Preface

This is a masters thesis conducted at the Department of Computer and Information Science at the Norwegian University of Science and Technology. The project period was January 20 to June 23. The project assignment was specified by the Norwegian Centre for Electronic Patient Records (NSEP).

The technical teacher responsible for the project assignment was Øystein Nytrø. The project was supervised by Dr.Ing.Amund Tveit from the Norwegian University of Science and Technology.

Acknowledgements The author expands gratitude towards the following persons: Amund Tveit, Ole Edsberg and Thomas Brox Røst for their suggestions and feedback on the project, Anders Grimsmo for the contributions of medical expertise and David Scott Hamnes for proof-reading a draft of this thesis.

Problem statement

The electronic patient record is primarily used as a way for clinicians (and sometimes also lawyers) to remember what has happened during the care of a patient. Can it also be an information source for medical research? To help answer that question, this project aims to evaluate the usefulness of applying data mining methods on the patient record. The data mining method considered is clustering, which will be applied to data from a general practitioner database.

The candidate should first design a number of clustering cases, for example two. Each case should contain a selection of data from the database and an idea of what type of information we can hope to discover from it. The candidate should then perform the necessary preprocessing steps to create suitable data sets. The candidate should design a number of clustering approaches, implement them and perform carefully planned experiments to evaluate the methods' effectiveness on the data sets. An important part of the evaluation would be review of results by clinical experts, for example general physician Anders Grimsmo and rheumatologist Arild Faxvaag.

Abstract

The electronic patient record is primarily used as a way for clinicians to remember what has happened during the care of a patient. The electronic record also introduces an additional possibility, namely the use of computer based methods for searching, extracting and interpreting data patterns from the patient data. Potentially, such methods can help to reveal undiscovered medical knowledge from the patient record.

This project aims to evaluate the usefulness of applying clustering methods to the patient record. Two clustering tasks are designed and accomplished, one that considers clustering of ICPC codes and one that considers medical certificates. The clusterings are performed by use of hierarchical clustering and k-means clustering. Distance measures used for the experiments are Lift correlation, the Jaccard coefficient and the Euclidian distance. Three indices for clustering validation are implemented and tested, namely the Dunn index, the modified Hubert Γ index and the Davies-Bouldin index. The work also points to the importance of dimensionality reduction for high dimensional data, for which PCA is utilised. The strategies are evaluated according to what degree they retrieve well-known medical knowledge owing to the fact that a strategy that retrieves a high degree of well-known knowledge are more likely to identify unknown medical information compared to a strategy that retrieves a lower degree of known information.

The experiments show that, for some of the methods, clusters are formed that represent interesting medical knowledge, which indicates that clustering of a general practitioner's record can potentially constitute a contribution to further medical research.

Contents

List of Figures

List of Tables

1 Introduction

The electronic patient record is primarily used as a way for clinicians (and sometimes also lawyers) to remember what has happened in the care of a patient. Can it also be an information source for medical research? To help answer that question, this project aims to evaluate the usefulness of applying data mining methods on the patient record. The data mining method considered is clustering, which will be applied to data from a general practitioner database.

1.1 Motivation.

Today, paper-based patient records are rapidly being replaced by electronic patient record systems. The main reason for this change is to improve the usability of the patient record as an information source during health care. The change also introduces an additional possibility, namely the use of computer-based methods for searching, extracting and interpreting data patterns from the patient data. This possibility can potentially involve a huge increase in the efficiency and consistence of the information scanning and can help medical researchers find potential relevant research problems by firstly generating a large set of hypotheses and then utilise these hypotheses to select interesting problems and to carry out clinical research to validate or reject the hypotheses.

1.2 Objective.

The aim of this thesis is to investigate how clustering methods can be applied to tabular data in a patient record. The thesis attempts to examine how different clustering strategies succeed in clustering patient record data. The strategies will be evaluated according to what degree they retrieve well-known medical knowledge. This is due to the assumption that a strategy that retrieves a high degree of well-known knowledge are more likely to generate credible hypotheses compared to a strategy that retrieves a lower degree of known information.

1.3 Background - data mining

This section gives a short introduction to the area of data mining. Section 1.3.1 gives a general introduction to knowledge discovery in databases, while Section

1.3.2 describes the concept of clustering. The information given in this section constitutes essential background material for this thesis.

1.3.1 Knowledge discovery in databases

Knowledge Discovery in Databases(KDD), explored in (HK01), is a process that attempts to uncover significant data patterns in large amounts of data. The KDD process can be divided into the following three principal steps:

- *Data preprocessing.* A substantial part of the knowledge discovery process is to prepare the data for data mining. The miner must decide what data is relevant for the task at hand, and how this data is best represented. Also, the data extracted is likely to contain missing or erroneous values. In such cases strategies of how to deal with those values must be considered. Moreover, methods such as normalisation are commonly used for scaling such that all values for an attribute fall within a specified range, for instance to avoid having large-value attributes dominate low-value attributes.

- *Data mining.* Intelligent methods are applied in order to extract patterns from the preprocessed data set. A generic name for such intelligent methods are *data mining* methods. Data mining methods can be either supervised or unsupervised. An unsupervised method implies that the mining does not rely upon any predefined classes or class-labeled training examples, but rather seeks to group objects based on their internal similarity structure. In contrast, supervised data mining methods make use of predefined information and use this information to guide the mining. Clustering, which is the method utilised in this thesis, is an example of an unsupervised data mining method. A further clarification of the main concepts of clustering is shown in Section 1.3.2.

- *Result evaluation and representation.* The mining methods reveal some internal structure in the data set that potentially contains interesting and valid knowledge. An important fact to keep in mind is that the mining tasks tend to indicate structures or patterns in a data set even if the structure does not reflect any meaningful information. This emphasises the importance of a validation or evaluation of the results. Possible validation methods includes both human inspection of the conceptual structure by an expert and automatic validation of the data structure by implemented quality measures, from where the first method should never be omitted. An important part of the finishing KDD step is to prepare an intuitive and clear representation of the results, due to the fact that the KDD process is

> often worked out to potentially reveal knowledge of interest for experts in domains other than computer science.

This thesis considers all three steps in the KDD process implemented on the patient journal.

1.3.2 Clustering

The intention behind clustering analysis is to structure and partition a data set based on information which is implicitly given in the data. The result of a clustering process is a grouping of objects in the data set, where the object of a group has a high degree of similarity with the other objects in the group and a low degree of similarity with objects in other groups.

A classical clustering example is reported in (Fis36). The objective in this article is to group different types of iris flowers in terms of the two measurements petal width and petal length. Three types of irises were considered for the task. Figure 1 shows a plot of 40 of the flowers in the data set, where each iris is represented as a point in two dimensional space. The example indicates that the parameters chosen were suitable to differentiate the iris type *Iris setosa* from the others, but did not differentiate the other two iris types well. However, if a clustering algorithm is applied that seeks to group the data objects into groups of similar size, the chosen features include enough information to separate the iris types.

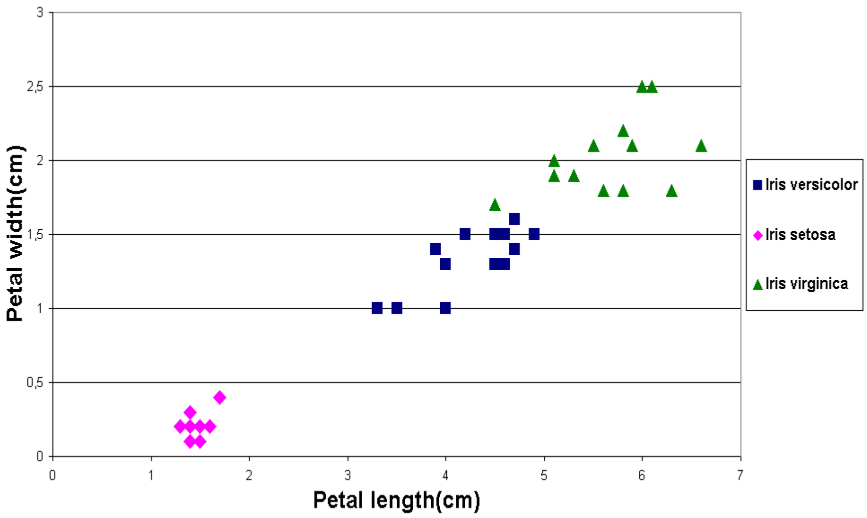


Figure 1: Clustering of iris types

An important field within medicine, for which improvements can probably be indicated by the use of clustering analysis, is the development and continuous updating of diagnoses. In this context clustering analysis can be applied to group patients with the same diagnosis, such that the group of patients with one specific diagnosis may be divided into subgroups. *These subgroups can indicate a diversity in the diagnosis and reveal a potential set of ailments covered by a joint diagnosis in the present diagnosis system.* This discovery can actuate the development of separate diagnoses and guidelines for the particular ailments.

The preprocessing step to prepare a clustering task is understood to include the decision of which features that relevant to the clustering task. The features should be suitable to separate or group objects in regard to relevant concepts and they should also be uncorrelated to maximize the amount of information represented. The data structure utilised is most commonly a vector. For instance, the feature vector used in the iris example above was

$$x = \begin{bmatrix} p\_length \\ p\_width \end{bmatrix}$$

In addition to the preprocessing step, two factors largely determine the clustering results, namely the choice of *distance measure* and the choice of *clustering algorithm*. A distance measure is used to calculate the distance between two objects in the data set. For instance, given the two iris flowers

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, x_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

and the distance measure

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{2} (x_{1_i} - x_{2_i})^2}$$

where x_{1_i}, y_{1_i} are the i-th coordinates of x_1 and x_2 this gives a distance of $\sqrt{2}$ between x_1 and x_2.

A *clustering algorithm* takes a set of data objects as input and outputs a grouping, or a hierarchy of groupings, of the objects. There are several main categories of clustering algorithms, and the choice of algorithm depends on the data available and the purpose of the clustering. Some algorithms tend to make clusters of similar size while other algorithms make clusters of dissimilar size, some algorithms tend to make spherical clusters while other algorithms form elongated clusters, some algorithms are sensitive to outliers and so on. However, the algorithms have an important common characteristic, which is that their output is highly dependent of how the input data is preprocessed and represented, and also of how the distances between objects are calculated. Most algorithms also require input parameters from the user, for instance the desired number of clusters or threshold values for deciding if an object belongs to or does not belong to a cluster. Another characteristic of most clustering algorithms is that they always impose a clustering structure to a data set, even if there is no meaningful way to cluster the data.

1.4 Background - health care

The thesis is carried out in cooperation with the Norwegian Centre for Electronic Patient Records (NSEP). The clustering tasks designed and implemented in this thesis are applied on a general practitioner's record for which access has been given through NSEP.

In Norway, the GPs constitute the so-called first line service, which means that all contact between the patient and public medical services starts with a consultation between the patient and the general practitioner. The GP carries out the basic examination of the patient and sends the patient to hospital or to other specialists if necessary. The general practitioner's patient record mirrors this organisation of the medical service. The record is general and reflects the incidence and the treatment of diseases and medical complaints in an area. In contrast, a specialist record will cover the treatment of specific diseases over a larger area.

When mining patient record data the question of data security and sensitivity is central. Security and privacy issues must be concerned before medical data are released for research objectives like data mining. The research should not violate any existing laws regulating security of sensitive personal information. Research which makes use of personal sensitive information presupposes that all research participants are aware of the content of existing laws and have signed on security and confidentiality agreements. In Norway the use of patient record data for research like data mining is regulated by the Personal Data Act(PDA), the Health Personnel Act(HPA) and the Personal Health Data Filing System Act(PHD).

1.5 Primary plan

To explore the problem, two clustering tasks will be designed and carried out, one that considers the clustering of diagnosis codes and one that considers medical certificates. This task includes the following subtasks:

- To define and prepare a sensible attribute selection and data representation of the data relevant to the clustering tasks.

- To define and perform relevant preprocessing on the data sets prior to clustering analysis

- To implement and test different clustering algorithms for which each is tested with different parameters. Algorithms implemented in this work are hierarchical clustering and k-means clustering. Three strategies are implemented for hierarchical clustering, namely the minimum distance strategy, the maximum distance strategy and the average distance strategy. Distance measures implemented are Lift correlation, the Jaccard coefficient and the Euclidian distance.

- To implement and test different clustering validation indices on the achieved clustering results. Indices implemented are the Dunn index, the Hubert Γ index and the Davies-Bouldin index.

- To evaluate and discuss the results obtained through the experiments.

1.6 Main conclusions

This thesis demonstrates that the application of clustering methods to a patient record can identify established medical information and therefore also potentially identify previously undiscovered knowledge. However, only a minority of the methods tried out in this thesis managed to reveal known information. The method that best identified meaningful clusters was the hierarchical clustering given by the maximum distance merge strategy and Lift correlation as the distance measure.

This thesis emphasises a primary challenge associated with all clustering tasks, namely the almost infinite number of possible combinations of attributes, preprocessing methods, distance measures and clustering algorithms. The sensible choices are not absolutes but depend on parameters such as the shape or size of the clusters underlying the data set, which are parameters likely to be unknown to the computer scientist prior to the clustering process. This thesis also demonstrates that clustering quality is difficult to measure automatically, which stresses the importance of evaluation by an expert.

1.7 Related work

Previous research includes several approaches of applying clustering methods to medical record data. (BWH01) attempts to compare given diabetes treatment to existing guidelines by clustering. The clustering is based upon information about the frequency and regularity of specific treatments. The data used in this work is extracted from a hospital journal. (Tsu01) also deals with knowledge discovery in a hospital journal. The aim of this work was to discover patterns among patients examined for bacterial infections. This work emphasises the preprocessing stages and argues that preprocessing constitutes 70 % of the total work. In (ML00) the potential relations between causes of death and demographical information are investigated. Again, a substantial part of the work deals with the preprocessing stage of the knowledge discovery process. (WL02) attempts to utilise clustering to identify homogenous groups of care episodes in a hospital journal.

The tasks mentioned so far do all treat medical data, however, the author of this thesis has no knowledge of other research concerning the application of clustering methods to the general practitioner's record.

1.8 How to read this thesis

The intended reader for this thesis is a masters student or a professional with basic computer science skills and with some previous knowledge of machine learning and clustering. This thesis may also be of interest to medical personnel.

This thesis has 5 main sections. Section 2 gives an introduction to clustering in general, and a detailed description of existing algorithms, configurations, preprocessing methods and clustering validation methods used in this thesis. Section 3 describes the patient record on which the clustering methods are applied, and also describes tools utilised in this thesis. Section 4 describes the plan for the experiments which includes feature extraction, data preprocessing, clustering procedure and clustering evaluation for each clustering task. Section 5 describes the results from the experiments. Section 6 explores, explains and discusses discoveries or unexpected results obtained through the accomplishment of the clustering tasks. Section 7 summarises the findings of this thesis.

2 Clustering methods

This section has two main objectives. Firstly, the section argue for the choice of which methods to utilise during the clustering procedures. Types of methods are clustering algorithms, distance measures, cluster validity/quality measures and preprocessing methods. Secondly, the concrete methods utilised in the thesis are explored for each type of method. Section 2.1 deals with clustering algorithms, Section 2.2 deals with the distance measures, Section 2.3 treats the validity indices while Section 2.4 deals with methods for preprocessing.

2.1 Clustering algorithms

According to (HK01) and (H.D02), there are several conceptual groups of clustering algorithms. The main groups are hierarchical clustering, which creates a hierarchical decomposition of the data set; partitioning methods, which partition the data into non-overlapping groups; density-based methods, which consider the density in the neighborhood of an object and grid-based methods, which partition the object space into a finite number of cells to improve the processing time. The two clustering algorithms implemented in this thesis were *hierarchical clustering* and the partitioning method *k-means clustering*.

The main reason why hierarchical clustering was selected was the known lack of knowledge of the number of clusters underlying in the data set, and the size and shape of those clusters. Hierarchical algorithms do not require any input parameters from the user but can potentially give an idea of what the most correct number of clusters in the data set may be. Moreover, the algorithm can be run with several different strategies to how to select objects to merge or split. Each of these strategies will tend to find clusters of different shapes and sizes, such that the quality of the results obtained by the different strategies will indicate the shape or size of the clusters hidden in the data set. These observations can then be used to guide the choice of algorithms or input parameters suitable for the data set. The hierarchical clustering method is also well-known and frequently used. The main disadvantage associated to hierarchical clustering is that objects are never moved between clusters, which can potentially hinder the algorithm to find the ideal clusters.

The k-means algorithm has its strength in the iterative moving of data objects between the clusters and was therefore believed to be strong where the hierarchical algorithm was weak. Due to this repeated object replacements the computation is more costly than the computation associated to the hierarchical clustering. The k-means method is therefore suitable only for data sets of size small-medium, which was the expected size of the data sets designed in this thesis. K-means tend to form spherical clusters of similar size. The disadvantage of k-means clustering is the need for calculating a mean object to represent each cluster. The method therefore does not work for categorical attributes, which are attributes that take a number of discrete values with no internal order. Due to the calculation of mean values, the k-means algorithm also suffer from outliers.

Other algorithms were also considered but refused due to a variety of reasons. The hierarchical clustering algorithms Birch and Cure were rejected due to a limiting branch structure and the demand for crucial input parameters respectively. The partitioning method Clarans, the density-based method DBscan and Denclue and the grid-based method Wave-cluster were also rejected due to influential input parameters such as the number of clusters, branching factor, neighborhood radius, cluster radius and the number of grids. The two selected algorithms are explored in the following paragraphs.

Hierarchical clustering As mentioned, the hierarchical clustering algorithms create a hierarchy of clusterings in the given set of data objects. The algorithm can be either *agglomerative* or *divisive*. An agglomerative clustering algorithm starts with each object forming a separate group, and merge two and two groups until all the objects belong to one group. The divisive approach starts with all the objects in the same cluster, and splits a cluster until each object forms its own cluster. The agglomerative approach is the one used in this thesis.

There are several strategies for how to choose which two clusters to merge next. Three strategies are considered in this thesis: the minimum distance strategy, the maximum distance strategy and the average distance strategy.

- The minimum distance strategy merges the two clusters with the smallest minimum distance, where the minimum distance is given by

  $$d_{min}(C_i, C_j) = min|p - p'|, p \in C_i, p' \in C_j \tag{1}$$

  More intuitively, the minimum distance strategy merges the two clusters with the two closest objects that are not yet in the same cluster.

- The maximum distance strategy merges the two clusters with the smallest maximum distance, where the maximum distance is given by

  $$d_{max}(C_i, C_j) = max|p - p'|, p \in C_i, p' \in C_j \tag{2}$$

This means that the maximum distance strategy merges the two clusters with the smallest maximum distance between two objects connected one to each cluster.

- The average distance merges the two clusters with the smallest average distance between two objects belonging one to each cluster. The average distance is given by

$$d_{avg}(C_i, C_j) = 1/(n_i * n_j) \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'| \tag{3}$$

where n_i is the number of objects in cluster C_i.

K-means clustering The k-means clustering algorithm organises the objects into k partitions, where each partition represents a cluster. The algorithm first chooses k random objects to represent the k clusters. The other objects are assigned to the cluster for which the representing point is most similar. Then, for each cluster, a mean object is calculated based on the objects in the cluster, and all the objects in the data set are reassigned to the cluster with the most similar mean. The algorithm is repeated until there are no reassignments. The algorithm is described in Figure 2. The k-means algorithm can only be applied to data objects represented in such a way that a mean can be calculated.

```
Input:
        Number of clusters k and database containing n objects
Output:
        A set of k klusters that minimizes the mean square error

1. Choose k random objects as the initial cluster "means"

2. While changes

      a) assign each object to the same cluster as the most
         similar mean object

      b) calculate new mean for each cluster based on the
         objects assigned to that cluster
```

Figure 2: The k-means algorithm

2.2 Distance measures

The distance measures used in this thesis were the *Euclidian distance*, the *Jaccard coefficient* and *lift correlation*. This thesis also uses a strategy to deal with attributes of mixed type.

