DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DATA MINING AND KNOWLEDGE DISCOVERY IN MEDICAL INFORMATION SYSTEMS

by
Yunus DOĞAN

June, 2015
İZMİR

DATA MINING AND KNOWLEDGE DISCOVERY IN MEDICAL INFORMATION SYSTEMS

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Dokuz Eylül University

In Partial Fulfillment of the Requirements for the Degree of Doctor of

Philosophy in Computer Engineering

by

Yunus DOĞAN

June, 2015

İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **“DATA MINING AND KNOWLEDGE DISCOVERY IN MEDICAL INFORMATION SYSTEMS”** completed by **YUNUS DOĞAN** under supervision of **PROF. DR. ALP KUT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

[illegible]

Prof. Dr. Alp KUT

Supervisor

[illegible]	[illegible]
Asst. Prof. Dr. Derya BİRANT	Asst. Prof. Dr. Reyat YILMAZ
Thesis Committee Member	Thesis Committee Member
[illegible]	[illegible]
Prof. Dr. Ahmet Faik KAŞLI	Assoc. Prof. Dr. Aybars UĞUR
Examining Committee Member	Examining Committee Member

[illegible]

Prof. Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Alp Kut, for providing me with this research opportunity, for his patience, enthusiasm, motivation, and enormous knowledge. I greatfully thank to my thesis committee members, Asst. Prof. Dr. Derya Birant and Asst. Prof. Dr. Reyat Yılmaz for their valuable comments and advice in the process of my studies.

I am thankful to the great, current and previous, the members of the department of Computer Engineering, Assoc. Prof. Dr. Görkem Eskiizmir from the department of Head and Neck Cancers at Celal Bayar University and Prof. Dr. Nur Arslan from the department of Children's Diseases and Nutrition at Dokuz Eylül University for their practical and great help in my thesis.

Finally, and most importantly, I would like to thank my grandfather Musa Palit, grandmother Suna Palit, my mother Sebahat Dogan, my father Yaser Dogan, my sister Nergiz Dogan, her husband Kivanc Artun, and other members of my family. I could not work and live happily without their love, support, encouragement, sensitivity, and deep understanding.

Yunus DOĞAN

DATA MINING AND KNOWLEDGE DISCOVERY IN MEDICAL INFORMATION SYSTEMS

ABSTRACT

Hospital Information Management Systems are used in all public and private hospitals. Valuable data which is obtained from these big databases, where updated data are collected continuously during all day, is used for only some necessary reports and queries inside of the corporation, and this data may not be considered for an academic study, because it is not held as a clean data warehouse. In this thesis, it is underlined that this data should be an open source with standard protocols under definite boundaries for academic studies in order to supply the improvement of medical research, and it is mentioned that how this data is processed and valuable patterns are obtained.

There are studies about three branches, “medical laboratories”, “head and neck cancers” and “nutrition and diets” in this thesis. For these three branches, new information systems have been implemented by using the technologies of MSSQL database and ASP.NET with C# programming languages. Thus, it has supplied to collect clean and processable instances in them and, the technique of knowledge discovery and data mining was able to be applied for these information systems by using different algorithm approaches.

In the studies, clustering algorithms from data mining techniques have been implemented with different approaches and firstly, these algorithms have been used to analysis laboratory data sets which necessary permissions have been obtained to use. Secondly, head and neck cancer instances, which necessary permissions have been obtained to use, too, have been analysed by using clustering algorithms with different approaches. Lastly, the data set of nutrition and diets is optimized by hybrid quantum genetic algorithm.

Improved clustering algorithms have been compared with self-organizing map, k-means++ algorithms and its different approaches. Moreover, improved hybrid quantum genetic algorithm has been compared with traditional genetic algorithm and quantum genetic algorithm. As a result, the patterns, which have higher accuracy and performance than the traditional approaches, have been obtained.

Keywords: Medical information systems, data mining and knowledge discovery, medical information systems, clustering algorithms, classification algorithms, mapping algorithms and hybrid quantum genetic algorithm.

TIBBİ BİLİŞİM SİSTEMLERİNDE VERİ MADENCİLİĞİ VE BİLGİ KEŞFİ

ÖZ

Hastane Bilgi Yönetim Sistemleri kamu ve özel hastanelerinin tamamımda kullanılmaktadır. Yirmi dört saat kesintisiz olarak yeni tıbbi verilerin kaydedildiği bu geniş veri tabanlarında, biriken bu değerli veriler günümüzde sadece kurum içinde gerekli raporlama ve sorgular için kullanılmakta ve temiz veri ambarları şeklinde tutulamadığından her hangi bir akademik çalışma için değerlendirilemediğine tanık olmaktayız. Bu tezde tıbbi verilerin belirli sınırlar dâhilinde ve standart protokoller ile akademik çalışmalar için açık kaynak olması gerektiğine vurgu yapılacak, gerçekleştirdiğim tıbbi bilişim sistemleri ve tıbbi uygulamalarda bu verilerin işlenip nasıl kıymetli sonuçların elde edilebileceğinden bahsedilecektir.

Tezdeki çalışmalar “tıbbi laboratuarlar”, “yüz ve boyun kanserleri” ve “beslenme ve diyet” alanları üzerine yapılmıştır. Bu üç alan için MSSQL veri tabanı ve C# programlama dili ile ASP.NET kullanılarak yeni bilişim sistemleri geliştirilmiştir. Böylece, temiz ve işlenebilir verilerin bu sistemlere toplanması sağlanmış ve veri madenciliği ve bilgi keşfi tekniği bu bilişim sistemleri için farklı algoritma yaklaşımları ile uygulanabilmiştir.

Çalışmalarda veri madenciliği tekniklerinden farklı yaklaşımlara sahip kümeleme algoritmaları geliştirilmiştir. İlk olarak, bu algoritmalar etik kurul izinleri alınmış laboratuar verilerinin analizi için kullanılmıştır. İkinci olarak, yine etik kurul izinleri alınmış yüz ve boyun kanser verilerinin analizi için farklı yaklaşımlı kümeleme algoritmalar kullanılmıştır. Son olarak beslenme ve diyet veri kümesi melez kuantum genetik algoritması ile optimize edilmiştir.

Geliştirilen kümeleme algoritmaları kendi kendini düzenleyen haritalar, k-ortalama++ algoritmaları ve onların farklı yaklaşımları ile karşılaştırılmıştır. Bunun yanında, geliştirilmiş melez kuantum genetik algoritması, geleneksel genetik algoritma ve kuantum genetik algoritma ile karşılaştırılmıştır. Sonuç olarak da geleneksel yöntemlere nazaran daha hızlı ve doğruluğu daha yüksek desenler elde edilebilmiştir.

Anahtar kelimeler: Tıbbi bilişim sistemleri, veri madenciliği ve bilgi keşfi, kümeleme algoritmaları, sınıflandırma algoritmaları, haritalama algoritmaları ve kuantum genetik algoritma.

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Medical information systems have very important data sources to analysis. There are lots of patients, and their test results, diagnosis and treatments in these large data sources. The technique of data mining and knowledge discovery needs large data sources to obtain consistent and accurate patterns; medical information systems fulfil this requirement exceedingly. This thesis study is about data mining and knowledge discovery over medical information systems.

1.1 The Aim of the Thesis

The major aim of this thesis is to point out that various data mining algorithms can be applied for various medical information systems which have important and hidden knowledge for human being's healthcare. Another aim of this thesis is that running times of the data mining algorithms in medical information systems and accuracies of obtained consequences and patterns by data mining algorithms in medical information systems are very important for human being's healthcare; therefore, this thesis aims to point out that some additional differences at operations in traditional data mining algorithms can increase accuracy of the consequences, and patterns and decrease running time.

1.2 History and General Concepts

Increase and simplified methods to access information is the main reason behind the late 20th and 21st centuries' being called as communication era. This is caused by rapid improvement in technologies and becoming widespread of information networks. Internet is the most important development that expedited this process is the Internet (Akan, 1999).

In recent years, many factors that designate the life on earth has changed: Community structure, life style, production tools, products and services, type of products have dramatically changed. The main production of the past, manpower has been replaced with brain power. Mechanicals equipment and factories have been replaced with tools and factories equipped with information technologies. The most important resource in the world is human being and the information generated by the human. As the human and its scientific achievements have become more important, developed societies gave priority to the investment on human being rather than industry. Therefore, the concept of "people come first" emerged. These societies prefer marketing the information instead of industrial products. This provides them not only the societies have risk- and problem-free lives but also the opportunity to protect their rapidly contaminated environment. As a result of these policies, there has been a dramatic increase in producing information (Musoğlu, 2000).

Rapid improvements in medicine and information technologies result in collaboration and even parallel developments between these two disciplines. As scientific information gathered in medicine increases, information technologies plays an important role in health services management, data storage and sharing. In contemporary medicine, computers and computer controlled systems bring incredible extends to diagnosis and treatment, accelerates and significantly simplifies them. Correct, trustworthy and fast interpretations of complex and similar cases have been enabled by information technologies (Ay, 2009). Current data in medicine is huge and has vital importance. Big data in medicine has vital importance to analysis. Thanks to hospital information systems, this data is stored regularly. It is possible to benefit more from this database. Data mining studies on the data obtained from hospital and other medical databases can have an effective role for the specialists, hospital management and patients for better qualified health services (Kaya, Bulun, & Arslan, 2003).

1.3 Statement of Thesis

In this thesis, I studied clustering, classification and genetic algorithms to analysis data warehouses in various medical information systems.

Chapter 2 presents the descriptions of the major terms in data mining and knowledge discovery in medical information systems.

Chapter 3 submits the aims of the studies, which explain in next chapters in detail. Motivation and the reasons of preferred algorithms are summarized.

Chapter 4 provides a data mining application for a medical laboratory information system. A new approach, SOM++ is implemented and described in detail. This clustering algorithm is used for laboratory data as a case study. A manuscript describing this new algorithm in this chapter in detail, entitled “SOM++ Integration of Self-Organizing Map and K-Means++ Algorithms with The Sequential Assignment” has been submitted.

Chapter 5 provides a knowledge discovery application to optimize food receipts to obtain "Mediterranean Diet" program. A new approach, hybrid quantum genetic algorithm is implemented for the optimal diet receipt according to the some personal information like height, weight, age, chronicle illnesses, and etc. it is described in detail. A manuscript describing this new optimization algorithm in this chapter in detail, entitled “Mediterranean Diet Optimization by Hybrid Quantum Genetic Algorithm” has been submitted.

Chapter 6 provides a “head and neck cancers” automation and decision support system. A new mapping clustering approach is implemented and described in detail. Also, other new mapping clustering alternatives have been implemented and it is exposed that the regression mapping approach after weighted K-Means++ is the most successful at visualization of clusters and data points on coordinate system. This mapping approach points out that the visualization of the real distances between clusters and data points make decision diagnosis and treatments of patients easy.

Chapter 7 presents general discussion, conclusion and the future of this work.

CHAPTER TWO
MEDICAL DATA MINING

2.1 Data Mining and Knowledge Discovery in Medical Information Systems

Medical information science, standing at the intersection of medicine and information technologies, is a multidisciplinary branch of science that provides solutions to medical cases, decision techniques, biomedical data storage, and fast access to this data and optimum use of it. Medical information science also rationally investigates obtaining, compiling, sharing and application of medical information; determination of treatment and care methods for patients and improvement of these methods (Musoğlu, 2000; Saka, 2003). British Medical Informatics Society defines medical information science as: All the tools, abilities and knowledge that provides usage and sharing of available medical data in order to more effectively provide health services. As a branch of science has been recently developed and worldwide monitored by academic authorities. It investigates and teaches the methods of application of information technologies to health services, the meeting point of health, information technologies, psychology, epidemiology and engineering (ODTÜ Enformatik Enstitüsü Sağlık Bilişimi Anabilim Dalı, 2014).

Considering the high number of studies that are conducted in medicine, the difficulty and delay of applying this literature to practice, medical information technologies can be seen an important method to overcome these problems. In modern world, medicine and healthcare is one of the fields in which the information is used the most. The most intense use of information in the modern world fields As the measurement and visualization methods, testing, analysis and monitoring devices are improved and more widely used, medicine becomes richer and database and amount of information of patients increases fast. Medical information science and creation, compilation and sharing of these databases aim to determine and improve the treatment and care methods for patients (Ay, 2009; Kaya, Bulun, & Arslan, 2003).

The change in the use of information in medicine has affected healthcare service providers and the use of computers, data sharing; teamwork and application based on information concepts have become more common. In addition to providing direct medical services such as supporting patient care services, evaluation of the quality of the healthcare services; computers have begun being more commonly used in managerial and academic functions such as decision making, management, planning and medical research (Kaya et al., 2003)

Data mining is the search of rules and relationships in the existing large amounts of data, which enables us to make predictions for the future (Alpaydın, 2000; Güven, Bozkurt, & Kalıpsız, 2007). Improvements in the following four main areas: Result in new opportunities and research topics in finding data resources, questioning and compiling them, and doing analysis to produce information from them;

1) Technological developments enable us to process more data in less time.
2) Use of computers, therefore everyday more and more people work in digitally and produce more digital data.
3) Information technologies and internet infrastructure rapidly widespread around the world and develop a life independent of location and time.
4) Individuals adopt the culture of decision making with research and judgement based on scientific data. Data mining was born parallel to these improvements and is a hot research topic that focuses on finding undiscovered relationship among data (Güven et al., 2007).

Data mining is the process of discover hidden patterns in big databases. The problems that could take long time to solve with using traditional methods can be solved in a faster way using process of data mining. The main goal of the data mining is to find out hidden patterns of data, to increase the value of data, and transform data into information. In today’s world, data mining is widely used in various areas such as banking, marketing, insurance, telecommunication, stock market, health, industry, science and engineering (Küçüksille, Taşdelen, & Aydoğan, 2006).

Data mining and the knowledge discovery mainly consist of five stages:

1- Data Selection 2- Pre-processing 3- Transformation/Reduction

4- Data Mining 5- Interpretation/Evaluation (Baykal, 2006; Koyuncugil & Özgülbaş, 2009; Yıldırım, Uludağ, & Görür, 2008).

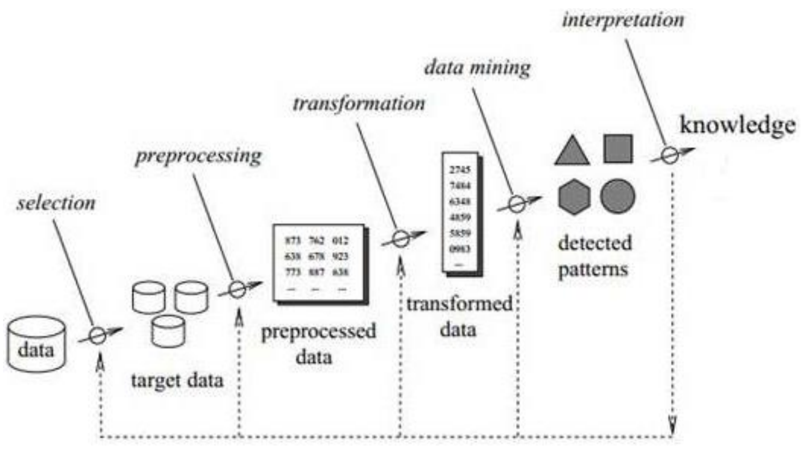


Figure 1.1 Steps of data mining and knowledge discovery (Sqldatamining, 2015)

2.2 Data Mining Methods

Data mining methods are classified into three groups based on their functions: "Classification and Regression", "Clustering", and "Association Rules" (Güven et al., 2007; Özekeş, 2003).

2.2.1 Classification and Regression

Classification and Regression are the two data analysis methods that can produce important data classes or create models that can predict future inclination. While classification predicts categorical variables, regression is used to predict continuous variables. The main techniques that are used in classification and regression are:

1) Decision Trees 2) Artificial Neural Networks 3) Genetic Algorithms

4) K-Nearest Neighbour 5) Memory Based Reasoning

6) Naive-Bayes (Güven et al., 2007; Kaya & Köymen, 2008; Özekeş, 2003).

2.2.2 Clustering

Clustering is the process of grouping data. Objects in the same group are more similar to each other than to those in other groups, clusters. Unlike classification, there are no data classes in clustering. In some applications, clustering can be pre-process of classification. Choice of clustering algorithm that is going to be used depends on the data type and the purpose. The main clustering methods are:

1) Partitioning methods
2) Hierarchical methods
3) Density-based methods
4) Grid-based methods
5) Model-based methods (Güven et al., 2007; Kaya & Köymen, 2008; Özekeş, 2003).

2.2.3 Association Rule Mining

Association rule mining is used to find association relationships among big data groups. Since collected and stored data amounts increase day by day, health institutions want to find out association rules in databases. Figuring out interesting association relationships in huge records of medical operations, make decision making process of health institutions more effective. The most typical use of association rules is market basket analysis. This method, by finding out association relationships among the applications to the individuals that apply to the health institution, analyses the determination and improvement of treatment and care methods applied to the patients (Güven et a.l., 2007; Jonsdottir, Hvannberg, Sigurdsson, & Sigurdsson, 2008; Özekeş, 2003).

2.3 Supervised and Unsupervised Data Mining

Data mining methods can be classified into two main categories: Supervised and unsupervised data mining. When data mining has a well-defined certain objective, it is called supervised data mining. If no special definition has been made for the objective and there is ambiguity, it is called unsupervised data mining (Baykal, 2003; Koyuncugil & Özgülbaş, 2009).

Table 2.1 Supervised and Unsupervised Data Mining Methods

Supervised Data Mining Methods	Unsupervised Data Mining Methods
• Decision trees • Neural networks • K-Nearest Neighbour • K-means Clustering • Regression Models • Rule Induction	• Hierarchical Clustering • Self-organized Maps

The commonly used methods in data mining are regression models, k-Nearest Neighbour and clustering methods. In addition, decision trees, association rules, and neural networks are the new generation techniques in data mining (Baykal, 2003; Koyuncugil & Özgülbaş, 2009; Yıldırım et al., 2008).

2.4 Challenging Problems (Major Issues) in Data Mining

Data mining consists of databases to provide raw data as input. This causes problems if databases do not have dynamic, incomplete, large and clear data. The problems can be classified into three main groups:

Limited Data: Databases are usually prepared for the purposes such as presenting properties or qualifications that provide simple learning. Thus, some properties that simplify learning task cannot be found. For instance, if the patient database does not include red blood cell test results of the patients, malaria diagnosis cannot be concluded from that database. Size of database: Database sizes are incredibly increasing. Database algorithm is created to deal with many small samples. Use of the same algorithm for hundreds times bigger samples requires to be very careful

Although, having big samples is an advantage in terms of prediction accuracy, errors due to lack of care cannot be ignored.

Outlying and missing data: The out of system errors that occur during data input or data collection are called noise. The higher the noise in data means the more difficult to obtain trustworthy results. To detect data with noise histogram, clustering and regression analysis are used. Missing data is caused by huge size or nature of the database. Things to do in case of missing data: Record or records that have missing data can be removed, the average value of the variable can be used instead of the missing data, or the best suitable value can be found out based on the existing data (Baykal, 2006; Akgöbek & Çakır, 2009).

2.5 Literature Review in Medical Data Mining and Knowledge Discovery

Chen et al. (2010) examined the risk factors of parenting stress. To do this, Taiwan Birth Panel Study research group obtained the data from a total of 206 mother gave birth to baby in the National Taiwan University between April 2004 and January 2005. Data mining with decision tree C5.0 depicting the classification of risk factors is better than the regression model (Chen, Hou, & Chuang, 2010).

Toussi et al. (2009) used a database of 463 type 2 diabetic patient records at Avicenne University Hospital in France to analyse physicians' therapeutic decisions using C5.0 decision-tree learning algorithm. 11 and 13 out of the 46 rules from the analysis of the database in "the French National Guidelines for the management of type 2 diabetes were about the choice of the type of treatment and the choice of pharmaco-therapeutic class of each drug, respectively. Only a few rules were extracted due to very few numbers of patient records. Similarities between the extracted rules and those added to the new clinical guidelines were detected (Toussi, Lamyl, Toumelin, & Venot 2009). Phillips-Wren, et al. (2008) assessed the utilization of healthcare resources by lung cancer patients. This study has been carried on the records of 4365 out of 1.238.895 patients who were benefited from health insurance after 65 years old in 1999 and monitored by physicians between

1997 and 2001. It was reported that data mining with the combination of decision trees and artificial neural networks is better identifier and predictor than solely logistic regression (Phillips-Wren, Sharkey, Dy, 2008).

Türe et al. (2009) aimed to determine a new prognostic index for the analysis of the subgroups of breast cancer. For this purpose, they conducted a study on 381 breast cancer patients diagnosed between 1997 and 2007. The techniques of cox regression analysis and decision-tree algorithms (C&RT, CHAID, QUEST, ID3, C4.5 and C5.0) were performed to obtain new prognostic indexes. C4.5 decision tree method performed better than other decision tree techniques and cox regression to identify risk groups (Türe, Tokatlı, & Omurlu, 2009).

Hu et al. (2006) evaluated non-invasive intracranial pressure using a database composed of 30-minute long measurements of arterial blood pressure, intracerebral pressure, cerebral blood flow velocity, from nine traumatic brain injury patients to test database   The data warehouse for data mining is used to test and bigger data warehouse has been suggested for future works (Hu, Nenov, Bergsneider, & Martin, 2006).

Trifiro et al. (2009). In this study it is aimed to form a list of high priority pharmacovigilance cases in digital health records database. Using a data mining method, signal detection, 23 adverse drug effects have been identified. The most common ones among these are detected as, bulla skin loss, acute renal failure, anaphylactic shock, myocardial infarction rhabdomyolysis. New information and problems are noted for the safety of the drug and the necessity of updating the list is underlined (Trifiro et al., 2009).

Silva et al. (2008) predicted intensive care unit (ICU) organ failure using a database with a total of 25,215 daily records from 4425 patients in 42 European ICUs. Two data mining techniques: Multinomial logistic regression and artificial neural networks were compared. The technique of artificial neural networks was found as the best performer and can be used to develop intelligent clinical alarm monitoring (Silva, Cortez, Santos, Gomes, & Neves, 2008).

Jonsdottir et al. (2008) Data collected from 257 patients that are diagnosed with breast cancer have been used. A model selection tool (MTS) was used to find out the best algorithm for clustering. It was observed that all the algorithms performed similarly (Jonsdottir et al., 2008). Kahramanlı & Allahverdi (2008). An artificial immunization algorithm, Opt-aiNET (OPTBP), is used to develop a rule out of artificial neural network (ANN), and a rule set is formed for heart diseases.

The database of Clevealand Heart Diseases, which had 303 samples, was used from the University of California Irvine Machine Learning Repository and OPTBP was applied. This algorithm was noted to be successful; however, it is noted to generate too much data. It is indicated that a tool that would generate less data should be developed (Kahramanlı & Allahverdi, 2008).

Doğan & Türkoğlu (2008) used the On the biochemical test results of 472 patients association rule technique of data mining was used, and a decision support system for hypertroid diagnosis was developed and tested.. One to one correlation between the results of the developed system and the decision of the physicians clearly conformed the effectiveness and reliability of the developed system (Doğan & Türkoğlu, 2008).

Vepa (2009) used 45 normal, 42 systolic, 43 diastolic, total 130 heart beat records to classify heart murmurs. Data mining methods of support vector machine (SVM), k-nearest neighbour (KNN), multilayer perceptron (MLP) and neural networks were used. The best performance was obtained by SVM method and KNN method was the second (Vepa, 2009).

CHAPTER THREE

MOTIVATION

The major source of motivation in this thesis is to assemble two disciplines, computer science and medicine. Medical information systems achieve very rich inputs for computer sciences to analysis; that is, while deciding about diagnosis and treatments of patients by doctors, intelligent algorithms of data mining support doctors. Accordingly, interdisciplinary studies have been implemented in this thesis.

While literature researches about data mining algorithms, it has been observed that especially clustering algorithms and evolutional algorithms are open for improvement. Moreover, medical decision support systems involve high accuracy rate and performance. Thus, another motivation in this thesis has been to be able to notice implementing new approaches of traditional data mining algorithms.

Hospital Information Systems are used actively in almost all hospitals in Turkey and World. However, these systems have not got a suitable background to collect clean and utility data. These systems have been implemented for routine governmental operations; therefore, doctors archive information of their patients as hard copies and independently from hospital information systems. Thus, doctors cannot deduct patterns and consequences from these archive papers for medical researches as it should be. As a result, another motivation in this thesis has been to be able to obtain clean data bases in clean information systems.

There are studies about three branches, “medical laboratories”, “head and neck cancers” and “nutrition and diets” in this thesis. Next sections submit the reasons of choices of these three branches and implemented algorithms for these branches.

3.1 Medical Laboratories

Medical laboratories use a laboratory information system and collect all test results in data bases in this system. Laboratory information system module, which is a part of a hospital information system, covers the period from the acquisition of the medical data until the communication of this data. During this process, generated data is transferred to databases and safe stored in data centres. High number of patients and variability of the generated data, make data mining methods an effective way to process this data. Examination of the laboratory test results by an expert doctor makes it more possible to obtain meaningful information and provide treatment planning and management.

The recent developments in laboratory techniques and the increase in the results obtained out of tested materials make it harder for expert doctors to make decisions about making similar treatment proposals for patients that have similar test results. To uncover meaningful and hidden information in huge data storages and increase the pace and quality of the health system, special systems should be developed and used by data mining techniques.

3.2 Head and Neck Cancers

Cancer (malignant tumour or malignant neoplasm), is a group of diseases involving abnormal cell growth. Over 100 different known cancers affect humans beings nowadays (National Cancer Institute, 2014). This group of diseases invade and spread to other parts of the body (World Health Organization, 2014; NCI, 2014); as a result, life-threatening destructions in body are occurred. Early diagnosis is very important at recovery. The rate of fully recovery at the group of “head and neck cancers” is observed as approximately 100% at early diagnosis (American Cancer Society, 2014). Therefore, the branch of “head and neck cancers” has preferred to analysis by data mining algorithms and a decision support system has been implemented.

3.3 Nutrition and Diets

A healthy diet is important for maintaining or improving general health, and preventing many chronic health risks, such as obesity, diabetes, hypertension and cancer. A healthy diet involves consuming appropriate amounts of all nutrients and it needs to have a balance of fats, proteins, carbohydrates, energies, vitamins, and minerals. Nutrients can be obtained from many different foods, so there are various diets that may be considered healthy. Especially, Mediterranean Diet that we have mentioned in this approach is one of the healthiest diets applied worldwide. Its success was proved by several clinical studies.

3.4 Clustering Algorithms

Clustering algorithms are the base of the data mining analysis. The other data mining techniques, association rule mining and classification are fed back from clustering. If it is illustrated for classification; after a clustering analysis, a set of clusters is obtained and each data is assigned to a cluster in the set. That is to say, each data in data set attains a new feature as a target. This target feature is the cluster which the row data belongs to. As a result, classification algorithms can analysis this data set which is extended with a new feature after clustering analysis and classification analysis may need clustering analysis before itself.

If another scenario is illustrated for association rule mining; association rule mining algorithms are interested in frequencies and equivalences of each row data in data set while mining the associations. Therefore, if there is a big data set to discover hidden secrets in it, association rule mining algorithms use a kind of clustering approach by considering equivalence between association combinations. The clusters, which have the highest frequencies according to threshold for example minimum support and confidence values, are presents as result associations. As a result, association rule mining is a significant sub-technique of clustering techniques for big data.

3.4.1 SOM++ Integration of Self-Organizing Map and K-Means++ Algorithm

In SOM, initial weight values are assigned randomly, method performance is sensitive to these values and it is prohibitively slow in large-scale applications. In order to decrease the time complexity of SOM, I investigated different initialization procedures for optimal SOM and now propose K-Means++ as the most convenient method, given the proper training parameters.

This approach proposes a new clustering algorithm SOM++, which is composed by K-Means++ method followed by SOM clustering. The algorithm consists of two stages First, using K-Means++ method to determine the initial weight values instead of assigning randomly, then clustering task is done in the second stage by SOM as an unsupervised clustering method. The experimental results show that the proposed algorithm, SOM++, is considerably better than the conventional SOM based algorithms in terms of runtime, the rate of unstable data points and internal error. It generates similar clustering results with other SOM based clustering algorithms, however the use of it requires the smaller training time (Doğan, Birant, & Kut, 2013).

3.4.2 PSOMDM Parallel SOM by using Division Method

SOM algorithm requires the high execution times to train the map and this situation put a limit to its application in many high-performance data analysis application domains. For this reason, it is necessary to develop a parallel implementation of SOM by either partitioning the network among the processors (network partitioning) or by partitioning the input data across threads or multi-core processors (data partitioning).

This novel approach aims for faster clustering by using of SOM, entitled PSOMDM (Parallel SOM by using Division Method). PSOMDM is different from existing architectures in that it divides the map area constantly and lower number of data is clustered on different neurons by parallel method. PSOMDM has many advantages over conventional SOM based methods. The most remarkable advantage of PSOMDM is in saving training time for clustering large and complicated data sets by using special division method. Furthermore, the proposed division method provides higher accuracy by decreasing the number of unstable data points and internal errors (Doğan, Birant, & Kut, 2011).

3.4.3 RMWC Regression Mapping with Weighted Clustering

If there is not big data, association rule mining algorithms fails and cannot obtain any association. On such an occasion, clustering algorithms can use to analysis and mine the associations without consideration of the frequencies and equivalences, thus consequences as fuzzy associations can be obtained by explication centre points of clusters. In medical studies, a fuzzy structure is wanted by doctors, because certain decision about their patients must be done by the doctor.

A new approach for clustering, RMWC makes a successful visualization of data on two-dimensional map by using weighted clustering approach. Weighted clustering returns a tree from C4.5 decision tree algorithm and weight values are assigned to attributes in the data set according to the distances from the root of the returned tree. (Doğan, Birant, & Kut, 2011). Thus, doctors see only a map where all instances or clusters are located according to their real distances between each other. Then, they can interpret the pattern without any certain resolutions.

3.5 HQGA Hybrid Quantum Genetic Algorithm

Hybrid Quantum Genetic Algorithm is a new approach for genetic algorithm to optimize a problem which has a big data set in a big search space. In this approach, this algorithm has been implemented to obtain the personal optimized menu. It is based on the techniques of Quantum Genetic Algorithm and also, has extra methods to obtain the sensitive solution of values of optimum energy, protein, fat, carbohydrate and inorganic compounds with vitamins.

CHAPTER FOUR
LABORATORY INFORMATION SYSTEMS AND SOM++

4.1 Laboratory Information System

Today, laboratory test results are used to control early diagnosis and cure of many diseases. Using hospital information systems, these results are saved to databases and sent to data storage systems. These results are saved to the databases of the information systems of the hospitals. Laboratory information system (LIS), which is a part of hospital information system (HIS), provides examination of medical samples for doctors to solve patient cases, and displays and communicates test results. LIS contains the data that forms the richest content of HIS. This content archive can be accessed through the data storage in the hospital data centres by authorization. In this chapter, it is aimed to process the data that has been generated from three months data set of a private laboratory by data mining techniques, and to create a special system that is believed to serve medicine.

4.2 Medical Laboratory Tests

The usual biological samples that are taken from patients to diagnose a disease or to monitor the success of a treatment are blood, urine and stool. Medical laboratory tests are divided into 4 groups based on the samples taken and the type of examination

1. Laboratory analyses performed on blood samples

a. Biochemical tests; albumin, alkaline phosphatase, alanine aminotransferase (ALT), aspartate aminotransferase (AST), acid phosphatase, amylase, bilirubins, iron, iron binding capacity, phosphorus, gamma-glutamyl transpeptidase (GGT), glucose, globulin, HDL-cholesterol, LDL-cholesterol, calcium, chlorine, cholesterol, creatinine, creatinine phosphokinase, creatinine phosphokinase isoenzymes, lactate dehydrogenase (LDH), potassium, sodium, transferrin, triglycerides, urea, uric acid.

b. Hematology tests; basophils, eosinophils,erythrocytes, lymphocyte, leukocyte, monocytes.

c. Hormone tests; thyroid-stimulating hormone (TSH), triiodothyronine (T_3), free T_3, thyroxine (T_4), FT_4, estradiol (E2), progesterone, follicle-stimulating hormone (FSH), luteinizing hormone (LH), beta human chorionic gonadotropin (hCG), prolactin, testerone, dehydroepiandrosterone sulfate (DHEA-SO_4).

d. Tumor markers; alpha-fetoprotein (AFP), carcinoembryonic antigen (CEA), prostate-specific antigen (PSA), cancer antigen 15-3 (CA15-3), carbohydrate antigen 19-9 (Ca19-9), cancer antigen 125 (Ca125), cancer antigen 50 (Ca50), cancer antigen 72-4 (Ca72-4), neuron-specific enolase (NSE), squamous cell carcinoma antigen (SCC), beta-2 microglobulin, thyreglobuline.

2. Urine and and feces screening tests

a. Urine; bilirubine in urine, urine density, glucose in urine, blood in urine (hematuria), urine ketones, the pH of urine, protein in urine, urobilinogen.

b. Feces; fecal occult blood, fecal parasites.

3. Pregnancy test Human chorionic gonadotropin (HCG) level

4. Culture tests Culture; sputum, throat, nose, fecal, eye, urine, ear, wound culture.

Depending on the calibration of devices, the values of a healthy person in the results of above mentioned four main test groups are called as reference values. The results that are within the interval of the reference values are classified and stored as normal, whereas higher and lower results are classified and stored as high and low respectively. Reference values vary deeding on the age and sex of individuals (Middle East Technical University Medical Center, 2014).

4.3 Medical Laboratory Information System

Dokuz Eylül University Centre Medical Laboratory uses the laboratory information system which has been implemented by a software team where I have been. This laboratory examines thousands of people every day and LIS collects thousands of test results. Also, all steps are followed by LIS from taking blood to confirmation tests results by expert doctors and they are recorded in LIS. This rich database is precipitated to analysis by data mining algorithms significantly. Figure 4.2 shows a sample interface from LIS.

In the 3 months data sample of laboratory, in which medical samples are examined, there exist 39 properties and 650625 instances which belong to 26303 individuals. Pre-processing covers, the selection of patients and tests, and assignment of values between 0 and 1 to the data obtained after this selection by using min-max normalization method. During the process of data selection out of data sets, the instances that have null value and or values between reference values are excluded. After pre-processing, the study has been carried on 39 properties belonging to 18781 individuals. Since there is no identifying class information such as the result of the disease, unsupervised clustering method was selected.

Self-Organizing Map (SOM) has been applied on a 600X600 map; therefore, all instances stretch on 180000 cells. Then, K-Means algorithm has been applied on this map. Figure 4.1 shows that the maps of applications. In this application, it has been observed that SOM with a 600x600 map takes too many running time approximately 2 weeks; therefore, another approach as SOM++ instead of traditional SOM has been implemented and described in the next section in detail.

