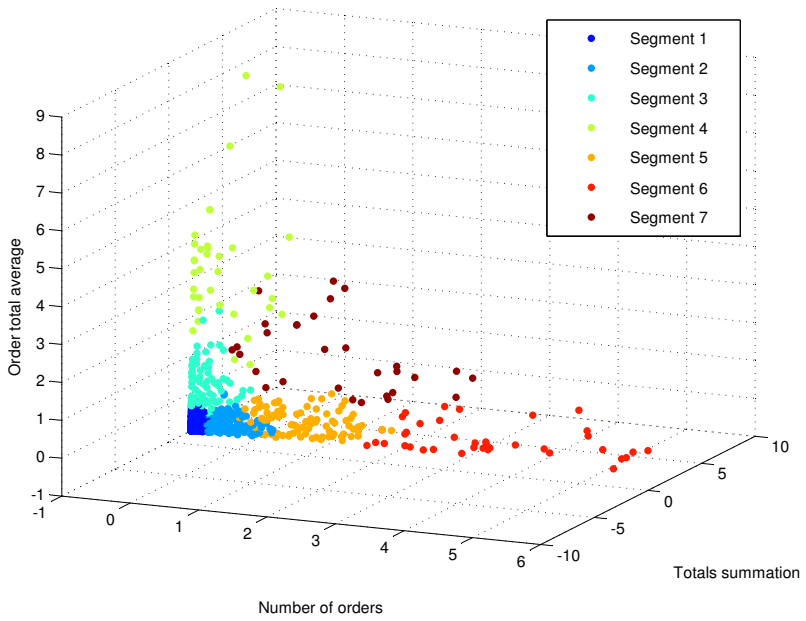


**Figure D.7:** Segmentation of eProcurement buyer data.

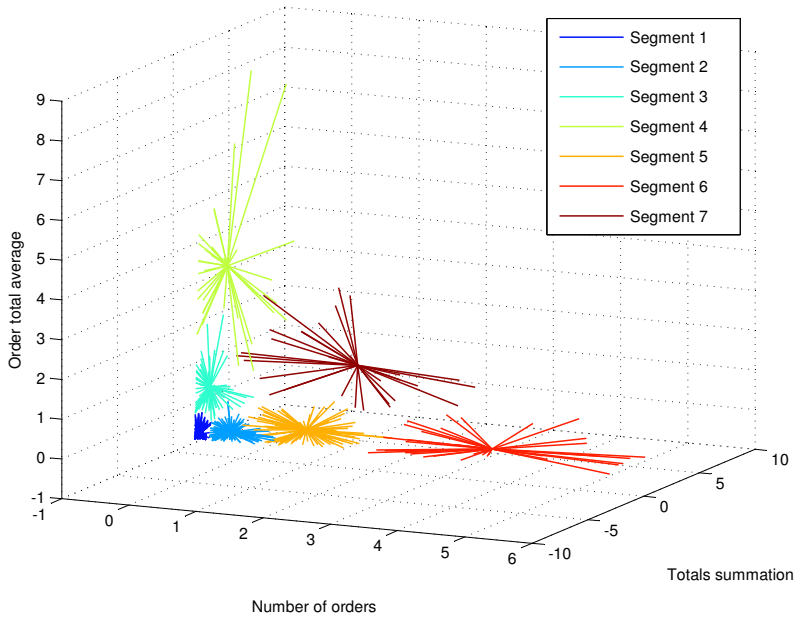


**Figure D.8:** eProcurement buyer data linked to their respective cluster centroids.

## D.3 eProcurement supplier segmentation results

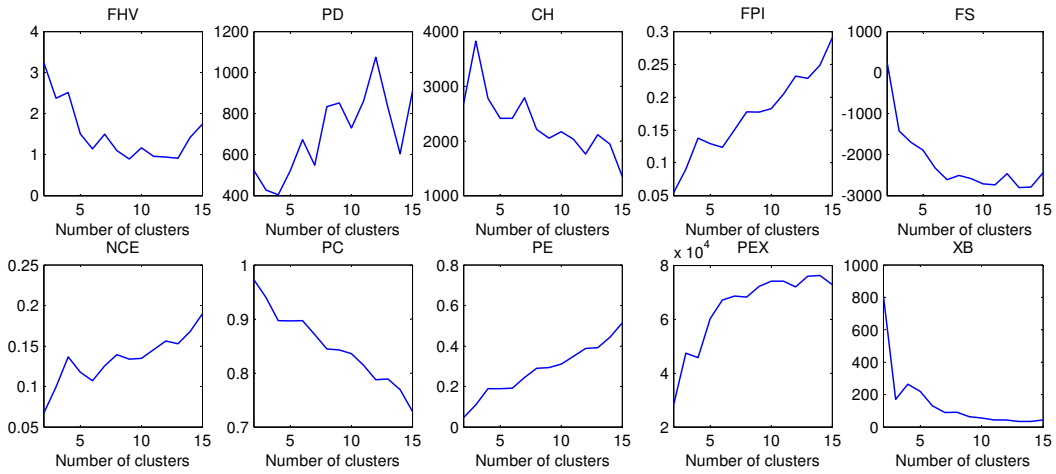


**Figure D.9:** Optimal number of clusters in eProcurement supplier data.

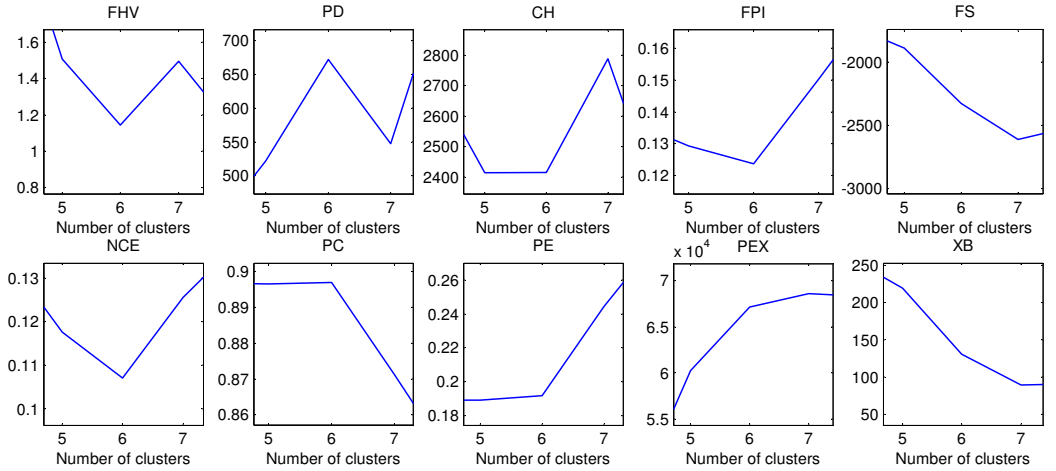