Euclidian distance was implemented in this thesis due to the fact that both (HK01) and (H.D02) claim that this is one of the most popular and traditional distance measures. The Jaccard coefficient was described by (HK01) as a distance measure suitable for asymmetric binary variables, which is binary variables where the outcomes of the states are not equally important. This coefficient was therefore considered appropriate for the data sets consisting of binary variables denoting the presence or absence of a condition. The lift correlation was previously used at another project at NSEP, and the results from that work were promising. Therefore, lift correlation was also tested out in this thesis[1].

A patient journal is supposed to contain a large share of both numerical and categorical attributes and it would therefore be an advantage to be able to handle data objects represented by attributes of mixed types. A possible approach to this challenge is to combine several distance measures such that for each data type an appropriate distance measure is used. This thesis performs clustering to data sets including both interval-scaled, binary and categorical values. Details about the measuring of distances for these data sets are explained in Section 4 related to the description of the clustering tasks. The other three distance measures used are described in the following.

Euclidian distance The Euclidean distance is defined in (H.D02) as

$$d(x, y) = \sqrt{\sum_{i=1}^{l} (x_i - y_i)^2} \tag{4}$$

where x, y are feature vectors, x_i, y_i are the i'th coordinates of x and y and l is the length of the vector. The Euclidian distance is applicable only for feature vectors with values coded in numerical values.

Jaccard coefficient As mentioned, the Jaccard coefficient is best suited for asymmetric binary variables where, by convention, the most important outcome

[1] Thanks to Ole Edsberg for the idea of using Lift correlation and for the lending of source code that implements it.

is coded by 1 and the least important outcome is coded by 0. The Jaccard coefficient is defined in (HK01) as

$$d(x, y) = (r + s)/(q + r + s) \tag{5}$$

where q is the number of variables that equal 1 for both objects x and y, r is the number of variables that equal 1 for object x but that are 0 for object y and s is the number of variables that equal 0 for object x but equal 1 for object y.

Lift correlation This thesis compares Lift correlation to traditional clustering distance measuring on binary feature vectors. The distance between two objects are the inverse of the correlation, given by

$$corrdist_{ij} = f_i f_j / f_{ij} * numf \tag{6}$$

where f_i is the number of features that is true for object i, f_j is the number of features which is true for object j, f_{ij} is the number of features which is true for both object i and object j and num_f is the total number of features.

2.3 Cluster validity measures

A characteristic of the clustering algorithms considered in this thesis is that they always impose a clustering structure to a data set, even if there is no meaningful way to cluster the data. This fact emphasis the importance of clustering validation. The use of automatical methods to measure the quality of the clustering structure, such as the density of clusters, the distance between clusters and the accordance between the proximity matrix and the clustering, can be a part of the validation. It should be emphasised that automated validation by today's existing measures must never replace human inspection.

This thesis tests out three indices for automatic measuring of clustering quality. These indices are the *Dunn index*, the *Modified Hubert* Γ *statistic* and the *Davies-Bouldin index*. These three indices are founded on different components of the clustering structure and therefore constitute an adequate basis for evaluating the clusters. Moreover, according to (TK99) and (CR98) all three indices are well-known and widely used methods. The indices are described in the following paragraphs.

Dunn index The Dunn index measures the relationship between the diameter of the clusters and the distance between the clusters. The dissimilarity between two clusters C_i and C_j is defined as

$$d(C_i, C_j) = min_{x \in C_i, y \in C_j} d(x, y) \tag{7}$$

That is, the dissimilarity between two clusters equals the distance between the two most similar objects not in the same cluster. The diameter of a cluster is described as

$$diam(C) = max_{x,y \in C} d(x, y) \tag{8}$$

which means that the diameter of a cluster is the distance between its two most distant vectors. The Dunn index is then defined as

$$D_m = min_{i=1,\ldots,m}\{min_{j=1,\ldots,m}(d(C_i, C_j)/max_{k=1,\ldots,m} diam(C_k))\} \tag{9}$$

where m is the number of clusters.

The Dunn index will be large for clusterings with small cluster diameters and large distance between the clusters, such that increased index indicates increased clustering quality. A disadvantage with the Dunn index is that outliers are likely to increase the value of the maximum diameter. Another disadvantage with the index is that the density of the clusters is not considered, so a large diameter will be considered negative even if the clusters are dense.

Modified Hubert Γ statistic The modified Hubert Γ statistic is founded on the relationship between the proximity matrix and the clustering. The statistic compares the proximity matrix p with a matrix q that holds the distance between the cluster means for each pair of clusters.

In this thesis, p is the $n * n$ proximity matrix defined such that p_{ij} is the dissimilarity between object i and object j, found by use of the dissimilarity measure used for the clustering. q is a $n*n$ matrix defined such that q_{ij} is the dissimilarity between $mean(C(i))$ and $mean(C(j))$, where $C(i)$ is the cluster to which object i is assigned and the dissimilarity measure used is the same as the one used for the clustering. n is the number of objects in the clustering. The modified Hubert Γ statistic can then be defined as

$$Gamma = 1/(n^2 var(p) var(q))(\sum_{i=1}^{n}\sum_{j=1}^{n}(p_{ij} - mean(p))(q_{ij} - mean(q))) \tag{10}$$

As was the case for the Dunn index, the Hubert Γ statistic is also supposed to increase when the clustering quality increases.

Davies-Bouldin index The Davies-Bouldin index calculates the average similarity between each cluster and its most similar one. Because the ideal clustering has a smallest possible similarity between the clusters, this study seeks to minimize the Davies-Bouldin index.

The Davies-Bouldin index is defined as

$$DB_m = 1/m \sum_{i=1}^{m} R_i \tag{11}$$

where m is the number of clusters and R_i is defined as

$$R_i = max_{j=1,...,m,j \neq i} R_{ij}, i = 1, ..., m \tag{12}$$

R_{ij} is a measure for the similarity between clusters C_i and C_j and is defined as

$$R_{ij} = (s_i + s_j)/d_{ij} \tag{13}$$

where s_i measures the dispersion of cluster C_i and d_{ij} measures the dissimilarity between clusters C_i and C_j.

In this thesis s_i is defined as

$$s_i = (1/n_i \sum_{x \in C_i} d(x, mean(C_i))^2)^{1/2} \tag{14}$$

where n_i is the number of objects in C_i and $d(x, y)$ is the distance between x and y found by use of the same dissimilarity measure as used for the clustering. d_{ij} is defined in this thesis as the dissimilarity between the calculated means for clusters i and j, found by use of the same dissimilarity measure as used for the clustering.

2.4 Preprocessing methods

This section explores methods utilised during the preprocessing stage. Section 2.4.1 describes the main steps of principal component analysis, Section 2.4.2 describes methods for normalisation and Section 2.4.3 deals with the treatment of missing values.

2.4.1 Dimensionality reduction

Principal component analysis(PCA) was performed in this work to reduce the dimensionality of the data set. According to (I.T02), the idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated attributes, while preserving as much as possible of the variation contained in the data set. To achieve this, PCA transforms the existing attributes into a new set of uncorrelated attributes that are ordered so that the first attributes maintain most of the variation present in all of the original variables. Then, a number of attributes can be selected which balances the benefit of reduced dimensionality with the inconvenience of lost variance information.

In this thesis the PCA was carried out according to the following:

1. The input data were normalised to prevent attributes with a wide range of values from dominating attributes with a smaller range of values.

2. For each feature in the data set, the mean value for that feature was calculated. The mean value was subtracted from the feature values.

3. The covariance matrix for the data set was calculated.

4. The eigenvalues and eigenvectors for the covariance matrix were calculated.

5. The desired number n of attributes were decided. To make a sensible choice, the eigenvalues were assessed. The eigenvalues reflect the degree of variation in the corresponding eigenvectors, which causes a reasonable number of features to be indicated, for instance, by a flattening of the values or by the occurrence of knees.

6. The matrix achieved after accomplished step 5 was then multiplied by the matrix obtained after carried out step 2.

The result after completing these steps is a data set of a dimensionality n which is selected by the user.

2.4.2 Normalisation

According to (HK01), an attribute is normalised when its value is scaled to fall inside a small specified range. Normalisation was used in this work to avoid attributes with potentially large values from outweighting attributes with smaller values. The normalisation methods considered in this thesis were *linear min-max normalisation* and *z-score normalisation.*

Linear min-max normalisation Linear min-max transformation performs a linear transformation on the values such that they fall inside a predefined range but still preserves the relationship among the original values. The new values are given by

$$v' = \frac{v - min_A}{max_A - min_a}(newmax_A - newmin_A) + newmin_A \tag{15}$$

where min_A and max_A are the minimum and maximum values respectively of the original data and $newmin_A$ and $newmax_A$ are the extreme points of the normalised range (HK01).

Z-score normalisation In z-score normalisation the values for an attribute are normalised based on the mean and standard deviation of the values. The normalised value v_z for a value v is given by

$$v_z = (v - mean_v)/stdev_v \tag{16}$$

The values achieved after normalisation will have a mean value that equals 0 and a standard deviation that equals 1.

2.4.3 Strategies to deal with missing values

All clustering tasks designed in this thesis encountered the problem of missing values. The strategies selected to deal with missing values differ among the clustering tasks. Each strategy is therefore described separately in Section 4 in connection with the relevant clustering task.

3 Materials

This section describes the material utilised in this thesis. Section 3.1 gives an introduction to the general practitioner's record from which the data sets for clustering are extracted. Section 3.2 gives a short introduction to a graph visualisation tool utilised in this thesis, while Section 3.3 gives a short description of a statistical programming language utilised for the principal component analysis.

3.1 Patient record system

The patient journal database from where the data used in this thesis is collected is from a general practitioner's office in a medium size Norwegian community. The database is a part of a *ProfDoc Vision Allmenn* journal system. The system is designed to cover all information storage needs that may arise in a general practitioner's office. These needs include storage of patient information, contact information, medication, vaccinations, medical allergies, diagnoses, laboratory tests and results, referrals, medical certificates, discharge summaries and correspondence. More information about the ProfDoc systems can be found on (Pro).

The Norwegian Centre for Electronic Patient Journals (NSEP) possesses a copy of all data stored in this journal system in the period 1992 to 2003. During this period, the number of patients who consulted the general practitioner's office was 10 859. The number of contacts that occurred during this period was 482 906. The clustering tasks explored in this thesis were applied on the NSEP data. The term 'contact' covers several types of incidents which cause information to be entered into the journal; here, consultations constitute the main subgroup.

A substantial part of this thesis deals with the distribution of and correlation between codes encoded by use of the International Classification of Primary Care (ICPC) code standard, which is a system for coding symptoms, complaints, diagnoses and process during primary care. The ICPC coding standard is further explored in Appendix C.

The ICPC codes in the journal system are linked to the contacts. For contacts such as consultations, phone calls or sick calls, the general practitioner enters into the journal one or several ICPC codes which denote the diagnoses or complaints that caused the contact (Gri05). The most common practice is to enter only one contact code, which is the case for 82% of the contacts for which contact codes are entered in the NSEP journal.

The total number of ICPC codes stored in the NSEP journal is 348 994, where the distribution of codes among the ICPC main code-groups is given in Figure 3, and the percentual distribution of codes among ICPC main groups per year is given in Figure 4.

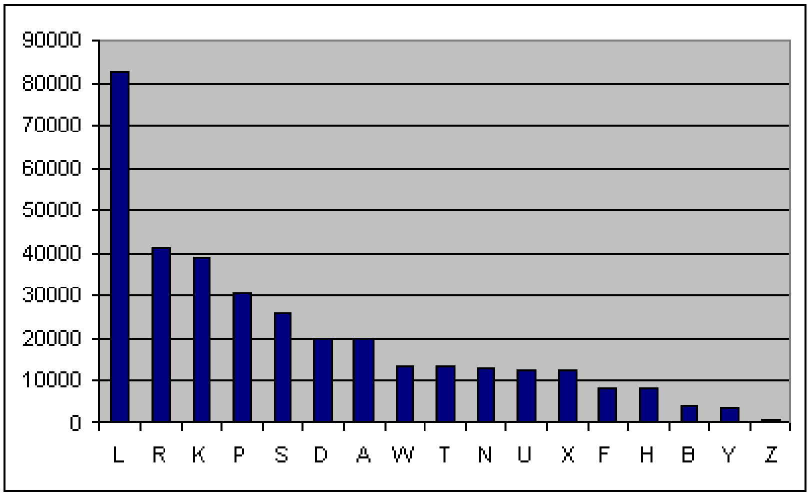


Figure 3: Distribution of ICPC code-groups

Figure 3 demonstrates that a substantial part of the codes entered belong to ICPC code-group L, which cover musculoskeletal ailments. The L-group is followed by the group of respiratory ailments which is less than half the size of the musculoskeletal group. The size of the succeeding groups decreases gradually as indicated by the figure.

Figure 4 displays the development of ailments and coding during the period 1992-2003.

The most significant alterations are an increased number of psychological ailments(P) and a decreased number of respiratory ailments(R).

One of the clustering tasks implemented in this thesis deals with medical certificates. The data set includes 48 038 temporary medical certificates and 2 786 long-term medical certificates. The number of diagnoses for which the certificates are written is 49 150 and 3 269 for the temporary and long-term groups respectively. 53% of the temporary certificates are prescribed for women and 47% are prescribed for men. The corresponding numbers for long-term certificates are 59% and 41%. The distribution of diagnoses codes for which the medical certificates are prescribed are given in Figure 5, where Figures 5a and 5b show the distribution for temporary and long-term certificates respectively.

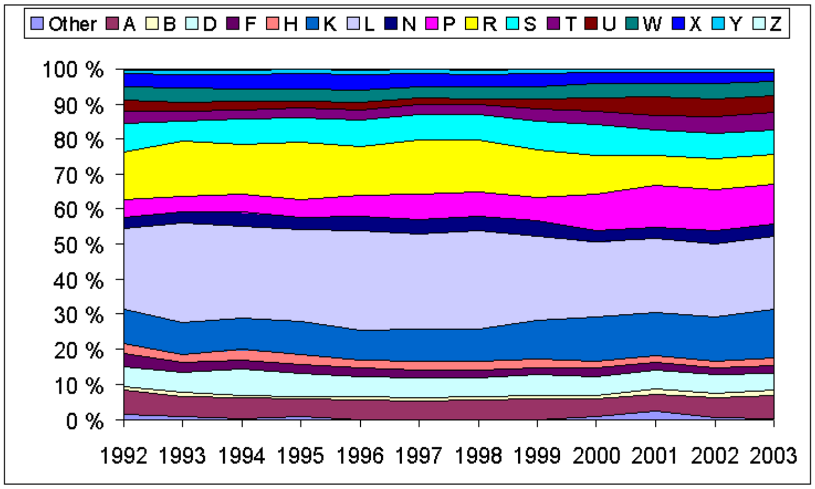


Figure 4: Distribution of ICPC code-groups per year

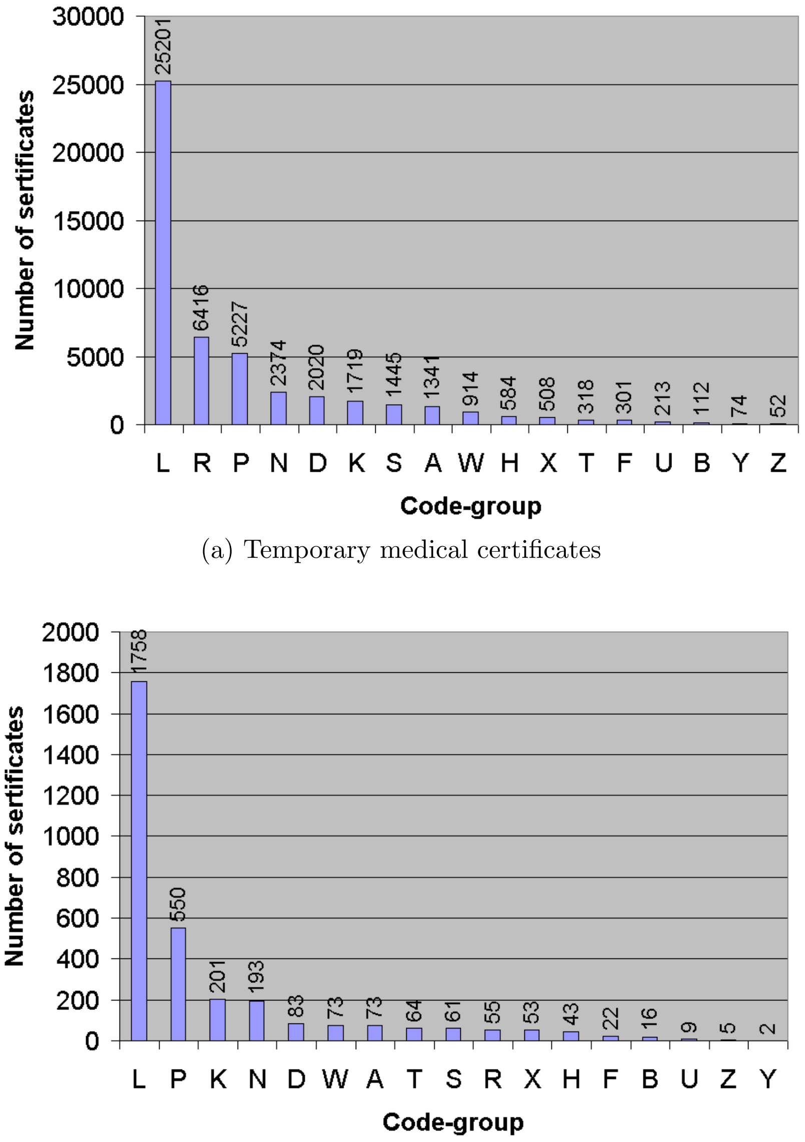
3.2 Visualisation tool - GraphViz

GraphViz is an open source graph visualisation program. GraphViz includes several main graph layout programs that creates hierarchical directed graphs, undirected graphs and graphs with a radial and circular layout. The layout program used in this thesis is *Neato*, which is based on the Kamada-Kawai algorithm proposed in 1989. Neato draws undirected graphs by constructing a virtual physical model of the graph. The virtual model consists in placing a string between each pair of nodes, where the length of the string is set to the length of the path between the nodes. Neato then runs an iterative solver that virtually pulls and contracts the strings such that they move the nodes until a low-energy configuration is achieved. GraphViz is further explored at (Gra)

GraphViz was used in this thesis to visualise the clustering results. A graph structure can potentially help indicating the distribution of objects by calculating a graph configuration that aims to preserve the all-to-all distances between the objects.

3.3 Statistical tool - R

This thesis makes use of principal component analysis to reduce the dimensions of the data set before applying clustering algorithms. The principal component analysis(PCA) was performed by use of libraries provided by the programming language R, which is an open source statistical programming language based on



(a) Temporary medical certificates

(b) Long-term medical certificates

Figure 5: Distribution of code-groups among medical certificates

the S and S/Plus programming languages. The R language is further explored at (The). Rpy, a Python interface to the R programming language, was utilised to execute the R-functions.

From the beginning, methods offered by Numerical Python were used during the PCA analysis. However, the results returned by the Numerical Python methods deviated from those returned by the corresponding R methods. The R libraries were preferred to the corresponding methods offered by Numerical Python both because R is a well-known and thoroughly tested statistical language and because the results obtained by use of R methods appeared to be correct during the clustering analysis.