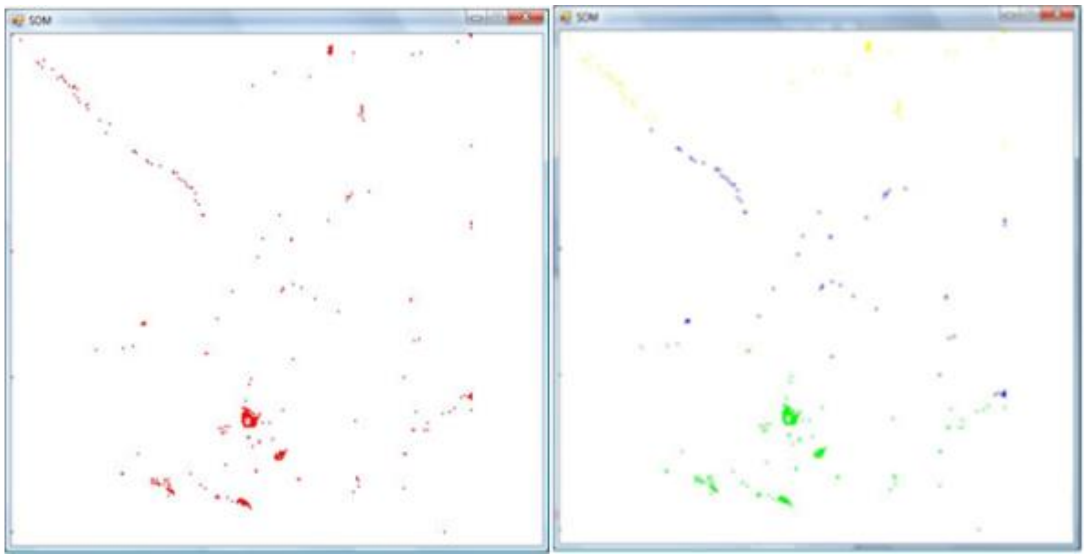
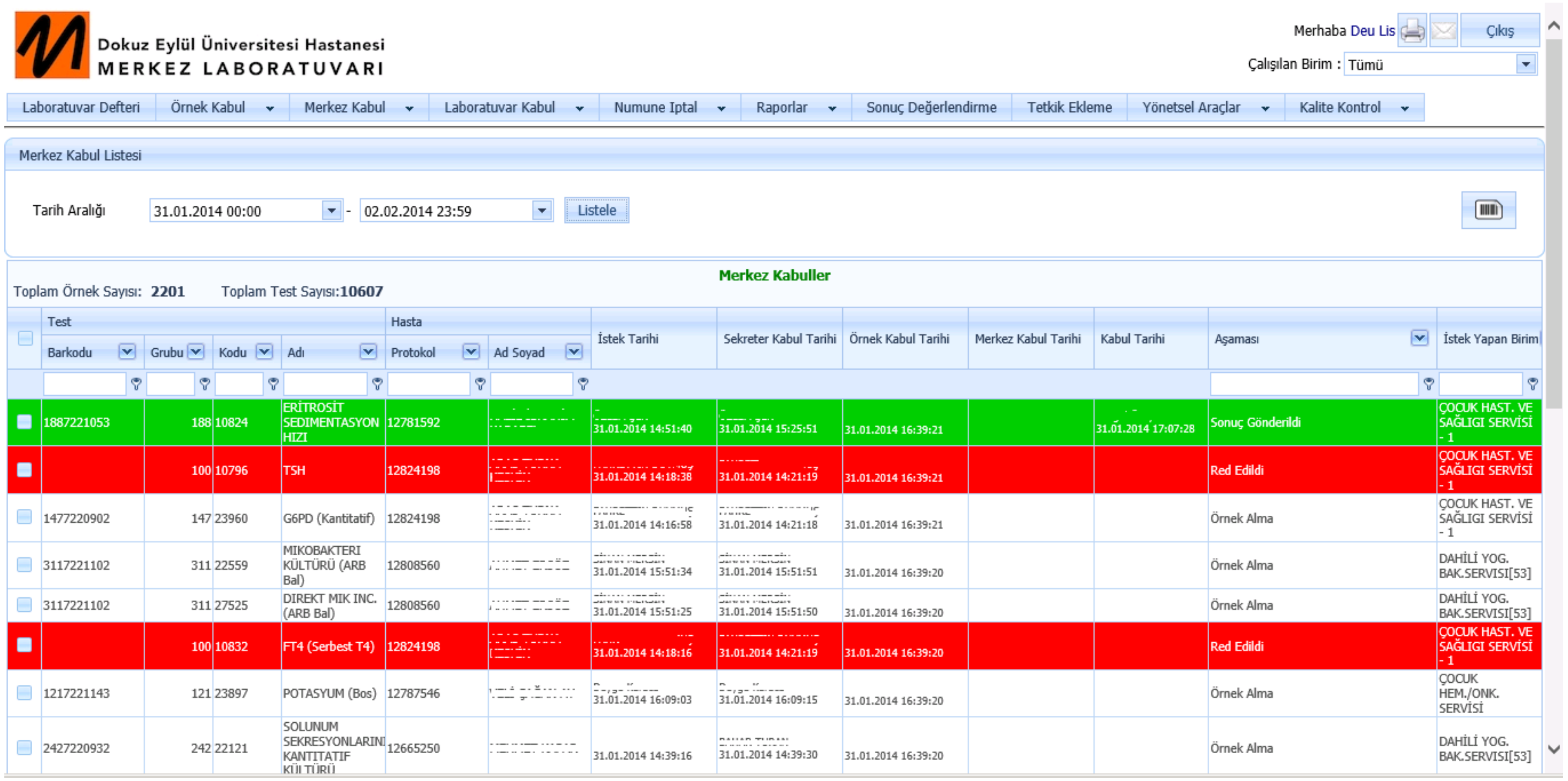


Figure 4.1 Implementation of SOM on 600x600 map, then K-Means k = 3 on this map


Dokuz Eylül Üniversitesi Hastanesi
MERKEZ LABORATUVARI

Merhaba Deu Lis | Çıkış

Çalışılan Birim : Tümü

Merkez Kabul Listesi

Tarih Aralığı 31.01.2014 00:00 - 02.02.2014 23:59 Listele

Merkez Kabuller

Toplam Örnek Sayısı: **2201** Toplam Test Sayısı:**10607**

Test				Hasta								
Barkodu	Grubu	Kodu	Adı	Protokol	Ad Soyad	İstek Tarihi	Sekreter Kabul Tarihi	Örnek Kabul Tarihi	Merkez Kabul Tarihi	Kabul Tarihi	Aşaması	İstek Yapan Birim
1887221053	188	10824	ERİTROSİT SEDIMENTASYON HIZI	12781592	[redacted]	[redacted] 31.01.2014 14:51:40	[redacted] 31.01.2014 15:25:51	31.01.2014 16:39:21		31.01.2014 17:07:28	Sonuç Gönderildi	ÇOCUK HAST. VE SAĞLIGI SERVİSİ - 1
	100	10796	TSH	12824198	[redacted]	[redacted] 31.01.2014 14:18:38	[redacted] 31.01.2014 14:21:19	31.01.2014 16:39:21			Red Edildi	ÇOCUK HAST. VE SAĞLIGI SERVİSİ - 1
1477220902	147	23960	G6PD (Kantitatif)	12824198	[redacted]	[redacted] 31.01.2014 14:16:58	[redacted] 31.01.2014 14:21:18	31.01.2014 16:39:21			Örnek Alma	ÇOCUK HAST. VE SAĞLIGI SERVİSİ - 1
3117221102	311	22559	MIKOBAKTERI KÜLTÜRÜ (ARB Bal)	12808560	[redacted]	[redacted] 31.01.2014 15:51:34	[redacted] 31.01.2014 15:51:51	31.01.2014 16:39:20			Örnek Alma	DAHİLİ YOG. BAK.SERVISI[53]
3117221102	311	27525	DIREKT MIK INC. (ARB Bal)	12808560	[redacted]	[redacted] 31.01.2014 15:51:25	[redacted] 31.01.2014 15:51:50	31.01.2014 16:39:20			Örnek Alma	DAHİLİ YOG. BAK.SERVISI[53]
	100	10832	FT4 (Serbest T4)	12824198	[redacted]	[redacted] 31.01.2014 14:18:16	[redacted] 31.01.2014 14:21:19	31.01.2014 16:39:20			Red Edildi	ÇOCUK HAST. VE SAĞLIGI SERVİSİ - 1
1217221143	121	23897	POTASYUM (Bos)	12787546	[redacted]	[redacted] 31.01.2014 16:09:03	[redacted] 31.01.2014 16:09:15	31.01.2014 16:39:20			Örnek Alma	ÇOCUK HEM./ONK. SERVİSİ
2427220932	242	22121	SOLUNUM SEKRESYONLARINI KANTITATIF KÜLTÜRÜ	12665250	[redacted]	31.01.2014 14:39:16	[redacted] 31.01.2014 14:39:30	31.01.2014 16:39:20			Örnek Alma	DAHİLİ YOG. BAK.SERVISI[53]

Figure 4.2 A sample web page of LIS in Dokuz Eylül University Centre Laboratory

4.4 SOM++ Integration of Self-Organizing Map and K-Means++ Algorithm

Cluster analysis is the process of grouping data into subsets such that each item in a cluster is more similar to the items in the same cluster than to the other items at the outside of the cluster. Generally, distance measures like Euclidean distance, Manhattan distance are utilized to evaluate the dissimilarity between data points. Cluster analysis is one of the most useful tasks in machine learning and data mining, and has been used in a variety of fields such as marketing, banking, medicine and telecommunication. It has been widely used in dimensionality reduction, information extraction, density approximation and data compression.

The K-means algorithm (MacQueen, 1967) is the most commonly used partitioning cluster algorithm with its easy implementation and its efficient execution time. Self-organizing map (SOM) (Kohonen, 1990) is an unsupervised, well-established and widely used clustering technique.

K-Means++ algorithm gives more successful results than standard K-Means at accuracy and consistency (Arthur & Vassilvitskii, 2007). Because, the K-Means algorithm works only to find a local optimum and this local optimum often becomes poor by using random initial centre points; however, K-Means++ starts with rational initial points, thus K-Means++ approximates the best clustering space. Also, K-Means++ outperforms at speed, too. K-Means++ guarantees O(logk) as the complexity time; however, K-Means has a complexity time as $O(n^{kd + 1} \log n)$, where k is the number of clusters, n is the number of data and d is the Euclidean distance between two clusters (Arthur & Vassilvitskii, 2007).

4.4.1 Clustering Algorithms

4.4.1.1 SOM

SOM is an unsupervised neural network which enables clustering according to data similarities and learns the patterns within the data itself, without any external supervision and without preliminary knowledge of the process. SOM is composed of multiple units called cells or “neurons” which can be further grouped into clusters using similarity measures. SOM consists of two layers of artificial neurons: an input layer and an output layer. The input layer is fed into feature vectors, so it is the same as the number of dimensions of the input feature vector. Output layer, also called the output map, is usually arranged in a regular two dimensional structure such that there are neighbourhood relations among the neurons. Every neuron in input layer is fully connected to every output neuron, and each connection has a weighting value attached to it.

The major goal of SOM is to determine the suitable weight values for the neurons according to dataset. The count of these weight values is equal to the number of attributes in dataset and each weight value corresponds to an attribute. At the beginning of the SOM algorithm, these weight values in all neurons are initialized randomly. Secondly, the best matching unit as the winner neuron is found by calculating Euclidean distance from each weight to the chosen sample vector which consists of the weights set in one of neurons. After finding the best matching unit, all vectors of the SOM are updated by using Gaussian function. These processes are repeated until a certain number of iterations. The details of Gaussian function and SOM algorithm are given in Section 4.4.2.

The complexity of SOM algorithm is O(NC), where N is the input vector size and C is the number of dataset presentation cycles. N contains n^2w as the multiply of the map size n^2 and the number of weights w. C contains n^2a as the multiply of the map size n^2 and the number of attributes a. The number of attributes is equal to the number of weights; therefore, the complexity of SOM algorithm obtains $O(N^2)$ (Roussinov & Chen, 1998). While simple SOM has been previously used in many applications, extended versions of SOM has also been proposed also proposed such as FSOM (Fast SOM) (Sagheer, Tsuruta, Maeda, Taniguchi, & Arita, 2006), ABSOM (Ant Based SOM) (Chi & Yang, 2006), and ESOM (Emergent SOM) (Poelmans, Elzinga, Viaene, Van Hulle, & Dedene, 2009).

4.4.1.2 K-Means and K-Means++

K-Means is a partitioning cluster algorithm by grouping n vectors based on attributes into k partitions, where k < n, according to the measure of Euclidean distance. The name of K-Means comes from the fact that k clusters are determined and the centre of a cluster is the mean of all vectors within this cluster.

The main concept of this method is to define k appropriate centroids, one for each cluster. The algorithm starts with k randomly generated centroids, then assigns vectors to the nearest centroid using Euclidean distance and re-computes the new centroids as means of the assigned data vectors. This process is repeated over and over again until vectors no longer changed clusters between iterations. At the end of the algorithm, every object is assigned to the only one cluster.

Similar to the other algorithms, K-means method also has some weaknesses. It is considered to be unstable; running the procedure several times gives several different cluster solutions, so multiple trials are necessary to determine better solution. In addition, it is difficult to specify the efficient number of clusters. Depending on its initial condition, the algorithm may be trapped in the local optimum.

One of the most important issues in K-Means method is the initialization procedure that ultimately determines the number of iterations to run before stopping. Arthur and Vassilvitskii propose a specific way to choose centres for the K-Means algorithm, instead of generating randomly. They call K-Means++, which weighs the data points according to their squared distance squared from the closest centre already chosen. Through new seeding method, K-Means++ yields a much faster, and consistently finds better clusters than K-Means (Arthur and Vassilvitskii, 2007).

When the performance of the K-Means++ algorithm were evaluated on four datasets and K-Means++ consistently outperformed K-Means, both by completing faster and by achieving a lower potential value. For example, on one dataset, K-Means++ terminates almost twice as fast while achieving potential function values about 20% better, on the larger dataset, it is obtained up to 70% faster and the potential value is better by factors of 10 to 1000. For this reason, I propose K-Means++ algorithm in this approach, instead of K-Means.

4.4.1.3 SOM + K-Means

There have been a number of studies comparing SOM with K-Means in the literature. Different authors point out different results, in other words, no definitive results have emerged and conclusions seem to be ambivalent. Some authors (Watts & Worner, 2009) conclude that the quantization errors produced by K-means is consistently lower, the information entropy of the target clusters consistently higher and the computational efficiency of K-means is also many orders of magnitude greater than the corresponding values for SOM. However, some authors (Bação, Lobo, & Painho, 2005) conclude that SOM outperforms in terms of three measures: Quadratic error, mean classification error and the structural coherence of the groups. Therefore, algorithms should be evaluated them under specific conditions and compared particularly with different problems and tasks.

In data analysis techniques, the capabilities of K-Means and SOM for clustering large datasets have already been confirmed in many studies. Even though these clustering methods have their superior features for cluster analysis, their combination as a two-stage method is generally much more powerful than individual methods. For this reason, applications based on SOM and K-Means as two-stage methods have also been proposed for different areas such as the computer security (Tjhai, Furnell, Papadaki, & Clarke, 2010), the healthcare (Leão, Neto, & Sousa, 2009), ecological modelling (Bedoya, Novotny, & Manolakos, 2009), and the financial sector (Deng, & Mei, 2010). All these studies firstly apply SOM to produce the map, and then use K-Means method to further define the boundary between the data by clustering of the SOM neurons. Nevertheless, differently from all these studies, our approach proposes the exact opposite way.

Recently, performing K-Means method after usage of SOM was also studied for different purposes as examples of the emergency planning to deal with extreme events such as earthquake, flood and fire (Yang & Rong, 2008), clustering meteorological data (Khedairia & Khadir, 2008), the biological wastewater treatment process (Aguado, Montoya, Borras, Seco, & Ferrer, 2008), the identification of day types of electricity load (Benabbas, Khadir, Fay, & Boughrira, 2008), and clustering text documents (Xinwu, 2008). Our proposed algorithm, SOM++ is a general clustering algorithm; as a result, it can be used in many different applications for different purposes.

In recent years, several studies have compared different SOM-based two-stage methods. For instance, while Souza et al. (Souza, Ludermir, & Almeida, 2009) compared SOMK (SOM+K-Means) with SOMAK (SOM + Ant K-Means), Chi & Yang (2008) compared both ABSOM (Ant-based Self-Organizing Map) with Kohonen's SOM individually, and SOM+K-Means with ABSOM+K-Means. Besides, Chiu et al. (Chiu, Chen, Kuo, & Ku, 2008; Chiu, Kuo, & Chen, 2009) compares four approaches simple K-Means, SOM+K-Means, PSO+K-Means (Particle swarm optimization (PSO)) and PSHBMO (Particle Swarm Optimization with Honey Bee Mating Optimization). As another example, the study of Corrêa & Ludermir (2006) about the comparison of several SOM-based two-stage approaches: SOM, SOM+KM (K-Means), SOM+W.KM (Weighted K-Means), SOM+AY (proposed by Azcarraga and Yap) and SOM+W.AY (Weighted AY Method), in terms of classification accuracy and runtime. To the best of our knowledge, however, this approach is the first in performing K-Means method before training neurons by usage of SOM to determine the initial weight values of SOM. Moreover, differently from other applications, I propose the integration of SOM with K-Means++ instead of K-Means for faster clustering.

4.4.2 SOM++ Algorithm

The study of Su et al. shows that the initializing the weight values increases the performance of SOM (Su, Liu, & Chang, 2002). SOM++ shows that initializing the weight values by K-Means++ (without K-Means clustering) increases the performance of SOM. Also, the study of Attik et al. mentions that the initializing the weight values by K-Means clustering increases the performance of SOM without any example or method (Attik, Bougrain, & Alexandre, 2005); SOM++ is a supportive and integral study of these studies with K-Means++ (without K-Means clustering) and a new sequential assignment algorithms. In this section, it is explained that our new algorithm SOM++, a two-stage clustering algorithm uses the combination of two data mining techniques, namely SOM and K-means++ clustering. SOM algorithm uses neurons for all points on its map and these neurons have weight values for all attribute values. Before showing the details of SOM++ algorithm, these weight values are indicated in the following part.

4.4.2.1 Weight Values

In SOM, input neurons are fully connected to output neurons, and each connection has a weighting value. In the initialization process of SOM, each neuron is associated with a random weight vector ($w_i = w_{i1}, w_{i2}, \ldots, w_{in}$), which has the same dimension (n) as the input vector ($x_i = x_{i1}, x_{i2}, \ldots, x_{in}$). Using the Euclidean measure, distance between the input vector and the incoming weight vector of each output map neuron is calculated. The output neuron with the smallest distance is declared the winner. After that, neuron weights are subsequently updated according to Formula 4.1 using a neighbourhood function Formula 4.2, which minimizes the overall distance between the neuron itself and its neighbours.

$$w_{ij}(t+1) = w_{ij} + h(t)(x_i - w_{ij}) \qquad (4.1)$$

where $w_{ij}(t)$ is the connection weight from input i to output neuron j at time t; x_i is element i of input vector x, and h is the neighborhood function.

$$h(t) = \alpha GF \qquad (4.2)$$

where α is the learning rate; GF is the Gaussian Function in Formula 4.3.

$$GF = \exp(-\sum \| wui\text{-}cui \|^2 / 2\rho^2(t)) \qquad (4.3)$$

where ϱ is the neighbourhood width parameter and GF uses the Euclidean distance between the winner unit wu and the current unit cu.

SOM method performance is sensitive to the randomly assigned initial weight values and it is prohibitively slowed in the large-scale applications. In order to decrease the time complexity of SOM, this paper proposes K-Means++ to determine the initial weight values, instead of random process. In this approach, K-Means++ centres are assigned as SOM weight values; thus, SOM will require fewer iterations. Since the K-means algorithm is more computationally efficient than SOM, the general solution will be faster.

The proposed SOM++ is a two-stage algorithm, which is a combination of SOM and the initialization method of centres in K-means++. Figure 4.3 and Figure 4.4 show pseudo codes for SOM++ algorithm. The algorithm starts to find the centre points for the all clusters by using the method of K-Means++ which initializes the centre points of the clusters. There must be two sets named as and D. Set D collects the centres of all clusters during the first part of SOM++ (step 1 and step 6 in Figure 4.3). Set collects the distances between each data and each centre (step 2 in Figure 4.3). The distances are calculated by using the Euclidean Distance. According to sum of squares of distances in set, the centres are obtained (step 3-4-5 in Figure 4.3). After obtaining k centres in set D (step 7 in Figure 4.3), the second part of SOM++ is started (Figure 4.4).

After these steps, all centre points for all clusters are collected in a set. These centres have the attribute values and these values must initialize the weight values of neurons on the map of SOM++. However, the most suitable method must be decided for locating these initial weight values.

If the locating method is not suitable according to the distances of neurons, the result of SOM++ is not different from the result of the standard SOM with the random initialization (Comparing the error rates is given in Section 4.4.4). Therefore, a new sequence assignment algorithm is implemented by considering the distances between K centres in set.

4.4.2.2 The Description of SOM++ Algorithm

The proposed SOM++ is a two-stage algorithm, which is a combination of SOM and the initialization method of centres in K-means++. Figure 4.3 and Figure 4.4 show pseudo codes for SOM++ algorithm.

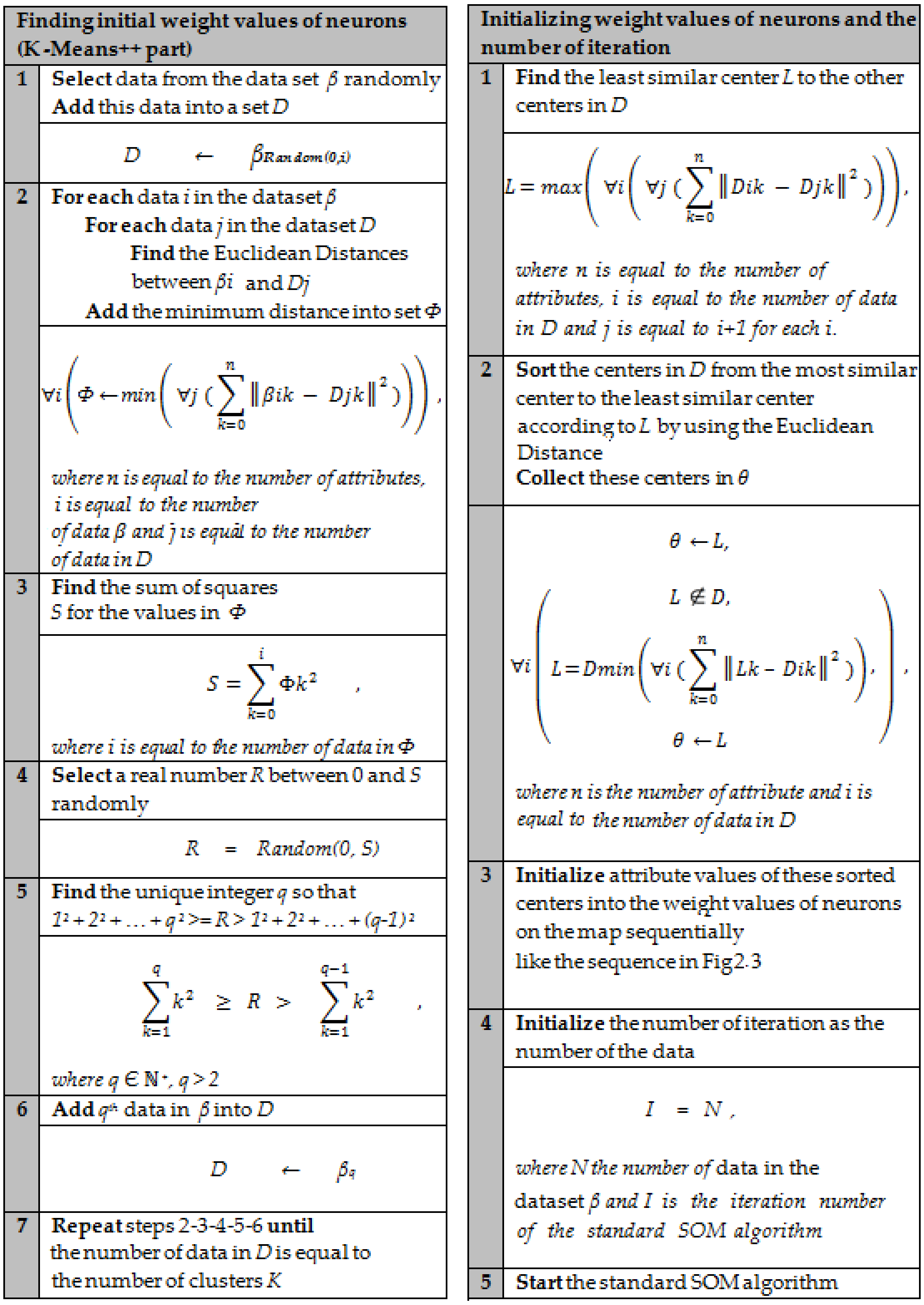
The algorithm starts to find the centre points for the all clusters by using the method of K-Means++ which initializes the centre points of the clusters. There must be two sets named as and D. Set D collects the centres of all clusters during the first part of SOM++ (step 1 and step 6 in Figure 4.3). Set collects the distances between each data and each centre (step 2 in Figure 4.3). The distances are calculated by using the Euclidean Distance. According to sum of squares of distances in set, the centres are obtained (step 3-4-5 in Figure 4.3). After obtaining k centres in set D (step 7 in Figure 4.3), the second part of SOM++ is started (Figure 4.4).

After these steps, all centre points for all clusters are collected in a set. These centres have the attribute values and these values must initialize the weight values of neurons on the map of SOM++. However, the most suitable method must be decided for locating these initial weight values. If the locating method is not suitable according to the distances of neurons, the result of SOM++ is not different from the result of the standard SOM with the random initialization (Comparing the error rates is given in Section 4.4.4). Therefore, a new sequence assignment algorithm is implemented by considering the distances between K centres in set.

Figure 4.4 and Figure 4.5 show the pseudo code of this sequence assignment algorithm. Firstly, the most different point in set D is calculated (step 1 in Figure 4.4) and according to the Euclidean Distance, a sorting operation is done by comparing to this outlier point. At the end of sorting operation, a new set which has sorted points according to the least similar point is obtained (step 2). The values in these points are assigned to the weight values of neurons as the initial values for SOM++ algorithm like the sequence in Figure 4.3 (step 3 in Figure 4.4). Before training operations in SOM++, the number of iteration is initialized as the number of total data (step 4 in Figure 4.4), because actually, the neurons on the map of SOM++ become trained at the beginning; therefore, the number of iteration must not start zero. The advantages of initializing of both the number of iteration and weight values of neurons with the values coming from K-Means++ are shown in the Section 4.4 and 4.5 in detail. Finally, training operations of neurons starts by using the standard SOM algorithm (step 5 in Figure 4.4).

The weight values become close to the final and decisive values by means of K-Means++; on the other hand, it is not enough singly, because the single aim of SOM algorithm is not to do clustering of data correctly. It is also a mapping algorithm; therefore, the places of the clusters win the importance in SOM algorithm. If locating of the weight values which come from the initializing method of K-Means++ is done randomly, the correct neuron cannot have the correct weight values, and the success of SOM++ algorithm is not realized certainly. In Section 4.4, the importance of the sequential assignment is tested with detail.

Our sequential assignment algorithm needs the sorted values according to the least similar value to each other in set. Then, locating operation starts from the neuron in [0, 0] which is the one of the furthest four neurons ([0, 0], [0, (n-1)], [(n-1), 0] and [(n-1), (n-1)] as n^2 is the number of neurons) from the centre on the map, because the top element in sorted values is the least similar value. The next values after the least similar value are located like the sequence in Figure 4.5. The steps on the left and right sides are done to down and inner-cross directions. The steps on the top and bottom sides are done to left and inner-cross directions; however, the steps on the centre side are done mix-cross directions. Finally, the furthest values are located in the furthest neurons on the map.



	Finding initial weight values of neurons (K-Means++ part)
1	**Select** data from the data set β randomly **Add** this data into a set D
	$D \leftarrow \beta_{Random(0,i)}$

2	**For each** data i in the dataset β
	  For each data j in the dataset D
	    Find the Euclidean Distances between βi and Dj
	  Add the minimum distance into set Φ

	$\forall i \left(\Phi \leftarrow min \left(\forall j \left(\sum_{k=0}^{n} \lVert \beta ik - Djk \rVert^2 \right) \right) \right),$ *where n is equal to the number of attributes, i is equal to the number of data β and j is equal to the number of data in D*
3	**Find** the sum of squares S for the values in Φ
	$S = \sum_{k=0}^{i} \Phi k^2 \quad ,$ *where i is equal to the number of data in Φ*
4	**Select** a real number R between 0 and S randomly
	$R = Random(0, S)$
5	**Find** the unique integer q so that $1^2+2^2+\ldots+q^2>=R>1^2+2^2+\ldots+(q-1)^2$
	$\sum_{k=1}^{q} k^2 \geq R > \sum_{k=1}^{q-1} k^2 \quad ,$ *where $q \in \mathbb{N}^+, q > 2$*
6	**Add** q^{th} data in β into D
	$D \leftarrow \beta_q$
7	**Repeat** steps 2-3-4-5-6 **until** the number of data in D is equal to the number of clusters K

	Initializing weight values of neurons and the number of iteration
1	**Find** the least similar center L to the other centers in D
	$L = max \left(\forall i \left(\forall j \left(\sum_{k=0}^{n} \lVert Dik - Djk \rVert^2 \right) \right) \right),$ *where n is equal to the number of attributes, i is equal to the number of data in D and j is equal to i+1 for each i.*
2	**Sort** the centers in D from the most similar center to the least similar center according to L by using the Euclidean Distance **Collect** these centers in θ
	$\forall i \left(\begin{array}{c} \theta \leftarrow L, \\ L \notin D, \\ L = Dmin \left(\forall i \left(\sum_{k=0}^{n} \lVert Lk - Dik \rVert^2 \right) \right), \\ \theta \leftarrow L \end{array} \right),$ *where n is the number of attribute and i is equal to the number of data in D*
3	**Initialize** attribute values of these sorted centers into the weight values of neurons on the map sequentially like the sequence in Fig2.3
4	**Initialize** the number of iteration as the number of the data
	$I = N,$ *where N the number of* data in the dataset β *and I is the iteration number of the standard SOM algorithm*
5	**Start** the standard SOM algorithm

Figure 4.3 a) Calculation of initial weight values of neurons (K-Means++ part) b) Initializing parameters part of SOM++ algorithm

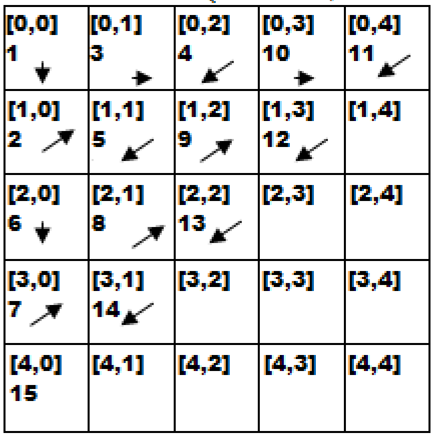
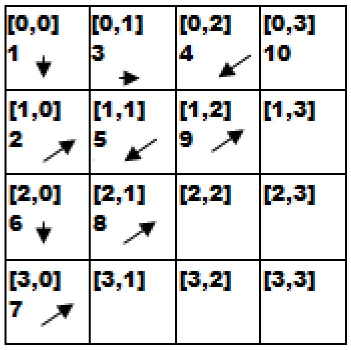
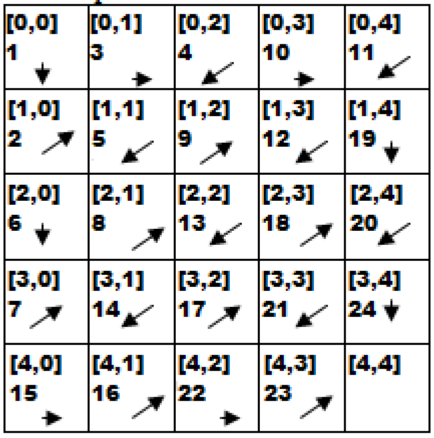
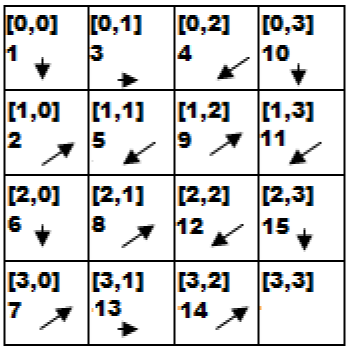
1	$x = 0, y = 0, n = 0, \forall i\ (M[x,y].W[i] = \theta[n].W[i])$, *where i is the number of the weights in a neuron, M is the map matrix, W is the weight array in a neuron, x and y are the indexes of M, and n is the index of the* θ
2 $\downarrow$	$y < BY \Rightarrow (y = y + 1,\ \ n = n + 1,\ \ \forall i\ (M[x,y].W[i] = \theta[n].W[i]))$, *where BY is the boundary of y in M*
3 $\nearrow$	$\exists x, y[(y < BY \wedge x < BX) \Rightarrow (x = x + 1, y = y - 1, n = n + 1, \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$, *where BX is the boundary of x in M*
4 $\rightarrow$	$x < BX \Rightarrow (x = x + 1,\ \ n = n + 1,\ \ \forall i\ (M[x,y].W[i] = \theta[n].W[i]))$, *where BX is the boundary of x in M*
5 $\swarrow$	$\exists x, y[(x < BX \wedge y < BY) \Rightarrow (x = x - 1, y = y + 1, n = n + 1, \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$,
6	**Repeat** 2-3-4-5 **until** $x = \begin{cases} 0 & , BX \bmod 2 = 0 \\ BX & , BX \bmod 2 = 1 \end{cases} \wedge\ y = \begin{cases} BY & , BY \bmod 2 = 0 \\ 0 & , BY \bmod 2 = 1 \end{cases}$  [0,0] 1 [0,1] 3 [0,2] 4 [0,3] 10 [0,4] 11 [1,0] 2 [1,1] 5 [1,2] 9 [1,3] 12 [1,4] [2,0] 6 [2,1] 8 [2,2] 13 [2,3] [2,4] [3,0] 7 [3,1] 14 [3,2] [3,3] [3,4] [4,0] 15 [4,1] [4,2] [4,3] [4,4] *where BX = BY = 4 and the first loop terminates after the first 15 steps*  [0,0] 1 [0,1] 3 [0,2] 4 [0,3] 10 [1,0] 2 [1,1] 5 [1,2] 9 [1,3] [2,0] 6 [2,1] 8 [2,2] [2,3] [3,0] 7 [3,1] [3,2] [3,3] *where BX = BY = 3 and the first loop terminates after the first 10 steps*
7 $\swarrow$	$\exists x, y[(x < BX \wedge y < BY) \Rightarrow (x = x - 1, y = y + 1, n = n + 1, \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$,
8 $\rightarrow$	$x < BX \Rightarrow (x = x + 1,\ \ n = n + 1,\ \ \forall i\ (M[x,y].W[i] = \theta[n].W[i]))$
9 $\nearrow$	$\exists x, y[(y < BY \wedge x < BX) \Rightarrow (x = x + 1, y = y - 1, n = n + 1, \forall i\ (M[x,y].W[i] = \theta[n].W[i])]$,
10 $\downarrow$	$y < BY \Rightarrow (y = y + 1,\ \ n = n + 1,\ \ \forall i\ (M[x,y].W[i] = \theta[n].W[i]))$
11	**Repeat** 7-8-9-10 **until** $x = BX - 1\ \ \wedge\ \ y = BY$  [0,0] 1 [0,1] 3 [0,2] 4 [0,3] 10 [0,4] 11 [1,0] 2 [1,1] 5 [1,2] 9 [1,3] 12 [1,4] 19 [2,0] 6 [2,1] 8 [2,2] 13 [2,3] 18 [2,4] 20 [3,0] 7 [3,1] 14 [3,2] 17 [3,3] 21 [3,4] 24 [4,0] 15 [4,1] 16 [4,2] 22 [4,3] 23 [4,4] *where BX = BY = 4 and the second loop terminates after the 24 steps*  [0,0] 1 [0,1] 3 [0,2] 4 [0,3] 10 [1,0] 2 [1,1] 5 [1,2] 9 [1,3] 11 [2,0] 6 [2,1] 8 [2,2] 12 [2,3] 15 [3,0] 7 [3,1] 13 [3,2] 14 [3,3] *where BX = BY = 3 and the second loop terminates after the 15 steps*
12	$x = x + 1, n = n + 1, \forall i\ (M[x,y].W[i] = \theta[n].W[i])$,

Figure 4.5 Sequential assignment of the initial weight values which come from K-Means++ to neurons.