**Figure D.10:** Local extrema near the optimal number of clusters in eProcurement supplier data.

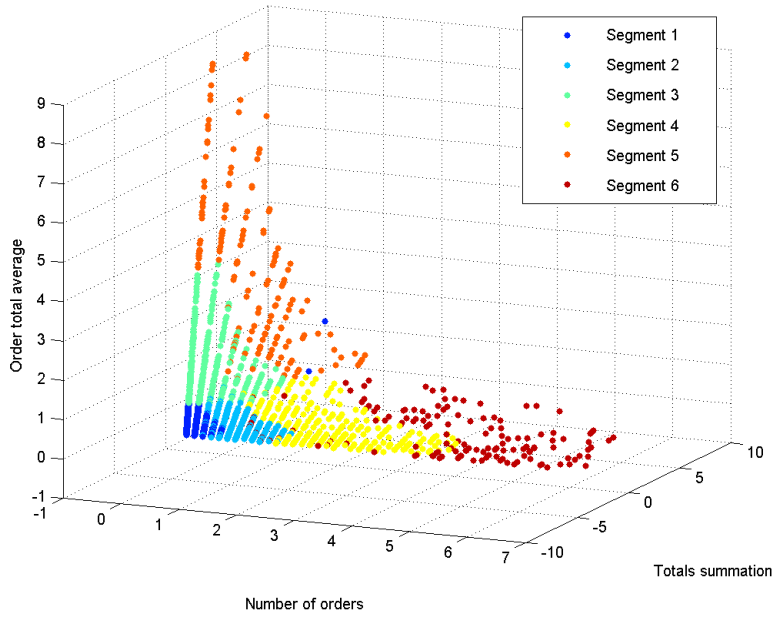


**Figure D.11:** Segmentation of eProcurement supplier data.

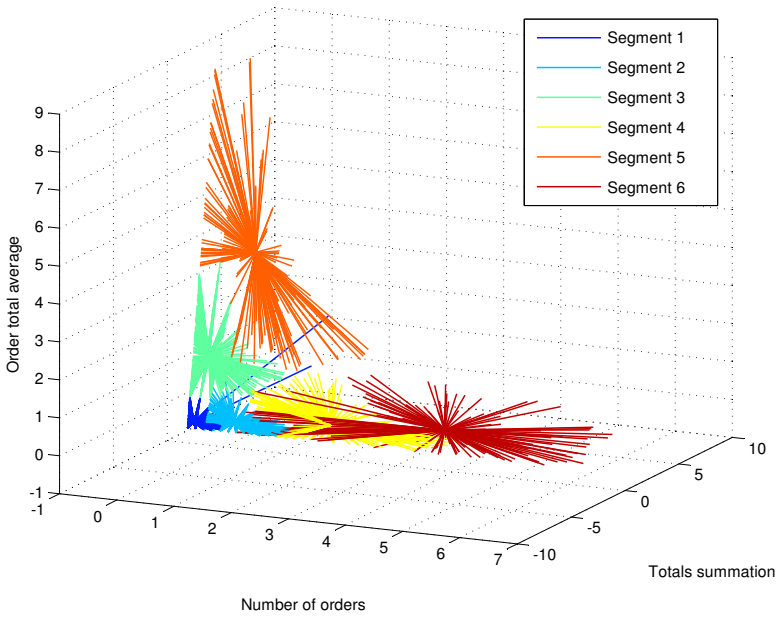


**Figure D.12:** eProcurement supplier data linked to their respective cluster centroids.