4 Experimental plan

This section contains a description of the design of the two clustering tasks. Section 4.1 deals with clustering of ICPC codes, while Section 4.2 treats clustering of medical certificates. For each task the selection and preprocessing of data sets and the procedure followed for the clustering are explained.

4.1 Clustering task 1: ICPC codes

The first designed clustering task was to group the ICPC codes based on their degree of mutual occurrences with the patients. Section 4.1.1 describes the data selection and preprocessing steps. Sections 4.1.2 and 4.1.3 describe the application of hierarchical clustering on the original data set and the PCA reduced data sets respectively, while Section 4.1.4 describes the application of k-means clustering on the PCA reduced data sets.

4.1.1 Data selection and preprocessing

For each patient, all ICPC codes registered for this patient were selected. The selected data was then filtered according to the following process:

1. Invalid codes were removed. In this case an invalid code was defined as a code which do not start with a letter followed by two or three digits.

2. Codes used for medication, treatment, procedure, test results, administration and referrals were removed. These codes have a digit element within the range 30-69.

3. Codes which occurred in less than one percent of the patients were removed.

This process resulted in a data set consisting of 10,104 patients and 227 distinct ICPC codes, where each patient was registered with at least one ICPC code. The data was represented by binary values, such that each code was represented by a vector of 10,104 binary values, from which each value represented a patient. A value of 1 indicated the presence of the code for this patient, while a value of 0 indicated the absence of the code for that patient.

4.1.2 Hierarchical clustering

The data set was clustered by use of hierarchical clustering. This was due to the following:

- Hierarchical clustering does not require computation of any mean object and therefore, in contrast to the k-means algorithm, work with all three distance measures implemented in this work. This characteristic renders possible a comparison of these three measures.
- Since a hierarchy of clusterings are produced, specifying the desired number of clusters is not necessary.

For each of the three merge strategies minimum, maximum and average distance, the clustering algorithm was run three times; that is, one run for each of the three dissimilarity measures Lift correlation, the Jaccard coefficient and the Euclidian distance. For each clustering the Dunn index was calculated. The Hubert Γ statistic and the Davies-Bouldin index require computation of mean objects for the clusters and do not work in combination with the Jaccard coefficient or the Lift correlation.

The nine clusterings were evaluated according to the following procedure:

1. Each clustering was evaluated by inspection. This evaluation aimed to validate the conceptual meaning of the clustering. The meaning was based on to what degree the clustering reflected known medical knowledge. For each section the clusterings achieved in this section were ranked.
2. The three merge strategies were evaluated firstly based on the inspection and secondly by the Dunn indices. This step aimed to identify if there was an indication of any overall best merge strategy for this data set and also if the observed quality and the measured quality seemed were consistent.
3. The three distance measures were evaluated firstly based on the inspection, and secondly by the Dunn indices. This step was carried out to see if there was any preferable distance measure for this data set and also to evaluate the consistency between the observed quality and the measured quality.

4.1.3 Hierarchical clustering on PCA reduced data set

Previous research has pointed to difficulties connected to clustering of high dimensional data ((RAR95) (Pat02)). This is further explored in Section 6. As mentioned, the data set used in this task consisted of 227 ICPC-codes registered on 10 104 patients, which resulted in a proximity matrix of dimension $227*10104$. It was therefore reasonable to assume that the dimension of the data set could be a problem combined with a distance measure based on a pairwise comparison of the features in the feature vector. The Euclidian distance is the only one of the distance measures used in this task which has this characteristic.

An interesting task would be to reduce the number of features and then compare the results given by use of the Euclidian distance on the full data set versus the reduced data set. In this thesis we have made use of principal component analysis, described in Section 2.4.1, as a method of dimensionality reduction. The steps carried out to investigate the influence of dimensionality reduction on the data set were the following:

1. 7 000 patients were randomly selected from the population of 10,104. This was done due to limitations of RAM during the PCA analysis.

2. The 7 000-size selection was clustered following the same procedure used for the 10 104-size data set. The results were compared with the result given by clustering the full data set. This aim of this step was to control the stability of the distribution of information in the data.

3. PCA was performed on the selected data set. According to the eigenvalues, a sensible choice of PCA reduced data sets of different dimensions was taken.

4. Hierarchical clustering by Euclidian distance was performed both on the 7,000-selection from the original set and on the PCA reduced data sets. The Dunn index, the Hubert Γ statistic and the Davies-Bouldin index were calculated for each clustering.

5. The clustering results for both the 7,000 set and the reduced sets were compared by inspection. The results were also compared due to the calculated quality measures, and the consistency between the observed and the calculated quality was evaluated.

4.1.4 K-means clustering on PCA reduced data set

The aim at this point was to explore if k-means clustering could give better results than hierarchical clustering when the desired number of clusters was determined. Because k-means clustering is based on the computation of mean objects for the clusters, the Euclidian distance is the only distance measure used in this clustering task suitable for k-means clustering. The strategy used is shown below:

1. The potential ideal number of clusters was selected based on the clustering performed in Sections 4.1.2 and 4.1.3.

2. K-means were run on the 10 104 feature set, the 7 000 feature set, and on the PCA reduced sets.

3. The clustering results were evaluated by inspection and by the quality measures. The consistency between observed and measured quality was evaluated.

4.2 Clustering task 2: Medical certificates

The aim of this clustering task was to explore the composition of patients who had reported sick for short and long periods. Clustering analysis applied to this type of information could help to reveal any potential conceptual grouping of the patients. Such a grouping could contain information about which patients get specific diseases.

General practitioner Anders Grimsmo has outlined the hypothesis that formed the foundation for this task. He suggested that diagnoses given for temporary medical certificates and long-term medical certificates often differed for the same patient. He indicated that the reason for this possible change of diagnosis could be that the second diagnosis was a correction of the first one, which again could indicate poor examination connected to prescription of temporary medical certificates.

The selected strategy to investigate this problem was to look into both the temporary and the long-term medical certificates for patients who have had a long-term medical certificate. The occurrences of diagnoses for each group were counted to reveal potential variations between the two groups. Clustering analysis was then performed separately on the two groups of medical certificates to evaluate whether the groups of patients formed were different for the two cases.

4.2.1 Data selection and preprocessing

For all cases where a temporary medical certificate and a long-term certificate were registered on the same patient, the two medical certificates were included in the data sets. The objects for the clustering application were the medical certificates and not the patients, so that one patient could contribute several times to a clustering.

After consulting with general practitioner Anders Grimsmo, the following features were selected from each medical certificate:

1. The age of the patient at the time when the medical certificate was written.
2. The sex of the patient.
3. The marital status of the patient.
4. The occupational status of the patient.
5. The diagnosis for which the medical certificate was prescribed.
6. The superior diagnosis class for the diagnosis.

Both data sets were preprocessed as described in the following:

1. Medical certificates for which date or ICPC code was missing were deleted.
2. The age was normalised. Both linear max-min normalisation and z-score normalisation were applied to the data. Due to outliers, the min-max normalisation resulted in poor spreading of the values, which would cause age to have little influence on the clustering result. The z-score normalisation gave better spreading and was therefore preferred.
3. The data set contained missing values for the attributes marital and occupational status. These attributes had values in the range 1-6, denoting different types of status. Two strategies were tested for the replacement of missing values, namely replacement by default value 0 and replacement by default value 1, where the last value denoted marriage for marital status and employed for occupational status. Value 0 was tested to keep the possibility to identify the patients with missing values and thereby render possible different ways of calculating the distance for these certificates. Value 1 was tested due to the fact that it occurred in respectively 0.55 % and 0.61 %

of the total number of instances of marital and occupational status in the database and therefore represented the most probable value for a random patient. The result was two different data sets, one where 0 was inserted for missing values and one where 1 was inserted.

4. Medical certificates for which the ICPC code was invalid were deleted
5. Medical certificates for which the combination of patient identification and ICPC code already existed in the data set were deleted

Because certificates were removed during the preprocessing stage, some patients could potentially have been deleted from one group but still be represented in the other. Therefore, the groups were compared and certificates associated with patients who were represented in only one group were deleted.

4.2.2 Counting of code occurrences

The first step in the investigation of the medical certificates was to verify if there were any differences between the diagnoses for the temporal and the long-term certificates. The percentual distribution of the occurrences of diagnosis codes and diagnosis classes where counted both for the temporal and the long-term medical certificates. The results were compared, and the differences were noted.

4.2.3 Clustering procedure

The second step was to perform clustering analysis on the data sets to reveal potential conceptual groups of patients. The clustering analysis was performed by use of hierarchical clustering. The distance between two objects was calculated according to the following:

- For interval-scaled values the Euclidian distance was used.
- For binary and nominal variables the distance was set to 1 if the values were different and 0 if the values were identical. For nominal variables of value 0 the value was recognised as a default replacement value and no distance was calculated for this variable.

The rules above constitute the basis weighting of the attributes. During the clustering process several variations on the distance measuring were tried out.

The steps in the clustering process followed are listed in Table 1. The first step was to compare the two normalisation methods min-max normalisation and z-score normalisation. The second step was to compare the two strategies used to replace missing values. The aim of these two steps was to find out if any of the strategies turned out better suited than the other for this clustering task. After evaluating the normalisation and replacement strategies, the third step was to attempt to achieve a meaningful clustering of the data set. A few test runs indicated that the basic weighting did not form conceptually interesting clusters, while double weighting of some of the attributes caused these attributes to dominate the other attributes completely. To test different combinations and weightings of attributes, the clustering algorithm was run several times with small changes in the attributes. In some runs a weighting of 1.5 was tested for selected attributes due to the dominance causes by the double weighting. Table 1 describes only the weighting that deviates from the basic weighting.

Step 1	**Normalisation**
Run 1	Basic weighting min-max normalisation of age
Run 2	Basic weighting, z-score normalisation of age
Step 2	**Replacing missing values**
Run 1	Basic weighting, default value 0 for occupational and marital status
Run 2	Basic weighting, default value 1 for occupational and marital status
Step 3	**Weighting of attributes**
Run 1	Basic weighting
Run 2	Code was weighted by 1.5
Run 3	Code and code-group was weighted by 1.5
Run 4	Marital status was omitted
Run 5	Occupational status was omitted
Run 6	Marital and occupational status were omitted
Run 7	Age was weighted by 1.5
Run 8	Marital status was weighted by 1.5
Run 9	Occupational status was weighted by 1.5
Run 10	Marital and occupational status were weighted by 1.5

Table 1: The clustering process for medical certificates

5 Experimental results

This section describes the experimental results achieved from executing the clustering tasks described in Section 4. Section 5.1 presents the results from the ICPC clustering task described in Section 4.1 while Section 5.2 presents the results from the medical certificate clustering described in Section 4.2.

The experiments were run on an AMD64 3500+ 2G RAM machine with Debian Linux 3.1 Testing OS.

5.1 Clustering task 1: ICPC codes

The results from the ICPC clustering task are described in the following 5 subsections. Section 5.1.1 contains the results from hierarchical clustering applied on the full data set. Section 5.1.2 contains the results from the hierarchical clustering of the PCA reduced data sets, while Section 5.1.3 contains the results from the k-means clustering of the PCA reduced data sets. For each of these sections, the results from the cluster inspection are described first followed by the results from the quality indices. Each section also includes a comment on the results. Section 5.1.4 contains a conceptual description of the clustering that best reflects known medical knowledge.

5.1.1 Hierarchical clustering

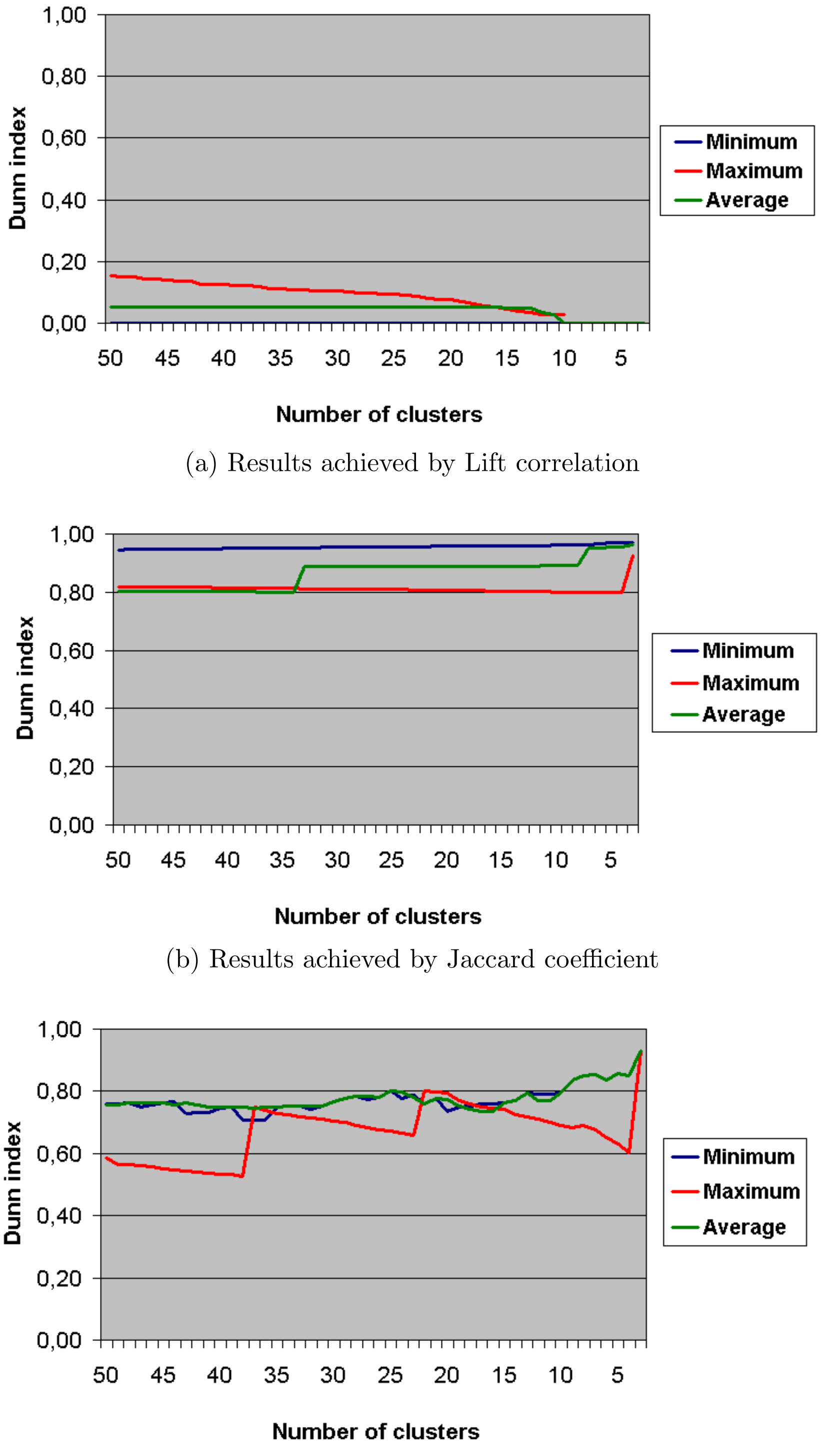
Quality observed through inspection The nine clusterings were observed through the last 30 iterations. Comments and ranking results from the inspection are given in Table 2.

Measured quality Figure 6 compares the three merge strategies for each of the three distance measures according to achieved Dunn index. The results for lift correlation, Jaccard and Euclid are given in Figure 6a, Figure 6b and Figure 6c respectively. As mentioned in Section 2.3, increased clustering quality, which indicates the existence of groups of highly correlated ICPC-codes in the data set, should cause increased Dunn index and Hubert Γ index and decreased Davies-Bouldin index.

The three distance measures are compared in Figure 7 for each of the three

Distance measure	**Merge strategy**	**Description of clustering**	**Ranking**
Lift correlation	Minimum distance	The clustering consists of one large cluster; the objects which are not contained in this cluster are single objects. For each iteration a new single object is merged into the large cluster.	6
	Maximum distance	Meaningful clusters of similar size, all clusters are growing. The algorithm stops merging at 9 clusters because it refuses to include two objects with no correlation to each other in the same cluster.	**1**
	Average distance	Clusters of variable size. Some large clusters which naturally could be grouped as two clusters. Half of the clusters contain 3 or fewer objects when the number of clusters are 30.	3
Jaccard	Minimum distance	The same problem as observed with minimum distance and Lift correlation. The single objects are not the same as for Lift correlation.	6
	Maximum distance	Meaningful clusters, all clusters are growing but the size is variable.	2
	Average distance	One cluster grows fast, the other clusters are of size 1-4.	4
Euclid	Minimum distance	The same problem as observed for minimum distance in combination with Lift correlation and Jaccard.	6
	Maximum distance	The same problem as observed for minimum distance with all distance measures. A few smaller clusters(2-4 objects) which do not seem sensible.	5
	Average distance	The same problem as observed with minimum distance. No smaller clusters.	6

Table 2: Inspection of hierarchical clusterings of original data set



(a) Results achieved by Lift correlation

(b) Results achieved by Jaccard coefficient

(c) Results achieved by Euclidian distance

Figure 6: Dunn comparison of merge strategies for the hierarchical clusterings

merge strategies according to achieved Dunn index. The results for minimum distance, maximum distance and average distance are given in Figures 7a, 7b and 7c respectively.

Comments on the results The results in Table 2 shows that correlation and Jaccard can give sensible results, while Euclid results in one large cluster for all merge strategies. Correlation seems to give the best results, with conceptual meaningful groups both in combination with maximum distance and average distance. Jaccard gives meaningful clusters combined with the maximum distance strategy.

Conversely, the maximum distance strategy gives the most sensible results for all distance measures. The minimum distance strategy gives the least meaningful results while the average distance strategy gives sensible results only combined with the Lift correlation, but not in the other cases.

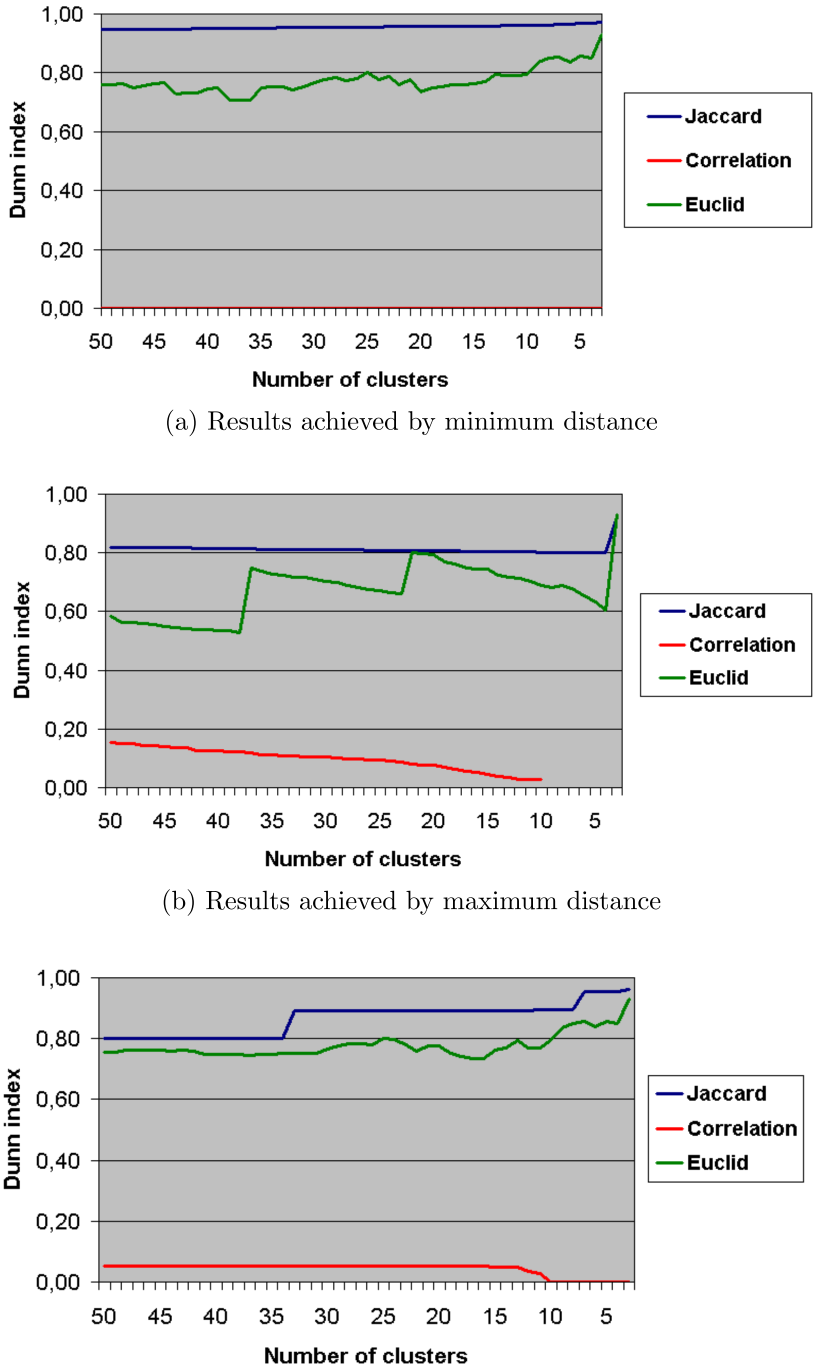
Figure 6 indicates that what is the best merge strategy changes vary for each distance measure. The results found during the inspection are reflected only in Figure 6a, while neither Figure 6b nor Figure 6c mirror the inspection results.This is further explored in Section 6.4.