4.4.3 Experimental Studies

The error rates and stability of the maps are tested by two datasets with 699 and 19020 instances. These datasets have an attribute which puts the correct classes of all data. However, the datasets are used without this attribute while SOM clustering. Because SOM is a clustering algorithm and does not need a target attribute while training neurons. This attribute is used while testing the stability of the maps.

After training neurons without the target attribute, all correct classes for each data are taken and because a neuron can contain more data than one, containing the elements of the same class or not could be showed with coloured neurons. For the visual compares, the dataset with 699 instances is used. This dataset contains 10 attributes and it is about Breast Cancer Wisconsin (BCW) (Wolberg, 1992). There are two certain classes and each neuron which contains the elements of the same class is shown by a different colour on maps.

On all maps in the following figures, there are three different coloured neurons as silver, grey and black. The silver and grey neurons show the correct clustered neurons. In the other words, these neurons contain the elements of the same class; however, the black neurons contain the elements of the different classes and the number of black neurons shows the instability of the map.

For the tests of error rates, the dataset with 19020 instances is used together with the dataset with 699 instances. This dataset has also two certain classes and 11 attributes. It is about MAGIC Gama Telescope (MGT) (Bock, 2007).

For the tests of training time, the dataset with 18781 instances is used. This dataset contains the results of laboratorial tests in the Hospital of Dokuz Eylül University between years of 2008 and 2009. These results are put in 390 attributes for each patient. This dataset is large for both the number of attributes and the numbers of instances. Therefore, the tests of training time are implemented with this datasets to observe the performance of all combinations of SOM, K-Means and K-Means++ algorithms.

Finally, all compares and tests are done by the implementation which is written with C# programming language at Microsoft Visual Studio 2008 platform.

4.4.3.1 Visual Compares

The aim of the visual compares is to find the version of SOM algorithm which has both the least indecisive neurons and the best allocation classes visually. In these compares, all SOM algorithms for different versions use 5x5, 10x10, 15x15 and 20x20 neurons.

In Figure 4.6, there are 5x5 neurons for standard SOM algorithm and updating weight values is done 3x699 times to train neurons. As a result, the number of indecisive neurons is 11 and they locate in the middle of the map and between two cluster areas correctly.

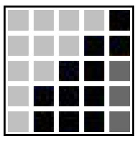


Figure 4.6 Standard SOM algorithm processes over 5x5 neurons until 3x699 iterations.

In Figure 4.7, there are 5x5 neurons again for three versions of SOM algorithm. On the first map, the standard SOM algorithm is used with initialized weight values by the centre points which are returned from K-Means clustering. On the second map, standard SOM algorithm is used with initialized weight values by the initial centre points of K-Means++ without K-Means clustering. On the third map, standard SOM algorithm is used with the initialized weight values by the centre points which are returned from K-Means++ clustering. The weight values are located in neurons sequentially and updating weight values is done with 3x699 iterations. As a result, the number of indecisive neurons is near the number which comes from the standard

SOM and they locate in the middle of the map and between two cluster areas correctly.

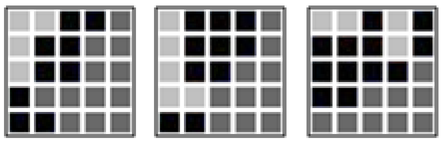


Figure 4.7 SOM algorithm for each pattern processes over 5x5 neurons until 3x699 iterations. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

There is already a pre-treatment because of K-Means clustering, calculating the initial centre points of K-Means++ or K-Means++ clustering. Therefore, in Figure 4.8, the iteration numbers at the beginning of SOM algorithms are changed from 0 to 699 unlike the versions in the Figure 4.7. However, there are not any great changes for the number and places of the indecisive neurons.

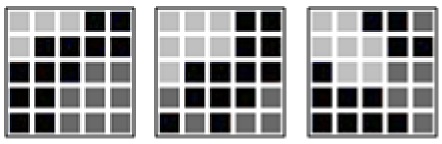


Figure 4.8 SOM algorithm for each pattern processes over 5x5 neurons until 3x699 iterations. The number of initialized iteration is 699. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

The visual compares are done with 10x10 neurons under the same conditions to catch changes. In Figure 4.9, there is the map for standard SOM algorithm and the number of indecisive neurons is 16.

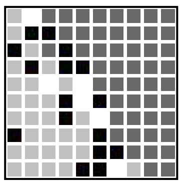


Figure 4.9 Standard SOM algorithm processes over 10x10 neurons until 3x699 iterations.

In Figure 4.10, standard SOM algorithm is used with initialized weight values by the centre points which are returned from K-Means clustering on the first map and K-Means clustering does not need all 10x10 clusters. Therefore, some neurons do not match with any data. The second and third maps shows that the standard SOM algorithms with the versions of K-Means++ have lower number of black neurons than the standard SOM. However, the distinction between them is not clear yet.

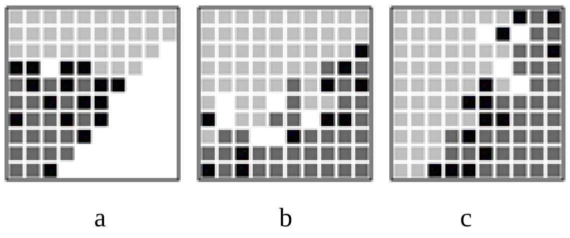


Figure 4.10 SOM algorithm for each pattern processes over 10x10 neurons until 3x699 iterations. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

In Figure 4.11, the same events in Fig 4.10 are implemented with the changing the number of initialized iteration of SOM as 699 to clear the distinction between the patterns of the standard SOM and the other versions. However, there is not any great changing with 10x10 neurons again.

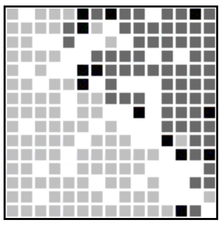


Figure 4.11 Standard SOM algorithm processes over 15x15 neurons until 3x699 iterations.

The visual compares are done with 15x15 neurons under the same conditions to see changes better. In Figure 4.11, there is the map for standard SOM algorithm and the number of indecisive neurons is 13.

The second and third maps in Figure 4.12 shows that SOM algorithms with the versions of K-Means++ have lower number of black neurons than the standard SOM.

This result is started to be seen clearer with 15x15 neurons. However, the distinction between new versions of SOM algorithm is not clear enough with 13, 10 and 11 black neurons.

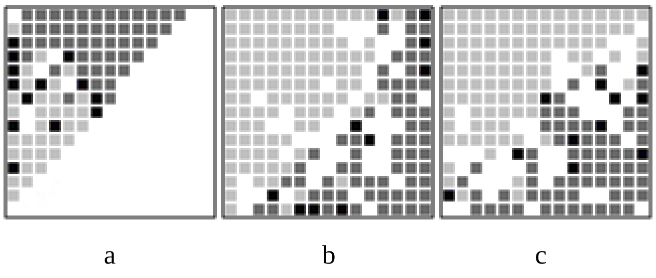


Figure 4.12 SOM algorithm for each pattern processes over 15x15 neurons until 3x699 iterations. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

In Figure 4.13, the same algorithms in Fig 4.12 are implemented with the changing the number of initialized iteration of SOM as 699 to clear the distinction between the patterns of the standard SOM and the other versions. Halving of the number of indecisive neurons is supplied by using K-Means++ without K-Means clustering and 15x15 neurons on the second map with 6 black neurons.

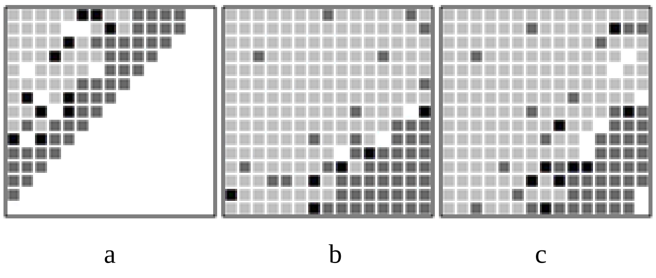


Figure 4.13 SOM algorithm for each pattern processes over 15x15 neurons until 3x699 iterations. The number of initialized iteration is 699. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

The visual compares are done with 20x20 neurons under the same conditions to see the best distinction between all versions.

In Figure 4.14, there is the map for standard SOM algorithm and the number of indecisive neurons is 9.

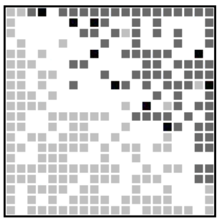
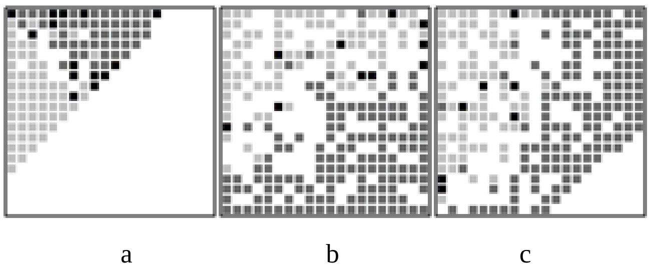


Figure 4.14 Standard SOM algorithm processes over 20x20 neurons until 3x699 iterations.

On the first map in Figure 4.15, standard SOM algorithm is used with initialized weight values by the centre points which are returned from K-Means clustering on the first map and K-Means clustering does not need all 20x20 clusters. Therefore, many neurons do not match with any data. Also, the number of the black neurons is not lower than the number on the map of the standard SOM.

On the second map in Figure 4.15, standard SOM algorithm is used with initialized weight values by the initial centre points of K-Means++ without K-Means clustering. The number of the black neurons is not lower than the number on the map of the standard SOM again.

On the third map in Figure 4.15, standard SOM algorithm is used with initialized weight values by the centre points which are returned from K-Means++ clustering. The number of the black neurons is lower than the number on the map of the standard SOM; however, it cannot be seen clearly. Therefore, the iteration numbers at the beginning of SOM algorithms are changed as 699 in the following tests to catch the stability clearer.



a b c

Figure 4.15 SOM algorithm for each pattern processes over 20x20 neurons until 3x699 iterations. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

Because many neurons cannot match with any data, decreasing of the indecisive neurons cannot be observed clearly on the first map in Figure 4.16. The success by using standard SOM algorithm with initialized weight values by the initial centre points of K-Means++ without K-Means clustering for 15x15 neurons like on the second map in Figure 4.13 gets higher by using 20x20 neurons like on the second map in Figure 4.16. As a result, the number of black neurons is observed as only 2. The same observation is done with the standard SOM algorithm with initialized weight values by the centre points which are returned from K-Means++ clustering on the third map in Figure 4.16. The number of the black neurons is obtained as only 3.

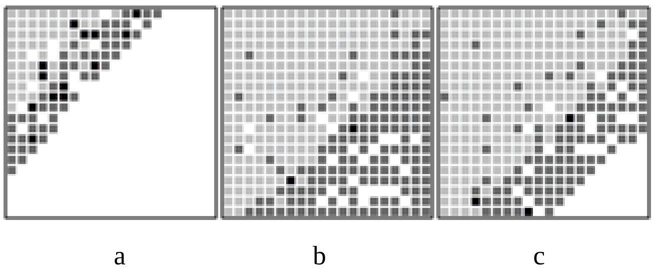


Figure 4.16 SOM algorithm for each pattern processes over 20x20 neurons until 3x699 iterations. The number of initialized iteration is 699. a) K-Means + SOM, b) K-Means++ (without K-Means Clustering) + SOM, c) K-Means++ (with K-Means Clustering) + SOM.

4.4.3.2 Visual Compares According to The Sequential Assignment Algorithm at The Beginning Maps

Before starting to train neurons in SOM algorithm, initializing the weight values of neurons by a pre-treatment supplies a stability to the map at the beginning immediately. In the following versions, the distinction, between the sequential assignments according to the similarities of the centre points which are returned from K-Means clustering algorithms and the random assignments of them, is observed, too.

In Figure 4.17, the beginning map for the standard SOM algorithm with 20x20 neurons is shown. The initial weight values are assigned randomly in the version of the standard SOM; therefore, the map is observed indecisively.

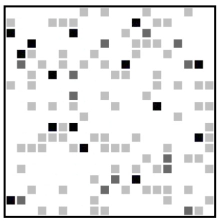


Figure 4.17 Standard SOM at the beginning phase by using 20x20 neurons without any training.

On the first map in Figure 4.18, there are neurons with initialized weight values by the centre points which are returned from K-Means clustering. On the second map, there are neurons with initialized weight values by the initial centre points of K-Means++ without K-Means clustering. On the third map, there are neurons with initialized weight values by the centre points which are returned from K-Means++ clustering.

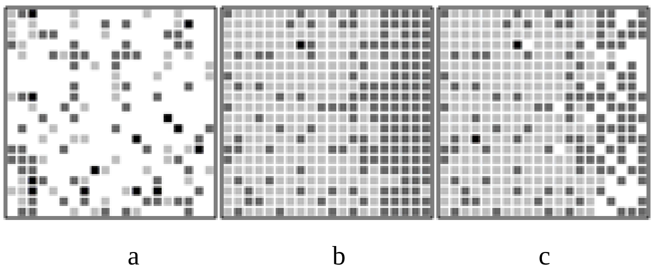


Figure 4.18 SOM is at the beginning phase by using 20x20 neurons without any training. a) K-Means, b) K-Means++ (without K-Means Clustering), c) K-Means++ (with K-Means Clustering).

These weight values are not located by using the sequential assignment and these neurons on the maps are at the beginning phase of SOM algorithm. However, it is seem that the instability cannot be prevented in these examples.

In Figure 4.19, the sequential assignment is used and neurons are located according to the similarities. The importance of the sequential arraignment is observed from the maps in Figure 4.19.

It is observed at the visual tests that the successes of the new versions of SOM algorithm have more stability and less indecisive neurons than the standard SOM. Also, visually, it can be obtained that using K-Means++ without K-Means clustering has a near success together with using K-Means++ with K-Means clustering for the sequential initialized weight values and initialized number of iteration of SOM as the number of data (699 in the previous examples).

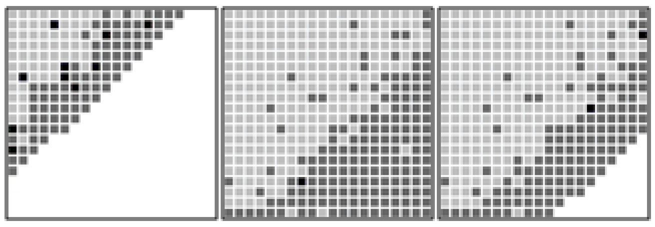


Figure 4.19 SOM is at the beginning phase by using 20x20 neurons without any training and with initialized weights by the sequential assignment. a) K-Means, b) K-Means++ (without K-Means Clustering), c) K-Means++ (with K-Means Clustering).

The numerical comparisons of the new versions are done by calculating the error rates at the phase of training neurons in SOM algorithm.

4.4.3.3 The Comparison of Indecisive Neurons for the Versions of SOM

The indecisive neurons are shown by black neurons on the previous maps. They mean that irrelevant data instances are in same neurons on the map and if the number of these neurons is high, the map is not consistent.

The numbers of indecisive neurons in the previous visual compares are collected like the graph in Figure 4.20. This graph shows that because the numbers of total neurons for the first 5x5 neurons are low, the numbers of indecisive neurons are low, too. Therefore, the numbers of indecisive neurons are higher for 10x10 neurons and the versions of SOM could be compared according to the numbers of the indecisive neurons for 10x10, 15x15 and 20x20 neurons.

The steadiest algorithms are SOM++ (or SOM + K-Means++ (without K-Means)) and SOM + K-Means++ according to the graph. Both of these algorithms use both initializing iteration number as the number of data and the sequential assignment algorithm.

As a result, the most successful algorithm is SOM++ with the least number of indecisive neurons as 6 for 15x15 neurons and only 2 for 20x20 neurons.

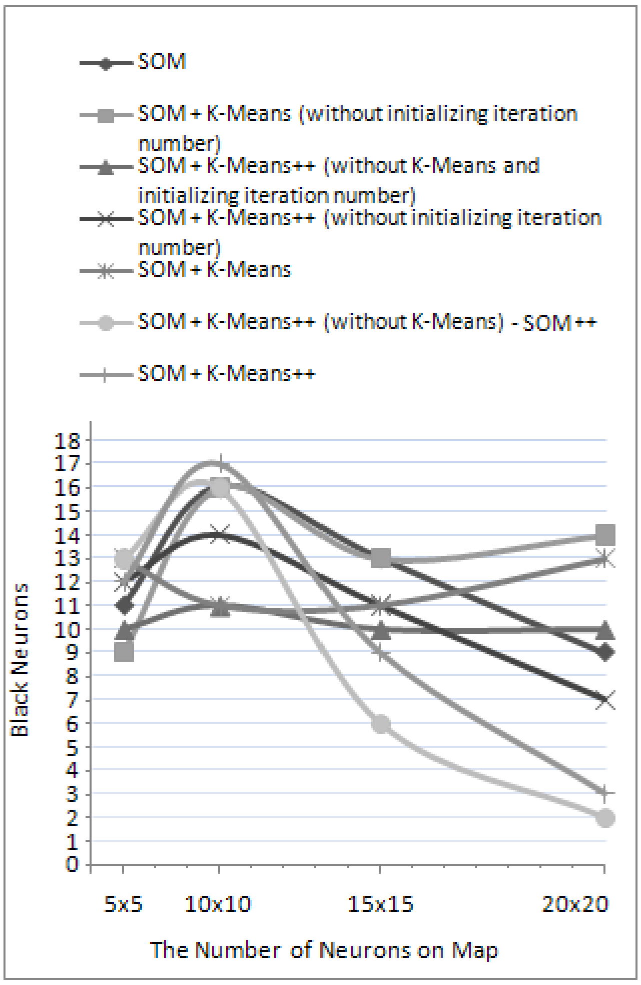


Figure 4.20 The comparison of indecisive neurons.

4.4.3.4 The Error Rates

The error rates are obtained for two different datasets with 699 (BCW dataset) and 19020 (MGT dataset) instances. Table 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 are about BCW dataset, and Table 4.7, 4.8, 4.9 and 4.10 are bout MGT dataset.

The error rates are taken until the end of third iteration (3 x number of data), because the values of rates decrease too less than zero after third iteration. The rates are calculated according to the Eq. 4.1. Actually, this equation updates the weights of neurons on the map for each data; however, the differences between the previous values of the weights and the updated values of weights are calculated by Eq. 4.1 and the total of these differences gives the error rate.

The compares between error rates are done both among themselves of the versions of algorithms according to the different numbers of neurons and the different numbers of iterations, and with other versions of algorithms according to the same numbers of neurons and the same numbers of iterations. Also, the compares among themselves of the versions are done according to initialized number of iteration at SOM algorithm as the number of data and initialized weight values by the sequential assignment.

The first results of error rates are done by using the dataset BCW with 699 instances. In Table 4.1, the error rates are collected by using the standard SOM. It is too remarkable that the error rates after the first iteration are over 1.0. The error rates do not decrease very much when the number of neurons increases and the error rates for 1x699 iterations and higher number of neurons come over 1.0.

Table 4.1 Error rates for standard SOM according to the number of neurons and the number of iterations. (for BCW dataset)

Iteration Number	**5x5 neurons**	**10x10 neurons**	**15x15 neurons**	**20x20 Neurons**
1th	11.65542	7.51092	6.39301	5.64841
2nd	0.75749	0.15426	0.06443	0.04129
3rd	0.07828	0.01595	0.00651	0.00344

The following three tests are about the standard SOM algorithm with initialized weight values by the centre points which are returned from K-Means clustering. In *a* column in Table 4.2, there are the results for the sequential assignment; however, there is not the initialized number of iteration as the number of data.

The test in b column in Table 4.2 is done without the initialized weight values by the sequential assignment and with the initialized number of iteration as the number of data, 699. The error rates get higher without the sequential assignments and the rate after the first 1x699 iterations is obtained over 1.0.

The test in c column in Table 4.2 is done with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data, 699. The error rates get the lowest values under these conditions. When the number of neurons gets greater than 5x5, the error rates comes as undefined values; because K-Means clustering does not need all clusters for 699 data. Therefore, the error rates are taken for only 5x5 neurons in these three tests.

Table 4.2 a) Error rates for K-Means + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. b) Error rates for K-Means + SOM with the initialized number of iteration as the number of total data (699) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c) Error rates for K-Means + SOM with the initialized number of iteration as the number of total data (699) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for BCW dataset)

Iteration Number	**a (5x5 neurons)**	**b (5x5 neurons)**	**c (5x5 neurons)**
1th	0.23121	1.28005	0.07629
2nd	0.03068	0.08460	0.01359
3rd	0.00650	0.03274	0.00451

The following three tests are about the standard SOM algorithm with initialized weight values by the initial centre points of K-Means++ without K-Means clustering.

Table 4.3 Error rates for K-Means++ (without K-Means clustering) + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for BCW dataset)

Iteration Number	**5x5 neurons**	**10x10 neurons**	**15x15 neurons**	**20x20 neurons**
1th	0.42720	0.49013	0.53707	0.27704
2nd	0.01000	0.01873	0.01157	0.00993
3rd	0.00518	0.00177	0.00056	0.00035

In Table 4.3, there are the results for the sequential assignment without the initialized number of iteration as the number of data. The values are 0.42720, 0.01000 and 0.00518 for K-Means++ without K-Means clustering + SOM while using 5x5 neurons. The values for K-Means + SOM are 0.23121, 0.03068 and 0.00650 while using 5x5 neurons. K-Means++ without K-Means clustering uses all neurons; therefore, there are defined error rates for 10x10, 15x15 and 20x20 neurons. The values get lower while increasing the number of neurons.

The test in Table 4.4 is done without the initialized weight values by the sequential assignment and with the initialized number of iteration as the number of data as 699. The error rates get higher without the sequential assignments and the rate after the first 1x699 iterations is obtained over 1.0 like the results of K-Means + SOM.

Table 4.4 Error rates for K-Means++ (without K-Means Clustering)+ SOM with the initialized number of iteration as the number of total data (699) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for BCW dataset)

Iteration Number	**5x5 neurons**	**10x10 neurons**	**15x15 neurons**	**20x20 neurons**
1th	1.61001	0.68055	0.46300	0.20368
2nd	0.08828	0.01924	0.00785	0.00397
3rd	0.03508	0.00733	0.00303	0.00152

The test in Table 4.5 is done with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data as 699. The error rates get the lowest values like the error rates of K-Means + SOM under the same conditions. It is observed that when using K-Means, the value after the first 1x699 iterations is 0.07629; however, when using K-Means++ without K-Means clustering, the value after the first 1x699 iterations is 0.28725. This difference is closed when the number of iteration increases.

Table 4.5 Error rates for K-Means++ (without K-Means clustering) + SOM with the initialized number of iteration as the number of total data (699) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for BCW dataset)

Iteration Number	**5x5 neurons**	**10x10 neurons**	**15x15 neurons**	**20x20 neurons**
1th	0.28725	0.05095	0.02495	0.00168
2nd	0.00252	0.00058	0.00030	0.00023
3rd	0.00125	0.00007	0.00003	0.00002

The following three tests are about the standard SOM algorithm with the initialized weight values by the centre points which are returned from K-Means++ clustering. In “a” rows in Table 4.6, there are the results for the sequential assignment without the initialized number of iteration as the number of data. It is observed that the values for 5x5 and 10x10 neurons are close with the values for using K-Means++ without K-Means clustering under the same conditions.

The results in “b” rows in Table 4.6 are done without the initialized weight values by the sequential assignment and with the initialized number of iteration as the number of data, 699. The error rates get higher without the sequential assignments and also, the rate after the first 1x699 iterations is obtained over 1.0; therefore, it can be determined that initializing weight values by the sequential assignment is necessary.

The results in “c” rows in Table 4.6 are done with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data. These error rates get the lowest values like the other error rates of the other versions (c column in Table 4.2 and Table 4.5) under the same conditions.

Table 4.6 a) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. b) Error rates for K-Means++ (with K-Means Clustering)+ SOM with the initialized number of iteration as the number of total data (699) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized number of iteration as the number of total data (699) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for BCW dataset)

Types	**Iteration Number**	**5x5 neurons**	**10x10 neurons**
a	1th	0.43723	0.47991
	2nd	0.05145	0.01509
	3rd	0.01086	0.00112
b	1th	1.81375	0.67474
	2nd	0.09418	0.01972
	3rd	0.03524	0.00777
c	1th	0.28313	0.08267
	2nd	0.00795	0.00029
	3rd	0.00298	0.00024

Finally, it is observed that the error rates with using the all new combinations are lower than the error rates with using the standard SOM. However, the best combination which has the lowest error rates comes from the version with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data. Also, it is observed that there is not a great distinction between the versions with K-Means++ without K-Means clustering and K-Means++ with K-Means clustering. When using K-Means for 5x5 neurons, the value after the first 1x699 iterations in c column in Table 4.2 is 0.07629; however, when using K-Means++ without K-Means clustering for 5x5 neurons, the value after the first 1x699 iterations in Table 4.5 is 0.28725 and when using K-Means++ with K-Means clustering for 5x5 neurons, the value after the first 1x699 iterations in c rows in Table 4.6 is 0.28313.

The following results of error rates are done by using the dataset MGT with 19020 instances to test the new versions of SOM for high number of data. In Table 4.7, the error rates are collected by using the standard SOM. It is too remarkable that the error rates after the first iteration (1x19020) are too high with the values of 147.94405 for 5x5 neurons and 91.29000 for 10x10 neurons because of the random beginning of weight values. However, when the number of iterations gets higher, the error rates get lower and the values approach zero.

Table 4.7 Error rates for standard SOM according to the number of neurons and the number of iterations. (for BCW dataset)

Iteration Number	5x5 neurons	10x10 neurons
1th	147.94405	91.29000
2nd	$46x10^{-09}$	$74x10^{-10}$
3rd	$24x10^{-17}$	$42x10^{-18}$

The following three tests are about the standard SOM algorithm with initialized weight values by the centre points which are returned from K-Means clustering. In “a” rows in Table 4.8, there is the sequential assignment; however, there is not the initialized number of iteration as the number of data. This combination has more successful results than the standard SOM; however, after the first iteration (1 x 19020), the error rates come high as 1.85885 for 5x5 neurons and 0.32190 for 10x10 neurons.

The next test in “b” rows in Table 4.8 is done without the initialized weight values by the sequential assignment and with the initialized number of iteration as the number of data as 19020. The error rates come very low from the first 1x19020 iterations as $32x10^{-09}$ for 5x5 neurons and $50x10^{-10}$ for 10x10 neurons. However, the assignment is not done sequentially in this test and if the sequential assignment is used for initializing the weight values, the error rates can become lower.

Table 4.8 a) Error rates for K-Means + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. b) Error rates for K-Means + SOM with the initialized number of iteration as the number of total data (19020) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c) Error rates for K-Means + SOM with the initialized number of iteration as the number of total data (19020) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for MGT dataset)

Types	**Iteration Number**	**5x5 neurons**	**10x10 neurons**
a	1th	1.85885	0.32190
	2nd	$71x10^{-11}$	$25x10^{-11}$
	3rd	$26x10^{-18}$	$43x10^{-19}$
b	1th	$32x10^{-09}$	$50x10^{-10}$
	2nd	$17x10^{-17}$	$27x10^{-18}$
	3rd	$99x10^{-26}$	$14x10^{-26}$
c	1th	$11x10^{-10}$	$88x10^{-12}$
	2nd	$53x10^{-19}$	$57x10^{-20}$
	3rd	$55x10^{-28}$	$38x10^{-28}$

The test in “c” rows in Table 4.8 is done with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data as 19020. The error rates have the lowest values as excepted. After the first 1x19020 iterations, the error rate comes as $11x10^{-10}$ for 5x5 neurons and $88x10^{-12}$ for 10x10 neurons. If these results are compared to the results of the standard SOM, it is observed that the error rate for 5x5 neurons comes from 147.94405 to $11x10^{-10}$ and the error rate for 10x10 neurons comes from 91.29000 to $88x10^{-10}$. Also, if the error rates after the second and third 1x19020 iterations are compared, it is seen that the success ascends near 1010 times.

The following three tests are about the standard SOM algorithm with initialized weight values by the initial centre points of K-Means++ without K-Means clustering. In “a” rows in Table 4.9, there is the sequential assignment; however, there is not the initialized number of iteration as the number of data. The values under the same conditions are 1.30422, $44x10^{-11}$ and $37x10^{-20}$ for K-Means++ without K-Means clustering + SOM while using 10x10 neurons. The values for K-Means + SOM are

0.32190, $25x10^{-11}$ and $43x10^{-19}$ while using 10x10 neurons. Although the first results are too different, the difference between the other results is not great.

The test in "b" rows in Table 4.9 is done without the initialized weight values by the sequential assignment and with the initialized number of iteration as the number of data, 19020. The error rates come very low from the first iterations (1 x 19020) as $36x10^{-09}$ for 5x5 neurons and $55x10^{-10}$ for 10x10 neurons. These values are very close with the error rates of the version with K-Means clustering in b rows in Table 4.8. The assignment is not done sequentially in this test and if the sequential assignment is used for initializing the weight values, the error rates can become lower like the error rates of the version with K-Means clustering in c rows in Table 4.8.

Table 4.9 a) Error rates for K-Means++ (without K-Means clustering) + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. b) Error rates for K-Means++ (without K-Means Clustering)+ SOM with the initialized number of iteration as the number of total data (19020) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c) Error rates for K-Means++ (without K-Means clustering) + SOM with the initialized number of iteration as the number of total data (19020) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for MGT dataset)

Types	**Iteration Number**	**5x5 neurons**	**10x10 neurons**
a	1th	1.34450	1.30422
	2nd	$13x10^{-10}$	$44x10^{-11}$
	3rd	$38x10^{-18}$	$37x10^{-20}$
b	1th	$36x10^{-09}$	$55x10^{-10}$
	2nd	$20x10^{-17}$	$30x10^{-18}$
	3rd	$11x10^{-25}$	$17x10^{-26}$
c	1th	$49x10^{-10}$	$28x10^{-11}$
	2nd	$25x10^{-18}$	$16x10^{-20}$
	3rd	$13x10^{-26}$	$15x10^{-29}$

The test in "c" rows in Table 4.9 is done with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data, 19020. The error rates have the lowest values as excepted. After the first iterations (1 x 19020), the error rate comes as $49x10^{-10}$ for 5x5 neurons and $28x10^{-11}$ for 10x10 neurons. If these results are compared to the results of the standard SOM, the great enhancement is observed that the error rate for 5x5 neurons comes from

147.94405 to 49x10^{-10} and the error rate for 10x10 neurons comes from 91.29000 to 28x10^{-10}. Also, if the error rates after the second and third 1x19020 iterations are compared, it is seen that the success ascends near 1010 times. If the results in c rows in Table 4.9 are compared to the results of the version with K-Means clustering, it is observed that the error rates are very close.

The next three tests are about the standard SOM algorithm with the initialized weight values by the centre points which are returned from K-Means++ clustering. In “a” rows in Table 4.10, there is the sequential assignment; however, there is not the initialized number of iteration as the number of data. The values under the same conditions are 1.94547, 19x10^{-10} and 15x10^{-18} for K-Means++ with K-Means clustering + SOM while using 10x10 neurons. The values for K-Means + SOM are 0.32190, 25x10^{-11} and 43x10^{-19} while using 10x10 neurons. Although the first results are too different, the difference between the other results is not great. The values for K-Means++ without K-Means clustering + SOM are 1.30422, 44x10^{-11} and 37x10^{-20} while using 10x10 neurons. These values are closer to the results in “a” rows in Table 4.10.

The test in “b” rows in Table 4.10 is done without the initialized weight values by the sequential assignment and with the initialized number of iteration as the number of data, 19020. The error rates come very low from the first 1x19020 iterations as 33x10^{-09} for 5x5 neurons and 49x10^{-10} for 10x10 neurons. These values are very close with the error rates of the other two versions in b rows in Table 4.8 and Table 4.9. The assignment is not done sequentially in this test and if the sequential assignment is used for initializing the weight values, the error rates can become lower like the error rates of the other two versions in c rows in Table 4.8 and Table 4.9.

The last test in “c” rows in Table 4.10 is done with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data as 19020. The error rates have the lowest values as excepted. After the first 1x19020 iterations, the error rate comes as 45x10^{-11} for 5x5 neurons and

$31x10^{-11}$ for 10x10 neurons. If these results are compared to the results of the standard SOM, it is observed that the error rate for 5x5 neurons comes from 147.94405 to $45x10^{-11}$ and the error rate for 10x10 neurons comes from 91.29000 to $31x10^{-11}$.

Table 4.10 a) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. b) Error rates for K-Means++ (with K-Means Clustering)+ SOM with the initialized number of iteration as the number of total data (19020) and without the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. c) Error rates for K-Means++ (with K-Means clustering) + SOM with the initialized number of iteration as the number of total data (19020) and the initialized weights by the sequential assignment according to the number of neurons and the number of iterations. (for MGT dataset)

Types	**Iteration Number**	**5x5 neurons**	**10x10 neurons**
a	1th	1.52600	1.94547
	2nd	$11x10^{-10}$	$19x10^{-10}$
	3rd	$42x10^{-19}$	$15x10^{-18}$
b	1th	$33x10^{-09}$	$49x10^{-10}$
	2nd	$18x10^{-17}$	$28x10^{-18}$
	3rd	$10x10^{-25}$	$15x10^{-26}$
c	1th	$45x10^{-11}$	$31x10^{-11}$
	2nd	$45x10^{-19}$	$51x10^{-21}$
	3rd	$20x10^{-27}$	$21x10^{-28}$

Also, if the error rates after the second and third 1x19020 iterations are compared, it is seen that the success ascends near 1010 times, again. If the results in c rows in Table 4.10 are compared to the results of the other two versions in c rows in Table 4.8 and Table 4.9, it is observed that the error rates are very close. As a result, according to the tests about the error rates, the combination with both the initialized weight values by the sequential assignment and the initialized number of iteration as the number of data must be used at K-Means clustering + SOM, K-Means++ (without K-Means clustering) + SOM and K-Means++ clustering + SOM.