The results in Figure 7 show that the Jaccard coefficient gives the best measured clustering results for all strategies, while correlation gives the overall worst measured quality. These results do not correspond to the results found by cluster inspection, and indicates that the Dunn index is probably not sensible when the calculation is based on distances found by use of different distance measures. This problem is explored in Section 6.4.

5.1.2 Hierarchical clustering on PCA reduced data sets

A subset of 7000 patients was extracted from the full data set. Applying clustering on this data set gave results comparable to those given by clustering the full data set, which indicated that the smaller set hold the information contained in the full data set.

Feature selection The eigenvalues found during the PCA analysis are shown in Figure 8. This figure indicates several natural choices for the reduced number of features. The first eigenvalue is considerably larger than the number two values, which indicates that the first values holds a substantial part of the information.



(a) Results achieved by minimum distance

(b) Results achieved by maximum distance

(c) Results achieved by average distance

Figure 7: Dunn comparison of distance measures for the hierarchical clusterings

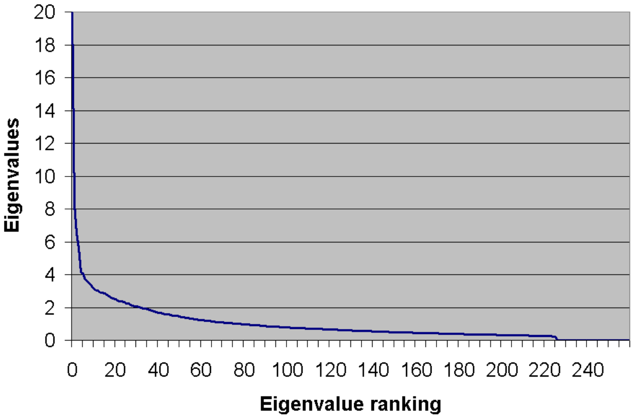


Figure 8: Eigenvalues found during PCA of original data set

The curve drops sharply until value 6, then the curve starts to flatten out. The curve starts to flatten out further from eigenvalue seven, which indicates that the vectors that correspond to the sixth highest eigenvalue hold considerably more significant information compared to the vector corresponding to the seventh highest eigenvalue. From eigenvalues 12 the eigenvalues starts decreasing even more linearly. At 226 there is an evident knee in the values, and values succeeding 226 are close to zero. An interesting question was whether clustering with 266 features would give almost identical results as clustering with 7000 features. 266, 12, 6 and 1 were therefore considered as interesting choices for the number of features in the PCA reduced data sets.

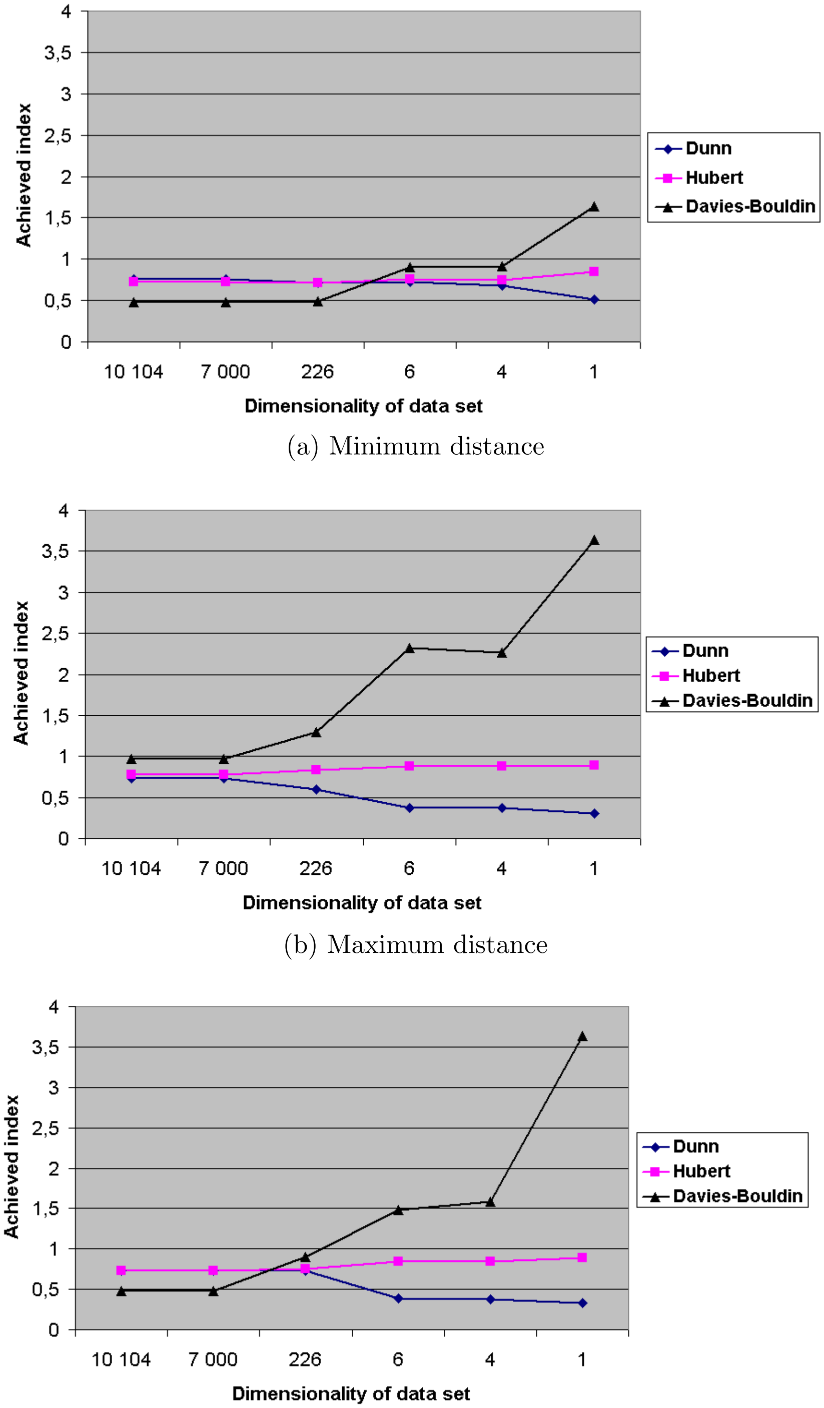
Clustering validation by inspection The results from the manual inspection of the clusterings are given in Table 3.

Clustering validation by quality measures The Dunn index, the Hubert Γ index and the Davies-Bouldin index for the PCA reduced clusterings are given in Figure 9. To reduce the amount of information only the indices for the size-16 clustering are shown. The reason for this choice is further explored in Section 5.1.3.

Comments on the results The results in Table 3 emphasis the problems faced in Section 5.1.1 associated with minimum distance. Both for the original data set and for all the PCA reduced sets only one cluster is growing. With maximum distance the clustering tendency increases for the reduced number of features and some of the clusters are conceptually meaningful. The clustering tendency

Strategy	**Features**	**Description of clustering**	**Ranking**
Minimum distance	7000	One cluster is growing, the other clusters are single objects.	6
	226	Identical to the result obtained for 7000 features.	6
	12	One cluster is growing, the other clusters are single objects except from a few 2-3 objects clusters.	5
	6	Same situation as observed for 12 features.	5
	1	One large cluster, several smaller clusters. Smaller clusters seem meaningless.	4
Maximum distance	7000	One large cluster, one small not sensible cluster, the rest of the clusters are single objects.	5
	226	Identical to the result obtained for 7000 features.	5
	12	One large cluster, a few smaller clusters which includes both sensible and not sensible ones. More single objects than observed with 6 or 4 features.	2
	6	One large cluster, a few clusters of size 10-30, the rest of the clusters are small (1-3). Some of the medium size clusters seem sensible.	**1**
	1	Several large clusters. However, the clusters do not have any conceptual meaning.	4
Average distance	7000	One large cluster, the remaining clusters are single objects.	6
	226	Same situation as observed for 7000 clusters.	6
	12	Similar situation as observed for 7000 clusters, some smaller clusters.	5
	6	One large cluster, several smaller clusters which are both sensible and not sensible.	3
	1	Several large clusters, but the clusters seem meaningless.	4

Table 3: Inspection of hierarchical clustering of PCA reduced data set



(a) Minimum distance

(b) Maximum distance

(c) Average distance

Figure 9: Quality indices for hierarchical clusterings size-16 of PCA reduced sets

for average distance is slightly less than for maximum distance, and the achieved clusters are less meaningful.

These results are not mirrored in Figure 9. The Dunn indices and the Hubert Γ indices are similar for all three merge strategies. The Dunn index indicates an insignificant decrease in clustering quality when the number of features goes down, while the Hubert Γ index in that case indicates an insignificant increase in the quality. Contrary, the Davies-Bouldin index shows an important decrease in clustering quality when the number of features goes down. This is further explored in Section 6.4.

5.1.3 K-means clustering on PCA reduced data sets

Based on the previous clustering results, 16 was chosen as a sensible number of clusters. When the number of clusters was 16, most clusters represented one and only one concept. For clusterings that contained more than 16 clusters, several clusters represented the same idea and a merge would have been preferred. For clusterings that contained less than 16 clusters, several clusters were too large to represent one single concept. Both these problems existed also in the size-16 clustering. However,the size-16 clustering were considered to best counterbalance the two weaknesses. K-means was then applied on the 10 104 data set, the 7 000 data set and on the PCA-reduced sets of size 226, 12, 6, 4 and 1.

Clustering validation by inspection The results from the inspection of the k-means clusterings are given in Table 4.

Clustering validation by quality measures The results from calculating the Dunn index, the Hubert Γ index and the Davies-Bouldin index for the clusterings are given in Figure 10.

Comments on the results The results given in Table 4 indicates that the clustering tendency increases when the number of features decreases. The meaningfulness of the clustering increases when the number of features decreases. However, with too few features a significant amount of the information is lost, and the clusters seem meaningless. The most sensible clustering is achieved by use of 6 features.

The results in Figure 10 do not reflect these trends. The Dunn index indicates a

Data set	Iterations	Description of clustering	Ranking
10104	6	One large cluster. Some smaller clusters with no perceived concept.	4
7000	6	One large cluster. Some smaller clusters with no perceived concept.	4
226	9	Similar to 10104 and 7000 result.	4
12	7	Several medium size (5-20) clusters, some of them with a clear concept. Conceptual clusters contain outsiders.	2
6	29	Several medium size clusters, most of them with a clear concept. Conceptual clusters contains fewer outliers than observed with 12 features. Some larger (30+) clusters cover more than one concept.	**1**
1	7	Clusters are of similar size but the clustering seems random.	3

Table 4: Inspection of k-means clustering of the PCA reduced data set

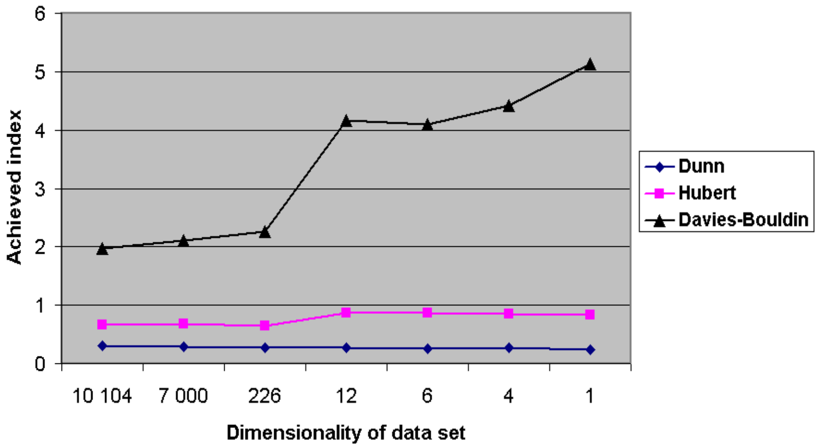


Figure 10: Quality indices for the k-means clusterings of PCA reduced sets

slightly decrease in quality and the Davies-Bouldin index indicates a significant decrease in quality when the number of features decrease. The Hubert Γ index indicates a slightly increased quality when the number of features decrease.

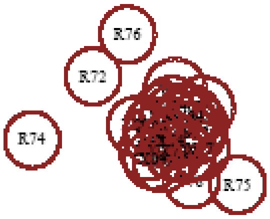
5.1.4 Conceptual description of the clusters

From all the clustering results achieved, the clustering that best indicated established medical knowledge of correlation between ICPC codes was the one achieved by use of the full 10104 data set, hierarchical clustering with maximum distance merge strategy and lift correlation as distance measure. As explained in Section 5.1.3, the number of clusters that best reflected the medical concepts was 16. The clustering achieved by use of these parameters is explored in this subsection.

For each pair of clusters, the average distance between the objects in the two clusters was calculated. According to theses average distances,an approximation of the cluster distribution were constructed by use of the GraphViz spring model described in 3.2. Figure 11 displays this approximation. The size of each node reflects the size of the cluster it represents, while the labels indicates the conceptual meaning of the clustering.

A GraphViz approximation of the distribution of objects was also constructed for the single clusters. However, it is harder to preserve the correct all-to-all distances when the number of objects, and consequently the number of distances, increases. The objects therefore seem more scattered then when they belong to a small cluster.

The mean squared distance (MSD) for all objects in this clustering was 1.1836. The descriptions of the clusters are ordered according to increased internal mean squared distance(MSD) for the single cluster.



Cluster 1: Female diseases
Size: 25
MSD: 0.0966
ICPC codes: P78 R72 R74 R75 R76 W01 W03 W10 W11 W12 W14 W301 W78 W84 W94 X01 X02 X06 X07 X08 X14 X17 X72 X74 X84

Most of the codes in this cluster are from the main groups W, which covers pregnancy, childbirth and family planning, and group X, which covers the female genital system. The cluster also contains some codes from the main group R which treats the respiratory system. The distribution figure indicates that the

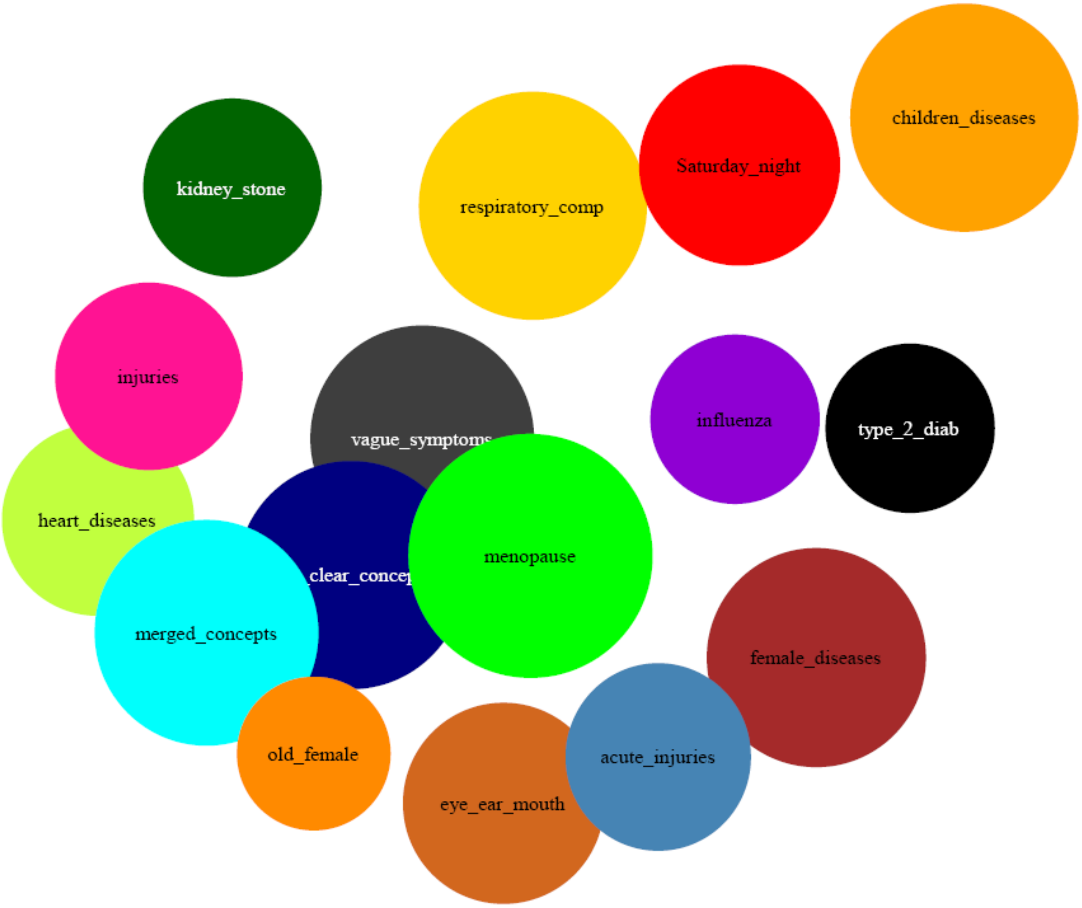
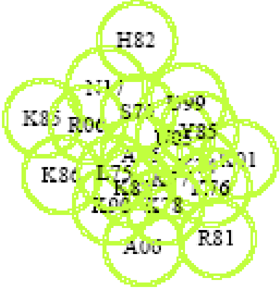


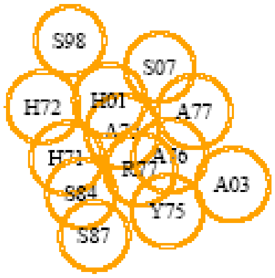
Figure 11: The most significant clustering distributed according to intracluster distance

codes from the W and X groups are strongly correlated, while the codes from the R group are more peripheral.