4.4.3.5 Training Times

The tests of training times are implemented by using the dataset with 18781 instances and 390 attributes. This large dataset is produced the distinctions on performance of trainings between simple SOM, SOM++ (with the initialization method of K-Means++), SOM + K-Means++ and SOM + K-Means absolutely. The computer, which tests the performances of the algorithms for this large dataset, has 3.24 GB of RAM and Intel(R) Core(TM) 2 Duo CPU E6550 @ 2.33 GHz. This average computer trains 600x600 neurons with this large dataset which has a large attribute set for three weeks.

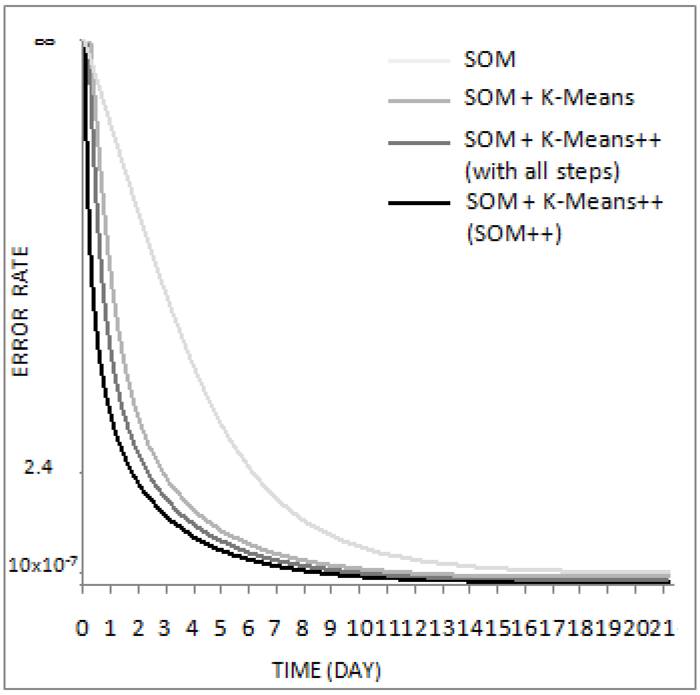


Figure 4.21 the error rates - time (day) graphic for the versions of SOM

The Figure 4.21 shows that the simple SOM trains 600x600 neurons since the first moment; however, SOM++ (with the initialization method of K-Means++), SOM + K-Means and SOM + K-Means++ need a preparation time for SOM. Therefore, the error rates of SOM are observed stably until a few times for these versions of SOM. These times are less than an hour for SOM++, about 2 hours for SOM + K-Means++ and about 3 hours for SOM + K-Means, because there are lots of attributes and instances in the dataset. After these times, the simple SOM starts with initialized weight values and iteration values for these versions.

At the end of 7 days, the simple SOM gives about 2.4 of the error rate; however, SOM++ gives a less error rate than $10x10^{-7}$ before 8 days. Also, the error rate for SOM + K-Means++ is taken a less error rate than $10x10^{-7}$ before 9 days and the error rate for SOM+ K-Mean is taken a less error rate than $10x10^{-7}$ before 10 days. These results show that the versions of SOM have better performances than the simple SOM. However, SOM++ has the best performance. Because of the complexity of SOM algorithm as $O(N^2)$ where N is the input vector size (N is 390 x 18781 and N^2= 53,649,618,668,100), SOM algorithm gets the result map with minimum error rates after some days.

Also, it is observed that the initialization method of centre points at K-Means++ accelerates K-Means, because K-Means++ with all steps needs about 2 hours to cluster a large dataset. However, K-Means needs about 3 hours.

4.4.4 Comparison Results

The accuracy, about which of the SOM versions must be used, is taken by the visual tests. It is observed that the SOM versions of K-Means++ without K-Means clustering and with K-Means clustering have the least number of indecisive neurons in the visual tests.

If the versions of K-Means++ with K-Means clustering and without K-Means clustering have a close success, a comparison of time can do the distinction between them and it is observed that K-Means++ (without K-Means clustering) + SOM has the best results. Consequently, our experimental results empirically prove that K-Means++ (without K-Means clustering) + SOM (SOM++ with its short name) is best suited to data clustering due to its high speed and lower error rates as compared with other SOM based techniques.

4.5 PSOMDM Faster Parallel Self Organizing Map by Using Division Method

The Self Organizing Map (SOM) (Kohonen, 1985) is a neural network model that is capable of clustering high-dimensional input data. It produces a two-dimensional “feature map” that can be useful in detecting and analyzing features in the input space. SOM model has been successfully applied in a number of disciplines including pattern recognition, image classification, and document clustering.

Experimental results with various test data sets and comparison results show that PSOMDM can be efficiently used for clustering large datasets with SOM, somehow making technique move toward the more powerful unsupervised technique. In the experimental studies, PSOMDM is implemented by dividing data processing into a number of parallel threads and multi-core processors. The threads or multi-cores of parallel SOM model process over the same neurons by partitioning data.

4.5.1 Literature Review

The SOM performs clustering by means of unsupervised competitive learning. The neurons in the SOM are usually arranged in a two dimensional lattice and each neuron get information from the input layer and from the other neurons in the map. The SOM has the complexity O(NC), where N is the input vector size and C is the number of dataset presentation cycles. N contains n^2w as the multiply of the map size n^2 and the number of weights w. C contains n^2a as the multiply of the map size n^2 and the number of attributes a. Usage of parallel SOM supplies lower complexity than $O(N^2)$.

Several studies have been done related to parallel SOM by using different techniques and different architectures. Garcia et al. proposed a speculative approach which fits better than traditional Map-Partitioning strategies due to a better exploitation of the memory hierarchy (Garcia, Prieto, & Pascual-Montano, 2006). Yu et al. proposed a multistage modular self-organizing map (SOM) model which can be used for parallel web text clustering (Yu, Wang, & Lai, 2007). Gorgonio and Costa presents an approach for efficient cluster analysis in distributed databases using SOM and K-Means (Gorgonio & Costa, 2008; Gorgonio & Costa, 2010). They also proposed a SOM-based strategy for cluster analysis in distributed databases (Gorgonio & Costa, 2008). Another approach of parallel SOM is about mining massive datasets by an unsupervised parallel clustering on a GRID (Faro, Giordanoa, & Maioranaa, 2011).

Liping & Wensheng (2009) analyzed a variety of software and hardware environment of designing artificial neutral networks on clusters (Liping & Wensheng, 2009). Dagli et al. (2009) proposed a SOM model, called ParaSOM, which utilizes a feature called a cover region, in which individual neurons "cover" whole regions of the input space, and not just a single vector (Dagli, Bryden, Corns, Gen, Tumer, & Süer, 2009).

Parallel SOM and its derivations were used in many different areas. For example, the study about dynamics of soil organic matter and mineral nitrogen in soil was given as an application which used parallel SOM for geological researches (Semaoune et al., 2011). The study of M. Takatsuka and M. Buis presents the parallel implementation of SOMs, particularly the batch map variant using Graphics Processing Units (GPUs) through the use of Open Computing Language (OpenCL) (Takatsuka & Bui, 2010). Parallel SOM was also used for the computer security, the healthcare, ecological modelling, the financial sector and other areas which need clustering.

Differently from the previous studies, this study proposes a new SOM model which is implemented in a parallel way and using a different division method to gain greater computational efficiency. To the best of our knowledge, this study is the first on proposing a division the area into small areas for parallel processing and different SOM architecture as defined in the next section.

4.5.2 PSOMDM Approach

SOM consists of two layers of artificial neurons an input layer and an output layer. The input layer is fed into feature vectors, so it is the same as the number of dimensions of the input feature vector. Output layer, also called the output map, is usually arranged in a regular two dimensional structure such that there are neighbourhood relations among the neurons. Every neuron in input layer is fully connected to every output neuron, and each connection has a weighting value attached to it.

The major goal of SOM is to determine the suitable weight values for the neurons according to dataset. The count of these weight values is equal to the number of attributes in dataset and each weight value corresponds to an attribute. At the beginning of the SOM algorithm, these weight values in all neurons are initialized randomly. Secondly, the best matching unit as the winner neuron is found by calculating Euclidean distance from each weight to the chosen sample vector which consists of the weights set in one of neurons. After finding the best matching unit, all vectors of the SOM are updated by using Gaussian function. These processes are repeated until a certain number of iterations. The loop accrues over the all neurons.

Usage of Parallel SOM supplies lower complexity than $O(N^2)$. According to the number of parallel threads or multi-core processors, parallel SOM accelerates the process. However, the threads or cores of parallel SOM process over the same neurons; therefore, the solutions which come from parallel SOM are at risk about lower accuracy. This new approach of parallel SOM by using division method has higher accuracy and speed. Because the random choosing the data in the training phase of SOM is abandoned and the data are chosen in the same order for each time. The accuracy problem is passed over at connecting the subparts of maps.

This new approach firstly divides the area by four and this standard SOM processes for all little areas by parallel processing. Thus, datasets are divided for all processes and the complexity becomes lower.

PSOMDB consists of standard SOM (SSOM). This algorithm is used for 2x2 neurons stably in each phase. PSOMDB starts training with SSOM with 2x2 neurons and usage of all dataset. After that, the recursive structure of PSOMDB is activated and recursively, SSOM for 2x2 neurons is processed with usage of divided datasets. The following Figure 4.22 shows the process-flow of PSOMDB for 4x4 neurons.

There are four parallel maps and they are trained for 2x2 maps. At the end of the PSOMDB, a 4x4 map is obtained after combining operation in the same order before splitting.

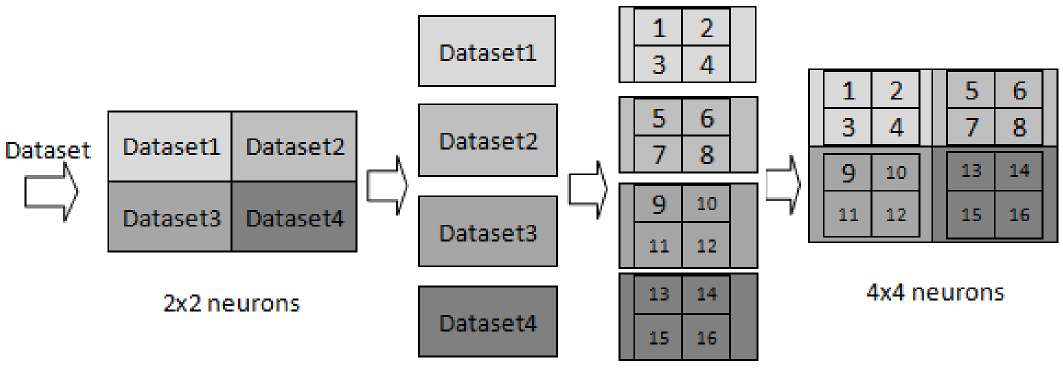


Figure 4.22 The process-flow of PSOMDB for 4x4 neurons.

The complexity of traditional SOM is O(N^2). However, this formula is obtained by the assumption of the multiply of the map size and weight numbers equal to the multiply of the number of instances and the number of attributes in the dataset. Therefore, if N^2 is splitted to NxC as N is the total size of map and C is the total size of dataset, the changes of SOM speed according to the different datasets and its effects appears in more detail. When this formula is splitted sub-components, these following formulas in Formula 4.4 and Formula 4.4 are obtained;

$$T \text{ x } A = C \tag{4.4}$$

$$M \text{ x } W = N \tag{4.5}$$

where T is the number of instances in the dataset, A is the attribute number for an instance in the dataset, M is the total neuron number in the map and W is the total number of weight variables for a neuron in the map.

$$T \text{ x } A \text{ x } M \text{ x } W = \alpha \tag{4.6}$$

where α is the total time for traditional SOM algorithm.

When PSOMDB is processed for 4x4 neurons (M = 16), SSOM algorithm is processed for 2x2 neurons (M = 4) and all dataset in the first phase. Secondly, the datasets are divided into four pieces for each neuron according to the proximities to the weight values of neurons. All proximities calculations in the algorithm are done by Euclidean distance formula;

$$\text{Euclidean Distance} = \sqrt{\sum_{k=0}^{d} (Xk - Yk)^2} \tag{4.7}$$

Four pieces of dataset are used for training by SSOM algorithm in parallel. Traditional SOM is processed for 2x2 neurons (M = 4) for each piece again. After these pieces are trained, these pieces are joined and the map with 4x4 neurons is obtained. The process time calculation is done the following formulas

$$\mathrm{T\ x\ A\ x} \frac{\mathrm{M}}{2\mathrm{x}2}\ \mathrm{x\ W} = \frac{\mathrm{T\ x\ A\ x\ M\ x\ W}}{4} = \beta \tag{4.8}$$

$$\frac{\mathrm{T}}{4} \mathrm{x} \frac{\mathrm{A\ x\ M\ x\ W}}{4} = \frac{\mathrm{T\ x\ A\ x\ M\ x\ W}}{16} = \sigma \tag{4.9}$$

$$\frac{5\ \mathrm{x\ T\ x\ A\ x\ M\ x\ W}}{16} = \sigma + \beta \tag{4.10}$$

If it is assumed that SSOM is processed for all dataset and 4x4 neuron, process time of SSOM is found by Formula 4.6. For 4x4 neurons, PSOMDB is processed for 2x2 neurons firstly and the process time of this phase is calculated by Formula 4.7. The result shows that this phase is equals to quarter of SSOM. However, PSOMDB continues to be trained for 4x4 neurons and in parallel and for four pieces of datasets, PSOMDB is processed for 2x2 neurons again.

The process time of this phase is calculated by Formula 4.8. Totally, the process time is calculated by Formula 4.9 and it shows that PSOMDB takes less time than SSOM as 5/16 times or nearly 1/3 times. This difference is kept even if the size of the dataset gets larger. Also, if the map size gets larger, new components add to Formula 4.6 and because the dataset is divided again and again, these process times are approximate to zero a lot like. Thus, this difference is kept. Theoretically, this is possible; however, the machine, where PSOMDB is processed, must have enough numbers of cores to supply the parallel processing.

In section 4.5.3, experimental studies, these formulas are compared to the results when the real world datasets are used for PSOMDB.

4.5.3 Experimental Studies

Our proposed approach was proved by three different datasets from UCI machine learning (Lichman, 2013) and successful results are observed. The datasets are “Iris”, “CPU” and “Segment- challenge”. These datasets are for classification operation; however, for accuracy tests, clustering algorithms are tested by usage of classification datasets. Because training phase takes the dataset without the target attribute, then test phase compares the derivation of pattern to the correct values in target attribute.

“Iris”, “CPU” and “Segment-challenge” have different features: Iris is a small-sized dataset. It has 150 instances and 5 attributes. The last attribute is the target attribute; therefore, this column is deleted in the training phase. Thus, PSOMDB is trained with 4 attributes. CPU is a middle-sized dataset. It has 209 instances and 7 attributes. Like the content in the Iris dataset, the last attribute is the target attribute; therefore, training operation of PSOMDB is implemented without this column. The third dataset, Segment-challenge is a larger dataset. It has 1500 instances and 21 attributes. Like the content in the other two datasets, the last attribute is the target attribute and PSOMDB is trained with 20 attributes.

PSOMDB and SSOM algorithms are implemented in Visual Studio 2010 platform by usage of C# programming language. The implementation gives the results about speeds as Milliseconds with the result map. The comparison of result maps shows that there are the same result maps with the same weight values. Therefore, the accuracy does not change with PSOMDB. However, it is observed that the performance of PSOMDB is better than the performance of SSOM.

Table 4.11 shows that there are three different datasets as Iris, CPU and Segment challenge and two different clustering algorithms as PSOMDB and SSOM. The algorithms are processed for 4x4 neurons. The performance results of PSOMDB are given for all components. The first five rows in Table 4.11 are results for PSOMDB and the last row is the results for SSOM because SSOM is processed for 4x4 neurons at one phase. However, the results of PSOMDB are taken for 2x2 neurons firstly. Then, for each cell, {(0,0),(0,1),(1,0),(1,1)}, PSOMDB is processed in parallel and for each cell, different results are obtained.

The process time of the cell, which takes the biggest time, determines the total process time. For example, (0,1) cell for Iris takes the biggest time as 376 ms, (1,0) cell for CPU takes the biggest time as 641 ms and (0,1) cell for Segment-challenge takes 3021 ms. Finally, Table 4.11 shows that the total performances for each datasets are more successful than the performances of SSOM.

Table 4.11 The speed times of PSOMDB and SSOM for three datasets (Iris, CPU and Segment-challenge) as millisecond

Datasets	Iris	CPU	Segment-challenge
2x2 map calculation (ms)	**441**	**747**	**3427**
(0,0) cell calculation for 2x2 map (ms)	353	490	2990
(0,1) cell calculation for 2x2 map (ms)	**376**	562	2986
(1,0) cell calculation for 2x2 map (ms)	336	**641**	**3021**
(1,1) cell calculation for 2x2 map (ms)	339	454	2998
Total time (ms)	**817**	**1388**	**6448**
SSOM (ms)	**1784**	**2755**	**11870**

The process time of the cell, which takes the biggest time, determines the total process time. For example, (0,1) cell for Iris takes the biggest time as 376 ms, (1,0) cell for CPU takes the biggest time as 641 ms and (0,1) cell for Segment-challenge takes 3021 ms. Finally, Table 4.11 shows that the total performances for each datasets are more successful than the performances of SSOM.

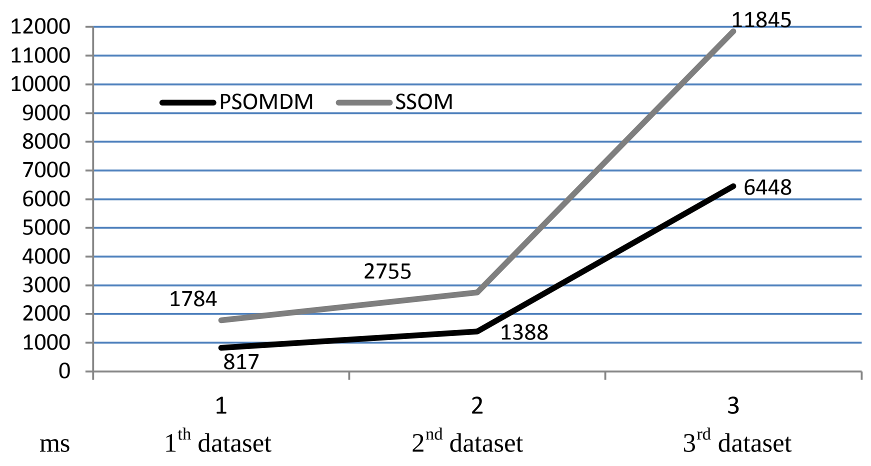


Figure 4.23 Speeds of SSOM and PSOMDB for UCI datasets as Milliseconds.

Figure 4.23 shows that PSOMDB is more successful than traditional SOM because the performances decrease nearly half and half. For Iris dataset, the process time decreases from 1784 to 817, for CPU dataset, the process time decreases from 2755 to 1388 and for Segment-challenge dataset, the process time decreases from 11845 to 6448. The most important gain is that when there is a large dataset, PSOMDB increases the training performance and gives the same result map in less time.

CHAPTER FIVE
MEDITERRANEAN DIET OPTIMIZATION BY HQGA

5.1 Diet Optimization

Dietary nutrient needs depend on age, sex, weight, and height. Aim of this approach is offering an optimized Mediterranean dietary menu specific to a person whose age, sex, weight, height, anamnesis information, eating habits, and daily activities are known. A modified version of Hybrid Quantum Genetic Algorithm has been developed to generate an optimized menu which supplies daily nutrient needs by offering minimum amount of foods without unnecessary consuming. There are a lot of different approaches to obtain an optimized menu in literature (Lv, 2009; Mino & Kobayashi, 2009; Pei & Liu, 2009; Wang, Lee, Hsieh, Hsu, & Chang, 2009), and they are compared with our approach in Section 5.7. Diet optimization methods of Hybrid Quantum Genetic Algorithm have been implemented as a WCF web service. The expert system application which calls the methods of the WCF web service has been implemented as an ASP.NET web site and has been published on a web server (The-mediterranean-diet. Retrieved May 01, 2015). All implementations of algorithms have been coded on the Visual Studio 2013 platform.

5.2 Mediterranean Diet

The Mediterranean diet has been consumed traditionally in the Mediterranean region for thousands of years. This diet is patterned as an intangible cultural heritage of Italy, Portugal, Spain, Morocco, Greece, Cyprus and Croatia by UNESCO in 2013. The potential health benefits of the Mediterranean diet were initially observed in the Seven Countries Study in the early 1950s and it formally described by Keys in the 1960s (Keys et al.,1986).

The traditional Mediterranean diet is known to be one of the healthiest dietary patterns in the world, is characterized by a high intake of virgin olive oil, fruits, vegetables, other plant proteins and fibers (walnuts and legumes), unrefined whole grains, and fish; a moderate intake of dairy products, eggs, and lean meats; moderate wine consumption with meals; and low red meat, processed meats, refined carbohydrate, and sweet intake (Trichopoulou, 2001; Trichopoulou & Lagiou, 1997; Willett et al., 1995). This cultural model for healthy eating is presented graphically as pyramid in Figure 5.1. Total fat in this diet is 25% to 35% of energies, with saturated fat at 8% or less of energies.

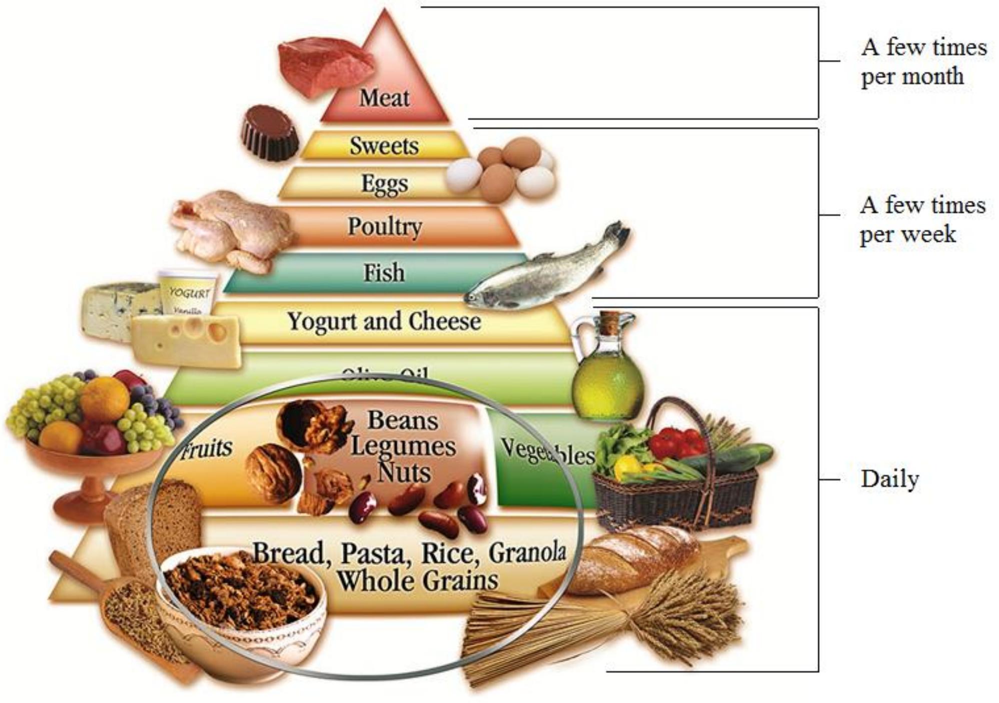


Figure 5.1 Mediterranean diet pyramid (Mediterranean diet pyramid, 2009)

It is a nutritionally balanced diet including B-complex vitamins, vitamins A and D, polyunsaturated fatty acids (PUFA), flavones (phytochemicals) and other antioxidants which are all known to have many beneficial effects including anti-inflammatory, anti-carcinogenic activities and cardio-protective and neuro-protective effects. In addition to its functionally active ingredients, since it has also low energy content, this style nutrition has been shown to be effective in weight loss in obese population (Roswall et al., 2014).

The most striking effect of this diet is reduction of cardiometabolic disease risks. Recently, in the Prevención con Dieta Mediterránea (PREDIMED) study, a large intervention trial, significant reductions in incident cardiovascular disease and diabetes mellitus were shown in the Mediterranean diet groups as compared to the low-fat control group (Estruch et al., 2013). Another study shows that, this healthy diet reduces the risk of all-cause and cardiovascular mortality (Reedy et al., 2014).

Mediterranean diet has been shown to be protective for some cancers or premalign lesions in adult populations, including hepatocellular cancer (Turati et al., 2014), colorectal adenomas and cancers (Schwingshackl & Hoffmann, 2014; Whalen et al., 2014), breast cancer (Castelló et al., 2014), oropharyngeal cancers (Filomeno et al., 2014) and prostate cancers (Schwingshackl & Hoffmann, 2014). Moreover, in a recent meta-analysis, adherence to Mediterranean diet strictly, has been found associated with the lowest risk for overall cancer mortality (Schwingshackl & Hoffmann, 2014).

Neuro-protection and prevention of chronic degenerative brain diseases in elderly population are other important health effects of Mediterranean diet (Tangney et al., 2014; Wengreen et al., 2013). Especially higher intakes of nuts, whole grains and legumes have found associated with more protective effects (Wengreen et al., 2013).

The traditional Mediterranean diet provides substantial protection against type 2 diabetes (Martínez-González et al., 2008). Also, the Mediterranean diet is associated with lower blood pressure, blood sugar, and triglycerides according to an important meta-analysis (Kastorini et al., 2011).

Olive oil is considered as the principle source of fat in the Mediterranean diet and some studies show that olive oil consumption improves cholesterol regulation and it has anti-hypertensive effects (Covas, 2007). Mediterranean diet also includes health effects of red wine that contains flavonoids with powerful antioxidant properties (Baron-Menguy et al., 2007).

5.3 Methodology

5.3.1 Description of the Dataset

In scope of this study, nutrition data were obtained from the USDA National Nutrient Database for Standard Reference, Release 27 (USDA, 2014). The database contains nutrition values of 8,618 different foods grouped by main categories. We have eliminated all the food and categories incompatible with Mediterranean diet. The resulting database contains 2,586 food and 19 categories to use in menu optimization as seen in Table 5.1.

Table 5.1 Food categories used in our study

Id	Category Name	Breakfast	Lunch	Dinner	Snack	Frequency
1	Baby Foods					
2	Baked Products	√	√	√	√	●●●
3	Beef Products		√	√		●
4	Beverages	√	√	√	√	●●●
5	Breakfast Cereals					
6	Dairy and Egg Products	√	√	√	√	●●●
7	Ethnic Foods		√	√		●●●
8	Fast Foods					
9	Fats and Oils	√	√	√		●●●
10	Finfish and Shellfish Products		√	√		●●●
11	Fruits and Fruit Juices	√			√	●●●
12	Grains and Pasta		√	√		●●●
13	Herbs and Spices		√	√		●●●
14	Lamb, Veal, and Game Products		√	√		●
15	Legumes and Legume Products		√	√		●●●
16	Meals, Entrees, and Side dishes		√	√		●●●
17	Nut and Seed Products	√			√	●●●
18	Pork Products		√	√		●
19	Poultry Products		√	√		●●
20	Sausages and Luncheon Meats					
21	Snacks					
22	Soups, Sauces, and Gravies	√	√	√	√	●●●
23	Sweets	√	√	√	√	●
24	Vegetables and Vegetable Products	√	√	√	√	●●●

Four levels of frequency score are assigned to all food categories depending on the degree of correspondence with the Mediterranean diet recommendations. An empty value in frequency column means entire category is never used in menu optimization. Scores ●●●, ●●, and ● are used for the categories which should be consumed daily, a few times per week and a few times per month, respectively (Hammond, 2013). Frequency score of some foods is assigned in food level. For example, while Dairy and Egg Products category assumed as can be consumed daily, Egg Products in this category is scored as can be consumed a few times per week.

Some foods cannot be consumed at each meal. For instance, it is not customary and not appropriate according to the logic of Mediterranean diet to eat beef products for breakfast. So we mark the categories can be eaten at breakfast, lunch, dinner and snack. 26 features of each food are evaluated to optimize a dietary menu for a specific profile. These 26 features are energy, carbohydrate, protein, fat, 11 types of vitamins, and 10 types of minerals given in Table 5.2.

Table 5.2 Evaluated nutritive values

Macronutrients	**Vitamins**	**Minerals**
✓ Energy	✓ Vitamin A	✓ Calcium
✓ Carbohydrate	✓ Vitamin C	✓ Iron
✓ Protein	✓ Vitamin E	✓ Magnesium
✓ Fat	✓ Vitamin K	✓ Phosphorus
✓ Water	✓ Vitamin B1	✓ Potassium
	✓ Vitamin B2	✓ Sodium
	✓ Vitamin B3	✓ Zinc
	✓ Vitamin B5	✓ Copper
	✓ Vitamin B6	✓ Manganese
	✓ Vitamin B9	✓ Selenium
	✓ Vitamin B12	

5.3.2 Energy Calculation

A healthy diet requires consuming the right amount of food and taking appropriate amounts of nutrients. Personal information consists of age, sex, weight, and height are taken into account to determine Recommended Dietary Allowances (RDA) established in the Dietary Reference Intake (DRI) (Institute of Medicine [IOM], 1997, 1998, 2000, 2001, 2002, 2004, 2011). RDA is the daily dietary intake level of a nutrient considered sufficient by the Food and Nutrition Board to meet the requirements of nearly all (97–98%) healthy individuals in each life-stage and gender group.

There are two known formulae to calculate Basal Metabolic Rate (BMR) and Resting Metabolic Rate (RMR) give how much energy burned in a day based on a general activity level. The BMR can be calculated using Harris-Benedict equations (Frankenfield, Muth, & Rowe, 1998). Eq. 5.1 is used for men and Eq. 5.2 is used for women.

$$P = (13.75\times m + 5\times h + 6.76\times a + 66)\ \text{kcal/day} \tag{5.1}$$

$$P = (9.56\times m + 1.85\times h + 4.68\times a + 655)\ \text{kcal/day} \tag{5.2}$$

where, P is basal metabolic total energy, m is the weight in kg, h is the height in cm, and a is the age in year.

The RMR can be calculated using Mifflin’s equation given below (Mifflin et al., 1990):

$$P = (10.0\times m + 6.25\times h + 5.0\times a + s)\ \text{kcal/day} \tag{5.3}$$

where, P is total energy at complete rest, m is the weight in kg, h is the height in cm, and a is the age in year, where s is +5 for males and −161 for females.

As BMR and RMR only represent resting energy expenditure, an adjustment must be made to reflect the activity level. This is done by multiplying the BMR or RMR by an activity factor (Dietary Reference Intakes [DRI] tables, 2010). The Physical Activity Coefficients given in Table 5.3 corresponds to which activity level the user has: Sedentary, Low Active, Active, or Very Active. The explanation of the activity levels are listed below:

- **Sedentary** Typical daily living activities (e.g., household tasks, walking to the bus)
- **Low Active** Typical daily living activities + 30 - 60 minutes of daily moderate activity (ex. walking at 5-7 km/h)
- **Active** Typical daily living activities + at least 60 minutes of daily moderate activity

- **Very Active** Typical daily living activities + at least 60 minutes of daily moderate activity + an additional 60 minutes of vigorous activity or 120 minutes of moderate activity

Table 5.3 Physical Activity Coefficients (PAC)

	Sedentary	Low Active	Active	Very Active
Boys 3 – 18 y	1.00	1.13	1.26	1.42
Girls 3 – 18 y	1.00	1.16	1.31	1.56
Men 19 y +	1.00	1.11	1.25	1.48
Women 19 y +	1.00	1.12	1.27	1.45

Estimated Energy Requirement (EER) is calculated for boys and girls between 3-18 years by using Eq. 5.4 and 5.5, respectively.

$$EER= (88.5-61.9\times a+PAC \times (26.7\times m+ 903\times h) + s) \text{ kcal/day} \quad (5.4)$$

$$EER= (135.3-30.8\times a+PAC \times (10.0\times m+ 934\times h) + s) \text{ kcal/day} \quad (5.5)$$

where, m is the weight in kg, h is the height in cm, and a is the age in year, where s is +20 for 3-8 years children and +25 for 9-18 years boys and girls.

EER is calculated for men and women 19 years and older by using Eq. 5.6 and 5.7, respectively.

$$EER= (662-9.53\times a+PAC \times (15.91\times m+ 539.6\times h)) \text{ kcal/day} \quad (5.6)$$

$$EER= (354-6.91\times a+PAC \times (9.36\times m+ 726\times h)) \text{ kcal/day} \quad (5.7)$$

where, m is the weight, h is the height, and a is the age.

Table 5.4 Recommended macronutrient proportions by age

	Carbohydrate	Protein	Fat
1–3 years	45–65%	5–20%	30–40%
4–18 years	45–65%	10–30%	25–35%
19 years +	45–65%	10–35%	20–35%

EER is obtained by applying corresponding Physical Activity Coefficient according to user’s sex, age and activity level on appropriate EER formula. Energy need for a pregnant is considered 300 kcal more than a female while for lactation period 500 kcals more. The Institute of Medicine has established ranges for the percentage of energies in the diet that should come from carbohydrate, protein, and fat as shown Table 5.4 (Gebhardt & Thomas, 2002). These ranges are the minimum and maximum percentages of each macronutrient that should comprise total caloric intake and they are taken into account both chronic disease risk reduction and intake of essential nutrients. Energy yield of macronutrients can be seen in Table 5.5 (DRI, 2010).

Table 5.5 Energy yield of macronutrients

	Carbohydrate	**Protein**	**Fat**	**Alcohol**
Energy	4 kcal /g	4 kcal /g	9 kcal /g	7 kcal /g

According to the advices of dieticians, the energy distribution of daily 3 meals should be about 343 for breakfast, lunch and dinner respectively. Most of the dieticians advice 3 snacks which are not exceeding 200 kcal, after two hours following main meals to complete our energy account, ensure that the blood sugar balance in our bodies, reduce the lack of attention and concentration and avoid eating too much in main meals. In this case, energy distribution should be about 314131. An optimized dietary menu can be generated for every meal, daily, weekly or monthly by using this energy distribution rule.

5.4 The Diet Optimization as Knapsack Problem

The diet optimization problem (DOP) contains a big search space with foods and their attributes. The target values of energy, carbohydrate, fat, protein and other minerals-vitamins cause many combination with an aspect of linear search principles. Big search space and many combinations have leaded DOP to evolutionary optimization techniques (Garey & Johnson, 1990).

DOP is a Knapsack Problem (KP). Both of two have target fitness value as a capacity like weight and energy. The target weight at KP is equal to the target energy at DOP. Thus, the maximum values of boxes at KS are equal to the optimum values of carbohydrate, fat, protein and other minerals-vitamins. Therefore, KP has been assumed the kernel point at this study.

The type of DOP is “0-1 KP” which restricts the number xi of duplicates of each kind of item to zero or one; because all food are represented as a gene in a chromosome and the value of each gene can be zero or one. It means that this food should have stayed on the menu or should not. 0-1 KP is formulated as the formula below;

$$\text{Optimum of } \sum_{i=0}^{n}(V_i X_i) \qquad \text{Subject to } \sum_{i=0}^{n}(c_i X_i) \leq \mathrm{C} \tag{5.8}$$

where X is the chromosome, Xi 0-1 gene, Vi represent optimum value of each carbohydrate, fat, protein and other minerals-vitamins of ith food, ci is the value of energy of ith food, and C is the target energy (Garey & Johnson, 1990).

0-1 KP is a NP-Complete problem, because Dynamic Programming, Meet-in-the-middle or other not-evolutionary algorithms run in pseudo-polynomial times (Garey & Johnson, 1990). The run time cannot be predicted at the evolutionary optimization solutions. The target value, dynamic search space, and parametric boundaries as generation number and population size, supply flexibility about run-time. In this study, the evolutionary optimization solutions like Simple Genetic Algorithm, Quantum Genetic Algorithm and Hybrid Quantum Genetic Algorithm are concentrated to test adaptability for DOP.

5.5 Simple Genetic Algorithm (SGA)

SGA is a natural selection and optimization technique based on genetic evolutionary argument. These processes are simulated based on evaluation principles of Lamarck and Darwin (Rabinovich & Wigderson, 1999).

The first step of SGA is to initialize of the first population P(0) by 0 and 1 randomly. All individuals in this population evaluate by using the fitness function and a generation number is determined (Jebari & Madiafi, 2013). After this step, SGA triggers a loop where there are methods which are repeated respectively. The first method is selectionOperator() which obtains the first two fittest individuals according to the fitness function. The second method is crossoverOperator() which obtains the hybrid-new individuals by crossing over these individuals (Sivanandan & Deepa, 2008). The third method, mutationOperator() is to supply to get rid of local optimum by changing a random gene of random number of individuals. The fourth method is calculateAdaptability() which supplies to obtain a new adaptable population, and finally, findBestIndividual() method calculates fitness values of all individuals according to fitness function. The pseudo code below shows all steps of SGA (Kaya, Uyar, & Tekin, 2011);

```
 Simple Genetic Algorithm Pseudo Code
Begin
   • Choose an initial population of individuals P(0);
   • Evaluate the fitness of all individuals P(0);
   • Choose a maximum number of generations $t_{max}$;
   • While (not satisfied $t < t_{max}$)
     Begin
        • t = t+1;
        • Select parents for offspring production; (with
          selectionOperator() method)
        • Apply reproduction and mutation operators ;(with
          crossoverOperator() and mutationOperator() methods)
        • Create a new population of survivors P(t);(with
          calculateAdaptability() method)
        • Evaluate P(t);(with findBestIndividual() method)
     End
   • Return the best individual of P(t);
End
```

The complexity of the selectionOperator() that requires reordering, if the class of the possible fitness functions to varying fitness functions is restricted, is O(NlogN), where N is the size of the population (Eiben & Smit, 2011).

SGA has been implemented in C# in Visual Studio .NET 2013 platform and has run with a sample of the food dataset. The results with comparisons between SGA and other approaches are given the next section.

5.6 Quantum Genetic Algorithm (QGA) and Hybrid Quantum Genetic Algorithm (HQGA)

The base structure of QGA is similar to the structure of SGA. A quantum selection operation that includes a quantum fitness evaluation unit is used instead of selectionOperator() in QGA to solve the problem using a “global search” (Hey, 1999). QGA is based on the concepts of qubits and superposition of states of quantum mechanics. The smallest unit which stored information in a quantum computer is called a quantum bit (qubit) (Han & Kim, 2000). A qubit does not get only ‘1’ or ‘0’; it may get a value between 0 and 1. The state of a qubit (Ψ) is represented like the formula below;

$$|\Psi> = \alpha|0> + \beta|1> \tag{5.9}$$

where α and β are complex numbers that specify the probability amplitudes of the corresponding states. $|\alpha|^2$ gives the probability of ‘0’ state for the qubit and $|\beta|^2$ gives the probability of ‘1’ state for the qubit. For normalization of the state, the following formula is obtained;

$$|\alpha|^2 + |\beta|^2 = 1 \tag{5.10}$$

The attractive point in QGA is that when there is a system of m-qubits, the event of 2^m states at the same time is expected of the system; however, it collapses to a single state in the act of observing a quantum state. This situation is interested in the complexity of QGA directly and the complexity collapses to O(1) (Malossini, Blanzieri & Calarco, 2008). An m-qubits representation is defined like the formula below;

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array} \right], \tag{5.11}$$

This formula shows the advantage which the qubits can represent any superposition of states. If there is, for instance, a two-qubits system with two pairs of amplitudes such as;

$$\left[\begin{array}{c|c} 1/\sqrt{5} & 1/2 \\ 2/\sqrt{5} & \sqrt{3}/2 \end{array}\right], \tag{5.12}$$

The state of the system is represented as;

$$(1/2\sqrt{5})\ |00> \ + (\sqrt{3}/2\sqrt{5})\ |01> \ + (1/\sqrt{5})\ |10> \ + (\sqrt{3}/\sqrt{5})\ |11> \tag{5.13}$$

The formula in Eq. 5.13 shows that the probabilities to represent the state |00>, |01>, |10> and |11> are 1/20, 3/20, 4/20, 12/20, sequentially. As a result, the two-qubits system in Eq. 5.12 has four states information simultaneously. In DOP, 2586 foods means 2586 qubits and 22586 probabilities can be considered in the same time.

$$P(t) = \{p1, p2,p3,..., pn\}\ , pi = [\alpha 1, \alpha 2, \alpha 3, ..., \alpha m; \beta 1, \beta 2, \beta 3, ..., \beta m] \tag{5.14}$$

In QGA, the first operation is initialization of each qubit (α, β) with 1/√2 in chromosomes in the population identically. In Eq. 5.14, n is the number of foods (2586) and m is the selected population size. The second step is to make P(t) by observing the states with makeOperator() method. This method uses a randomization technique between 0 and 1 to assign the value of qubits. The pseudo code of makeOperator() is like the following method (Malossini, Blanzieri & Calarco, 2008);

```
makeOperator(individual as x, length of x as l) Pseudo Code
Begin
   •  i = 0;
   •  While (not satisfied i < l)
      Begin
         •  i = i + 1;
         •  If random[0,1) > $|\alpha_i|^2$
         •  Then $x_i$ = 1;
         •  Else $x_i$ = 0;
      End
End
```

The next operations are that determining a generation number, evaluating all chromosomes and storing the global optimum chromosome. Until generation number is reached, makeOperator() is used again, then evaluating and storing the global individual are repeated. In this loop, the most important operation is updateOperator() method.

All qubits are updated according to the best individual to approximate it angular. Updating formula of qubits is like the formula in Eq. 5.15 (Malossini et al., 2008).

$$\begin{bmatrix} \alpha_t \\ \beta_t \end{bmatrix} = \begin{bmatrix} Cos(\Delta\theta_t) & -Sin(\Delta\theta_t) \\ Sin(\Delta\theta_t) & Cos(\Delta\theta_t) \end{bmatrix} \cdot \begin{bmatrix} \alpha_{t-1} \\ \beta_{t-1} \end{bmatrix}, \qquad (5.15)$$

$\theta = s(\alpha t, \beta t)$. $\Delta\theta t$, the value $s(\alpha t, \beta t)$ and orientation of rotation angle $\Delta\theta t$ are obtained from the following lookup table (Malossini, Blanzieri & Calarco, 2008).

Table 5.6 Lookup table of θ_t, where f is the fitness function, $s(\alpha_t, \beta_t)$ is the sign of θ_t, and b_t and x_t are the tth bits of the best solution B and the binary solution X, in sequence.

x_t	b_t	$f(X) \geq f(B)$	$\Delta\theta_t$	$s(\alpha_t, \beta_t) > 0$	$s(\alpha_t, \beta_t) < 0$	$\alpha_t = 0$	$\beta_t = 0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	0	0	0	0	0
0	1	True	0.05π	-1	+1	±1	0
1	0	False	0.01π	-1	+1	±1	0
1	0	True	0.025π	+1	-1	0	±1
1	1	False	0.005π	+1	-1	0	±1
1	1	True	0.025π	+1	-1	0	±1

The pseudo code of updateOperator() is like the following method;

```
updateOperator(individual as x) Pseudo Code
Begin
    • i = 0;
    • While(not satisfied i < l)
      Begin
          • i = i + 1;
          • Determine $\theta_i$ with the lookup table;
          • Obtain $(\alpha'_i, \beta'_i)$ as $[\alpha'_i\ \beta'_i]^T = U(\theta_i)[\alpha_i\ \beta_i]^T$;
      End
      x = x';
End
```

QGA has been implemented in C# in Visual Studio .NET 2013 platform and has run with a sample of the food dataset. The results with comparisons between QGA and other approaches are given the next section. The pseudo code of QGA is like the following method;

```
Quantum Genetic Algorithm Pseudo Code
Begin
   •  t = 0;
   •  Initialize Q(t);
   •  Make P(t) by observing Q(t) states;(with makeOperator()
      method)
   •  Evaluate P(t);
   •  Choose a maximum number of generations $t_{max}$;
   •  Store the best individual B among P(t);
   •  While (not satisfied $t < t_{max}$)
      Begin
         •  t = t+1;
         •  Make P(t) by observing Q(t-1) states;
         •  Evaluate P(t);
         •  Update Q(t);(with updateOperator() method)
         •  Store the best individual B among P(t);
      End
End
```

In this study, QGA has broadened by using repairOperator() and searchOperator() methods. The repairOperator() runs to approach the optimum energy which is the most important parameter of DOP. The searchOperator() runs to approach the optimum protein, fat and carbohydrate linearly. The aim of this hybrid structure is to obtain the optimum results at less generation number and population size.

```
repairOperator(individual as x, length of x as l, weights as w,
capacity as C) Pseudo Code
Begin
   •  overfilled = false;
   •  If $\sum_{i=0}^{l} w_i x_i > C$
   •  Then overfilled = true;
   •  While (overfilled)
      Begin
         •  Select an ith item from data set;
         •  $x_i = 0$;
         •  If $\sum_{i=0}^{l} w_i x_i \le C$
         •  Then overfilled = false;
      End
   •  While (not overfilled)
      Begin
         •  Select an ith item from data set;
         •  $x_i = 1$;
         •  If $\sum_{i=0}^{l} w_i x_i > C$
         •  Then overfilled = true;
      End
      $x_i = 0$;
End
```

The results with comparisons between HQGA and other approaches are given the next section and it has been observed that more successful results had been obtained by using HQGA with extra two methods.

```
Hybrid Quantum Genetic Algorithm Pseudo Code
Begin
   •  t = 0;
   •  Initialize Q(t);
   •  Make P(t) by observing Q(t) states;(with makeOperator()
      method)
   •  Repair P(t);(with repairOperator() method)
   •  Evaluate P(t);
   •  Choose a maximum number of generations $t_{max}$;
   •  Store the best individual B among P(t);
   •  While (not satisfied t < $t_{max}$)
      Begin
         •  t = t+1;
         •  Make P(t) by observing Q(t-1) states;
         •  Repair P(t);(with repairOperator() method)
         •  Evaluate P(t);
         •  Update Q(t);(with updateOperator() method)
         •  Store the best individual B among P(t);
      End
   •  Local Search for the best individual B;(with searchOperator()
      method)
End
```

HQGA has been implemented in C# in Visual Studio .NET 2013 platform and has run with a sample of the food dataset like the other algorithms. The pseudo code of HQGA is like the method above.

The searchOperator() is expressed in detal in the following pseudo code;

```
searchOperator(individual as x, weights w) Pseudo Code
Begin
   •  Determine the minimum as $v_1$ and the maximum ones as $v_2$
      from the target protein, carbohydrate  and fat;
   •  If v2 = 'protein'
   •  Then Select the i^th item which has the highest protein
      while $x_i$ = 1;
           Determine the coefficient as c to equate the target
           protein;
           $w_i = w_i \times c$;
   •  Else If v2 = 'carbohydrate'
   •  Then Select the i^th item which has the highest carbohydrate
      while $x_i$ = 1;
           Determine the coefficient as c to equate the target
           carbohydrate;
```

```
•       $w_i = w_i \times c$;
•  Else If v2 = 'fat'
•  Then Select the ith item which has the highest fat
   while $x_i = 1$;
        Determine the coefficient as c to equate the target fat;
        $w_i = w_i \times c$;
•  If v1 = 'protein'
•  Then Select the ith item which has the highest protein
   while $x_i = 0$;
        Determine the coefficient as c to equate the target
        protein;
        $x_i = 1$; $w_i = w_i \times c$;
•  Else If v1 = 'carbohydrate'
•  Then Select the ith item which has the highest carbohydrate
   while $x_i = 0$;
        Determine the coefficient as c to equate the target
        carbohydrate;
        $x_i = 1$; $w_i = w_i \times c$;
•  Else If v1 = 'fat'
•  Then Select the ith item which has the highest fat
   while $x_i = 0$;
        Determine the coefficient as c to equate the target fat;
        $x_i = 1$; $w_i = w_i \times c$;
End
```

5.7 The Comparison Results of SGA, QGA and HQGA

SGA, QGA and HQGA have been run on the computer, which has 8 GB of RAM and Intel(R) Core(TM) i7-3770T CPU @ 2.50 GHz, to test the performances of these algorithms. The same dataset, which had been taken from the food dataset as a sample, has become input to SGA, QGA and HQGA in separately. Comparisons have been expressed on 3D graphics at 3 different subjects as total value, total energy and time on the next figures. Generation numbers and population size have varied from 1 to 50 for both these two parameters. It means that different variations have been obtained for 2500 probabilities. The target energy has been assumed as 1000 and while reaching to the value of 1000, total nutrition values of foods have had to possess maximum numbers. The expected total value has been approximately 5000.

In Figure 5.2, there are the variations of SGA and QGA. The first variation in Figure 5.2 is for SGA. When population size has been stable and while generation number has been increasing, becoming slight of the variation of total value has been observed. By consequence, the inference is able to obtain that total value is influenced by population size more than generation number. At each increment of population size, higher values than the previous ones have been attained and after 20 for population size, it has been observed that total value has been 4000 approximately. The second variation in Figure 5.2 is for QGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total value. While generation number and population size have been increasing, it is clear that the variation of total value has been observed. However, this 3D graphic is different from SGA's graphic and it shows a heterogenic variation with irregular increments and decrements of inclined. The reason of this situation is said as that QGA had been tried to get rid of local optimums.

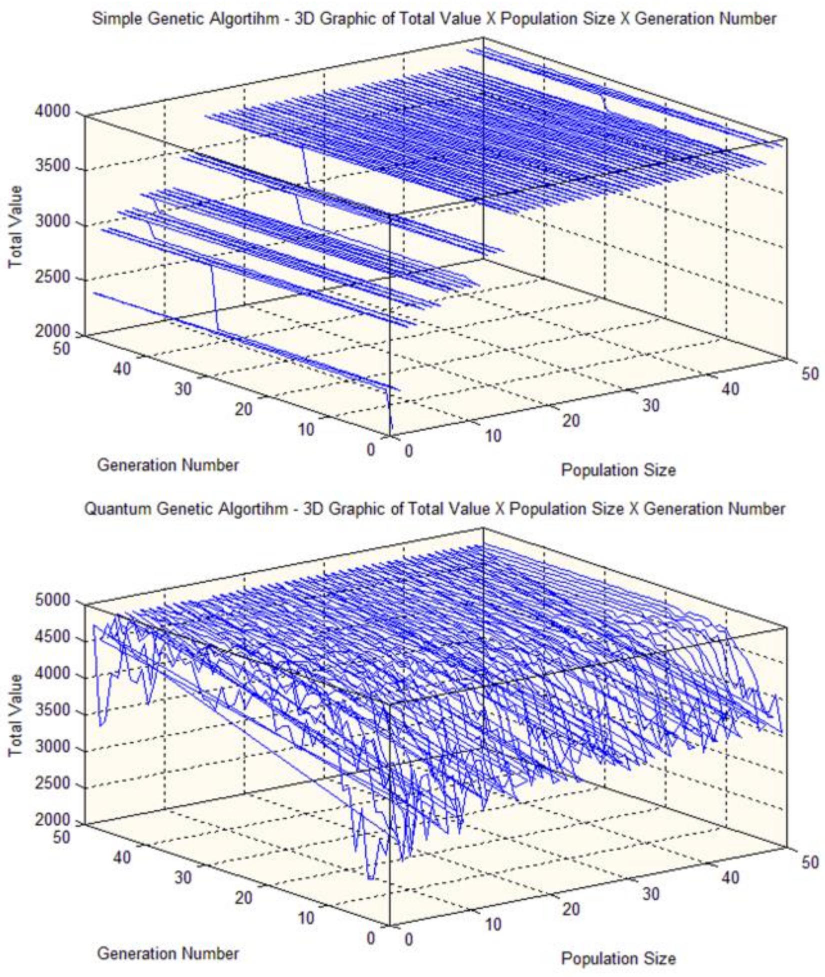


Figure 5.2 3D graphics of total value x population size x generation number (SGA & QGA)

Another inference is obtained that at low values of population size and generation number, the total value for QGA is higher than the total value for SGA. Also, QGA has higher the total values than SGA at all probability of generation number and population size and the total value is obtained as 3000 approximately at the first generation. Finally, the total value converges 5000 by increment of generation number. The variation in Figure 5.3 is for HQGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total value. However, while generation number and population size have been increasing, it is not clear that the variation of total value has been observed. HQGA reaches the target value as 5000 approximately except the generation number between 1 and 10 and population size between 1 and 10. It means that the value of approximately 5000 can be obtained at the lower probabilities.

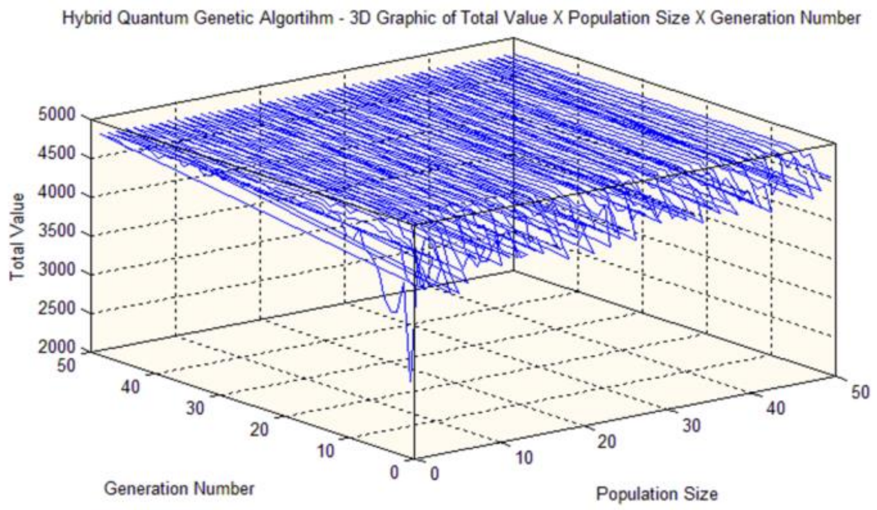


Figure 5.3 3D graphic of total value x population size x generation number for HQGA

In Figure 5.4, there are the variations of SGA and QGA. The first variation in Figure 5.4 is for SGA. When population size has been stable and while generation number has been increasing, becoming slight of the variation of total energy has been observed. By consequence, the inference is able to obtain that total energy is influenced by population size more than generation number. At each increment of population size, higher energies than the previous ones have been attained and after 20 for population size, it has been observed that total energy has been 1000. The second variation in Figure 5.4 is for QGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total energy. While generation number and population size have been increasing, it is clear that the variation of total energy has been observed. However, this 3D graphic is different from SGA's graphic and it shows a heterogenic variation with irregular increments and decrements of inclined. The reason of this situation is said as that QGA had been tried to get rid of local optimums.

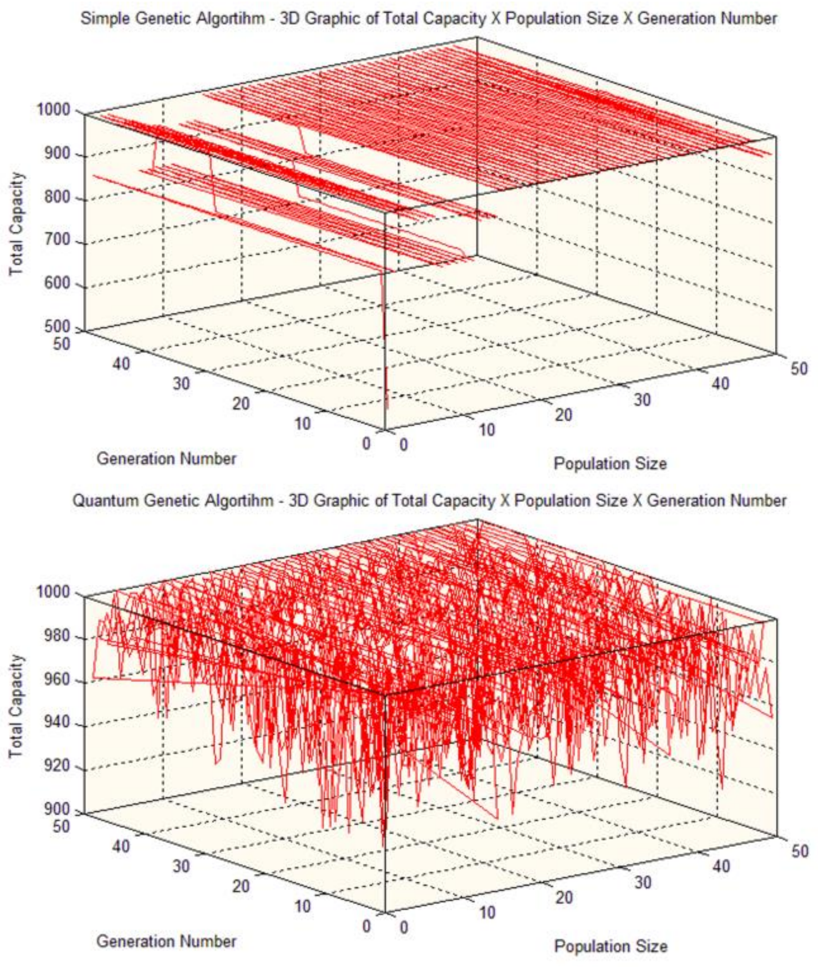


Figure 5.4 3D graphics of total capacity x population size x generation number (SGA & QGA)

The variation in Figure 5.5 is for HQGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total energy. However, while generation number and population size have been increasing, it is not clear that the variation of total energy has been observed; it shows a heterogenic variation with irregular increments and decrements of inclined.

The reason of this situation is said as that HQGA had been tried to get rid of local optimums. While QGA is at energy of 920, HQGA is at energy of 940 and earlier, HQGA reaches to the target energy, 1000.

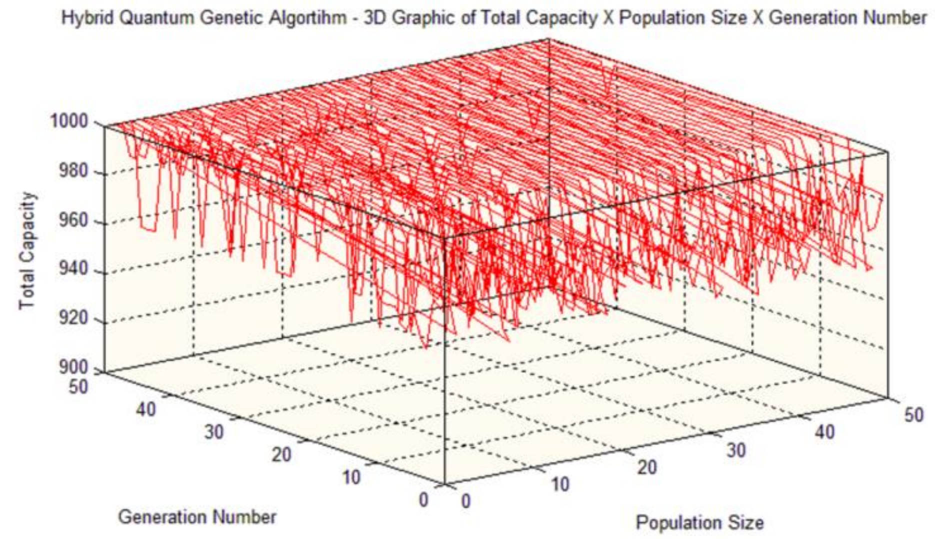
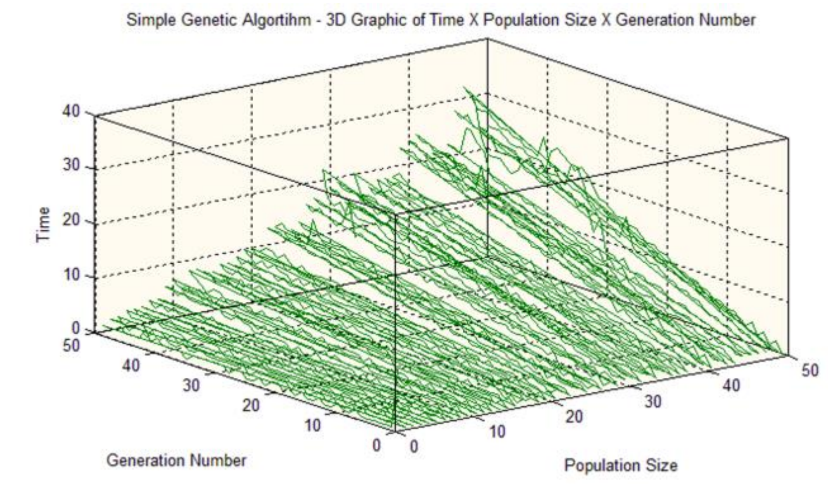


Figure 5.5 3D graphic of total capacity x population size x generation number for HQGA

The first 3D graphic in Figure 5.6 is for SGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total time. While both of these parameters have been increasing, total time has been increasing, too. However, all operations have taken very little time in level of millisecond (ms). Maximum time of running of SGA has been 32 ms for 50 of population size and 50 of generation number. The second 3D graphic in Figure 5.6 is for QGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total time. While both of these parameters have been increasing, total time has been increasing, too. However, all operations have taken very little time in level of millisecond (ms). Maximum time of running of QGA has been 2000 ms for 50 of population size and 50 of generation number. It is higher than the total time of SGA. The 3D graphic in Figure 5.7 is for QGA. In this 3D graphic, it has been observed that both population size and generation number have been ascendant at the variation of total time. While both of these parameters have been increasing, total time has been increasing, too. However, all operations have taken very little time in level of millisecond (ms).



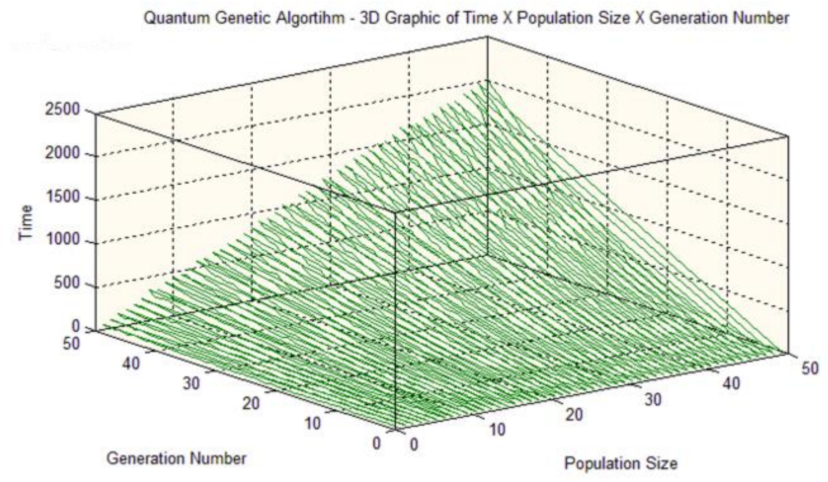


Figure 5.6 3D graphics of time x population size x generation number (SGA & QGA)

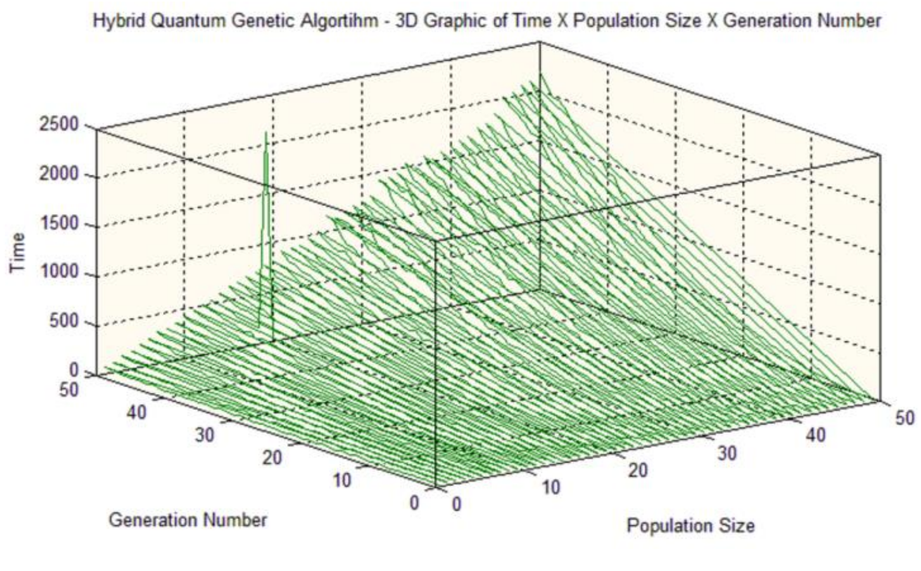


Figure 5.7 3D graphic of time x population size x generation number for HQGA

Maximum time of running of HQGA has been 2300 ms for 50 of population size and 50 of generation number. It is higher than the total time of SGA and QGA. The reason of this situation, HQGA has 2 different methods as repairOperator() and searchOperator(). (At the possibility of population size as 20 and generation number as 50, there has been an outlier increment which is different from the homogeny variation. It has been assumed that the situation of the computer at that moment had caused; therefore, any other inference has not been done from this outlier increment).

5.8 The Application of Diet Optimization and Experimental Studies

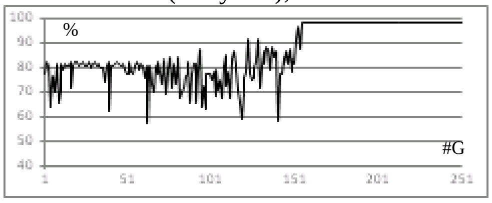
An application built with Web services is a service-oriented application. ASP.NET Web Services (ASMX) has been available for building Web services since .NET was first released. Then Microsoft introduced the new service model WCF that provides a number of benefits over ASP.NET Web Services. WCF service is a practical approach while looking at the current development trend. Therefore, the diet optimizer application has been developed as a WCF Web Service.

An Asp.Net web application has been implemented and published on the Internet. After the implementation phase, the algorithm has been tested with 20 different profiles. These profiles contain age, gender, pregnancy and lactation features.

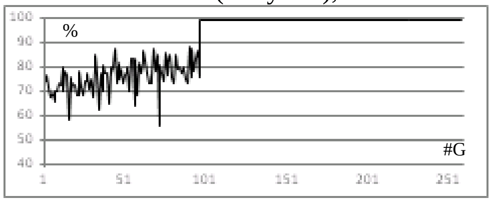
The energies, which have varied from a profile to another profile, have become the targets of the algorithm. For these target energies, HQGA has worked with the population size as 30 and unlimited generation number. HQGA has terminated at 145th generation on average and produced the optimal menus at 100% success rate as seen in Figure 5.8 and Figure 5.9, respectively.

The generated optimal menus guarantee the optimal energy, carbohydrate, protein and fat intake according to daily requirements. Recommended intake values are obtained from different sources which are explained in detail in section 5.3. Therefore, the application produces the optimal values between ranges.