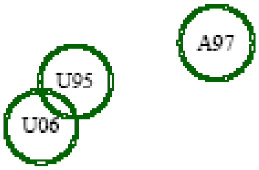
Cluster 2: Heart diseases
Size: 20
MSD: 0.1071
ICPC codes: A06 A96 H82 K01 K74 K76 K77 K78 K85 K86 K89 K90 L75 N17 R06 R81 S70 U05 U99 Y85

This cluster is dominated by codes from the main group K. The cluster emphasize the known correlation between hypertension and serious heart diseases such as ischaemic heart disease with and without angina and heart failure. The cluster also covers the code for death, which indicates that heart diseases are the kind of illness strongest correlated with death. Due to the fact that most patients with heart diseases are old people, this cluster also contains some codes denoting problems associated with advanced age. Such problems are for instance femur fracture and problems associated with the urinary tract.



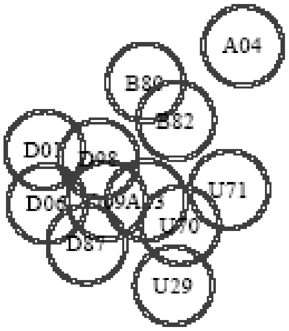
Cluster 3: Children's diseases
Size: 13
MSD: 0.1284
ICPC codes: A03 A72 A76 A77 H01 H71 H72 R77 S07 S84 S87 S98 Y75

This cluster contains codes for illness associated to children, such as Chickenpox, impetigo (milk blotch), inflammation of the ear and spasmodic croup. Fever is also contained in this group and indicates that fever is strongly correlated to children's diseases.



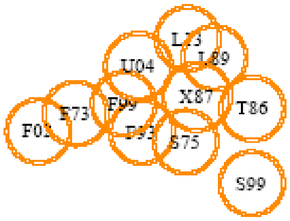
Cluster 4: Kidney stone
Size: 3
MSD: 0.1300
ICPC codes: A97 U06 U95

This very small cluster contains the codes for haematuria and kidney stone. The fact that these two problems are separated from the other codes can indicate a low degree of correlation between these codes and the other codes.



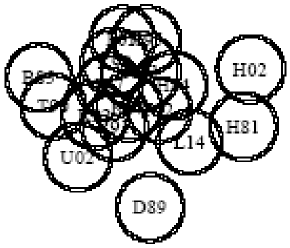
Cluster 5: Vague symptoms
Size: 12
MSD: 0.1444
ICPC codes: A04 A13 B80 B82 D01 D06 D09 D87 D98 U29 U70 U71

This cluster covers general and vague symptoms such as weakness/tiredness, abdominal pain and nausea. The cluster also contains anaemia, stomach function disorder and urinary complaints and diagnoses which are probable reasons for such symptoms.



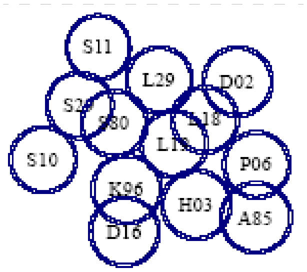
Cluster 6: Elderly female
Size: 11
MSD: 0.1551
ICPC codes: F02 F73 F93 F99 L13 L89 S75 S99 T86 U04 X87

The codes in this cluster denotes mainly chronic and less to moderately problems associated with elderly female (and to some degree male) patients. Examples are diverse eye complaints, hip complaints, skin complaints, incontinence, hypothyroidism and vaginal prolapse.



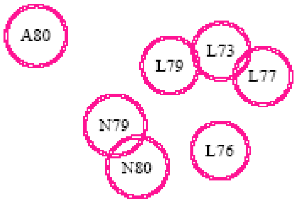
Cluster 7: Type 2 diabetes
Size: 17
MSD: 0.1931
ICPC codes: B85 D89 F05 F92 H02 H81 H84 K07 K92 K94 K95 L14 R91 S101 S97 T90 U02

Complaints associated to diabetes dominate this cluster. Such complaints are visual disturbance, leg/thigh symptoms, vascular disease, chronic ulcer skin and urinary urgency.



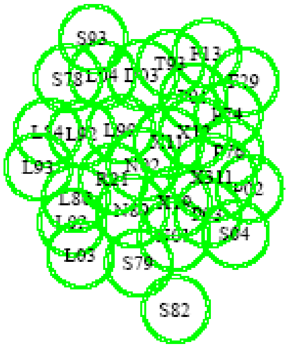
Cluster 8: No clear concept
Size: 13
MSD: 0.2203
ICPC codes: A85 D02 D16 H03 K96 L18 L19 L29 P06 S10 S11 S29 S80

This cluster does not seem conceptually meaningful. The cluster contains codes for tinnitus, hemorrhoids, musculoskeletal complaints, skin infections among others. The cluster is not dominated by any type of codes.



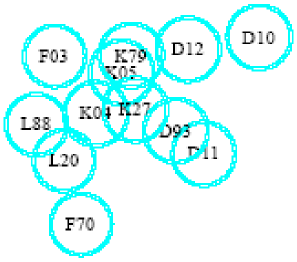
Cluster 9: Injuries
Size: 7
MSD: 0.2315
ICPC codes: A80 L73 L76 L77 L79 N79 N80

This cluster covers injuries as a consequence of accidents and contains fractures, strains, concussion and other head injuries.



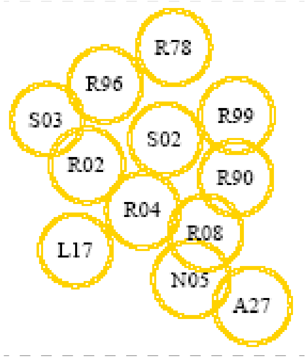
Cluster 10: Menopausal symptoms
Size: 30
MSD: 0.2319
ICPC codes: D03 F13 F29 L02 L03 L04 L84 L86 L92 L93 L99 N01 N02 N89 P01 P02 P03 P74 P76 R21 S04 S78 S79 S82 S93 T93 X11 X12 X19 X311

This cluster includes comparatively many codes for symptoms. Examples are heartburn, headache, feeling anxious, feeling depressed and symptoms and complaints associated with eye, back, chest and throat. Two potential relations are indicated by this cluster. Firstly, the cluster contains codes for anxiety disorder and depressive disorder. Patients with these diagnoses will possibly feel physical symptoms, which explains the high incidence of symptoms. Secondly, this cluster includes the code for menopausal symptom, which indicates the relation between diverse symptoms/complaints, depressive and anxiety disorder and the climacteric.



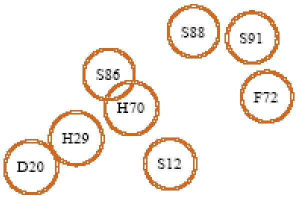
Cluster 11: Merged concepts
Size: 12
MSD: 0.2413
ICPC codes: D10 D11 D12 D93 F03 F70 K04 K05 K27 K79 L20 L88

This cluster represents several concepts and is probably a result of some unfortunate merges. Some codes denote digestive problems such as vomiting, diarrhoea and constipation, some codes covers irregular heart beat conditions and some codes denote eye problems. Surprisingly, the code for rheumatoid arthritis, which is evidently strongly correlated to musculoskeletal symptoms/complaints, is grouped in this cluster.



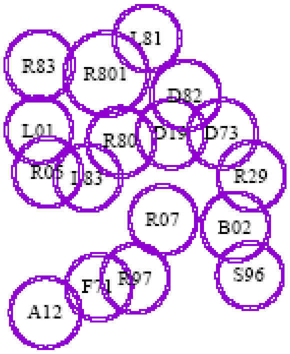
Cluster 12: Respiratory complaints
Size: 12
MSD: 0.2655
ICPC codes: A27 L17 N05 R02 R04 R08 R78 R90 R96 R99 S02 S03

Complaints and diseases in the respiratory system dominate this cluster. The cluster also contains a few outlier codes that apparently do not share any conceptual meaning with the rest of the cluster, such as foot/toe complaints.



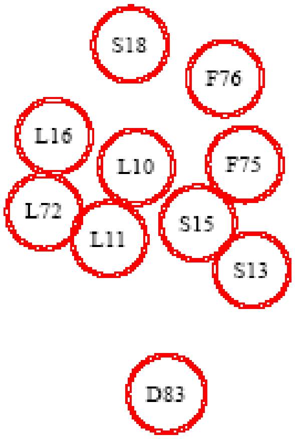
Cluster 13: Eye, ear, mouth
Size: 8
MSD: 0.2913
ICPC codes: D20 F72 H29 H70 S12 S86 S88 S91

The concept of this cluster is vague. The codes denote allergic or inflammation conditions associated to eye, ear, mouth and skin. The cluster includes insect bite which could indicate a correlation between insect bite and allergic reactions.



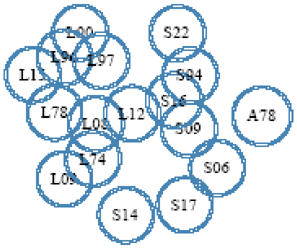
Cluster 14: Influenza
Size: 17
MSD: 0.2918
ICPC codes: A12 B02 D19 D73 D82 F71 L01 L81 L83 R05 R07 R29 R80 R801 R83 R97 S96

Complaints associated with influenza dominate this cluster. Such complaints are coughing, sneezing, nasal congestion, musculoskeletal symptoms, vomiting and enlarged lymph glands. The cluster also contains some outliers such as teeth/gum disease and acne.



Cluster 15: Saturday night injuries
Size: 10
MSD: 0.3050
ICPC codes: D83 F75 F76 L10 L11 L16 L72 S13 S15 S18

This cluster includes injuries typically arising as a result of drunk persons fighting. The cluster covers cuts and foreign bodies in eye, mouth and skin, animal/human bites and diverse fractures and complaints in the elbow, wrist, ankle and forearm.



Cluster 16: Acute injuries
Size: 17
MSD: 0.3194
ICPC codes: A78 L08 L09 L12 L15 L74 L78 L90 L96 L97 S06 S09 S14 S16 S17 S22 S94

This cluster has some similarities with the Saturday night cluster in that this cluster is also dominated by acute injuries such as fractures, strains, burns, bruises, contusions and blisters. However, this cluster contains codes only for conditions related to muscles, skeleton and skin. The cluster also includes some chronical complaints such as osteoarthrosis of the knee.

5.2 Clustering task 2: Medical certificates

This section contains the results from the clustering of medical certificates. Section 5.2.1 presents the results from the comparison of code occurrences for temporary and long-term medical certificates. Section 5.2.2 presents the results from the hierarchical clustering procedure.

The number of patients represented in both groups after completed preprocessing was 1314.

5.2.1 Counting

The percentual distribution of the occurrences of codes for temporary and long-term medical certificates per main ICPC code-group is shown in Figure 12.

The figure reveals some great variances between the temporary and the long-term medical certificates. The number of long-term certificates prescribed for musculoskeletal illness (ICPC main group L) is 21% larger compared to the number of prescribed temporary certificates. The L group constitute respectively 43% and 53% of the certificates, which means that a total of 10% of the temporary certificates not for musculoskeletal illness have the code changed to L when the long-term certificate is written. The number of certificates prescribed for psychological illness (ICPC main group P) increases 63% from the temporary group to the long-term group. This growth corresponds to 6% of the total number of certificates. For circulatory illness (ICPC main group K), the increase is 50% from temporary to long term certificates, which corresponds to 3% of the total number of certificates.

Conversely, the number of certificates prescribed for respiratory illnesses decreases from 18% of the total number of certificates to less than an eighth of the initial share, namely 2% of the total. The number of certificates prescribed for digestive illness decrease 50% from temporary to long term illness. There are also smaller variances in the other code-groups as shown in the figure.

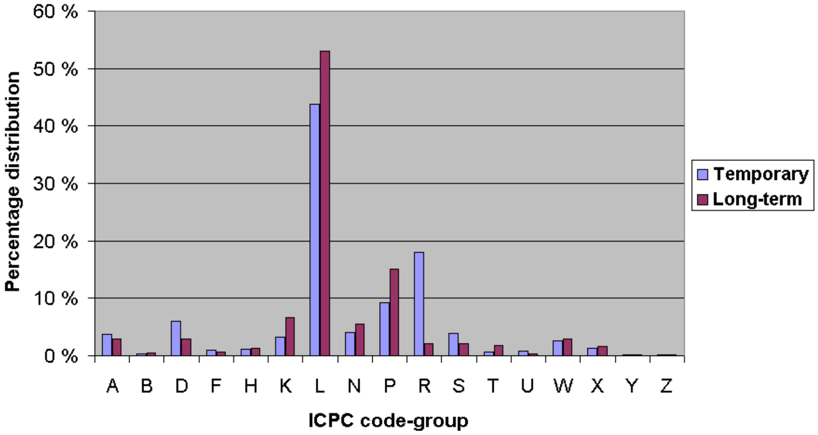


Figure 12: Percentual occurrence of main code-groups

The occurrences of each code were also counted for both groups of certificates. To find significant variances, only codes that occurred with at least 0.5% of the certificates in both groups were taken into consideration. Among these, the twelve codes with the greatest variance were extracted. These codes are displayed in Figure 13.

The most striking variance is connected to the use of the code P76, which denotes depressive disorder. The frequency for the use of this code increases 134% from temporary to long-term certificates. The growth in the use of code P78 (tiredness) is 132%. There is a decrease in the use of code P02, but this decrease does not outweigh the increased use of P76 and P78. This indicates that psychological ill patients probably often get their temporary medical certificates prescribed for non-psychiatry-related complaints.

Regarding the use of musculoskeletal codes there is an increase of about 100% for both L86 (back syndrome with radiating pain), L97 (benign/unspecified musculoskeletal tumor) and L99 (other musculoskeletal disease) from temporary to long-term certificates.

L81 (musculoskeletal injury) and P02 (acute stress reaction) are the only two codes which decrease significantly from temporary to long-term use. Not surprisingly, both these codes denote acute situations.

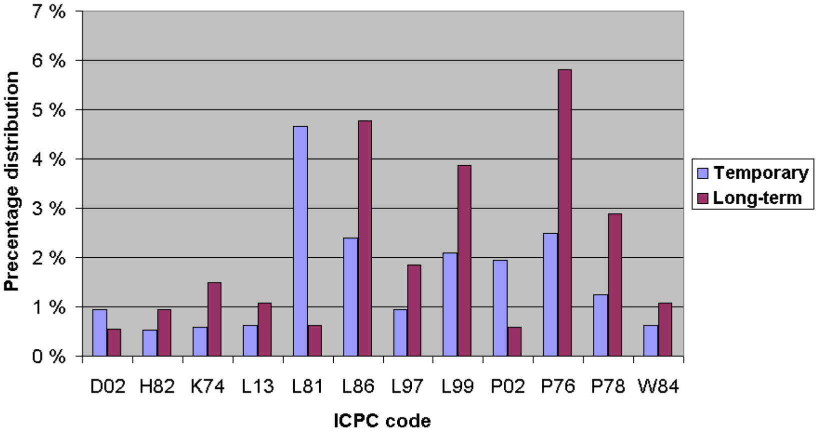


Figure 13: Percentual occurrence of single codes

5.2.2 Hierarchical clustering

According to the plan explored in Section 4.2.3, the first step was to compare the two normalisation methods described in Section 2.4.2. To avoid the values obtained by z-score normalisation to dominate the other features, the results from the z-score normalisation were multiplied by 0.7 to reduce the range of the values. The clusterings achieved from these step indicated that min-max normalisation in higher degree than the z-score normalisation gave clusters with similar mean age. This finding indicated that min-max normalisation caused the age to influence the clustering minimal. The range and standard deviation for the values after normalisation where therefore calculated for both normalisation strategies. The results from this calculation are given in Table 5.

The relatively high standard deviation for z-score normalisation indicates that

method	**data set**	**min**	**max**	**mean**	**std dev**
original	temporary	17	102	44.23	12.14
original	long-term	18	99	45.91	12.48
linear	temporary	0	1	0.32	0.14
linear	long-term	0	1	0.34	0.15
z-score	temporary	-1.57	3.33	0	0.7
z-score	long-term	-1.57	2.98	0	0.7

Table 5: Results from normalisation

this method manages to separate the values, while the linear normalisation suffered from outlier values and therefore did not separate well the inlier values. The z-score normalisation was therefore, based both on the clustering results and on the results from the calculation, considered best suited for the task.

The next step was to select a strategy for missing value replacement. The clusters obtained when missing values were replaced by 1 were all dominated by married, employed patients, which was the attribute values that 1 denoted. Contrary, when missing values were replaced by 0 clusters were formed which were dominated by patients with different types of marital and occupational status. This is probably due to the relatively high fraction of missing values in the original data set; the shares of missing values for temporary certificates were respectively 10% for marital status and 9% for occupational status, while the corresponding values for long-term certificates were both 12%. The insertion of 1's causes a substantial growth of the already dominating value 1, and causes the groups of certificates that contain other values to be relatively smaller compared to the 1 groups. The 1's will therefore dominate all clusters in a clustering as long as the clusters are not of extremely variable size. 0 was therefore chosen as the default value for replacement of missing values.

The z-score normalisation and the 0 replacement strategy were then used in the further clustering procedure. The data sets were clustered according to the plan described in Table 1, and the results were inspected.

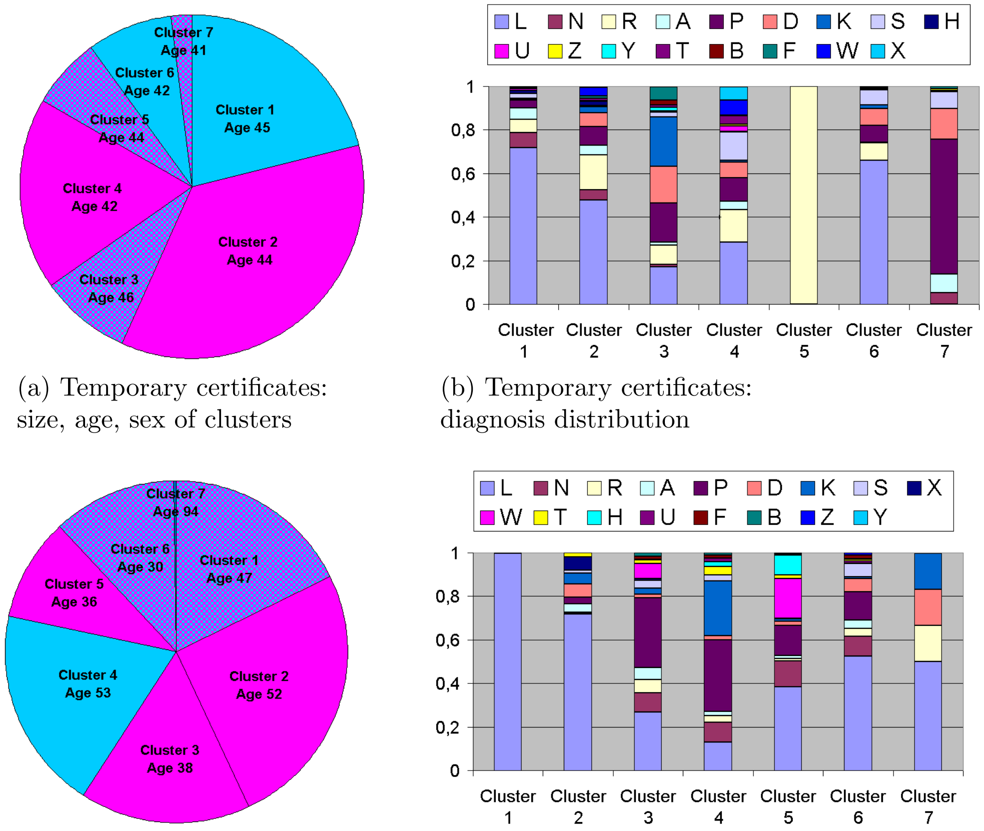
Generally, the clusters did not seem to reflect interesting concepts. When the basis weighting was used, the clusters were conceptually similar. The average age was similar for each cluster, the standard deviation of the age was similar, the compositions of marital and occupational status were similar, and most groups contained a fraction of the musculoskeletal patients and small fractions of several other groups. When variations are made in the weighting, the resulting clusters were dominated by the attributes corresponding to the heaviest weighting. For instance, emphasising the codes or the code-groups results in clusters representing usually only one, and only seldom a few code-groups. The situation for the other attributes in these code-group clusters are similar to what was observed for basis weighted attributes; the attributes are equally distributed among the clusters.