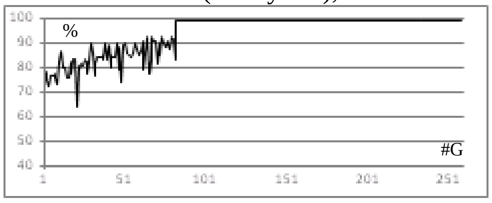
Profile 1 Child (1-3 years), 1300 calories



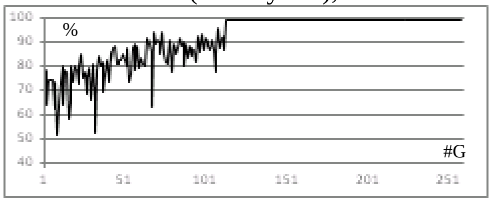
Profile 2 Child (4-8 years), 1800 calories



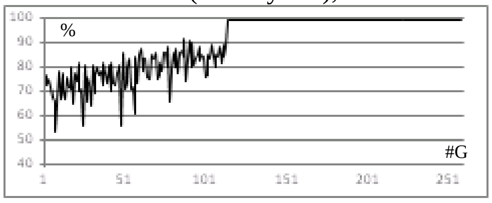
Profile 3 Male (9-13 years), 2500 calories



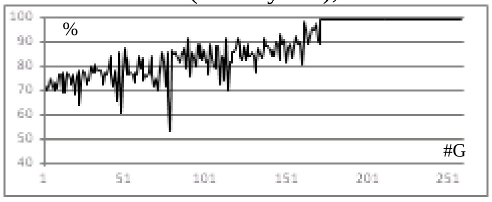
Profile 4 Male (14-18 years), 3000 calories



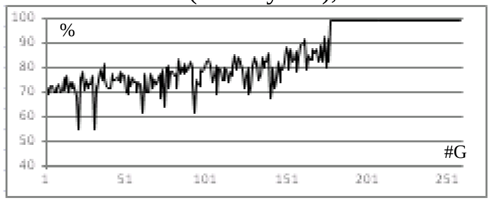
Profile 5 Male (19-30 years), 2900 calories



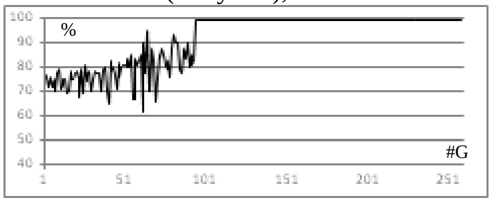
Profile 6 Male (31-50 years), 2900 calories



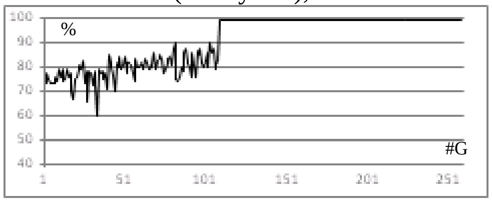
Profile 7 Male (51-70 years), 2300 calories



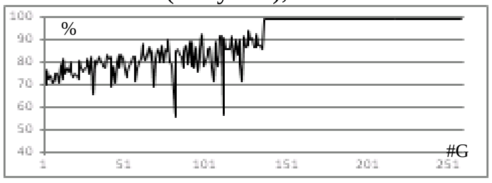
Profile 8 Male (71- years), 2300 calories



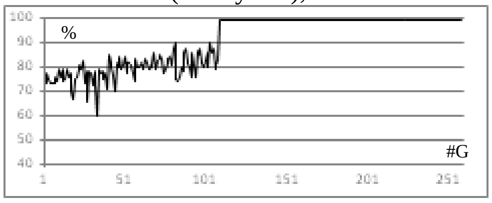
Profile 9 Female (9-13 years), 2200 calories



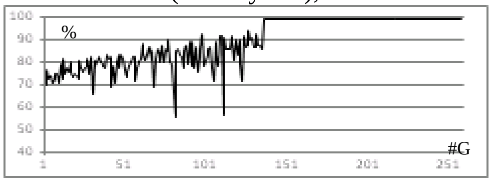
Profile 8 Male (71- years), 2300 calories



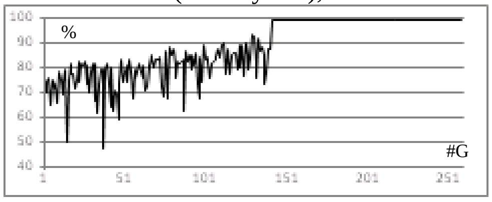
Profile 9 Female (9-13 years), 2200 calories



Profile 10 Female (14-18 years), 2200 calories



Profile 11 Female (19-30 years), 2200 calories



Profile 12 Female (31-50 years), 2200 calories

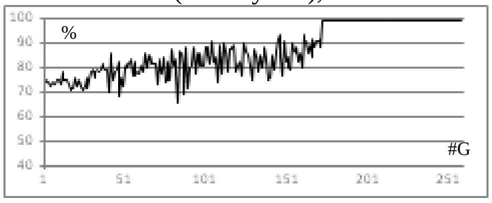


Figure 5.8 The variations according to 10 profiles which have different parameters. "Success rate (%) x Number of generations (#G)"

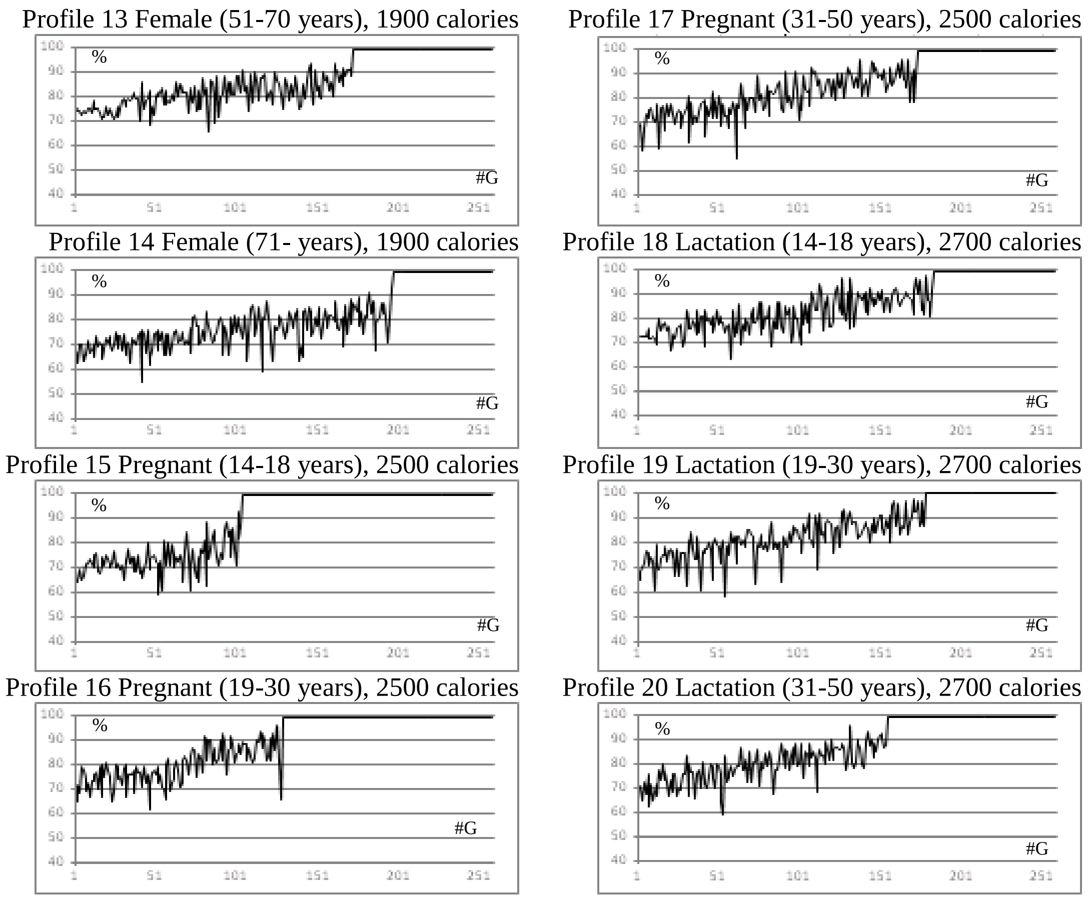
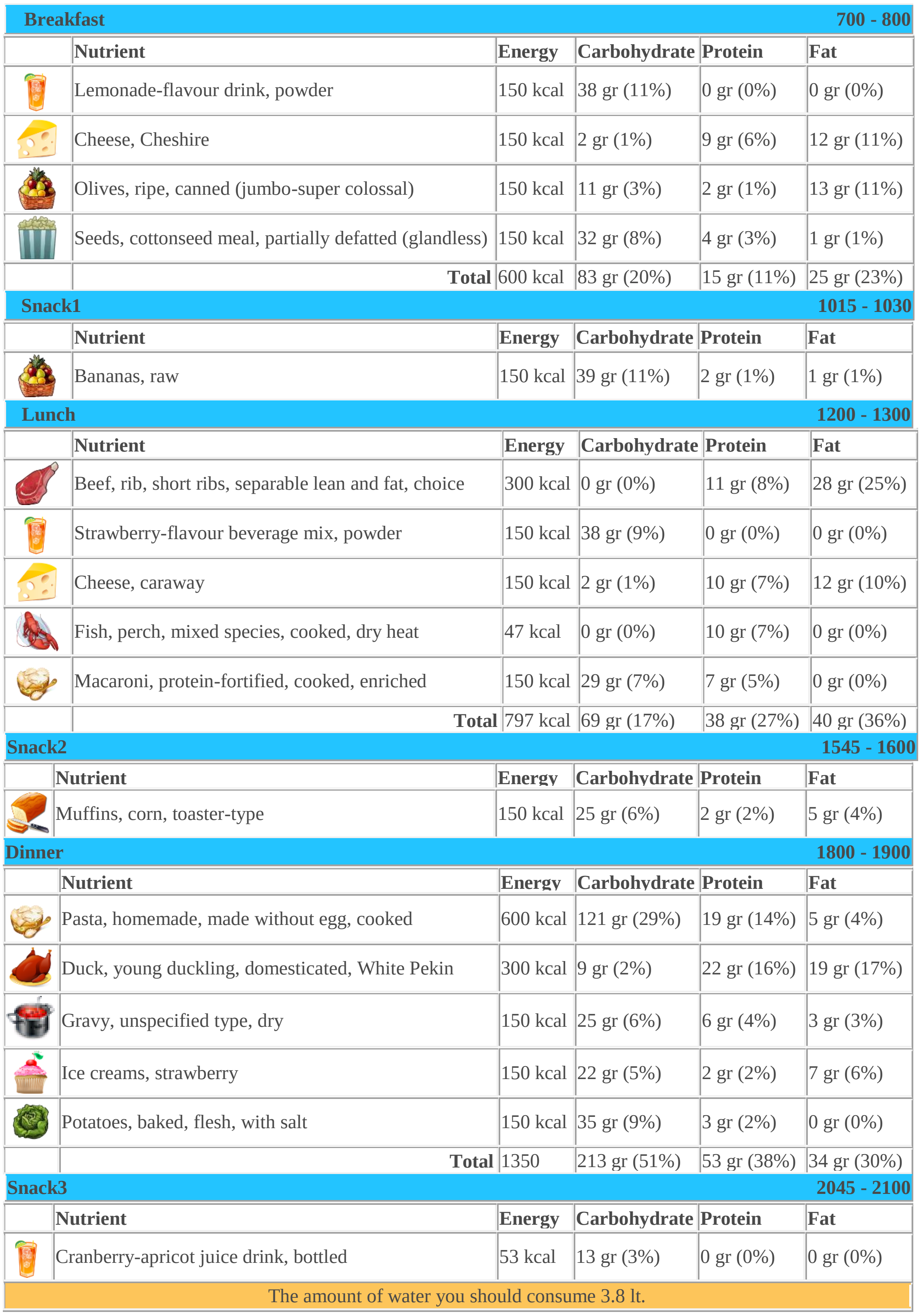


Figure 5.9 The variations according to 10 profiles which have different parameters. "Success rate (%) x Number of generations (#G)"

For the analysis of the menus, two users U1 and U2 who have different features have been handled. The different physical information of two users, U1 and U2 is gathered from the Nutrition Advisor web application as shown in Table 5.7. After individual nutritional analysis, daily nutritional requirement ranges and the values obtained from the optimized menus for U1 and U2 are listed in Table 5.8. The sample menu optimized for U1 is shown in Figure 5.10.

Table 5.7 Users' physical and health condition

	Sex	**Age (y)**	**Weight (kg)**	**Height (cm)**	**Physical Activity**
U_1	Female (Lactation)	30	75	170	Low Active
U_2	Male	55	82	178	Active
	Anamnesis	Don't Eat	Extra Physical Activities		
U_1	-	Egg	Cycling, 12-13.9 mph, moderate (45 min. – 451 kcal) General cleaning (60 min. – 263 kcal)		
U_2	Diabetes	-	Running, 7 mph (8.5 min mile) (30 min – 470 kcal)		



Breakfast					700 - 800
	Nutrient	Energy	Carbohydrate	Protein	Fat
	Lemonade-flavour drink, powder	150 kcal	38 gr (11%)	0 gr (0%)	0 gr (0%)
	Cheese, Cheshire	150 kcal	2 gr (1%)	9 gr (6%)	12 gr (11%)
	Olives, ripe, canned (jumbo-super colossal)	150 kcal	11 gr (3%)	2 gr (1%)	13 gr (11%)
	Seeds, cottonseed meal, partially defatted (glandless)	150 kcal	32 gr (8%)	4 gr (3%)	1 gr (1%)
	Total	600 kcal	83 gr (20%)	15 gr (11%)	25 gr (23%)

Snack1					1015 - 1030
	Nutrient	Energy	Carbohydrate	Protein	Fat
	Bananas, raw	150 kcal	39 gr (11%)	2 gr (1%)	1 gr (1%)

Lunch					1200 - 1300
	Nutrient	Energy	Carbohydrate	Protein	Fat
	Beef, rib, short ribs, separable lean and fat, choice	300 kcal	0 gr (0%)	11 gr (8%)	28 gr (25%)
	Strawberry-flavour beverage mix, powder	150 kcal	38 gr (9%)	0 gr (0%)	0 gr (0%)
	Cheese, caraway	150 kcal	2 gr (1%)	10 gr (7%)	12 gr (10%)
	Fish, perch, mixed species, cooked, dry heat	47 kcal	0 gr (0%)	10 gr (7%)	0 gr (0%)
	Macaroni, protein-fortified, cooked, enriched	150 kcal	29 gr (7%)	7 gr (5%)	0 gr (0%)
	Total	797 kcal	69 gr (17%)	38 gr (27%)	40 gr (36%)

Snack2					1545 - 1600
	Nutrient	Energy	Carbohydrate	Protein	Fat
	Muffins, corn, toaster-type	150 kcal	25 gr (6%)	2 gr (2%)	5 gr (4%)

Dinner					1800 - 1900
	Nutrient	Energy	Carbohydrate	Protein	Fat
	Pasta, homemade, made without egg, cooked	600 kcal	121 gr (29%)	19 gr (14%)	5 gr (4%)
	Duck, young duckling, domesticated, White Pekin	300 kcal	9 gr (2%)	22 gr (16%)	19 gr (17%)
	Gravy, unspecified type, dry	150 kcal	25 gr (6%)	6 gr (4%)	3 gr (3%)
	Ice creams, strawberry	150 kcal	22 gr (5%)	2 gr (2%)	7 gr (6%)
	Potatoes, baked, flesh, with salt	150 kcal	35 gr (9%)	3 gr (2%)	0 gr (0%)
	Total	1350	213 gr (51%)	53 gr (38%)	34 gr (30%)

Snack3					2045 - 2100
	Nutrient	Energy	Carbohydrate	Protein	Fat
	Cranberry-apricot juice drink, bottled	53 kcal	13 gr (3%)	0 gr (0%)	0 gr (0%)

The amount of water you should consume 3.8 lt.

Figure 5.10 The output of generated menu for U1

The fitness function in the application aims to obtain the keep organic and inorganic compounds in target interval values. In Table 5.8, there are energy, carbohydrate, protein and fat compounds between the target intervals and they are at 100%.

The local search method in the application supplies to obtain these target values. Other values are considered in fitness function; however, when these attributes were included in the local search, the performance decreased too much. Therefore, energy, carbohydrate, protein and fat as vital needs are considered as the kernel target in the application.

Table 5.8 Nutrition values obtained by optimized menus and the interval of requirements

		U_1		**U_2**	
		Value	Range	Value	Range
Macronutrients	Energy (kcal)	3099	2959-3413	2829	2770-2913
	Carbohydrate (g)	414	414-478	339	323-339
	Protein (g)	140	140-162	139	139-146
	Fat (g)	103	82-95	106	103-108
Vitamins	A (UI)	2511	4749-5478	3172	3613-3800
	C (mg)	112	132-152	111	108-114
	E (mg)	26	21-24	20	18-19
	K (mcg)	314	99-114	170	152-145
	B1 (mg)	2	2-2	1	1-2
	B2 (mg)	2	2-2	2	2-2
	B3 (mg)	22	19-21	23	19-20
	B5 (mcg)	5	8-9	5	6-6
	B6 (mg)	2	2-3	2	2-2
	B9 (mcg)	403	548-632	345	482-507
	B12 (mcg)	4	3-4	7	3-3
Minerals	Calcium (mg)	1083	1096-1264	1141	1204-1267
	Iron (mg)	18	10-11	12	10-10
	Magnesium (mg)	319	340-392	333	506-532
	Phosphorus (mg)	1324	767-885	1728	843-887
	Potassium (mg)	4434	5590-6447	3945	5660-5953
	Sodium (mg)	601	2740-3161	391	2770-2913
	Zinc (mg)	12	13-15	13	13-14
	Copper (mg)	3	1-2	2	1-1
	Manganese (mg)	3	3-3	4	3-3
	Selenium (mg)	68	77-88	94	66-70

5.9 Literature Comparisons

In literature, there are several studies which aim to recommend healthy menu according to user’s personal activities, energies and nutrition intakes, and/or personal information. Some of these studies have been exemplified for our study. On the following paragraphs, these studies are mentioned and the differences and our supplements are exposed clearly.

Lv (2009) proposed a multi-objective mathematic model for nutritional diet optimization program based on quantum genetic algorithm (QGA) to generate computer-aided diet schedules for people. QGA was compared with the random and backtracking algorithms and it is observed that the effectiveness and superiority of QGA are considerable evident. The differences of our study and Lv (2009)'s study are that as HQGA, which provides repair and search methods, is used for optimization in our study, Lv (2009) uses QGA without any improvements about local search. Therefore, as HQGA reaches to 100% for energy, carbohydrate, protein and fat, QGA remains under the success of HQGA.

Wang, Lee, Hsieh, Hsu, & Chang (2009) developed an intelligent healthy diet planning multi-agent for healthy diet planning. The study provides a semantic analysis of healthy diet status for people based on the pre-constructed ontology by domains experts and results of fuzzy inference. Experimental results show that the proposed system enables an intelligent behaviour able to generate the more suitable healthy diet for a given human being. Also, Pei & Liu (2009) used Compromise Differential Evolutionary algorithm to solve the issue of multi-objective nutrition diet decision and acquired more stable, more accurate results than that derived in genetic algorithms.

After all, these successful approaches present healthy receipts certainly; however, our study contains only the Mediterranean foods and presents a Mediterranean diet. The benefits on human being of the Mediterranean diet are proved by a lot of articles and publications definitely (Baron-Menguy et al., 2007; Castelló et al., 2014; Covas, 2007; Estruch et al., 2013; Filomeno et al., 2014; Kastorini et al., 2011; Keys et al.,1986; Martínez-González et al., 2008; Reedy et al., 2014; Roswall et al., 2014; Schwingshackl & Hoffmann, 2014; Tangney et al., 2014; Trichopoulou, 2001; Trichopoulou & Lagiou, 1997; Turati et al., 2014; Wengreen et al., 2013; Whalen et al., 2014; Willett et al., 1995). In section 5.2, the benefits and affects over physiologic of the Mediterranean diet are exposed in detail. Also, our study considers 22 parameters of vitamins and minerals in the fitness function distinctively in addition. Thus, beside optimum energy, carbohydrate, protein and fat, HQGA endeavours to obtain the fittest receipt for vitamins and minerals as much as possible.

Mino & Kobayashi (2009) considered user's activities as an important factor to recommend recipes, and then aim to propose a recipe recommendation method for a diet considering user's personal activities on his/her schedule in the calculation of energy intake. This study is similar to our approach by considering user's activities. However, they use linear programming to recommend more healthy recipes among the selected recipes. Therefore, as they cannot perform in a reasonable running time, our study is published as a web application and returns menus in a short time.

Finally, Personal Mediterranean Diet Optimization by HQGA (PMDOH), which provides repair and search methods for sensitive optimization of energy, carbohydrate, protein and fat, considers 22 parameters of vitamins and minerals beside energy, carbohydrate, protein and fat in the fitness function. This expert system holds a database where there are only Mediterranean foods. The database has obtained by a process of a data preparation on USDA National Nutrient Database. After taking the personal inputs of general health condition and eating habits beside age, sex, weight, height and daily activities, PMDOH selects foods from the database by eliminating and considers returning a healthy receipt entirely, too.

CHAPTER SIX
HEAD AND NECK CANCERS & RMWC

6.1 Head and Neck Cancers Automation System

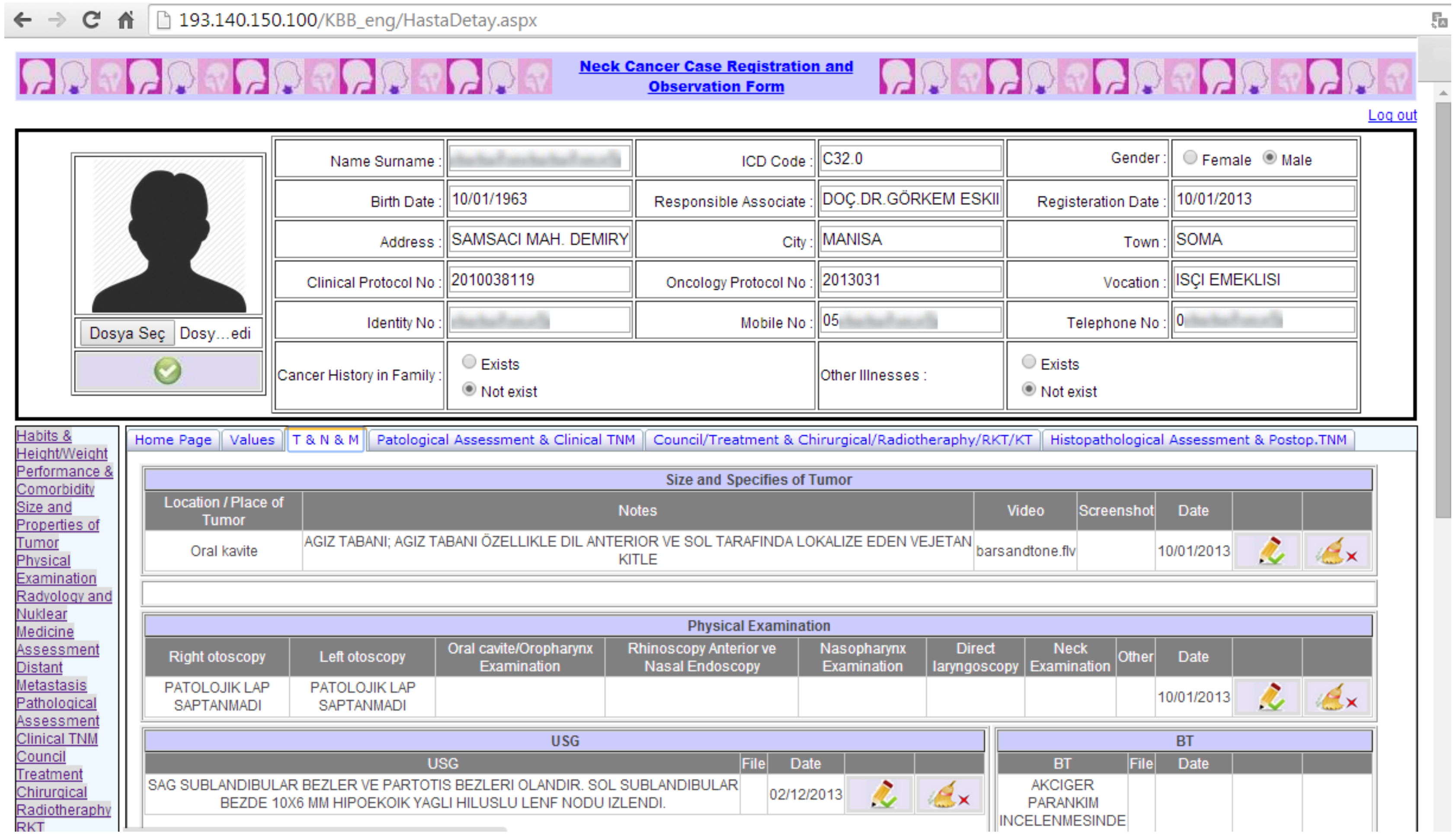
The policlinic of Celal Bayar University Head and Neck Cancers uses the head and neck cancers automation system which has been implemented by me. This automation system records patients and their following features;

Demographic and medical data of patients are detailed in the following paragraphs. Content; Name Surname, ICD Code, Gender (Female/Male), Birth Date, Responsible Associate, Registration Date, Address, City, Town, Clinical Protocol No, Oncology Protocol No, Vocation, Identity No, Mobile No, Telephone No, Cancer History in Family (Exists, Not exist) and Other Illnesses (Exists, Not exist).

Habits & Height/Weight, Performance & Comorbidity, Size and Properties of Tumor, Physical Examination, Radyology and Nuklear Medicine Assessment, Distant Metastasis, Pathological Assessment, Clinical TNM, Council, Treatment, Chirurgical, Radiotheraphy, RKT, KT, Histopathologic Examination, Postoperation TNM. (T means that the size of the original –primary- tumor and whether it has invaded nearby tissue, N means that nearby -regional- lymph nodes that are involved, M means that distant metastasis -spread of cancer from one part of the body to another-)

ECOG, Carnovsky, Charlson Index, Cumulative, Illness Score, these formula determines the person's life situation and create a guiding treatment parameters on what direction it should be yet.

This system collects these important information sets. This rich database is precipitated to analysis by data mining algorithms significantly. Figure 6.1 shows a sample interface from this automation system.


193.140.150.100/KBB_eng/HastaDetay.aspx

Neck Cancer Case Registration and Observation Form

Log out

Dosya Seç Dosy...edi

Name Surname :	[redacted]	ICD Code :	C32.0	Gender :	Female [✓ Male]
Birth Date :	10/01/1963	Responsible Associate :	DOÇ.DR.GÖRKEM ESKII	Registeration Date :	10/01/2013
Address :	SAMSACI MAH. DEMIRY	City :	MANISA	Town :	SOMA
Clinical Protocol No :	2010038119	Oncology Protocol No :	2013031	Vocation :	ISÇI EMEKLISI
Identity No :	[redacted]	Mobile No :	05[redacted]	Telephone No :	0[redacted]
Cancer History in Family :	Exists [✓ Not exist]	Other Illnesses :			Exists [✓ Not exist]

Habits & Height/Weight
Performance & Comorbidity
Size and Properties of Tumor
Physical Examination
Radyology and Nuklear Medicine Assessment
Distant Metastasis
Pathological Assessment
Clinical TNM
Council Treatment
Chirurgical Radiotheraphy
RKT

Home Page | Values | T & N & M | Patological Assessment & Clinical TNM | Council/Treatment & Chirurgical/Radiotheraphy/RKT/KT | Histopathological Assessment & Postop.TNM

Size and Specifies of Tumor

Location / Place of Tumor	Notes	Video	Screenshot	Date		
Oral kavite	AGIZ TABANI; AGIZ TABANI ÖZELLIKLE DIL ANTERIOR VE SOL TARAFINDA LOKALIZE EDEN VEJETAN KITLE	barsandtone.flv		10/01/2013		

Physical Examination

Right otoscopy	Left otoscopy	Oral cavite/Oropharynx Examination	Rhinoscopy Anterior ve Nasal Endoscopy	Nasopharynx Examination	Direct laryngoscopy	Neck Examination	Other	Date		
PATOLOJIK LAP SAPTANMADI	PATOLOJIK LAP SAPTANMADI							10/01/2013		

USG

USG	File	Date		
SAG SUBLANDIBULAR BEZLER VE PARTOTIS BEZLERI OLANDIR. SOL SUBLANDIBULAR BEZDE 10X6 MM HIPOEKOIK YAGLI HILUSLU LENF NODU IZLENDI.		02/12/2013		

BT

BT	File	Date		
AKCIGER PARANKIM INCELENMESINDE				

Figure 6.1 A sample web page from “Head and Neck Cancers Automation System”

6.2 Different Approaches of Mapping Clustering

Clustering algorithms are the base of the data mining analysis. The other data mining techniques, association rule mining and classification are fed back from clustering. If it is illustrated for classification; after a clustering analysis, a set of clusters is obtained and each data is assigned to a cluster in the set. That is to say, each data in data set attains a new feature as a target. This target feature is the cluster which the row data belongs to. As a result, classification algorithms can analysis this data set which is extended with a new feature after clustering analysis and classification analysis may need clustering analysis before itself.

If another scenario is illustrated for association rule mining; association rule mining algorithms are interested in frequencies and equivalences of each row data in data set while mining the associations. Therefore, if there is a big data set to discover hidden secrets in it, association rule mining algorithms use a kind of clustering approach by considering equivalence between association combinations. The clusters, which have the highest frequencies according to threshold for example minimum support and confidence values, are presents as result associations. As a result, association rule mining is a significant sub-technique of clustering techniques for big data.

Additionally, if there is not big data, association rule mining algorithms fails and cannot obtain any association. On such an occasion, clustering algorithms can use to analysis and mine the associations without consideration of the frequencies and equivalences, thus consequences as fuzzy associations can be obtained by explication centre points of clusters.

6.2.1 K-Means++ + Quantum Genetic Algorithm

K-Means++ algorithm is a successful clustering algorithm. Its inspirer algorithm is K-Means and K-Means++ is more consistent than K-Means by returning a same pattern at each running. It is detailed in Section 4.4.1.2. Traditional K-Means++

algorithm submits cluster numbers for each instance; however, there is no data about visualization in returned values. In this study, K-Means++ is mapped on two-dimensional map.

SOM algorithm is a clustering and mapping algorithm. That is, cluster centroids, which are trained for all instances, are mapped according to their neighbourhoods. SOM is used an artificial neural network structure at background while training. In this study, a new K-Means++ approach with a visualization property has been implemented; in other words, K-Means++, which has a purer logic than SOM, has been improved with a mapping characteristic by using quantum genetic algorithm. After K-Means++ obtains the centroids, quantum genetic algorithm is used and this successful evolutionary algorithm runs until the fittest map is obtained according to the neighbourhoods of the centroids between each other. The centroid values have a multi-dimensional structure; therefore, the distances between the centroids according to the Euclidean distance formula are the most important reference while mapping on a two dimensional pattern of these multi-dimensional centroids.

6.2.1.1 Algorithm of K-Means++ + QGA

The first phase of this mapping approach is the traditional K-Means++. This algorithm returns the centroids as multi-dimensional and 0-1 normalized instances. The second phase is that obtainment of a matrix where exits the distances between all centroids according to the Euclidean distance formula. After that, the standard deviation values between all distance values in the matrix are calculated according to the following formula;

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2} , \qquad (6.1)$$

where N is the total number of clusters, x_i is ith distance value and x is the mean value of the all distances between a current cluster and the others.

For example; K-Means++ returns 4 centroids with 5 attributes and the centroid instances like the following values;

Table 6.1 An example of centroid instances and values

Clusters \ Attributes	A1	A2	A3	A4	A5
C1	0.6	0.1	0.9	0.7	0.4
C2	0.5	0.2	0.3	0.6	0.1
C3	0.4	0.1	0.5	0.7	0.2
C4	0.7	0.3	0.6	0.8	0.1

Then, the distances between the centroid instances according to the Euclidean distance and a matrix is obtained like the following example matrix;

Table 6.2 The distances between the example centroids according to the Euclidean distance

Clusters	C1	C2	C3	C4
C1	0.0000	0.6928	0.4898	0.4898
C2	0.6928	0.0000	0.2828	0.4242
C3	0.4898	0.2828	0.0000	0.3999
C4	0.4898	0.4242	0.3999	0.0000

Quantum genetic algorithm is used an approach which is based on standard deviation values between the centroid instances. If the standard deviation of a centroid instance is the least standard deviation value in the others, this instance is located on the centre of the map. Accordingly, the centroid instances with the little standard deviation values are located near the centre of the map and the other centroid instances are located near the boundary lines of the map. Quantum genetic algorithm appoints the locations of the centroid instances by stages from the centre to the boundary of the map. These standard deviation valued are obtained and an array where the standard deviation values between the centroid instances is calculated like the following example array;

Table 6.3 The standard deviation values according to the distances of the example centroids

Clusters	The Standard Deviation Values
C1	0.3358
C2	0.3185
C3	0.2410
C4	0.2557

At this phase, one of the two ways in the algorithm is followed according to the number of centroids. If the number of the centroids is even, the four points on the centre of the map is calculated and determined without quantum genetic algorithm. Firstly, the centroids are sorted according to the standard deviation values and the four centroids with the least standard deviations are selected. Then, the first one and the fourth one are located on the centre four points of the map crossly like the points on the following example map;

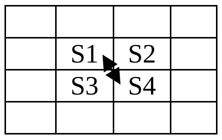


Figure 6.2 The four centroids with the least standard deviations are located on the centre of the map

In Figure 6.2, there is a map with 4x4 = 16 points. S1 is the centroid which has the least standard deviation value and S4 is the centroid which has the fourth least standard deviation value. These two centroids are located crossly because S1 must be located farther than S4 considering S2 and S3. Thus, the distance between S1 and S4 is obtained $\sqrt{2}$ unit when the distance between S1 and S2 is 1 unit, and the distance between S1 and S3 is 1 unit.

If the number of the centroids is odd, the nine points on the centre of the map is calculated and determined without quantum genetic algorithm. Firstly, the centroids are sorted according to the standard deviation values and the nine centroids with the least standard deviations are selected. Then, the first one is located on the centre point of the map as the other eight ones are located according to the distances between the standard deviation values of the first one and these eight ones. The second, third, fourth and fifth ones are located according to the protection the distance between them and the centre point as 1 unit. It means that the second centroid is located on the left-side of the centre point; the third centroid is located on the up-side of the centre point; the fourth centroid is located on the down-side of the centre point and the fifth centroid is located on the right-side of the centre point.

There is the assumption that the second one and the fifth one must be located on the furthest points; therefore, they are located on left and right sides of the centre point.

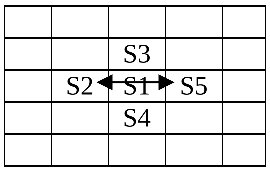


Figure 6.3 The five centroids with the least standard deviations are located on the centre of the map

In Figure 6.3, there is a map with 5x5 = 25 points. S1 is the centroid which has the least standard deviation value and it is located on the centre of the map. The other centroids, S2, S3, S4 and S5 must be located by the protection the distances between S1 and them as 1 unit. S2 is the centroid which has the second least standard deviation value as S5 is the centroid which has the fifth least standard deviation value. These two centroids are located linearly because S2 must be located farther than S5 considering S3 and S4. Thus, the distance between S2 and S5 is obtained 2 unit, when the distance between S2 and S3 is $\sqrt{2}$ unit, the distance between S2 and S4 is $\sqrt{2}$ unit, the distance between S5 and S3 is $\sqrt{2}$ unit and the distance between S5 and S4 is $\sqrt{2}$ unit. The next four centroids with the least standard deviations are located on the vertexes on the centre of the map. The assumption in this phase is that the sixth one is located on the nearest vertex of S2 and S3; the seventh one is located on the nearest vertex of S2 and S4; the eight one is located on the nearest vertex of S3 and S5 and the ninth one is located on the nearest vertex of S4 and S5. Thus, the nine centroids which have the least standard deviation values are located on the map with the consistency of each other. The initialization of the map with locations of the centre nine centroids is determined and a map is obtained like the following one;

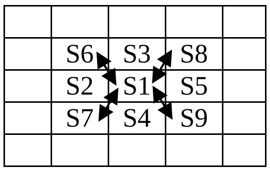


Figure 6.4 The next four centroids with the least standard deviations are located

In Figure 6.4, the distances between S1 and S6-S7-S8-S9 separately are obtained $\sqrt{2}$ unit, when the distance between S1 and S2-S3-S4-S5 is 1 unit.

In the final phase, quantum genetic algorithm is used for the empty and frame points which are located on the near centre points. After that, the other outer frame points find the centroids by quantum genetic algorithm. These loop operations are terminated after the centroids are assigned to the frame points outermost.

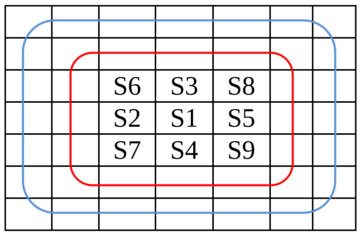


Figure 6.5 The process of quantum genetic algorithm after assignment of the centre centroids

In Figure 6.5, there is an example map with 7x7 = 49 points. Because the number of the points is odd, the centroids which have the least standard deviation values are assigned to the nine points on the centre of the map. After this introduction phase, the next least standard deviation values are assigned by quantum genetic algorithm like the red frame on the figure above. Then, it is repeated for the outer frame points like the blue frame on the figure. Until all centroids are assigned, quantum genetic algorithm is run for all frames separately.

The number of frame points differs according to the total number of points. If the number of points is odd, k*8 centroids are taken from the sorted centroids list according to the standard deviation values for all frames separately. After the introduction assignments, k initializes as 2 and it adds into 1 for each frame.

If the number of points is even, k*8 + 4 centroids are taken from the sorted centroids list according to the standard deviation values for all frames separately. After the introduction assignments, k initializes as 1 and it adds into 1 for each frame.

6.2.1.2 Experimental Results of K-Means++ + QGA

This new approach was tested with its alternative, SOM by the usage of three datasets which have different characteristics. These datasets were obtained from UCI machine learning repository (UCI Machine Learning Repository, 2011)

The first dataset, Iris has 4 attributes and 150 instances which are divided by 3 clusters. The second dataset, Glass Identification has 9 attributes and 214 instances which are divided by 7 clusters. The third dataset, Pima Indian Diabetes has 8 attributes and 768 instances which are divided by 2 clusters. All these datasets have numerical values in their attributes. However, their cluster attributes have been deleted while testing. For all data sets, 3 different numbers of clusters have been observed as 4, 9 and 16. According to the sum of square error (SSE) in Formula 6.2, it can be seen that K-Means++ + QGA have had less SSE values than SOM.

$$\mathrm{SSE} = \sum_{k=0}^{n} (\mathrm{X} - \mathrm{X}')^2 \qquad (6.2)$$

Table 6.4 SSE values of K-Means++ + QGA and SOM

	The Number of Clusters	**K-Means++ + QGA The Sum of Square Error (SSE)**	**SOM The Sum of Square Error (SSE)**
Iris	4	0.09	0.31
	9	0.46	0.63
	16	0.73	2.61
Diabetes	4	0.06	0.51
	9	0.42	0.99
	16	0.81	4.98
Glasses	4	0.07	0.32
	9	0.45	0.82
	16	0.77	3.76

6.2.2 SOM + SOM

Decision Support Systems use different data mining algorithms. Classification algorithms gives either a certain class as in Decision Trees and Artificial Neural Networks techniques or alternative classes as in Naive Bayes and Fuzzy Classification techniques for a new instance.

Given certain rules and consequences at Medical Information Systems by data mining algorithms is an un-solicited status. Because, decision support systems for doctors are used for only a support and these software systems must not pass judgement on patients while deciding of diagnosis, treatments and etc. Therefore, clustering algorithms may be more helpful than classification algorithms at cancer branches especially.

Doctors can discover the probabilities at deciding of diagnosis, treatments and etc. by visualize the consequence pattern of clustering algorithms. SOM algorithm returns a map to visualize the pattern; however it is not enough to interpret the neighbourhoods of clusters, because traditional SOM ignores the distances between centres of clusters. Therefore; a new approach for mapping the clusters, which is based on that after applying SOM with all instances, applying again SOM by giving the centre points from the first SOM operation as instances, is applied. The algorithm is given in detain in next section.

6.2.2.1 Algorithm of SOM + SOM

SOM algorithm as a clustering technique takes only cluster number as a parameter and returns a cluster map where all clusters are neighbours with each other. However, this pattern is not the expected map because, the distances between the returned centre points are not equal to each other actually and these clusters must be located on different cells of the map. If it is illustrated by the following patterns, the differences between points can be observed;

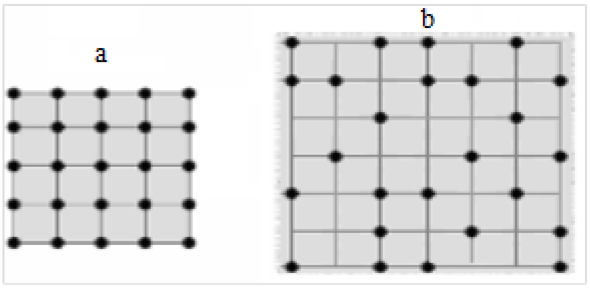


Figure 6.10 a- 5X5 = 25 clusters returned by SOM, b- 25 expected clusters with different distances between centre points.

The aim in Figure 6.10 is to point out the differences between the consequence pattern of SOM with the neighbours of all points in “a” and the expected patterns with real distances of all points in “b”.

There are two main steps of SOM + SOM algorithm to obtain a pattern like the map in “b” in Figure 6.10. The first step is that all instances are used to train the map. After training operation, the clusters are returned with x and y coordinates.

For example; SOM algorithm is run on a map with 5x5 = 25 points, all instances train the all points on the map and 25 clusters are returned with x and y coordinates from (0,0) to (3,3). Thus, a pattern like “a” map in figure above is obtained.

These 25 clusters have 25 centroid instances which have calculated multi-dimensional values related to the original instances in the data set as well as x and y coordinates. The second step of SOM + SOM is that these 25 multi-dimensional values generate a new data set where there are these 25 instances and these 25 instances are used to train a new map with 100x100 = 10000 points. It is assumed that all centroids must be located on a map with x and y coordinates from (0,0) to (99,99) to compare with other approaches on a standard 100x100 map. This number of 100 can be changed according to observations of instances and the number of clusters. As a result, these 25 instances spreads on the 100x100 map and the distances between these centroid instances are changed from 1 to different values. Thus, a pattern like “b” map in figure above is obtained.

6.2.2.2 Experimental Results of SOM + SOM

This approach was tested by the usage of Iris dataset from UCI machine learning repository (UCI Machine Learning Repository, 2011). The visualizations of SOM + SOM patterns are submitted in the next figures step by step. On the next each figure, the sizes of maps increase and the centroid points spread clearer.

In the following Figure 6.11, “a” presents returned 4x4 = 16 centroids by SOM algorithm and the distances between all centroids equals to 1 unit there. Then, these 16 centroid instances are given SOM algorithm again. The another map “b” in Figure 6.11 presents 16 centroids on 25x25 map and “c” presents these centroids on 36x36 map. The another map “a” in Figure 6.12 presents 16 centroids on 49x49 map, “b” presents these centroids on 64x64 map and “c” in Figure 6.12 presents 16 centroids on 81x81 map. These maps specify that these centroids spread on all over map with different distances between them by increasing the sizes of maps.

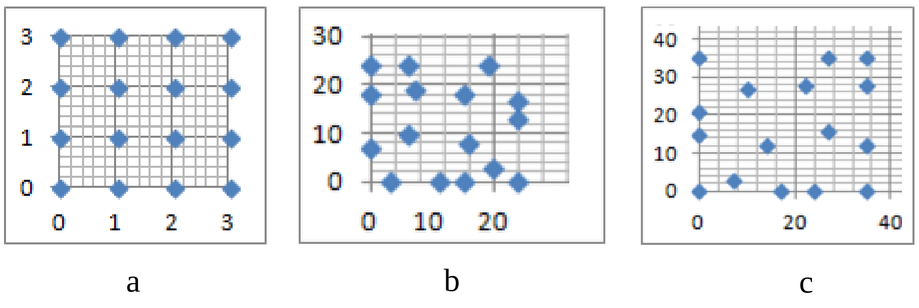


Figure 6.11 a) SOM with4x4 map b) SOM with 4x4 map + SOM with 25x25 map c) SOM with 4x4 map + SOM with 36x36 map

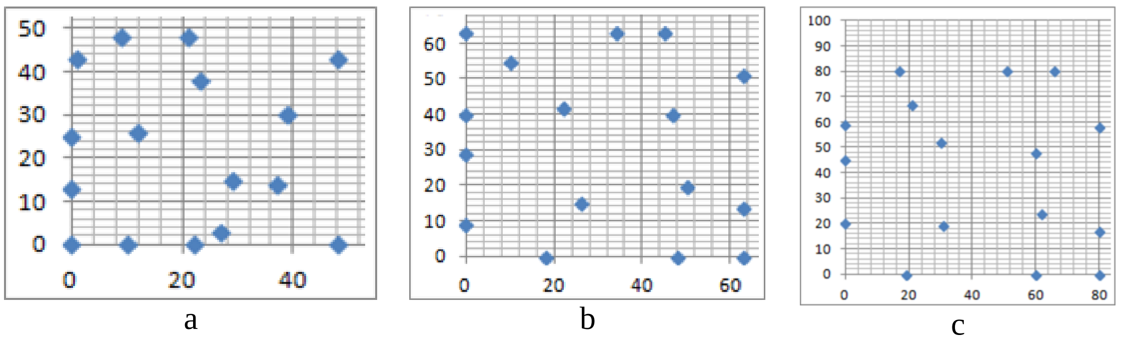


Figure 6.12 a) SOM with 4x4 map + SOM with 49x49 map b) SOM with 4x4 map + SOM with 64x64 map c) SOM with 4x4 map + SOM with 81x81 map

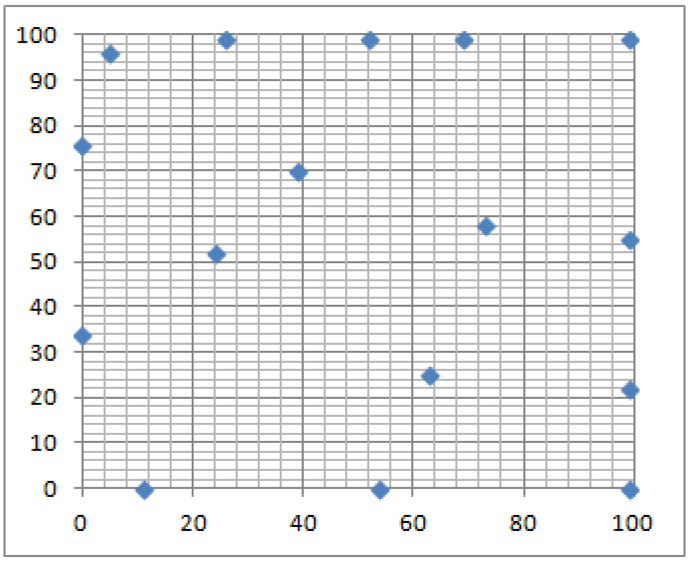


Figure 6.13 100x100 map

In Figure 6.13, all centroid instances are located clearly on 100x100 maps by SOM + SOM.

6.2.3 RMWC Regression Mapping with Weighted Clustering

In the field of cluster analysis, most clustering algorithms consider the contribution of each attribute for classification uniformly. In fact, different attributes (or different features) should be of different contribution for clustering result. In order to consider the different roles of each attribute, this chapter proposes a new approach for clustering algorithms based on weights, in which decision tree technique is used to assign the weights to each attribute. The comparison results show that novel approach improves the robustness of the traditional clustering algorithms. The experimental results with various test data sets illustrate the effectiveness of the proposed approach.

Cluster analysis is one of the main methodologies for analysing multivariate data and one of important branches of unsupervised pattern analysis. Clustering is to partition data into groups such that data within a group are more similar to one another than patterns in different clusters.

Traditional clustering algorithms assume that each attribute of the dataset plays a uniform contribution for cluster analysis. However, in real world, each attribute (or each feature) may have different effects on the clustering result. Some weight values should be assigned based on the relative importance of features, reflecting the different roles of them during the clustering process. More important features should be assigned a higher weight, while less important ones should have a lower weight, may even have a zero weight.

This section proposes a novel "weighted clustering using decision tree" approach for clustering data sets and for obtaining more consistent rule sets than traditional clustering algorithms. In this approach, decision tree technique is used to extract the contribution of each attribute for clustering result. By this approach, the dataset is firstly clustered by the classical clustering algorithm, then attributes are weighted according to the decision tree, then the dataset is clustered again by considering extracted weight values and finally rule sets are obtained using decision tree again.

Experimental results with various test data sets and comparison results show that this chapter presents a new approach of clustering via weights on the attributes, somehow making clustering move toward the more powerful unsupervised method. In the experimental studies, K-Means algorithm was used, which has widely used clustering algorithm among unsupervised learning algorithms. However, the proposed algorithm can be used other clustering algorithms and can be applied to data of any dimensions.

In order to consider the different contributions of features for clustering, several empirical studies have been done by using different techniques.

While Bao et al. proposed the usage of ReliefF algorithm to give the weights to each feature, Li et al. proposed the usage of atom clustering algorithm to determine weights and then both studies used fuzzy c-means (FCM) algorithm to cluster data. Nock and Nielsen also presented a different approach, in which weight calculations before clustering rely on the local variations of the expected complete log-likelihoods.

Most traditional clustering algorithms are limited to handling datasets that contain either numeric or categorical attributes. Han et al. proposed a model such that the weights for numerical features are calculated by Variable Precision Rough-Fuzzy Sets (VPFRS) algorithm, while the weights for categorical features are calculated by Variable Precision Rough Sets (VPRS) algorithm. In order to solve weighted clustering analysis problem, Chen used a framework which is based on decision-tree classify of little sample using a semi-supervised strategy. However, our approach is very different from this study, because it uses decision tree only to give a definition of cluster feature vector group for hybrid attributes.

In order to weight attributes before clustering, Song et al. proposed a pre-processing approach by calculating the coefficients of multiple correlations. Cheng et al. proposed a scheme to capture the local correlation structures to associate each attribute with an independent weighting vector and embed it in the subspace spanned by an adaptive combination of the dimensions. Chen presented a two-step projected clustering method for weighted clustering and evolutionary analysis of hybrid attributes data streams. Al-Razgan and Domeniconi addressed the problem of weighted clustering by using Locally Adaptive Clustering (LAC) algorithm.

To the best of our knowledge, this study is the first on weighted clustering using the levels of the decision tree and the calculations of the weights as defined in Section 6.2.3.1.

6.2.3.1 Weighted Clustering

The standard clustering operations are done by using of standard distance formula like Euclidean distance in Formula 4.7. However, standard clustering algorithms assume that all attributes are in the same priority and they ignore the priorities of attributes between each other. Weighted clustering method supplies the priorities by using different weight value for each attribute. In this study, weight values are calculated by a new approach using decision tree algorithm.

The decision tree algorithms discover the hidden priorities of attributes. In our approach, weight values are calculated from this tree by the following way; the root attribute has a weight value as 1, the attributes in the second level have weight values as 0.5 and the attributes in the other levels have weight values as the half of the previous values.

An attribute can appear in different levels. Therefore, all values in all levels for all attributes are found and these values are summed to calculate final weights. The calculation examples are given in the Section 6.3.2.2. The flow of this new approach is shown in the Figure 6.14 and its details are given below;

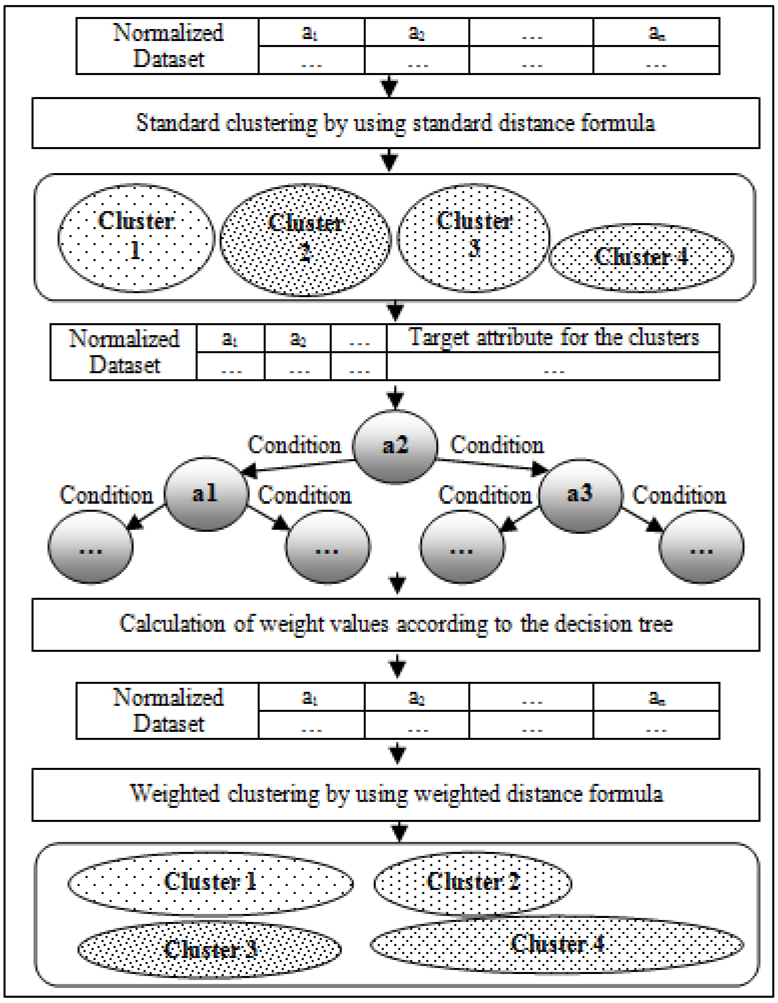

Normalized Dataset
a1
a2
...
an
Standard clustering by using standard distance formula
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Target attribute for the clusters
a2
a1
a3
Condition
Calculation of weight values according to the decision tree
Weighted clustering by using weighted distance formula

Figure 6.14 The flow of weighted clustering using decision tree.

1. All attributes in the dataset have the same priority at the beginning. Therefore, the attributes are normalized between 0 and 1.

2. The normalized dataset is clustered by using a standard clustering algorithm. For example, K-Means++ algorithm assumes that all attributes have the same priority while comparing the distances between two instances.

3. The aim of this approach is to obtain a space of consistent clusters. This consistency is supplied by using a decision tree algorithm and a weighted clustering. Because clustered instances have the cluster knowledge as a target, a new attribute is added to the dataset to put the cluster knowledge. As a result, a classification dataset is obtained for the decision tree.

4. The classification dataset is given to a decision tree algorithm. The decision tree algorithms discover the hidden priorities of attributes; the attributes from the root to the deepest node are listed as from the most important attribute to the least one by using Information Gain formula in Formula 6.3.

$$H(Ex) = -\sum_{i=0}^{n} p(xi)\log p(xi) \tag{6.2}$$

$$IG(Ex,a) = H(Ex) - H(Ex\,|a) \tag{6.3}$$

where $p(x_i)$ is the probability mass function of outcome attribute x_i, H(Ex) is the entropy of all dataset and H(Ex|a) is the entropy for a^{th} attribute.

5. According to the order of nodes in the tree, the weights are calculated and this knowledge is used in the standard clustering algorithm. For example, Weighted K-Means++ algorithm (WKM) uses Weighted Euclidean Distance (WED) in Formula 6.4, where Wk is the weight value of k^{th} attribute.

$$WED = \sqrt{\sum_{k=0}^{d} Wk\|Xk - Yk\|^2} \tag{6.4}$$

6. The normalized dataset without any target attribute in the first situation is used by the weighted clustering. Finally, a space of clusters which has more consistence rule set is obtained. The experimental studies given in Section 6.2.3.1.1 show the effectiveness of this new approach about accuracy and consistency.

When a new instance out of training dataset comes, its related cluster is found by comparison of the weighted distances between the central vector in each cluster and this new instance.

6.2.3.1.1 Experimental Studies of Weighted Clustering. The new approach to weighted clustering was tested by the usage of four datasets which have different characteristics. These datasets were obtained from UCI machine learning repository. The operation is clustering in this study; however, the experimental studies are done by the usage of classification datasets without target attributes. Thus, both the number of clusters and the correct cluster of each instance are known, and the success of this new approach can be tested correctly.

The first dataset, Iris has 4 attributes and 150 instances which are divided by 3 clusters. The second dataset, Glass Identification has 9 attributes and 214 instances which are divided by 7 clusters. The third dataset, Pima Indian Diabetes has 8 attributes and 768 instances which are divided by 2 clusters. The fourth dataset, Page Blocks has 10 attributes and 5473 instances which are divided by 5 clusters. All these datasets have numerical values in their attributes.

In experimental studies, K-Means++ algorithm is used for the clustering operations of this new approach. Also, before clustering operation, all datasets are normalized between 0 and 1 by usage of Min-Max Normalization given in formula below.

$$\beta = (\alpha - \theta) / (\mu - \theta) \tag{6.5}$$

where β is the normalized value, α is the processed value, μ is the maximum value of the processed attribute and θ is the minimum value of the processed attribute.

K-Means++ algorithm uses Euclidean Distance formula while comparing the distances between the centre points of clusters and the processed point. The original Euclidean formula assumes that all attributes have the same priority; therefore, the coefficients for all attributes are 1. In weighted clustering, these coefficients change. In experimental tests, the coefficients were calculated as Table I for Iris, Table II for Glass, Table III for Diabetes and Table IV for Page Blocks. As a result, the Euclidean distance, “E” is calculated as Formula 6.6 for Iris, Formula 6.7 for Glass, Formula 6.8 for Diabetes and Formula. 6.14 for Page Blocks. Firstly, normalized Iris dataset between 0 and 1 is given to the K-Means clustering with the standard Euclidean distance formula.

This dataset has 3 clusters in the real world; therefore, this attribute which contains the correct clusters is deleted before is being given to K-Means and the k parameter of algorithm is set as 3. After clustering operation, all instances in the dataset have a target attribute as their related clusters. This cluster knowledge is set in the dataset as a number like the 1st cluster, the 2nd cluster and the 3rd cluster. The second step is using a decision tree algorithm.

In experimental studies, C4.5 is used because this algorithm has a support for numerical data; thus, it would be used our test datasets which have numerical values in their attributes. Also, C4.5 has a pruning property. It means if there is a bough which gives the same result with another bough in the tree; the longest bough is deleted from the tree. Therefore, the most dominated attributes emerge and the number of identity attributes in the tree nodes is decreased. Furthermore, it means that weights of these attributes are decreased. Thus, the weights of the dominant attributes are increased according to the identity attributes. The new dataset for Iris with the target attribute which puts the values of clusters was given to C4.5 decision tree algorithm and the output tree given in Figure 1.2 was obtained.

The leaves in Figure 6.15 show that there are 50 instances in the 2nd cluster, 39 instances (34 + 3 + 2) and 61 instances in the 3rd cluster (52 + 9). This decision tree in this form obtains correct rule sets for 139 of 150 instances. Also, Figure 6.15

shows that there are 5 levels in the tree and the root, which indicates the most dominant attribute, becomes the 4th attribute. Therefore, the value of the 4th attribute in the first level of Figure 6.15 is 1 and the other values in the first level are 0. There is only the 3rd attribute in the second level; therefore, the value of the 3rd attribute in the second level is 0.5, while others are 0.

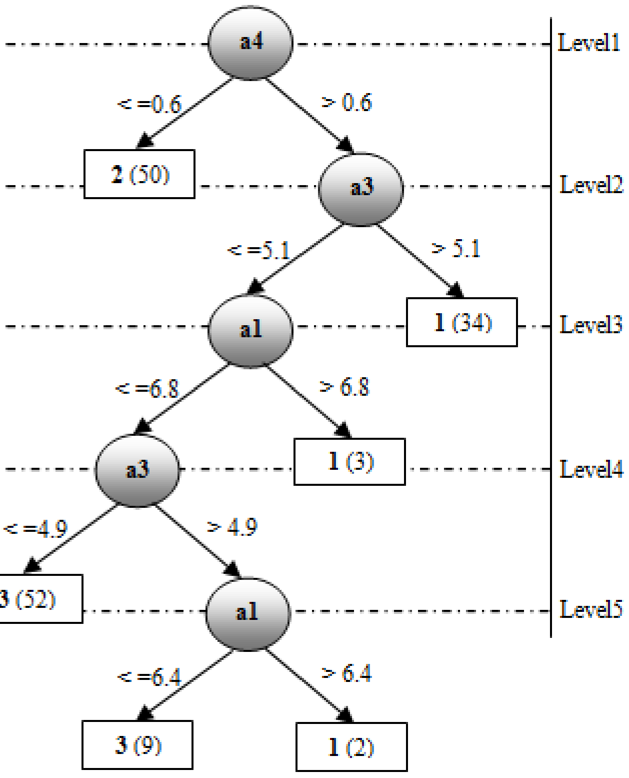


Figure 6.15 The decision tree of clustered Iris dataset by using C4.5.

The third level contains only the 1st attribute; therefore, the value of the 1st attribute in this level is 0.25 and the other values are 0. There is only the 3rd attribute in the fourth level; therefore, the value of the 3rd attribute in the fourth level is 0.125 and the other values are 0. The fifth level consists of the 1st attribute; therefore, the value of the 1st attribute in this level is 0.0625, while the other attribute values are 0. Finally, the total weights for all attributes are obtained (total weights column in Table 6.5) and then, Euclidean Distance, “E” in Formula 6.6 is used by K-Means++.

Table 6.5 Calculations of weight values for iris dataset

Attributes	**Levels in Decision Tree**					
	1	**2**	**3**	**4**	**5**	**Total weights**
a1	0	0.0	0.25	0.000	0.0625	0.3125
a2	0	0.0	0.00	0.000	0.0000	0.0000
a3	0	0.5	0.00	0.125	0.0000	0.6250
a4	1	0.0	0.00	0.000	0.0000	1.0000

$$E^2 = 0.3125 \times (\beta_1 - M_1)^2 + 0 \times (\beta_2 - M_2)^2 + 0.625 \times (\beta_3 - M_3)^2 + 1 \times (\beta_4 - M_4)^2 \quad (6.6)$$

After these steps, Iris dataset without a target attribute is clustered by weighted K-Means++. Finally, the result dataset with the target attribute, which contains new cluster values, is obtained. If this new dataset is given to C 4.5 again, the new decision tree obtains correct rule sets for 146 of 150 instances. The other three datasets are processed in the same steps. Then, Formula 6.7 according to Table 6.6, Formula 6.8 according to Table 6.7 and Formula 6.14 according to Table 6.8 are obtained. These Euclidean distance formulas, “E” are used by weighted K-Means.

Table 6.6 Calculations of weight values for glass dataset

Attributes	**Levels in Decision Tree**					
	1	**2**	**3**	**4**	**5**	**Total weights**
a1	0	0.0	0.25	0.125	0.0000	0.37500
a2	0	0.0	0.00	0.000	0.0000	0.00000
a3	1	0.0	0.00	0.000	0.0000	1.00000
a4	0	0.0	0.00	0.000	0.0000	0.00000
a5	0	0.0	0.00	0.125	0.0000	0.12500
a6	0	0.0	0.00	0.000	0.0625	0.06250
a7	0	0.0	0.00	0.000	0.0000	0.00000
a8	0	0.5	0.00	0.000	0.0000	0.50000
a9	0	0.5	0.25	0.125	0.0000	0.87500

$$\begin{gathered} E^2 = 0.375 \times (\beta_1 - M_1)^2 + 0 \times (\beta_2 - M_2)^2 + 1 \times (\beta_3 - M_3)^2 + 0 \times (\beta_4 - M_4)^2 \\ +0.125 \times (\beta_5 - M_5)^2 + 0.625 \times (\beta_6 - M_6)^2 + 0 \times (\beta_7 - M_7)^2 \\ + 0.5 \times (\beta_8 - M_8)^2 + 0.875 \times (\beta_9 - M_9)^2 \end{gathered} \quad (6.7)$$

As a result, Figure 6.16 shows that the percentages of correct and incorrect clustered instances After Weighted Clustering (AWC) are more successful than the percentages of correct and incorrect clustered instances After Simple Clustering (ASC), because the percentages of correct instances for weighted clustering are more approximate to 100% and the percentages of incorrect instances for weighted clustering are more approximate to 0%.

Table 6.7 Calculations of weight values for diabetes dataset

Attributes	**Levels in Decision Tree**						
	1	**2**	**3**	**4**	**5**	**6**	**Total weights**
a1	0	0.5	0.25	0.125	0.0000	0.00000	0.87500
a2	0	0.0	0.00	0.000	0.0625	0.00000	0.06250
a3	0	0.0	0.00	0.000	0.0625	0.00000	0.06250
a4	0	0.0	0.00	0.000	0.0000	0.03125	0.03125
a5	0	0.0	0.00	0.000	0.0000	0.00000	0.00000
a6	0	0.0	0.00	0.000	0.0000	0.00000	0.00000
a7	0	0.0	0.00	0.000	0.0000	0.00000	0.00000
a8	1	0.0	0.25	0.125	0.0625	0.03125	1.46875

$$\begin{aligned} E^2 = {} & 0.875 \times (\beta_1 - M_1)^2 + 0.0625 \times (\beta_2 - M_2)^2 + 0.0625 \times (\beta_3 - M_3)^2 \\ & + 0.03125 \times (\beta_4 - M_4)^2 + 0 \times (\beta_5 - M_5)^2 + 0 \times (\beta_6 - M_6)^2 \\ & + 0 \times (\beta_7 - M_7)^2 + 1.46875 \times (\beta_8 - M_8)^2 \end{aligned} \tag{1.10}$$

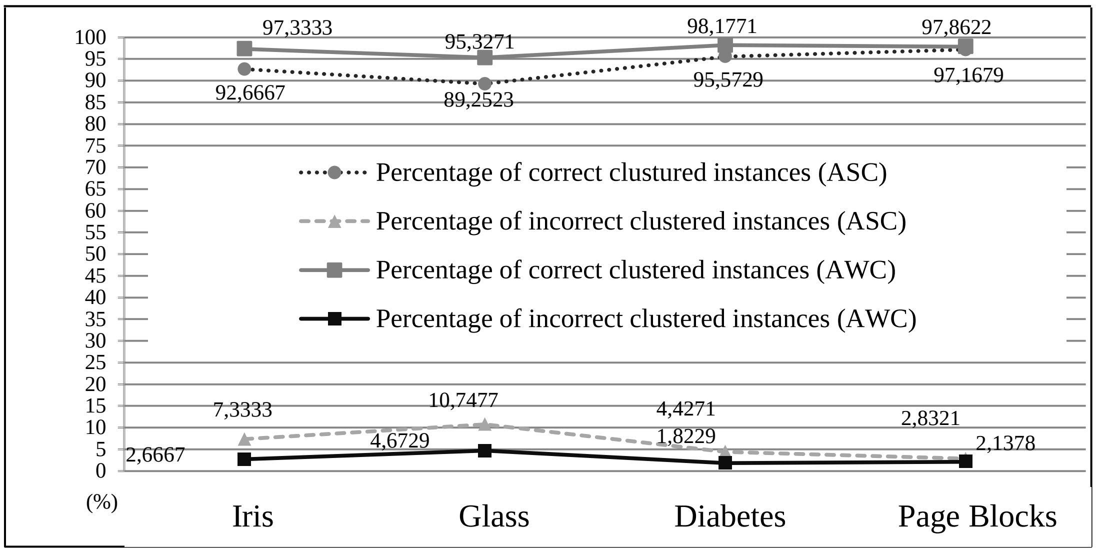


Figure 6.16 The accuracy percentages for all datasets

C4.5 algorithm for Iris and Glass datasets finds that the dominant attributes a4 for Iris and a3 for Glass are in only the root; however, this decision tree algorithm for Diabetes and Page Blocks finds that that the dominant attributes a8 for Diabetes and a5 for Page Blocks are in both the root and other levels of tree.

The weights of these attributes are calculated as the values greater than 1; thus, the differences between these attributes and the other attributes increase. On the other hand, C4.5 algorithm for all datasets finds identity attributes. Therefore, their weights become 0 and there is not any influence of these attributes over weighted clustering.

The identity attributes are a2 for Iris; a2, a4 and a7 for Glass; a5, a6 and a7 for Diabetes; a9 for Page Blocks.

Errors are generally analysed by four main methods Mean Absolute Error (MAE) (Formula 6.9), Root Mean Square Error (RMSE) (Formula 6.10), Relative Absolute Error (RAE) (Formula 6.12) and Root Relative Square Error (RRSE) (Formula 6.13). While comparing the errors, MAE and RAE give more general thoughts than others, because RMSE and RRSE enlarge the differences by using the operations of power and root.

$$\text{Mean Absolute Error} = \frac{\sum_{i=1}^{n} |\, pi - ti \,|}{n} \tag{6.9}$$

$$\text{Root Mean Square Error} = \sqrt{\frac{\sum_{i=1}^{n} (pi - ti)^2}{n}} \tag{6.10}$$

$$T = \frac{\sum_{i=1}^{n} ti}{n} \tag{6.11}$$

$$\text{Relative Absolute Error} = \frac{\sum_{i=1}^{n} |\, pi - ti \,|}{\sum_{i=1}^{n} |\, ti - T \,|} \tag{6.12}$$

$$\text{Root Relative Square Error} = \sqrt{\frac{\sum_{i=1}^{n} (pi - ti)^2}{\sum_{i=1}^{n} (ti - T)^2}} \tag{6.13}$$

where *pi* is the prediction, *ti* is the true value and *n* is the number of data.

Table 6.9 shows the error test results and comparison of simple and weighted clustering.

$$\begin{aligned} E^2 = {} & 0.00390625 \times (\beta_1 - M_1)^2 + 0.838867188 \times (\beta_2 - M_2)^2 + 0.046875 \times (\beta_3 - M_3)^2 \\ & + 0.30859375 \times (\beta_4 - M_4)^2 + 1.998046875 \times (\beta_5 - M_5)^2 \\ & + 0.436523438 \times (\beta_6 - M_6)^2 + 0.001953125 \times (\beta_7 - M_7)^2 \\ & + 0.001953125 \times (\beta_8 - M_8)^2 + 0 \times (\beta_9 - M_9)^2 \\ & + 0.03515625 \times (\beta_{10} - M_{10})^2 \end{aligned} \tag{6.14}$$

Table 6.8 Calculations of weight values for page blocks dataset

Attributes	Levels in Decision Tree											
	1	2	3	4	5	6	7	8	9	10	11	Total weights
a1	0	0.0	0.00	0.000	0.0000	0.00000	0.000000	0.0000000	0.00390625	0.000000000	0.000000000	0.003906250
a2	0	0.5	0.25	0.000	0.0625	0.00000	0.015625	0.0078125	0.00000000	0.001953125	0.000976563	0.838867188
a3	0	0.0	0.00	0.000	0.0000	0.03125	0.015625	0.0000000	0.00000000	0.000000000	0.000000000	0.046875000
a4	0	0.0	0.25	0.000	0.0000	0.03125	0.015625	0.0078125	0.00390625	0.000000000	0.000000000	0.308593750
a5	1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125	0.00390625	0.001953125	0.000000000	1.998046875
a6	0	0.0	0.25	0.125	0.0000	0.03125	0.015625	0.0078125	0.00390625	0.001953125	0.000976563	0.436523438
a7	0	0.0	0.00	0.000	0.0000	0.00000	0.000000	0.0000000	0.00000000	0.001953125	0.000000000	0.001953125
a8	0	0.0	0.00	0.000	0.0000	0.00000	0.000000	0.0000000	0.00000000	0.001953125	0.000000000	0.001953125
a9	0	0.0	0.00	0.000	0.0000	0.00000	0.000000	0.0000000	0.00000000	0.000000000	0.000000000	0.000000000
a10	0	0.0	0.00	0.000	0.0000	0.03125	0.000000	0.0000000	0.00390625	0.000000000	0.000000000	0.035156250

Table 6.9 Comparison Results of Error Tests

Tests	Datasets							
	Iris (ASC)	Iris (AWC)	Glass (ASC)	Glass (AWC)	Diabetes (ASC)	Diabetes (AWC)	Page Blocks (ASC)	Page Blocks (AWC)
Correctly Clustered Instances	139	146	191	204	734	754	5318	5356
Incorrectly Clustered Instances	11	4	23	10	34	14	155	117
Mean Absolute Error	0.0560	0.0181	0.0360	0.0140	0.0468	0.0212	0.0132	0.0102
Root Mean Squared Error	0.2111	0.1323	0.1692	0.1081	0.2006	0.1347	0.1048	0.0909
Relative Absolute Error (%)	12.8025	4.0845	17.6636	6.2650	10.5855	4.7528	4.6484	3.5249
Root Relative Squared Error (%)	45.1429	28.0739	53.1721	32.4420	42.6700	28.5376	27.8131	23.9287

6.2.3.2 Algorithm of RMWC

The aim of this algorithm is the same as the aims of the other mapping approaches and RMWC returns the centroids, which are multi-dimensional instances, on x and y coordinates of a map. However, there is a different approach while locating on the map. The distances between the centroids in K-Means++ + QGA or the training the all points on the map by SOM + SOM are not considered and important attribute/s in other words, the attributes, which have the assigned highest weight values by weighted clustering, are considered. Principal component analysis (PCA) has a same aim to visual the instances; however, this approach considers only two certain attributes while obtaining a two-dimensional map. In RMWC algorithm, all attributes have contributions for the visualization on a two-dimensional map and they are represented on the map, too.