To illustrate this problem, some of the achieved results will be explored in the following. The marital and occupational status were equally distributed in all the clusters and are therefore omitted from the figures to reduce the amount of information. Figure 14 shows the results from the size-7 clustering achieved by use of maximum distance merge strategy and basis weighting. Figure 14a shows the average age, the sex composition(pink/blue) and the fraction of certificates grouped in each cluster for the temporary certificates. As indicated by the figure, the clustering is composed of three large clusters and four smaller clusters. Two clusters are pure male clusters, two clusters are pure female and the remaining three clusters are of mixed sex. The average age is almost equal for all clusters. The distribution of codes for each cluster is given in Figure 14b. The code distribution indicates that diagnoses are distributed among the clusters and that most clusters are compounded from so many dissimilar codes that the probability of representing a concept is low.

The corresponding visualisation of the results obtained for the long-term certificates is given in Figure 14c and 14d. Figure 14c indicates clusters of more similar size except from cluster 7 which contains a negligible amount of the certificates. The average age varies among the clusters. The sex distribution is similar to the situation for the temporary certificates; two clusters are pure male, three are pure female and two are mixed. The code distribution shown in Figure 14d demonstrates compounded clusters in this case as well. As indicated by the results, these clusterings do not reveal any special conceptual ideas.

To explore the situation that arises when some features are more heavily weighted than the other, the size-7 clusterings achieved by use of maximum distance merge strategy and 1.5 weighting of code and code-group are shown in Figure 15. For these clusterings, all clusters were of mixed sex. The sex coding was therefore omitted in the visualisation of the results.

For both the temporary and the long-term certificates there are a few large clusters that dominate the remaining smaller clusters. This is probably due to the emphasis on code and code-group which causes the large group of musculoskeletal patients, and also the relatively large group of psychological patients to be grouped into single clusters. Both the average age and the male/female ratio vary less from cluster to cluster in these results compared to the results achieved by



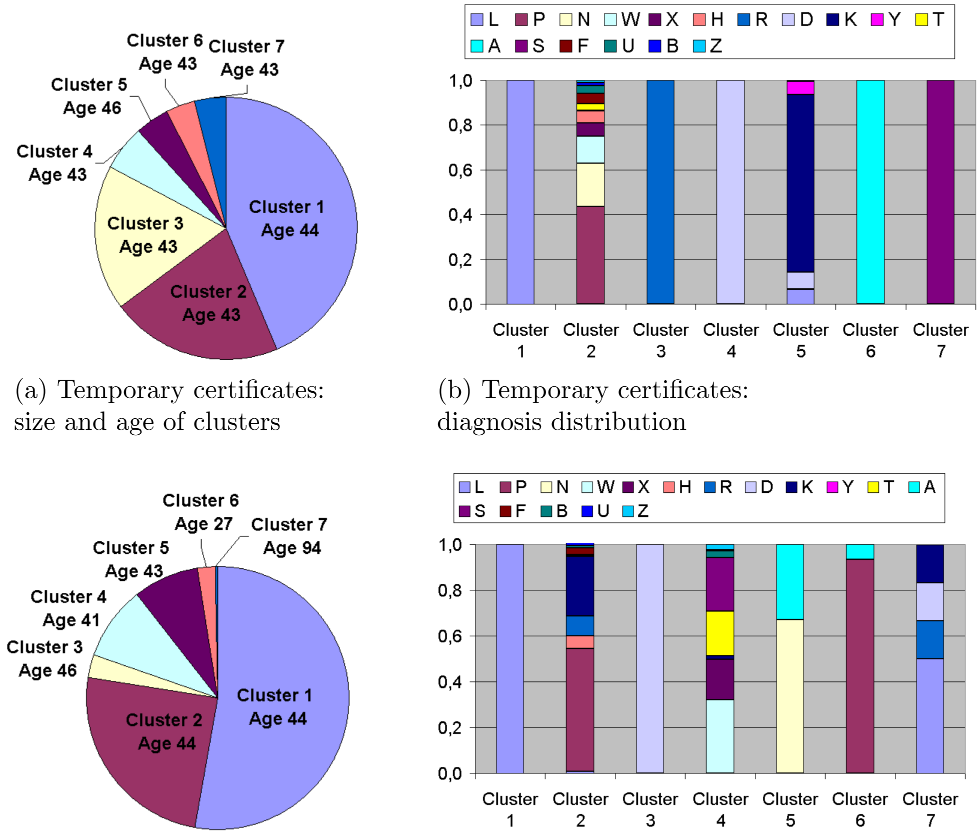
(a) Temporary certificates: size, age, sex of clusters

(b) Temporary certificates: diagnosis distribution

(c) Long-term certificates: size, age, sex of clusters

(d) Long-term certificates: diagnosis distribution

Figure 14: Results from clustering of medical certificates with basis weighting



(a) Temporary certificates: size and age of clusters

(b) Temporary certificates: diagnosis distribution

(c) Long-term certificates: size and age

(d) Long-term certificates: diagnosis distribution

Figure 15: Results from clustering of medical certificates with 1.5 weighting of code and code-group

basis weighting. On the other hand, the code-groups are less distributed among the clusters. This illustrates the observed difficulties associated to this clustering task; either the attributes balance each other and cause equally composed clusters, or some attributes dominate and minimize the influence from the resisting attributes.

As mentioned, these results were obtained by use of the maximum distance strategy. The use of minimum distance strategy caused the same problems as was observed for minimum distance clustering of ICPC codes; only one cluster was growing. For the average distance strategy, clusters of dissimilar size were formed. This caused the clusterings to contain some very small outsider groups with a clear concept, which could potentially be of greater interest than the larger clusters. Such clusters were for instance a group of male students with social diagnoses and a group of 60 year old male recipients of national insurance benefits reported sick for psychological ailments.

Several hypotheses can be made that attempt to explain the mainly uninteresting results from this clustering task. These hypotheses, and also suggestions of how to overcome the difficulties associated to each hypothesis, are explored in Section 6.6.

6 Discussion

The aim of this section is to explore, explain and discuss discoveries or unexpected results obtained through the accomplishment of the clustering tasks. Section 6.1 deals with the fitness of the selected methods. Section 6.2 discusses characteristics of the distance measures. Section 6.3 treats the merge strategies. The quality measures are discussed in Section 6.4. Section 6.5 consider the problem regarding clustering of high dimensional data and the fitness of the PCA method for this work. Finally, Section 6.6 discusses the clustering of medical certificates.

6.1 Regarding the choice of algorithms and methods

The choice of which clustering algorithms, distance measures and quality measures to implement was taken in advance of defining the clustering tasks, based on presumptions of how the clustering tasks would be designed. The separation of these two tasks was undoubtedly disadvantageous. Restricted knowledge of how the clustering task should be designed caused some incorrect assumptions.

Firstly, the data sets to cluster were believed to be smaller than they became. The assumption was that the clustering tasks should involve subgroups of patients. An example of such a group, which was emphasised by the supervisors and available GPs at NSEP, was the group of rheumatoid patients. However, the most interesting subgroups turned out to be too small to be used as foundation for stating credible medical hypotheses. A consequence of this was that during the implementation phase emphasis was placed on making the code easily read instead of making the program efficient. This caused much time to be spent later to optimise the code due to poor performance.

Secondly, previous knowledge of the Euclidian distance, the Manhattan distance and the Minowski distance caused the assumption that the distance between two objects would be calculated by a pairwise comparison of two corresponding feature vectors. This way of calculating distance would also render possible the calculation of defining for a group of objects a mean object which possesses the mean value for each feature. The choice of the k-means method, for which the calculation of mean objects is required, was based on this assumption. However, when the Jaccard coefficient and the Lift correlation turned out to give best results the use of k-means, which does not work in combination with these distance measures, becomes less reasonable. A possible substitute method for the k-means algorithm could be the k-medoid method, which bases the clustering on the calculation of medoid objects instead of mean object. However, due to the arbitrarily selection and testing of medoid objects the k-medoid method is known to be costly. Because of the combination of large data sets and costly computation the k-medoid method was not tested.

The lack of possibility to calculate mean objects also affected the utilisation of the implemented quality measures. Both the Hubert Γ index and the Davies Bouldin index require the calculation of mean objects, and could therefore not be calculated for the clusterings obtained by the Jaccard coefficient or the Lift correlation.

Another feature of the data sets which was unknown when the methods were chosen was the problem regarding the high dimension of the data, discussed in Section 6.5. When PCA reduction was performed to the data sets, the feature values were altered from binary values to interval-scaled values. Therefore, neither the Jaccard coefficient nor the Lift correlation worked for the PCA reduced data sets, which caused the reduced sets to be analysed only by the Euclidian distance. Even though the Euclidian distance was the measure that suffered most from the dimensionality problem, it would potentially have been an interesting task to cluster the reduced data sets not only by the Euclidian distance but also by the other distance measures.

6.2 Performance of the distance measures

This thesis performs a comparison between the results obtained by the three distance measures Lift correlation, Jaccard coefficient and the Euclidian distance for high dimensional, binary vectors. As explored in Section 5.1.1 the most sensible results were achieved by use of correlation, tight followed by the results achieved by Jaccard. The results achieved by the Euclidian distance were substantially worse than in the two other cases. This section is an attempt to explore and explain the results obtained in Section 5.1.1.

6.2.1 Fitness of the distance measures for binary vectors

The binary vectors used for the ICPC clustering have the characteristic that they are asymmetric, which means that the value 1 is considered more significant than the value 0 for a feature. This can easily be explained by the fact that the presence of a diagnosis for a patient is more significant than the absence of that diagnosis, since most patients lacks most diagnoses. A binary vector for an ICPC code that holds one value for each patient will therefore mainly contain 0's. An advantage with the Lift correlation and the Jaccard coefficient is that they emphasis the binary features of value 1 over the features of value 0. Instead of performing a pairwise comparison of the features in each vector such as most traditional distance measures like the Euclidian distance, these two measures counts the occurrences of 1's in common and for each of the two compared objects.

6.2.2 A comparison of the Lift correlation and the Jaccard coefficient

The Lift correlation and the Jaccard coefficient separate only in the way they combine the counted occurrences. As given in Equation 6 the correlation is calculated as

$$\frac{patwithcode1 * patwithcode2}{patwithbothcodes * featurestotally}$$

while the Jaccard coefficient, given in Equation 5, is calculated as

$$\frac{patwithcode1butnotcode2 + patwithcode2butnotcode1}{patwithcode1butnotcode2 + patwithcode2butnotcode1 + patwithbothcodes}$$

For both measures, a comparison between two objects without any shared attribute results in the greatest possible dissimilarity. Also, the dissimilarity decreases when the number of shared attribute values increases.

However, as mentioned in 5.1.1, the best results were achieved by use of Lift correlation. Why does correlation perform better than Jaccard? The difference in performance can probably be due to the fact that the probability of sharing the value 1 for a feature increases when the two compared vectors contains an equal number of 1's, compared with the situation where the same number of 1's is unequally distributed among the two vectors. This is illustrated by the two situations given in the following example:

Situation 1: Consider two vectors of length 4 which each contains 2 0's and 2 1's. The probability p_s that the two vectors share the value 1 for at least one feature equals 1 minus the probability p_n that the vectors do not share the value 1 for any feature. The number of combinations of values for one vector is the number of possible combinations of four elements divided by the number of possible combinations of each subgroup of equal values, which gives a total number of

$$\frac{4!}{2!2!} = 6$$

combinations of two 1's and two 0's. The total number of combinations of the two vectors is then

$$\frac{4!}{2!2!}\frac{4!}{2!2!} = 36$$

The number of combinations where no feature share the value 1 equals the number of possible combinations of the values in one single vector, since for each combination the inverse combination in the other vector will cause no shared elements. p_s is then given by

$$p_s = 1 - p_n = 1 - \frac{6}{\frac{4!}{2!2!}\frac{4!}{2!2!}} = \frac{5}{6}$$

Situation 2: Consider another situation with two vectors of length 4, where one vector contains one 1 and 3 0's, while the other vector contains 3 1's and one 0. Notice that the total number of 1's is the same in both situations. For each vector there are evidently 4 possible ways to combine the values, which gives a total number of combinations of the two vectors of $4 * 4$. As was the situation in the previous example, the number of combinations for which no feature shares the value 1 equals the number of possible combinations for a vector which was 4. Then, the probability p_s is given by

$$p_s = 1 - p_n = 1 - \frac{4}{4^2} = \frac{3}{4}$$

The correlation measure takes this variation in probability into consideration. By multiplying the numbers of occurrences of the two codes, the more equal the numbers, the greater the numerator. To equalize this increase, a greater number of common attributes is necessary to increase the denominator. The Jaccard coefficient does not take this into account. Objects with an equally distributed number of 1's will therefore generally achieve a smaller dissimilarity than objects with an unequally distribution of 1's due to the increased probability of shared attribute values.

6.2.3 Weaknesses of applying the Euclidian distance to binary vectors

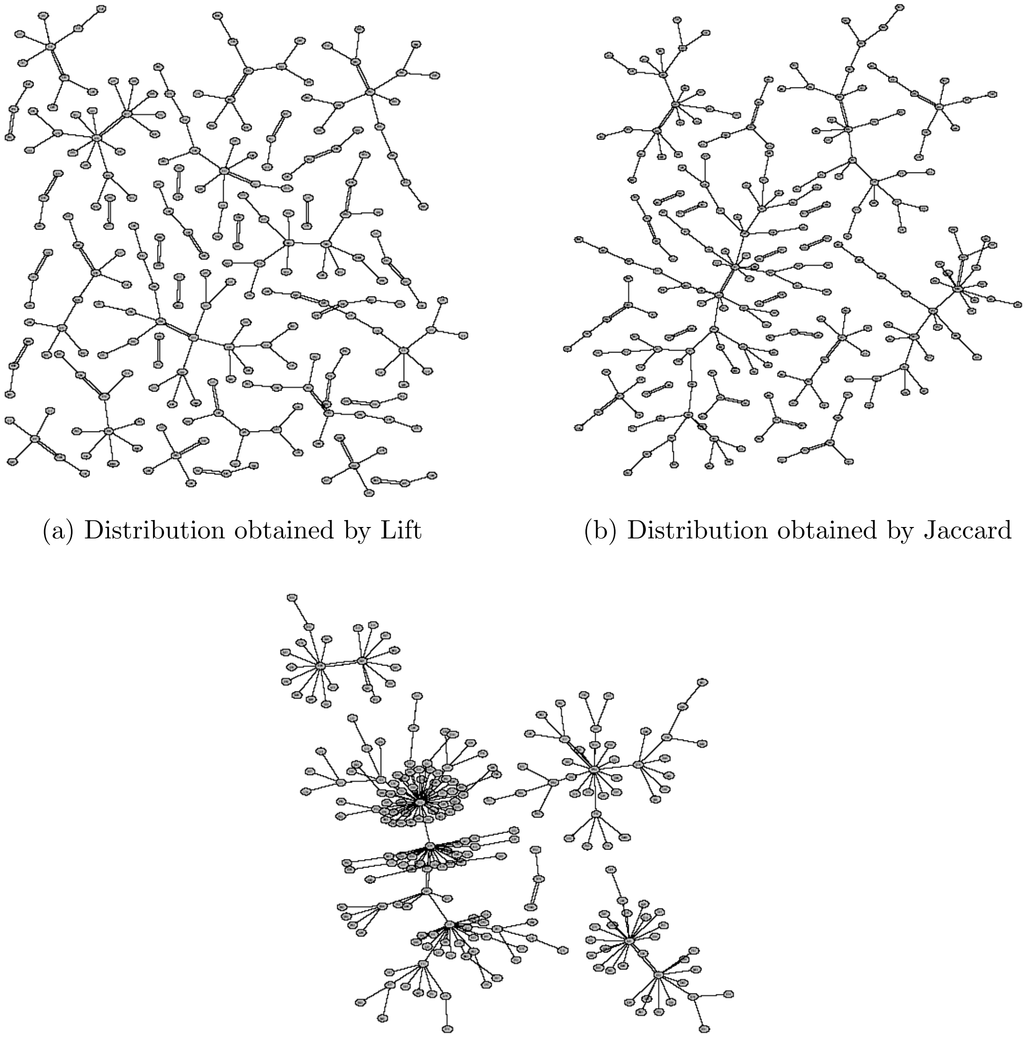
As mentioned, the Euclidian distance performs a pairwise comparison between each element in the two compared feature vectors. This causes several problems when the Euclidian distance is used to compare asymmetric binary variables. The Euclidian distance induces no difference between two elements of value 0 and two elements of value 1; both cases involve a dissimilarity of 0, even if the occurrence of two elements of value 1 represents a much more significant finding than that of two 0's. As a consequence of this, two objects who have numerous 1's in common are considered of equivalent similarity as two objects who has zero or a few 1's in common, if the number of features with shared value is the same. In the extreme case this will result in a measured dissimilarity of zero for two objects with no 1's in common. Contrary, for both the correlation and the Jaccard coefficient, the dissimilarity decreases when the number of 1's in common increases.

The Euclidian distance also causes an increased dissimilarity if the number of 1's is large in at least one of the two compared vectors, because in that case the probability increases that a feature is represented by a 0 in one vector and a 1 in the other. For clustering purposes this increased distance will often not be adequate, since it could be desirable that two codes which are widespread among the patients, or two patients who have unusually many diagnoses, are considered similar.

6.2.4 Distance measures and object distribution

How does the distance measure influence on the object distribution, and in what way does the distribution of objects affect the clustering results? To answer the first question, the distribution of data objects obtained by use of the three distance measures was attempted visualised by the GraphViz spring model described in Section 3.2. Figure 16 displays all the ICPC codes with a link to the code's nearest neighbor. From the figure it is evident that the distribution of data objects are highly related to the distance measure used. Figures 16a and 16b show the distribution by use of correlation and Jaccard coefficient respectively. These distributions are similar in that no objects seems to be the center for a clustering, most objects have from one to four relations to other objects. From the figures it seems probable that the objects will be equally distributed to any desired number of clusters. Figure 16c shows the distribution of objects attained by use of the Euclidian distance. Most of the objects are centered around a few central objects, such that as many as 40-50 objects share the same objects as their nearest neighbor. Obviously, the objects that share a common nearest neighbor will merge into the same cluster one by one. This is exactly the behavior observed through the clustering inspection in Section 5.1.1. Because the number of centre objects is small, and most objects belong to the same centre object, the clustering tendency achieved by use of the Euclidian distance is poor.

Figure 17 shows the 300 least distances and the objects connected by these 300 distances for each of the three distance measures. Figures 17a and 17b indicate one group of tightly connected objects, namely the group of W and X ICPC codes, and numerous less tightly connected objects scattered around. The Figure 17c on the other hand indicates that the 300 strongest connections are all between objects in the W and X ICPC groups, which again indicates less strong connections between the other objects and a poorer clustering tendency.