That is, the first phase of RMWC is the running weighted clustering with K-Means++ and the weighted clustering gives two patterns. The first pattern is the centroid instances of clusters and the second pattern is the weight values of attributes.

The second phase of RMWC is the mapping these centroid instances on a two-dimensional map. X coordinate of this map becomes the most dominant attribute values of these centroid instances. The most dominant attribute means the attribute which has the highest weight value. Y coordinate of this map is calculated by weighted arithmetic mean of the values at other attributes like the following formula;

$$\mathrm{Y} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}\ , \tag{6.15}$$

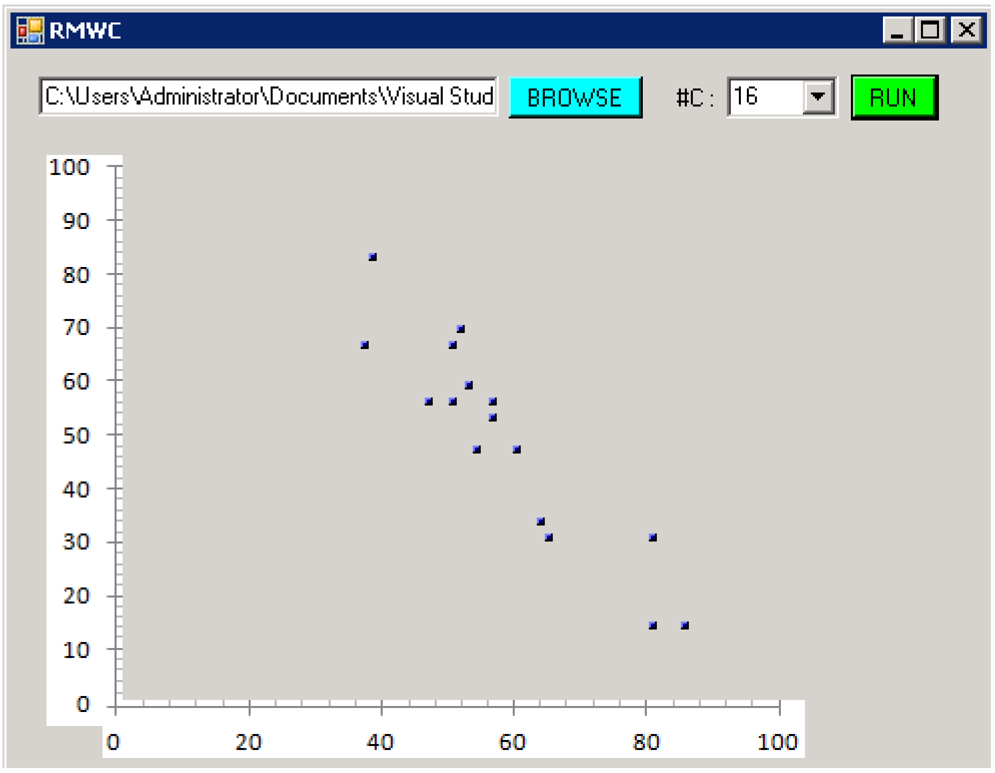
where n is the total number of attributes except the most dominant attribute, x_i is the value of ith attribute and w_i is the weight value of ith attribute.

For all centroid instances, x and y coordinate values are calculated separately. At last phase of the algorithm, these x and y values are normalized between 0 and 100 for a clear visualisation. As a result, multi-dimensional centroid instances are reduced on a two-dimensional structure.

6.2.3.3 Experimental Studies of RMWC

In experimental studies, the data set which contains the patients of head and neck cancers in Celal Bayar University. There is a sample of 50 patients from the data set to analysis and it tests for 16 clusters, because this data set has instances which have too different characteristics. Thus, each instance locates in single cluster virtually.

The features of the instances are age, gender, diagnostic place, the state of dead/survive and T, N and M values separately. (T the size of the tumour and whether it has invaded nearby tissue, N which is nearby lymph nodes that are involved, M that distant metastasis). The study is done on a 100x100 map and the following figure submits these clusters with distances between each other.


C:\Users\Administrator\Documents\Visual Stud

#C : 16

Figure 6.17 RMWC pattern on the 100x100 map of head and neck cancer data set

CHAPTER SEVEN
CONCLUSIONS

Clustering algorithms and evolutional algorithms are open for improvement. Moreover, medical decision support systems involve high accuracy rate and performance. This thesis points out that improved new approaches of traditional data mining algorithms increases success at deciding at diagnosis and treatments for data mining and knowledge discovery in medical information systems.

The first part of Chapter 4 introduces a new clustering algorithm SOM++. The significant difference between SOM++ algorithm and the standard SOM is that SOM++ does not start to initialize the weight values of neurons with random numbers. SOM++ uses the initializing centre points of clusters method in K-Means++. Eventually, each neurons represent a cluster and thus, SOM++ takes advantage of K-Means++. Separately, these significant initial values are not located in the neurons on the map and SOM++ has a special locating algorithm namely the sequential assignment.

Another difference between SOM++ algorithm and the standard SOM is that SOM++ initializes the starting number of iteration. Because the standard SOM starts with the random values in neurons, the number of iteration is declared as 0. However, because SOM++ starts with significant values in neurons, the number of iteration is declared as the number of total data. This initializing increases stability and decreases error rates.

In other words, SOM++ algorithm has many advantages over conventional SOM based methods. The most remarkable advantage of SOM++ is in saving training time for clustering large and complicated data sets by using K-Means++ algorithm in the weight initialization procedure of SOM. Furthermore, the rate of unstable data points decreases and internal error decreases.

In the second part of Chapter 4, parallel implementation of SOM has been devised by using a special division method to attempt for improving the performance of SOM in large-scale domains. It introduces a new clustering approach PSOMDM. The significant difference between PSOMDM algorithm, the standard SOM and parallel SOM is that PSOMDM divides the map area constantly and lower number data are clustered on different neurons by parallel method.

PSOMDM algorithm has many advantages over conventional SOM based methods. The most remarkable advantage of PSOMDM is in saving training time for clustering large and complicated data sets by using division method. Furthermore, the rate of unstable data points decreases and internal error decreases. For future work, the proposed algorithm, PSOMDM, can be used for the computer security, the healthcare, ecological modelling, the financial sector and another area which needs clustering its large data on a map successfully at accuracy, consistency and speed.

In Chapter 5, we have proposed a method to recommend a daily menu compatible with Mediterranean diet considering user's age, sex, weight, height, general health condition, eating habits, and daily activities. It was taken care of providing healthy menus considering the amount of 26 features' intake. Nutrition values have been gathered from the USDA National Nutrient Database and the Dietary Reference Intakes Tables of Health Canada has been used to determine daily intake values.

In test phases, it has observed that HQGA is the fittest algorithm for the DOP. Especially, the local search methods supplies successful results at 100% of the energy, carbohydrate, protein and fat as vital needs.

On the other hand, there are still many points where we should improve our method, for example, running time of the recommended algorithm can be improved and variation of food in our database can be increased. Also, weekly or monthly optimized menus can be generated by considering the recorded daily menus which have been followed by user.

In Chapter 6, All comparisons for the results of C4.5 algorithm according to the versions of ASC and AWC shows that the clusters after weighted clustering have less error scores and less incorrectly clustered instances. As a last result, if the weighted clustering is done after the second C4.5, it is observed that the error scores get higher. Because the number of the identity attributes increases and the little influence of these attributes is ignored incorrectly.

Traditional clustering algorithms suppose that all attributes have the same priority. This study proposes a new approach for clustering algorithms, using decision tree technique to assign a weight to each attribute.

This new approach in the design of unsupervised learning algorithms has allowed us to obtain dramatic improvements of clustering performances. The percentage of correct instances for weighted clustering using decision tree is approximately 97.2% on average, while it is around 93.6% for standard clustering algorithm.

The experimental studies show that if the aim of clustering is to obtain the most consistent rule set for a dataset at the end of clustering, the attributes must have different priorities for the clustering phase. These priorities are supplied by using weighted Euclidean distance formula in experimental studies. However, for this new approach, another weighted distance formula can be used. The single aim is to supply different weights for all attributes.

Other combinations of clustering and decision tree algorithms can be tested and compared in the next studies. Finally, the most successful pair will be tried to obtain and specially, for expert systems, a new approach will be produced.

REFERENCES

Aguado, D., Montoya, T., Borras, L., Seco, A., & Ferrer, J. (2008). Using SOM and PCA for analysing and interpreting data from a P-removal SBR. *Engineering Applications of Artificial Intelligence, 21 (6),* 919–930.

Akan, H. (1999). Tıp bilişimi. *Ankara Üniversitesi Tıp Eğitimi ve Bilişim Bülteni, 1 (1),* 10-20.

Akgöbek, Ö., & Çakır, F. (2009). Veri madenciliğinde bir uzman sistem tasarımı. *XI. Akademik Bilişim Konferansı Bildirileri, Harran Üniversitesi, Şanlıurfa.*

Alpaydın, E. (2000). Zeki veri madenciliği ham veriden altın bilgiye ulaşma yöntemleri. *Bilişim 2000 Eğitim Semineri*. Retrieved December 28, 2009 from http://www.cmpe.boun.edu.tr/~ethem/files/papers/veri-maden_2k-notlar.doc

Al-Razgan, M., & Domeniconi, C. (2006). Weighted clustering ensembles. *In: SIAM International Conference on Data Mining*, 258-269.

Arthur, D., & Vassilvitskii, S. (2007). K-means++ the advantages of careful seeding. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA*), 1027-1035.

Attik, M., Bougrain, L., & Alexandre, F. (2005). Self-organizing map initialization. *Artificial Neural Networks: Biological Inspirations – 15th ICANN 2005, Lecture Notes in Computer Science, 3696,* 357-362.

Ay, F. (2009). Uluslararası elektronik hasta kayıt sistemleri, hemşirelik uygulamaları ve bilgisayar ilişkisi, *Gülhane Tıp Dergisi, 51,* 131-136.

Bação, F., Lobo, V., & Painho, M. (2005). Self-organizing maps as substitutes for K-means clustering. *ICCS 2005 Lecture Notes in Computer Science, 3516,* 476-483.

Bao, Z., Han, B., & Wu, S. (2006). A general weighted fuzzy clustering algorithm. *ICIAR 2006, LNCS 4142,* 102 – 109.

Baron-Menguy, C., Bocquet, A., Guihot, A. L., Chappard, D., Amiot, M. J., Andriantsitohaina, R., et al. (2007). Effects of red wine polyphenols on postischemic neovascularization model in rats: low doses are proangiogenic, high doses anti-angiogenic. *The Faseb Journal, 21 (13)*, 3511–21.

Baykal, A. (2006). Veri madenciliği uygulama alanları. *Dicle Üniversitesi Ziya Gökalp Eğitim Fakültesi Dergisi, 7,* 95-107. Retrieved December 28, 2009 from http://www.dicle.edu.tr/suryayin/zgegitimder/tam_metinler/7pdf/07_09_Baykal.pdf

Baykal, N. (2003). Veri tabanı ve veri madenciliği. *Tıp Bilişimi Güz Okulu, 6-10 Ekim*, http://www.turkmia.org/eski/file/231verimadenciligi_baykal.ppt

Bedoya, D., Novotny, V., & Manolakos, E. S. (2009). Instream and offstream environmental conditions and stream biotic integrity importance of scale and site similarities for learning and prediction. *Ecological Modelling, 220 (19),* 2393-2406.

Benabbas, F., Khadir, M. T., Fay, D., & Boughrira, A. (2008). Kohonen map combined to the K-means algorithm for the identification of day types of Algerian electricity load. *IEEE Proceedings 7th Computer Information Systems and Industrial Management Applications*, *Ostrava, The Czech Republic*. 78-83.

Bock, R. K. (2007). *MAGIC Gama Telescope Dataset.* Retrieved June 10, 2014, from http://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope.

Castelló, A., Pollán, M., Buijsse, B., Ruiz, A., Casas, A. M., & Baena-Cañada, J. M. (2014). Spanish Mediterranean diet and other dietary patterns and breast cancer risk: case-control EpiGEICAM study. *British Journal of Cancer, 111*, 1454-1462.

Chen, H.Y., Hou, T.W., & Chuang, C.H. (2010). TBPS research group. *Expert Systems with Applications 37*, 598–601

Chen, X. (2008). Weighted clustering and evolutionary analysis of hybrid attributes data streams. *Journal of Computers, 3 (12)*, 60-67.

Chen, X. (2010). A semi-supervised weighted clustering framework facing to hybrid attributes data streams. *IEEE Proceedings of the 8th World Congress on Intelligent Control and Automation*, 5988-5993.

Cheng, H., Hua, K.A. & Vu, K. (2008). Constrained locally weighted clustering. *The Proceedings of the Very Large Database Endowment, 1 (1),* 90-101.

Chi, S.-C. & Yang, C.-C. (2008). A two-stage clustering method combining ant colony SOM and K-means. *Journal of Information Science and Engineering, 24,* 1445-1460.

Chi, S.-C., & Yang, C. C. (2006). Integration of ant colony SOM and K-means for clustering analysis. *Proceedings of Knowledge Based Intelligent Information and Information System, Lecture Notes in Computer Science, 4251,* 1-8.

Chiu, C.-Y., Chen, Y.-F., Kuo, I.-T., & Ku, H.-C. (2008). An intelligent market segmentation system using K-means and particle swarm optimization. *Expert Systems with Applications, 36 (3),* 4558-4565.

Chiu, C.-Y., Kuo, I.-T., & Chen, P.-C. (2009). A market segmentation system for consumer electronics industry using particle swarm optimization and honey bee mating optimization. global perspective for competitive enterprise. *Economy and Ecology, 12* (1).

Corrêa, R.F., & Ludermir, T.B. (2006). A hybrid SOM-based document organization system. *IEEE Proc. 9th Brazilian Symposium on Neural Networks (SBRN'06)*, 90-95.

Covas, M. I. (2007). Olive oil and the cardiovascular system. *Pharmacology. Researches, 55 (3)* ,175–186.

Dagli, C.H., Bryden, K.M., Corns, S.M., Gen, M., Tumer, K., and Süer, G. (2009). Parasom: An efficient self-organizing map for parallel multidimensional input processing and clustering. *Intelligent Engineering Systems through Artificial Neural Networks*, 7-20.

Deng, Q, & Mei, G. (2010). Combining self-organizing map and K-means clustering for detecting fraudulent financial statements. Proceeding. *IEEE International Conference on Granular Computing*, 126-131.

Doğan, Ş., Türkoğlu, İ. (2008). Hipertiroidi teşhisi için biyokimya parametrelerinden birliktelik kuralları çıkarımı. *16. IEEE Sinyal Isleme, Iletisim ve Uygulamalari Kurultayı 5(2)*, 162-169.

Doğan, Y., Birant, D., Kut, A. (2011). PSOMDM: Faster Parallel Self Organizing Map by Using Division Method. *2nd International Symposium on Computing in Science and Engineering*, 250-255.

Doğan, Y., Birant, D., Kut, A. (2011). A New Approach for Weighted Clustering Using Decision Tree. *International Symposium on Innovations in Intelligent Systems and Applications*, 54-58.

Doğan, Y., Birant, D., Kut, A. (2013). SOM++: Integration of Self-Organizing Map and K-Means++ Algorithms. *Machine Learning and Data Mining in Pattern Recognition 7988*, 246-259.

Dietary reference intakes tables (2010). Retrieved December 04, 2014, from http://www.hc-sc.gc.ca/fn-an/alt_formats/hpfb-dgpsa/pdf/nutrition/dri_tables-eng.pdf.

Eiben. A. E, Smit. S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation, 1(1)*, 19–31.

Estimated New Cancer Cases and Deaths by Sex, US, (2009). Retrieved Aug 07, 2014, from http://www.cancer.org/acs/groups/content/@research/documents/document/acspc-041780.pdf.

Estruch, R., Ros, E., Salas-Salvadó, J., Covas, M. I., Corella, D., Arós, F., et al. (2013). Primary prevention of cardiovascular disease with a Mediterranean diet. *N England Journal of Medicine, 368*, 1279-1290.

Faro, A., Giordanoa, D., & Maioranaa, F. (2011). Mining massive datasets by an unsupervised parallel clustering on a GRID. *Novel Algorithms and Case Study. Future Generation Computer Systems, 27 (6)*, 711-724.

Filomeno, M., Bosetti, C., Garavello, W., Levi, F., Galeone, C., Negri, E., et al. (2014). The role of a Mediterranean diet on the risk of oral and pharyngeal cancer. *British Journal of Cancer, 111*, 981-986.

Frankenfield, D. C., Muth E. R., Rowe W. A. (1998). The Harris-Benedict studies of human basal metabolism: history and limitations. *Journal of the American Dietetic Association, 98 (4),* 439–445.

Garcia, C., Prieto, M., & Pascual, M. (2006). A speculative parallel algorithm for self-organizing maps. *Parallel Computing, 33*, 615-622.

Garey, M., R., Johnson, D., S. (1990). Computers and intractability: a guide to the theory of NP-completeness. *White Freeman & Consulting, New York, NY, USA*, 247-248.

Gebhardt, S. E., Thomas, R. G. (2002). Nutritive value of foods. *U.S. Department of Agriculture, Agricultural Research Service, Home and Garden Bulletin 72*, 1-104.

Gorgonio, F. L., & Costa, J. A. F. (2008). Combining parallel self-organizing maps and K-means to cluster distributed data. *The 11th IEEE International Conference on Computational Science and Engineering – Workshops*, 53-58.

Gorgonio, F. L., & Costa, J. A. F. (2008). Parallel self-organizing maps with application in clustering distributed data. *International Joint Conference on Neural Networks*, 3276-3283.

Gorgonio, F. L., & Costa, J. A. F. (2010). PartSOM: A framework for distributed data clustering using SOM and K-means. *Self-Organizing Maps*, 43-64.

Güven, A., Bozkurt, Ö., Kalıpsız O. (2007). Veri madenciliğinin geleceği. *IX. Akademik Bilişim Konferansı Bildirileri, Dumlupınar Üniversitesi, Kütahya*, 150-158.

Hammond, P. (2013). *The Mediterranean diet cookbook*. 1-20.

Han, B., Gao, X., & Ji, H. (2006). A novel feature weighted clustering algorithm based on rough sets for shot boundary detection. *Fuzzy System and Knowledge Discovery*, 471-480.

Han, K., H., Kim, J., H. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. *IEEE Evolutionary Computation Proceedings of the 2000 Congress (2)*, 1354–1360.

Hey, T. (1999). Quantum computing: an introduction. *IEEE Computing & Control Engineering Journal, 10 (3)* 105–112.

Hu, X., Nenov, V., Bergsneider, M., Martin, N. (2006). A Data mining framework of noninvasive intracranial pressure assessment. *Biomedical Signal Processing and Control 1*, 64–77.

Institute of Medicine, Food and Nutrition Board. (1997). *Dietary reference intakes: calcium, phosphorus, magnesium, vitamin d, and fluoride*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files /Infographics/2014/DRIs.pdf.

Institute of Medicine, Food and Nutrition Board. (1998). *Dietary reference intakes for thiamin, riboflavin, niacin, vitamin b6, folate, vitamin b12, pantothenic acid, biotin, and choline*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files/Infographics/2014/DRIs.pdf.

Institute of Medicine, Food and Nutrition Board. (2000). *Dietary reference intakes for vitamin c, vitamin e, selenium, and carotenoids*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files/Infographics /2014/DRIs.pdf.

Institute of Medicine, Food and Nutrition Board. (2001). *Dietary reference intakes for vitamin a, vitamin k, arsenic, boron, chromium, copper, iodine, iron, manganese, molybdenum, nickel, silicon, vanadium, and zinc*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files/ Infographics/2014/DRIs.pdf.

Institute of Medicine, Food and Nutrition Board. (2002). *Dietary reference intakes for energy, carbohydrate, fiber, fat, fatty acids, cholesterol, protein, and amino acids (macronutrients)*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files/Infographics/2014/DRIs.pdf.

Institute of Medicine, Food and Nutrition Board. (2004). *Dietary reference intakes for water, potassium, sodium, chloride, and sulfate*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files/Infographics 2014/DRIs.pdf.

Institute of Medicine, Food and Nutrition Board. (2011). Dietary reference intakes for vitamin d and calcium. *National Academies Press, Washington, DC, USA*. Retrieved May 13, 2015, from http://www.iom.edu/about-iom/leadership-staff/boards/~/media/Files/Infographics/2014/DRIs.pdf.

Jebari, K. & Madiafi, M. (2013). Selection methods for genetic algorithms. *International Journal of Emerging Sciences, 3 (4)*, 333-344.

Jonsdottir, T., Hvannberg, E. T., Sigurdsson, H., Sigurdsson, S. (2008). The feasibility of constructing a predictive outcome model for breast cancer using the tools of data mining. *Expert Systems with Applications 34*, 108–118

Kahramanlı, H. & Allahverdi, N. (2008). Kalp hastalıkları veri tabanı için sınıflandırma kurallarının bulunması. *IEEE 16th Signal Processing, Communication and Applications Conference*, 1-4.

Kastorini, C. M., Milionis, H., Esposito, K., Giugliano, D., Goudevenos, J. & Panagiotakos, D. (2011). The effect of mediterranean diet on metabolic syndrome and its components. *Journal of the American College of Cardiology, 57 (11),* 1299–1313.

Kaya, E., Bulun, M., & Arslan, A. (2003). Tıpta veri ambarları oluşturma ve veri madenciliği uygulamaları. *Akademik Bilişim Konferansı Bildirileri*, 150-158.

Kaya, H., Köymen, K. (2008). Veri madenciliği kavramı ve uygulama alanları, *Doğu Anadolu Bölgesi Araştırmaları*, *6(2)*, 159-164.

Kaya, Y., Uyar, M., Tekin, R. (2011). A novel crossover operator for genetic algorithms: ring crossover. *Neural and Evolutionary Computing*, 1-4.

Keys, A., Mienotti, A., Karvonen, M. J., & et al. (1986). The diet and 15-year death rate in the seven countries study. *The American Journal of Epidemiol, 124 (6),* 903–15.

Khedairia, S., & Khadir, M. T. (2008). Self-organizing map and K-means for meteorological day type identification for the region of Annaba, Algeria. *IEEE Proc. 7th Computer Information Systems and Industrial Management Applications,* 91-96.

Kohonen, T. (1990). The Self-organizing map. *Proceeding. of the IEEE,* 78 (9), 1464–1479.

Koyuncugil, A. S., & Özgülbaş, N. (2009). Veri madenciliği: tıp ve sağlık hizmetlerinde kullanımı ve uygulamaları. *Bilişim Teknolojileri Dergisi, 2 (2),* 21-32.

Köktürk, F., Ankaralı, H. & Sümbüloğlu, V. (2009). Veri madenciliği yöntemlerine genel bakış. *Turkiye Klinikleri Journal of Biostatistics, 1 (1),* 20-5.

Küçüksille, E. U., Taşdelen & K., Aydoğan, T. (2006). Veri madenciliği ve uygulama programları. *Akademik Bilişim Konferansı Bildirileri*, 150-158.

Leão, A. C., Neto, A. D. D., & Sousa, M. B. C. (2009). New developmental stages for common marmosets (Callithrix jacchus) using mass and age variables obtained by K-means algorithm and self-organizing maps (SOM). *Computers in Biology and Medicine, 39 (10),* 853-859.

Li, J., Gao, X., & Jiao, L. (2005). A novel typical-sample-weighted clustering algorithm for large data sets. *Computational Intelligence and Security (CIS)*, 696–703.

Liping, Z., & Wensheng, G. (2009). Self-organizing maps neural networks on parallel cluster. *Information Science and Engineering (ICISE)*, 384–388.

Lv, Y. (2009). Multi-objective nutritional diet optimization based on quantum genetic algorithm. *International Conference on Natural Computation 2009*, 336-340.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *5th Berkeley Symposium on Mathematical Statistic and Probability, 1*, 281-297.

Malossini A., Blanzieri, E., Calarco, T. (2008). Quantum genetic optimization. *IEEE Transactions On Evolutionary Computation, 12 (2)*, 231-241.

Martínez-González, M. A., de la Fuente-Arrillaga, C., Nunez-Cordoba, J. M., Basterra-Gortari, F. J., Beunza, J. J., Vazquez Z., et al. (2008). Adherence to Mediterranean diet and risk of developing diabetes: prospective cohort study. *British Medical Journal, 336 (7657)*, 1348–51.

Mediterranean-diet-food-pyramid (2009). Retrieved May 01, 2015, from http://www.ambrosialgranola.com/mediterranean-diet-food-pyramid.html

Mifflin, M. D., St Jeor, S. T., Hill, L. A., Scott, B. J., Daugherty, S. A. & Koh, Y. O. (1990). A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition, 51 (2)*, 241–7.

Mino, Y., & Kobayashi, I. (2009). Recipe recommendation for a diet considering a user's schedule and the balance of nourishment. *Intelligent Computing and Intelligent Systems (ICIS 2009), IEEE International Conference, 3*, 383–387.

Musoğlu, E. (2000). *İki binli yıllar Türkiye'sinde sağlıkta bilgi stratejileri*, Retrieved Jun 01, 2000, from http://www.turkmia.org/tr/dokuemanlar/category/4-raporlar?download=47:ikibinli -yillar-turkiyesinde-saglikta-bilgi-stratejileri.

National Cancer Institute (2015). *Defining Cancer*. Retrieved May 09, 2015, from http://www.cancer.gov/about-cancer/what-is-cancer.

Nock, R., & Nielsen, F. (2006). On weighting clustering. *IEEE Transactions on Pattern Analysis and. Machine Intelligence, 28 (8)*, 1223-1235.

ODTU Enformatik Enstitüsü Sağlık Bilişimi Anabilim Dalı, Tıp Bilişimi Yüksek Lisans ve Doktora Programı Tanıtımı (n.d.), Retrieved May 01, 2015, from http://avicenna.ii.metu.edu.tr/Tarihce/EETarihce2.htm

Özekes, S. (2003), Veri madenciliği modelleri ve uygulama alanları. *İstanbul Ticaret Üniversitesi Dergisi, 2 (3)*, 1-23.

Pei, Z., & Liu, Z. (2009). Nutritional diet decision using multi-objective difference evolutionary algorithm. *International Conference on Computational Intelligence and Natural Computing, 1 (1)*, 77-80.

Phillips-Wren, G., Sharkey, P., Dy, S. M. (2008). Mining lung cancer patient data to assess healthcare resource utilization. *Expert Systems with Applications 35*, 1611–1619.

Poelmans, J., Elzinga, P., Viaene, S., Van Hulle, M. M. & Dedene, G. (2009). How emergent self organizing maps can help counter domestic violence. *IEEE Proceeding of 2009 WRI World Congress on Computer Science and Information Engineering (CSIE), 4*, 126-136.

Rabinovich., Y., Wigderson, A. (1999). Techniques for bounding the convergence rate of genetic algorithms. *Random Structures Algorithms, 14 (2)*, 111-138.

Reedy, J., Krebs-Smith, S. M., Miller, P. E., Liese, A. D., Kahle, L. L., Park, Y., Subar, A. F. (2014). Higher diet quality is associated with decreased risk of all-cause, cardiovascular disease, and cancer mortality among older adults. *Journal of Nutrition, 144,* 881-889.

Roswall, N., Angquist, L., Ahluwalia, T. S., Romaguera, D., Larsen, S. C., Østergaard, J. N., et al. (2014). Association between Mediterranean and Nordic diet scores and changes in weight and waist circumference: influence of FTO and TCF7L2 loci. *The American Journal of Clinical Nutrition, 100 (4),* 1188-1197.

Roussinov, D. G., & Chen, H. (1998). A scalable self-organizing map algorithm for textual classification: a neural network approach to thesaurus generation. *Communication and Cognition in Artificial Intelligence Journal, 15 (1-2),* 81-111.

Sagheer, A. El., Tsuruta, N., Maeda, S., Taniguchi, R., & Arita, D. (2006). Fast competition approach using self organizing map for lip-reading applications. *IEEE Proceeding of. International Joint Conference on Neural Network(IJCNN),* 3775-3782.

Saka, O. (2003). *Akdeniz Üniversitesi tıp bilişimi güz okulu*. Retrieved May 01, 2015, from http://liste.linux.org.tr/arsivler_2002-2004/lkd-duyuru/2003/Aug/0001.html

Schwingshackl, L. & Hoffmann, G. (2014). Adherence to Mediterranean diet and risk of cancer: a systematic review and meta-analysis of observational studies. *International Journal of Cancer, 135,* 1884-1897.

Semaoune, P., Sebilo, M., Derenne, S., Anquetil, C., Vaury, V., Ruiz, L., Morvan, T. & Templier, J. (2011). Dynamics of soil organic matter and mineral nitrogen in soil: investigation into a complex relationship. *European Geosciences Union General Assembly, 13,* 1-8.

Silva, A., Cortez, P., Santos, M. F., Gomes, L., Neves, J. (2008). Rating organ failure via adverse events using data mining in the intensive care unit. *Artificial Intelligence in Medicine 43*, 179-193.

Sivanandan, S., N., Deepa S., N. (2008). Introduction to genetic algorithm. *Springer-Verlag Berlin Heidelberg*, 1-13.

Song, Y-C., Meng, H-D., O'Grady, M. J. & O'Hare, G. M. P. (2007). Applications of attributes weighting in data mining. *IEEE Proceeding of Systems, Man & Cybernetics-United Kingdom & Republic of Ireland, 6th Conference on Cybernetic Systems*, 41-45.

Souza, J. R., Ludermir, T. B., & Almeida, L. M. (2009). A two stage clustering method combining self-organizing maps and ant K-means. *International Conference on Artificial Neural Networks - Lecture Notes in Computer Science, 5768*, 485-495.

Sqldatamining (2015). Retrieved May 01, 2015, from http://www.sqldatamining .com/data-warehousing/steps-of-the-knowledge-discovery-in-databases-process

Su, M.-C., Liu, T.-K., & Chang, H. T. (2002). Improving the self-organizing feature map algorithm using an efficient initialization scheme. *Tamkang Journal of Science and Engineering 5 (1),* 35–48.

Takatsuka, M. & Bui M. (2010). Parallel batch training of the self-organizing map using OpenCL, *Lecture Notes in Computer Science, 6444*, 470-476.

Tangney, C.C., Li, H., Wang, Y., Barnes, L., Schneider, J. A., Bennett, D. A., Morris, M. C. (2014). Relation of DASH- and Mediterranean-like dietary patterns to cognitive decline in older persons. *Neurology, 83*, 1410-1416.

The-mediterranean-diet (2015). Retrieved May 01, 2015, from http:// 193.140.150.100/the-mediterranean-diet/Login.aspx

Tjhai, G. C., Furnell, S. M., Papadaki, M, & Clarke, N. L. (2010). A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm. *Computers & Security 29 (6)*, 712-723.

Toussi, M., Lamy1 J. B., Toumelin, P. L., Venot, A. (2009). BMC Medical Informatics and Decision Making, 9:28.

Trichopoulou, A. (2001). Mediterranean diet: the past and the present. *Nutrition Metabolism and Cardiovascular Diseases,11*, 1-4.

Trichopoulou, A., Lagiou, P. (1997). Healthy traditional Mediterranean diet: an expression of culture, history, and lifestyle. *Nutrition Reviews, 55*, 383-389.

Trifiro, G., Pariente, A., Coloma, P.M., Kors, J. A., Giovanni, P., Ghada, M., S., et. all, (2009). Data mining on electronic health record databases for signal detection in pharmacovigilance: which events to monitor?, *Pharmacoepidemiology and Drug Safety 2009,* 1176-1184.

Turati, F., Trichopoulos, D., Polesel, J., Bravi, F., Rossi, M., Talamini, R., et al. (2014). Mediterranean diet and hepatocellular carcinoma. *Journal of Hepatol*, *60*, 606-611.

Türe, M., Tokatlı, F., Omurlu İ. K. (2009). The comparisons of prognostic indexes using data mining techniques and Cox regression analysis in the breast cancer data. *Expert Systems with Applications 36*, 8247–8254

UCI Machine Learning Repository (2011). Retrieved May 01, 2015, from http://archive.ics.uci.edu/ml, 2011.

United States Department of Agriculture Nutrient database for standard reference (2014). Retrieved May 01, 2015, from http://www.ars.usda.gov /Services/docs.htm?docid=8964.

Vepa, J. (2009). Classification of heart murmurs using cepstral features and support vector machines, *31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS),* 2539-2542.

Wang, M. H., Lee, C. S., Hsieh, K. L., Hsu, C. Y., & Chang, C. C. (2009). Intelligent ontological multi-agent for healthy diet planning. *Fuzzy Systems, FUZZ-IEEE, Korea*, 751-756.

Watts, M. J., & Worner, S. P. (2009). Estimating the risk of insect species invasion: Kohonen self-organising maps versus K-means clustering. *Ecological Modelling, 220 (6),* 821–829.

Wengreen, H., Munger, R. G., Cutler, A., Quach, A., Bowles, A., Corcoran, C., et al. (2013). Prospective study of dietary approaches to stop hypertension- and Mediterranean-style dietary patterns and age-related cognitive change: the cache county study on memory, health and aging. *The American Journal of Clinical Nutrition, 98,* 1263-1271.

Whalen, K. A., McCullough, M., Flanders, W. D., Hartman, T. J., Judd & S., Bostick, R. M. (2014). Paleolithic and Mediterranean diet pattern scores and risk of incident, sporadic colorectal adenomas. *The American Journal of Epidemiol, 181 (10),* 1-10.

Willett, W.C., Sacks, F., Trichopoulou, A., Drescher, G., Ferro-Luzzi, A., Helsing, E. & Trichopoulos, D. (1995). Mediterranean diet pyramid: a cultural model for healthy eating. *American Journal of Clinical Nutrition, 61 (6),* 2–6.

Wolberg, W. H. (1992). *Breast cancer wisconsin (original) dataset*. Retrieved May 01, 2015, from http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29.

World Health Organization (2014). *Cancer fact sheet N°297*. Retrieved February 1, 2014, from http://www.afro.who.int/index.php?option=com_docman&task=doc_download&gid=6139

X-Jolliffe, I. T. (2002). *Principal component analysis*. Retrieved May 15, 2015, from http://cda.psych.uiuc.edu/statistical_learning_course/Jolliffe%20I.%20Principal%20Component%20Analysis%20%282ed.,%20Springer,%202002%29%28518s%29_MVsa_.pdf.

Xinwu, L. (2008). Research on text clustering algorithm based on K_means and SOM. *IEEE Proceeding of. International Symposium on Intelligent Information Technology Application Workshops*, 341-344.

Yang, Y., & Rong, L. (2008). Establishment of the evaluation index system of emergency plans based on hybrid of SOM network and K-means algorithm. *IEEE Proceeding of 4th International Conference on Natural Computation*, 347-351.

Yıldırım, P., Uludağ, M., & Görür, A. (2008). Hastane bilgi sistemlerinde veri madenciliği. *Akademik Bilişim Konferansı Bildirileri,* 150-158

Yu, L., Wang, S., & Lai, K. K. (2007). Parallel web text clustering with a modular self-organizing map system. *Journal of Computational Information Systems, 3 (3),* 909-916.