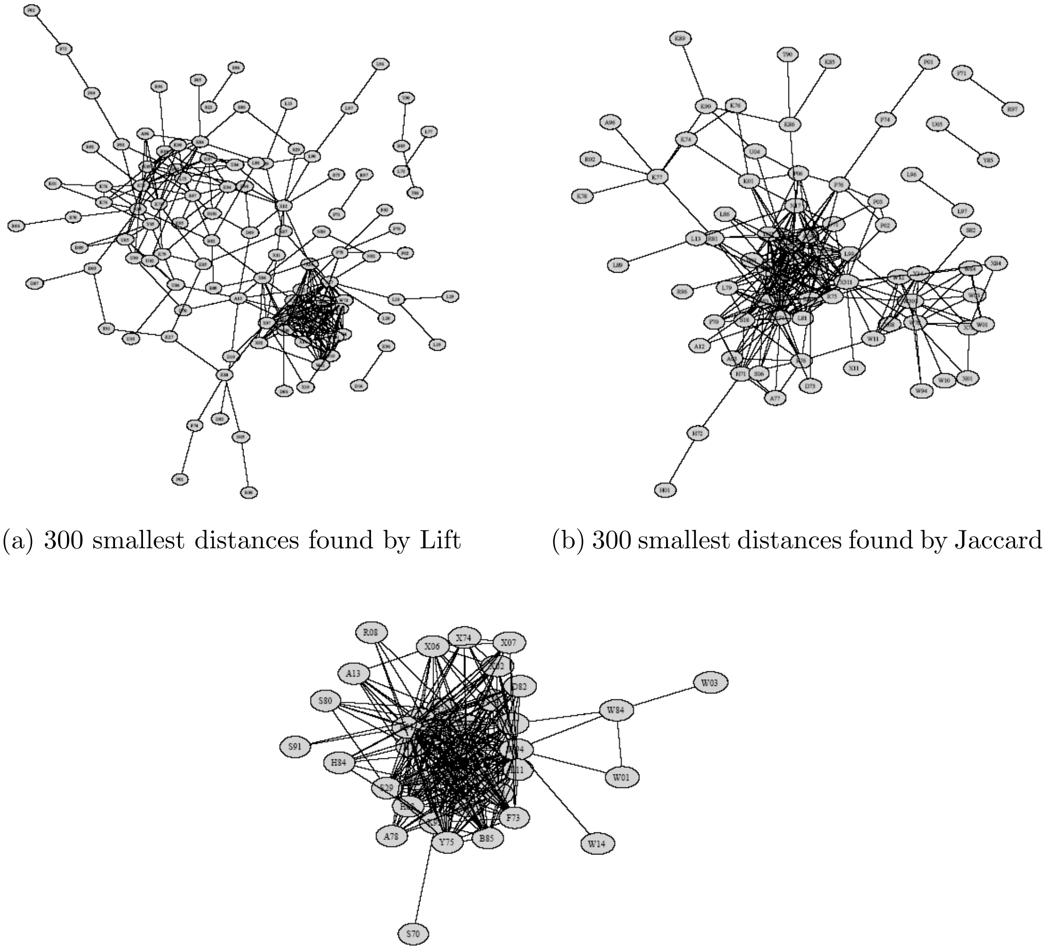


(a) Distribution obtained by Lift

(b) Distribution obtained by Jaccard

(c) Distribution obtained by Euclid

Figure 16: Each ICPC code linked to its most similar code



(a) 300 smallest distances found by Lift

(b) 300 smallest distances found by Jaccard

(c) 300 smallest distances found by Euclid

Figure 17: The 300 smallest distances including codes connected by the distances

6.3 Merge strategies

The comparison of merge strategies performed in Section 5.1.1 concluded that maximum distance gave the most sensible clustering results and the most similar size clusters. The use of average distance resulted in partially sensible clusters of variable size, where a substantial part of the clusters were small. The use of minimum distance caused all objects to merge one by one into the same cluster.

The results are intuitive and easy to explain. By use of the minimum distance strategy, the probability that an object's nearest neighbor is contained in a cluster increases when the size of that cluster increases. In the case of clustering ICPC codes, a predominance of the tightest connections belongs to the same two groups of codes, namely the W and X groups. Therefore, the minimum distance strategy will start the clustering by merging a substantial part of the X/W codes. This involves that only one cluster grows from the beginning. When this cluster increases, the probability that other objects will find their nearest neighbor into that cluster increases, which causes only that single cluster to grow. According to this behavior, the minimum distance strategy is likely to fail when the data set on which the clustering is applied contains a relatively tight connected subgroup of data objects. However, if elongated and well separated clusters are known to underlie the data set, the minimum distance strategy will potentially best identify these clusters.

When the maximum distance strategy is used, the probability that an object will merge into a specific cluster decreases when that cluster increases, because the average maximum distance between a random chosen external object and its most distant object in the cluster increases. This causes a cluster to stop growing and wait for other clusters to grow before it continues to expand. A potential drawback due to this behavior is that similar size clusters is preferred for tight clusters. This could possibly cause large clusters to be identified as two separate clusters or peripheral objects to be merged into small tight clusters.

The use of the average distance strategy could potentially avoid the drawbacks associated with both the minimum distance and the maximum distance strategy, by considering the tightness of a cluster when regulating the growth. According to this, the average distance strategy is supposed to best identify clusters in a data set of which we have no previous knowledge of neither clustering tendency nor the size or shape of the clusters. However, in the case of clustering ICPC codes the average distance strategy fails to identify the clusters. A probable reason for the shortcoming of the average distance strategy could be, as for the minimum distance strategy, the fact that all the smallest distances between objects belongs to the same group of codes. This causes a relatively large, very dense group. For external objects, the density of that cluster increases the probability of achieving a low average distance, and accordingly merge into the cluster.

6.4 Quality measures

The aim of this section is to suggest possible explanations to the results achieved in Section 5 regarding the measured clustering quality. The section discusses the usefulness of the quality measures tried out and of quality measures in general.

6.4.1 The main characteristics of the quality measures

As mentioned in Section 2.3, the three quality measures used in this work are founded on different characteristic of the clustering. The Dunn index and the Davies-Bouldin index both measure the ratio between the compactness of the clusters and the separation between clusters. These two measures separates in that the Davies-Bouldin index calculation utilizes the entire clustering structure, while the Dunn index calculation is based only upon at most 4 objects, namely the objects that constitute the minimum distance and the maximum diameter. The modified Hubert Γ index on the other hand measures the agreement between the proximity matrix and the clustering.

6.4.2 Regarding the measured quality of the hierarchical clusterings

This section discusses the quality indices achieved for the hierarchical clustering of the original data set given in Section 5.1.1. The section treats the index-based comparison of merge strategies and the index-based comparison of distance measures respectively.

The comparison of merge strategies Figure 6 displays the measured Dunn index for the results obtained by use of the three merge strategies. According to the Dunn index, only the results obtained by use of Lift correlation reflects the findings discovered by inspection. The results obtained by use of the Jaccard coefficient and the Euclidian distance do not mirror the results found by inspection. For the Jaccard coefficient, the minimum distance strategy gives the best results. This can be explained by intuition since the clustering criteria equals the numerator in the Dunn index. The Jaccard coefficient gives distances in the range 0-1. When the number of clusters goes down, the minimum distance will increase monotonically against 1. Also, the maximum diameter will always increase monotonically during an agglomerative hierarchical clustering process. The numerator for the last 50 clusterings would therefore be close to 1, while the denominator would also approach 1 but never exceed 1. This causes an index close to 1, like the one achieved for this case.

Also for the Jaccard/maximum distance strategy combination both the minimum distance and the maximum diameter will increase monotonically. However, only the maximum diameter will grow continuously. The growth of the minimum distance is unanticipated, which explains the results in the figure where the situation is that the minimum distance stay unaltered through 47 merges and then increases substantially.

Summarised: the minimum distance strategy guarantees a continuous growth in the numerator while the maximum distance strategy guarantees a continuous growth in the denominator.

For the Euclidian distance, neither the minimum distance strategy nor the average distance strategy managed to cluster the data set. The Dunn indices measured for these two strategies are similar and reflects the fact that when single objects are merged into the same large cluster one by one, the objects that are merged last are likely to have the largest distances to the objects in the cluster. The effect will therefore be similar to the effect achieved by use of the minimum distance strategy, where the minimum distance increases monotonically and therefore is relatively large for the last merge iterations. The roughness in the curves are probably due to erratic growth of the maximum diameter.

For the maximum distance strategy, the Euclidian distance manage to group a few smaller clusters in addition to the large one. The Dunn index curve is unstable. Since the maximum diameter is the clustering criteria this measure is supposed to increase monotonically. The curve reflects this theory in that the index decreases monotonically for all merges that do not cause a spring in the value. Due to the continuous growth in the maximum diameter, the springs must be induced by a huge increase in the minimum distance, as was the case for the combination Jaccard/maximum distance.

The comparison of distance measures Figure 7 shows an attempt to compare the three distance measures due to achieved Dunn indices. The Lift correlation, which gave the most sensible results according to the inspection, is the definitive loser in this comparison. The results shows clearly that a comparison of Dunn indices calculated by use of different distance measures is meaningless. This is attributed to the fact that the distances calculated by different measures are of very dissimilar range. The range for each of the distance measures are given in Table 6

Distance measure	Minimum distance	Maximum distance
Lift	0.056	2.15×10^9
Jaccard	0.5	1
Euclid	13.6	56

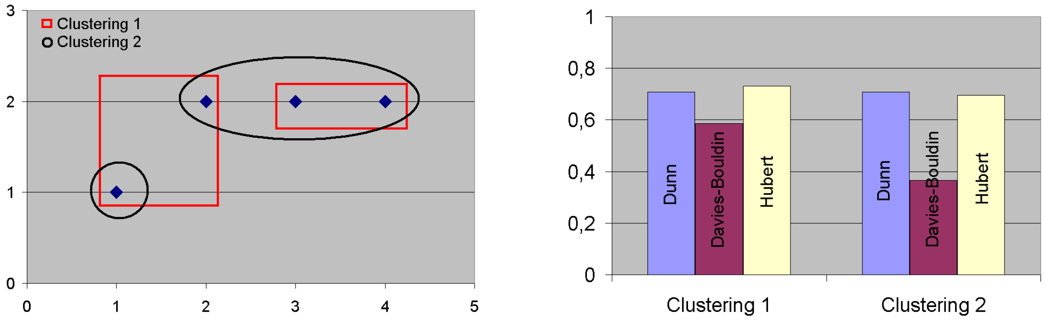
Table 6: Range of distances given for each distance measure

The minimum distance and the maximum diameter represent the extreme points of these ranges. The Dunn indices are therefore supposed to increase when the range of distance values decrease. Normalisation of the distance values was performed but did not result in more comparable results, probably because values close to the extreme points are overrepresented for all distance measures, which will cause a normalisation of a wide range to be likely to include more values close to the extreme points compared to an originally smaller range.

6.4.3 Antagonism between the indices

Observed antagonism in the results Figure 9 shows the Dunn index, the Davies-Bouldin index and the Modified Hubert Γ index for the PCA clusterings of size 16 for each of the three merge strategies. The figures indicate the same trends for all three strategies. The Modified Hubert Γ index shows a little increase in quality for decreased number of features, while the Dunn index and the Davies-Bouldin index indicates decreased quality for a decreased number of features. Corresponding results are obtained for the k-means clusterings, given in Figure 10. These results do not necessarily represent an antagonism, but can potentially reflect a situation where a decrease in the number of features causes a clustering that deviates less from the proximity matrix but nevertheless contains less compact or separated clusters. An example of such a situation is explored in the following paragraph.

An illustrating example of antagonism Consider the example in Figure 18. Figure 18a indicates two possible clustering structures, where the measured quality for the two situations is shown in Figure 18b. The statistic shows that



(a) Two clustering situations

(b) Measured quality per situation

Figure 18: Disagreement between quality measures

while the Dunn indices are equal for the two situations, both the Davies-Bouldin index and the Modified Hubert Γ index decrease from clustering 1 to clustering 2. Accordingly, the Dunn index indicates an unaltered clustering quality, the Davies-Bouldin index indicates an increased quality and the Modified Hubert Γ index indicates a decreased quality. The unaltered Dunn index can easily be explained; in the first case, the minimum distance equals 1 and the maximum diameter equals $\sqrt{2}$. In the second clustering the minimum distance equals $\sqrt{2}$ while the maximum distance equals 2. These metrics will give the same Dunn index:

$$Dunnindex = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2} = 0.707$$

The decreased Davies-Bouldin index indicates more compact and separated clusters in clustering 2 compared to clustering 1. This assertion is also verified by the data set; clustering 1 has an average density of $\frac{1+\sqrt{2}}{2} = 1.207$, while clustering 2 has an average density of $\frac{0+2}{2} = 1$. Also, the clusters in clustering 1 is less well separated compared to the clusters in clustering 2. Contrary, the decreased Modified Hubert Γ index indicates better clustering quality for clustering 1 than for clustering 2, which implies greater agreement between the proximity matrix and the clustering for clustering 1 than for clustering 2. To validate this, consider the proximity matrix p:

$$p = \begin{bmatrix} 0.00 & 1.41 & 2.24 & 3.16 \\ 1.41 & 0.00 & 1.00 & 2.00 \\ 2.24 & 1.00 & 0.00 & 1.00 \\ 3.16 & 2.00 & 1.00 & 0.00 \end{bmatrix}$$

and the mean matrices q_1 and q_2 for clustering 1 and 2 respectively:

$$q_1 = \begin{bmatrix} 0.00 & 0.00 & 2.06 & 2.06 \\ 0.00 & 0.00 & 2.06 & 2.06 \\ 2.06 & 2.06 & 0.00 & 0.00 \\ 2.06 & 2.06 & 0.00 & 0.00 \end{bmatrix} q_2 = \begin{bmatrix} 0.00 & 2.24 & 2.24 & 2.24 \\ 2.24 & 0.00 & 0.00 & 0.00 \\ 2.24 & 0.00 & 0.00 & 0.00 \\ 2.24 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

The sum of the element by element absolute value deviation from p is 9.63 for q_1 and 11.50 for q_2, which emphasises why the Modified Hubert Γ index considers clustering 1 as the best clustering.

6.4.4 Concluding remarks

Generally, these following characteristics of the three indices can probably help to explain the attained results. The Dunn index has a weakness in that both the numerator and the denominator are overly sensitive to changes in the clustering structure, such that even small changes in the clustering are likely to change the index and outliers will potentially affect the index substantially. The Davies-Bouldin index are more robust due to the consideration of the whole clustering. Both the Dunn index and the Davies-Bouldin index share the characteristic that they achieve best results for compact clusters and can therefore be misleading when applied to shell-formed or elongated clusters. The Modified Hubert Γ index do not take the geometry into consideration and are therefore applicable to clusters of all shapes.

The example explored in the previous section demonstrates an important property of the tested clustering quality measures: None of them are capable to define any overall best quality clustering. Each measure is based upon different features of the clustering and does therefore emphasis different quality characteristics. The quality measure should therefore be selected carefully based on the data set at hand. Preferably, several measures founded on different characteristics should be used.

Due to the weaknesses of each of the quality measures tested, and due to the lack of knowledge about both what characteristics each measure offered and what characteristics was desired emphasised, the quality measures did not produce any helpful information in this work. However, the results emphasis the importance of utilising human expertise in the clustering validation step.

6.5 Clustering of high dimensional data

This section deals with the problem of clustering high dimensional data. Section 6.5.1 explores the problem of high dimensional data, Section 6.5.2 explains the relation between binary variables and high dimensional data and Section 6.5.3 explores in what ways PCA can succeed or fail to make a reduced data set appropriate for clustering.

6.5.1 The curse of dimensionality

Clustering of high dimensional data encounters a well-known problem, namely *the curse of dimensionality*, first mentioned by (R.B61). The curse of dimensionality refers to the exponential growth of hypervolume as a function of dimensionality, which indicates that, for clustering purposes, the number of objects should increase exponentially with the dimension of the data space in order to retain a constant density of the data.

The curse of dimensionality is easily illustrated by an example. Imagine that the p dimensional space s is a hypercube of size 1 with total volume 1. Assume s is divided into boxes of size d, where d<1. The volume of each box is then d^p. From this we understand that the volume of each box goes rapidly towards zero when p increases. Evidently, the probability that a specific data object is contained in a box decreases exponentially with the increased dimensionality of the hypercube, which causes an exponentially decrease in the average density of points.

Due to the curse of dimensionality, an enormous amount of data objects is needed to achieve a sensible clustering when the number of features is great. This involves two problems. Firstly, such a large amount of data is often not available for clustering purposes. Secondly, most clustering algorithms encounter problems due to performance when trying to mine such large data sets. Therefore, a more sensible solution to the dimensionality problem is to try to reduce the dimensionality of the data set.

6.5.2 Binary data and dimensionality

Common for several clustering applications is the use of binary data for data representation. A noteworthy characteristic of binary data is that the number of features is often high compared to other data types, such as nominal or continuous data. This can be explained by the type of data on which the clustering is applied.

For instance, clustering is often applied for market basket analysis or web page analysis, where a binary value denotes the presence of a specific product in a store or a specific word in a language. Obviously, the number of features can potentially be great, which emphasises the need for dimensionality reduction of binary data.

In this work, the curse of dimensionality was encountered when clustering the ICPC codes. As mentioned in Section 4.1.1 the ICPC codes were represented by use of binary vectors. Each code was represented by a vector of length equivalent to the number of patients, which was 10 104 in this case. The number of codes was 227, which obviously results in a very low density of data points in the 10 104 dimensional space.

To attempt to reduce this problem, principal component analysis (PCA) was applied to the data set. As mentioned in Section 6.2, the Euclidian distance was the distance measure hardest stroken by the dimensionality problem. Also, the features returned by PCA were float values, which could be handled neither by Jaccard nor by correlation. Therefore, the PCA reduced data sets were only clustered by use of the Euclidian distance.

6.5.3 Shortcomings of PCA

From the results described in Sections 5.1.2 and 5.1.3 it is clear that PCA reduced the dimensionality substantially, but still very well preserved the information contained in the data set. However, the conceptual quality of the resulting clustering did not reach the quality observed by use of correlation distance measure or Jaccard distance measure. There are probably several reasons, from where two are explored in the following, to this restricted clustering quality obtained by use of PCA:

- Firstly, PCA does not always manage to project the data points in the original space into a space more appropriate for clustering. Consider the distribution of data point in the two dimensional space shown in Figure 19. The original axes are labeled a_1 and a_2, while the eigenvectors are labeled e_1 and e_2. As mentioned in 2.4.1, for each eigenvector a corresponding eigenvalue exists that reflects the degree of variation in the data point distribution along that eigenvector. Accordingly, the greater the eigenvalue, the more significant information is contained in the eigenvector. In Figure 19, the data points obviously form two clusters if they are projected to the original axis a_2. However, when the data point are projected to the transformed axes the distribution of data points are similar for both axes, which results in two eigenvalues of the same magnitude. The removal of any of the two new axes will therefore take away half of the information contained in the original distribution, which in practice means that neither of them could be removed. Therefore, the new space is of the same dimensionality than the original one, and, moreover, no projection to the new axes can be used to identify any clusters.

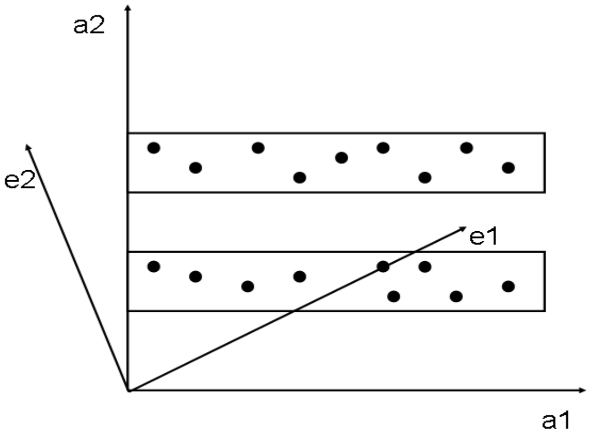


Figure 19: Unsuccessful PCA transformation

- Secondly, another shortcoming of PCA is due to the lacking discovery of subspace clusters. A subspace cluster is a cluster based upon only a subset of the original features or dimensions. In a high dimensional data representation, a great amount of the features are often irrelevant for the clustering. Furthermore, the inclusion of several redundant or irrelevant attributes increases the distance between objects. A conceptual meaningful group based upon only a small subset of the attributes can potentially be ruined by the distance introduced by all the irrelevant attributes. For instance, in Figure 19, subspace clustering could be used to identify the clusters although PCA failed. By eliminating the a_1 axis, the clusters would be discovered by projecting the data points to the a_2 axis.

For the clustering results in Sections 5.1.2 and 5.1.3, the first characteristic described above did probably restrict the quality of the results less than the second one. In the eigenvalues shown in Figure 8 there are a great variation in the magnitude of the values, which indicates that some eigenvectors contains significantly more information than other. Moreover, the clustering results shows that the dimension could be reduced from 7000 to 226 without any changes in the clustering. This demonstrates that the information preserved by the 226 subset is comparable to the information contained in the original 7000 set. However, when the dimension is further reduced, a gradually loss of information occurs. It is therefore likely to believe that the problem of preserving information increase proportional to the decreased number of dimensions.

When it comes to the second drawback, the lack of subspace cluster identification, this is probably the main contributor to the poor clustering quality. Clearly, in a binary vector of size 10104 where 1 indicates the presence and 0 the absence of a diagnosis, the number of 0's is of much greater magnitude than the number of 1's. Also, the fact that a patient had a diagnosis is much more significant than the fact that a patient never had that diagnosis. Equivalent weighting of 0's and 1's, which is the case when using Euclidian distance, would therefore probably result in a distance calculation in which the contribution from the 0's totally outperform the contribution from the 1's. This was further explored in Section 6.2.

6.6 Medical certificate clustering

As mentioned in Section 5.2.2 there are several possible reasons for why the clustering performed on the medical certificates did not manage to form significant groups. In the following, hypotheses of such reasons are stated. For each hypothesis, possible strategies of how to deal with the problems are discussed.

Hypothesis 1: There are no natural clusters in the data set. This hypothesis suggests that there is no clustering tendency in the data set. Such a hypothesis implies that interesting clusters will never appear regardless of the choice of attributes.

(Gri05) argue that there should exist potential groups of patients among the medical certificates, especially among the long-term certificates. As an example, he proposes that there should exist a potential partition between patients that have been reported sick due to real illness and patients that have been reported sick due to a lack of function or well-being in the working life. He suggests that such groups can be revealed by comparing the degree of subjectivity or objectivity for a diagnosis, such that patient groups with a clearly demonstrable ailment could be separated from patients with more vague or personally experienced diagnoses. A concrete potential group could be formed by uneducated but gifted housewives that started working when they were 35-40 in not very challenging jobs. According to (Gri05), a significant amount of patients reported long-term sick have this background. Typical diagnoses for this group are vague diagnoses such as depressive disorder or back syndrome.

Other potential groups of patients are groups that arise due to reorganised or closed companies. In a small or medium society like the one from where the data is collected a reorganisation or closed firm is likely to affect the sick community reporting statistic such that groups of patients with similar occupational background appear in the data set.

These arguments do not prove that there exist groups in the data set used for the clustering task. For instance, the data set can be to small and therefore not constitute a representative selection of the population. However, (Gri05) possesses detailed knowledge about the data set which indicates that it is likely to believe that such groups exist. Possible explanations of why such groups did not appear during the clustering performed in this work are explored in the following.

Hypothesis 2: There are no natural clusters based on the selected attributes. This hypothesis suggests that the data set potentially contains clusters, but that the selected attributes are unsuitable to form interesting clusters regardless of attribute weighting.

(Gri05) argues that the selection of attributes utilised for the medical certificate clustering seems sensible. However, for some of the attributes the definition of subgroups of attributes, or a ranging or detailing of the attributes could bring in more relevant information in the clustering process. He suggests some possible strategies that could be followed to include this additional relevant information about the selected attributes.

First, due to the recently described motives, he suggests that information should be included which describes the degree of subjectivity or objectivity of a diagnosis. For the musculoskeletal ailments typically objective diagnoses are reumatoid arthritis while more subjective diagnoses are back or neck syndrome. Corresponding examples from the set of psychological ailment are the objective disorders psychosis and schizophrenia compared to the subjective diagnoses depressive disorder or anxiety disorder.

Secondly, (Gri05) indicates that there is a correlation between occupation and kind of sickness, which points to the advantage of including more detailed information about a patient's occupation. For all medical certificates the occupation of the patient is registered. Unfortunately, the directives for these noting are not very strict, which results in an infinite number of possible employments. To utilise this information a manual inspection and a grouping of the values is required. This could help revealing potential correlations and could also help to identify the recently mentioned groups arised due to reorganisations in the working life.

Additionally, (Gri05) suggests some new attributes that could introduce new knowledge in the clustering. One of these attributes are hospital referrals. He argues that especially referrals prescribed some time in advance of the sickness reporting can indicate a seriously sick patient. He also mentions x-ray pictures as an indication of seriously sick patients.

Another possible attribute that could be included is the nationality of the patient. The patient journal system offers the possibility to register this information, but the normal practice is to omit it. Such information could possibly reveal interesting information due to the known connection between foreigners and both unemployment and social mechanism such as exclusion. There could also be correlations between nationality and ailments, the occurrences of for example type two diabetes are known to be frequent among some foreigner groups due to the changed diet related to the relocation.

Hypothesis 3: The uninteresting results arose due to the utilisation of unsuitable methods. This hypothesis suggests that the selected features can potentially be used to form interesting clusters, but that the choice of methods like the distance measure or the clustering algorithm caused the poor results.

The most crucial part of the clustering process is probably the distance calculation. The strategy for measuring the distances was probably to simple in this clustering task. The age was the only attribute for which the potential values of distance constituted a range. For the other attributes, the compared objects were considered either equal or not equal, the degree of similarity was not taken into account. A possible method that could help improving the distance measuring of an attribute could be to structure the values for an attribute in groups, hierarchies or ranges. For instance, the correlations between ICPC codes calculated in the ICPC clustering task could be utilised also in this clustering task to calculate diversified distances between codes and code-groups. A similar ranging could possibly be performed also for the marital and occupational status. Obviously, married or cohabitants could be considered more similar to each other than to widows or single people. Also, social security recipients and unemployed could be considered similar to each other and dissimilar to employed patients.

As mentioned in 4.2.3 the clustering algorithm utilised in this clustering task was hierarchical clustering. All three merge strategies were tested and especially the average distance strategy gave some interesting smaller clusters. This points to the importance of selecting an appropriate algorithm. It is likely to believe from the results achieved both by clustering and by counting that the potential interesting clusters in this data set are of varying size and that clustering algorithms such as hierarchical clustering with maximum distance that search for spherical clusters of similar size will fail. The minimum distance strategy and the average distance strategy could potentially identify clusters of dissimilar size, but, except from the few interesting clusters formed by average distance, they fail in this clustering task, probably due to the algorithmic lack of iterative object replacing.

A hierarchical agglomerative clustering algorithm that both takes categorical values and identifies clusters of varying size is the *Rock* algorithm. Rock introduces two new concepts:

- *neighbor:* an object has the number of neighbors equal to the number of other objects considerably similar to the object
- *link:* the link between two objects equals the number of common neighbors

The decision of which clusters to merge are based upon the number of links between the clusters.

Another type of algorithm that handles categorical values are the k-medoid algorithms, which is variations of the k-mean algorithm where the clustering are based on median points instead of mean points. However, these algorithms tend to form spherical clusters of similar size.

An interesting group of algorithms for this task is the group of density based algorithms. These algorithms consider regions of high density as clusters, and regions of low density as noise. This causes the low density regions not to be included in any cluster, which potentially can prevent outliers to weaken the cluster concepts. The density based algorithms consider only the local region of an object to decide if an object should grow into a cluster. This renders possible the growth of clusters of arbitrary shape. The density based algorithms were refused introductory in this work due to the number of influential user parameters. However, a rational choice of parameters could potentially have caused interesting results.

7 Conclusion

This thesis demonstrates that the application of clustering methods to a patient record can potentially identify well-known medical information and with that also potentially identify so far undiscovered significant knowledge. However, only a minority of the methods tried out in this thesis managed to reveal known information. Conclusions regarding the fitness of the tested methods are given in the following. It must be emphasised that the conclusions only consider the method's suitability for the data sets to which they are applied and that they are likely to lead to other conclusions when applied to other data sets.

The findings regarding merge strategies may be summarized thus:

- The maximum distance strategy showed the overall best performance compared to the average distance and the minimum distance strategy. The maximum distance tended to form conceptual meaningful clusters of similar size. Both the minimum distance strategy and the average distance strategy tended to form one single large cluster, the former to a greater degree than the latter.
- In this thesis the average distance strategy shows a tendency to identify smaller, conceptual and meaningful clusters that is not identified by the maximum distance strategy. This indicates that the minimum and the average distance strategies can potentially perform better than the maximum distance strategy when the natural clusters underlying the data set are elongated or of dissimilar size.

The findings regarding distance measures may be summarised thus:

- Lift correlation and the Jaccard coefficient performed better than the Euclidian distance for long binary vectors with a major portion of 0's.
- The clusterings obtained by correlation seemed more meaningful than the clusterings obtained by the Jaccard coefficient.

The findings regarding quality measures may be summarised thus:

- The quality measures evaluated in this thesis, namely the Dunn index, the modified Hubert Γ statistic and the Davies-Bouldin index neither agreed with each other nor with medical expertise in regard to what is considered a good clustering.

- The three measures are based on different clustering characteristics. The quality measured by different quality indices are will therefore at times contradict each other. This emphasises the importance of the user being conscious of what exactly the index shows when the results are interpreted.

- The measured results did not reflect the observed degree of meaning in the clusterings. This underlines the absolute necessity of human inspection and interpretation of the clusterings.

- The experiments demonstrate that the measured quality is highly dependent on the distance measured. This implies the following:
  - A measured quality is not absolute but rather relative compared to other values measured.
  - The comparison of quality measures calculated by use of different distance measure is not meaningful.

The findings regarding principal component analysis may be summarised thus:

- The PCA experiments demonstrated that a substantial reduction of the number of attributes potentially causes only an insignificant loss of information. For the ICPC clustering, the number of features was reduced from 7000 to 226 without having any influence at all on the results; the clusterings obtained by the two data sets were identical.

- PCA increased the conceptual clustering quality compared to the quality obtained by clustering the full data set.

- The clustering tendency increased when the number of features decreased, so that strategies that gave no more than one cluster for the full data set caused several sensible clusters for the reduced data set.

An additional finding in this thesis was that a significant part of the problems associated with clustering is the seemingly overwhelming number of available strategies. The number of possible attribute subsets to represent a data object is very large. There are a great number of available strategies for each preprocessing step as those for normalisation and for the replacement of missing values and there are plenty of distance measures and numerous clustering algorithms. The greatest challenge connected with a clustering task is to make the correct choices in the labyrinth of possible strategies.

Finally, one additional experience gained from this work was that it would be advantageous to design the applications areas and the concrete data sets in advance of the choice of clustering methods to increase the suitability of the selected methods to the data sets.

References

[BWH01] Rohan A. Baxter, Graham J. Williams, and Hongxing He. Feature selection for temporal health records. *Proceedings of the 5th Asia-Pacific Conference on Knowledge Discovery and Data Mining*, 2001.

[Com04] Wonca International Classification Committee. *Den internasjonale klassifikasjon for primærhelsetjenesten*. Wonca International Classification Committee, 2004.

[CR98] James C.Bezdak and Nikhil R.Pal. Some new indexes of cluster validity. *IEEE transactions on systems, man and cybernetics-part B: vol.28,No.3 June 1998*, 1998.

[Fis36] R. Fischer. The use of multiple measurements in taxonomic problems. *Annals of eugenics 7, pp 179-188,1936*, 1936.

[Gra] Graphviz - graph visualization software `http://www.graphviz.org/About.php`.

[Gri05] Anders Grimsmo, 2005. Personal correspondence.

[H.D02] Margaret H.Dunham. *Data Mining, introductory and advanced topics*. Prentice Hall, 2002.

[HK01] Jiawei Han and Micheline Kamber. *Data Mining, concepts and techniques*. Academic Press, 2001.

[HPA] The health personnel act `http://www.lovdata.no/all/nl-19990702-064.html`.

[ICP] Icpc-2 - international classification of primary care `http://www.kith.no/templates/kith_WebPage____1186.aspx`.

[I.T02] I.T.Jolliffe. *Principal Component Analysis, Second Edition*. Springer, 2002.

[ML00] A. Kandel M. Last, O. Maimon. Knowledge discovery in mortality records: an info-fuzzy approach. *Medical Data Mining and Knowledge Discovery, K. Cios (Ed.), Studies in Fuzziness and Soft Computing, Vol. 60, Springer-Verlag, pp. 211-235, 2001*, 2000.

[Pat02] Anne Patrikainen. Projected clustering of high-dimensional binary data. *PhD thesis*, 2002.

[PDA] The personal data act `http://www.ub.uio.no/ujur/ulovdata/lov-20000414-031-eng.pdf`.

[PHD] Personal health data filing system act `http://www.ub.uio.no/ujur/ulovdata/lov-20010518-024-eng.pdf`.

[Pro] Profdoc `http://www.profdoc.no`.

[RAR95] Dimitrios Gunopulos Rakesh Agrawal, Johannes Gerbe and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. 1995.

[R.B61] R.Bellmann. Adaptive control processes: A guided tour. *Princeton University Press*, 1961.

[The] R `http://www.r-project.org`.

[TK99] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition.* Academic Press, 1999.

[Tsu01] S. Tsumoto. Discovery of clinical knowledge in databases extracted from hospital information systems. *Medical Knowledge Discovery, 2000.*, 2001.

[WL02] Machiel Westerdijk and Martijn Ludwig. Product specification by finding homogeneous groups of care episodes in hospital data. *PCSE-conference*, 2002.

A Source code

This appendix lists the source code utilised in this work. Section A.1 contains a summary of the methods, while Section A.2 lists the Python code.

A.1 Method summary

```
FUNCTIONS

    PCA(inputarray, outputnum)
        performs principal component analysis of an array

    avg_link(clusters, pmatrix, ofile, freqcodes, binpats, codes,
        binproxmatrix, binproxmean, binproxvar, distmeasure,
       attrtype)
        implementation of the average distance strategy

    calcDistBetweenClusters(means, distmeasure)
        calculates the distance between the means of each cluster

    calculateDistance(pat1, pat2, distmeasure)
        controls the distance calculation

    calculateEuclid(pat1, pat2)
        calculation of Euclidian distance

    calculateJaccard(pat1, pat2)
        calculation of Jaccard coefficient

    calculateMeans(pats, clusters)
        calculation of means during k-means

    clustering(inputfile, outputfile, clustalg, strategy,
       numclusters, distmeasure, pca, numpcafeatures, attrtype)
        controls choices of clustering strategies

    comparePat(file1, file2)
        removes patients that do not occur in both temporary and
           long-term group

    countCodes(inputfile)
        counts total number and frequency of codes

    createBinVectors(patfile, codes)
        creates binary vectors

    davies_Bouldin(clusters, binpats, distmeasure)
```

```
        calculates the Davies-Bouldin index for a clustering

    dunn_index(clusters, binpmatrix)
        calculates the Dunn index for a clustering

    findCodes(patlist, freqcodes, binpats, codes)
        returns code and textual description for the most
           frequent codes.

    findIllformInfo(list, binpats)
        returns information about the medical sertificates

    hierarchical(pats, strategy, distmeasure, outputfile,
       freqcodes, binpats, codes, attrtype)
        implementation of pure hierarchical clustering

    kmeans(pats, meansnr, distmeasure, outputfile, freqcodes,
       binpats, codes, attrtype)
        implementation of the k-means algorithm

    max_link(clusters, pmatrix, ofile, freqcodes, binpats, codes,
        binproxmatrix, binproxmean, binproxvar, distmeasure,
       attrtype)
        implementation of the maximum distance strategy

    mod_Hubert_stat(clusters, binpats, p, pmean, pvar,
       distmeasure)
        calculates the modified Hubert gamma statistic for a
           clustering

    preprocessIllform(inputfile, outputfile)
        performs preprocessing of medical sertificates. Inserts
           default values, normalises age attributt,
        deletes fields that lacks crucial information

    quality_measure_help(binpats, distmeasure)
        calculates the proximity matrix and its mean and variance
            for the original binary matrix.
        Used for calculating Hubert statistic when the proximity
           matrix are changed due to PCA.

    single_link(clusters, pmatrix, ofile, freqcodes, binpats,
       codes, binproxmatrix, binproxmean, binproxvar, distmeasure
       , attrtype)
        implementation of the minimum distance strategy

    writeGraphViz(pmatrix, codes)
        writes information about clustering on a valid GraphViz
           input format
    writeResult(clusters, iteration, ofile, freqcodes, binpats,
       codes, binpmatrix, binpmean, binpvar, distmeasure,
       attrtype)
        writes results from the clustering procedure to file
```

A.2 Python source code

```
from Numeric import *
from rpy import *
from MLab import mean
import random
import math

def clustering(inputfile,outputfile,clustalg,strategy,numclusters
   ,distmeasure,pca,numpcafeatures,attrtype):
        '''
        controls choices of clustering strategies
        '''
        if attrtype=="binary_codes" or attrtype=="ICPCcluster":
                codes,freqcodes=countCodes(inputfile)
                pats=createBinVectors(inputfile,codes)

                #cluster ICPC not patients
                if attrtype=="ICPCcluster":
                        pats=transpose(pats)

        elif attrtype=="illform":
                print "illform worked"
                pats=[]
                infile=open(inputfile,"r")
                for line in infile:
                        localpat=[]
                        for item in (line.strip().split())[1:]:
                                localpat.append(item)
                        pats.append(localpat)
                freqcodes=0
                codes=0

        #PCA reduction
        if pca=="PCA":
                redpats=PCA(pats,numpcafeatures)
        else:
                redpats=pats[:]

        #select clustering algorithm
        if clustalg=="hierarchical":
                hierarchical(redpats,strategy,distmeasure,
                   outputfile,freqcodes,pats,codes,attrtype)
        elif clustalg=="kmeans":
                initMeans=random.sample(range(len(redpats)-1),
                   numclusters)
                kmeans(redpats,initMeans,distmeasure,outputfile,
                   freqcodes,pats,codes,attrtype)


def countCodes(inputfile):
        '''
        counts total number and frequency of codes
        '''
        opatfile=open(inputfile,"r")
        codes=[]
        for i,line in enumerate(opatfile):
                for code in line.strip().split():
                        if code not in codes:
                                codes.append(code)
        codes.sort()
        freqcodes=[0.0]
        freqcodes=freqcodes*len(codes)
        for i,code in enumerate(codes):
                opatfile.seek(0)
                for line in opatfile:
                        for patcode in line.strip().split():
                                if code==patcode:
                                        freqcodes[i]+=1
                                        break #several identical
                                           codes for one patient
                                           counts as 1
        opatfile.close()
        return codes,freqcodes


def createBinVectors(patfile,codes):
        '''
        creates binary vectors
        '''
        pats=[]
        opatfile=open(patfile)
        opatfile.seek(0)
        for line in opatfile:
                localpat=[]     #binary vector for one patient
                for code1 in codes:
                        pathascode = False
                        for code2 in line.strip().split():
                                if code1==code2:
                                        pathascode=True
                        if pathascode: localpat.append(1)
                        else: localpat.append(0)
                pats.append(localpat)
        opatfile.close()
        return pats


def kmeans(pats,meansnr,distmeasure,outputfile,freqcodes,binpats,
   codes,attrtype):
        '''
        implementation of the k-means algorithm
        '''
        means=[]
        for item in meansnr:
                means.append(pats[item])
        ooutput=open(outputfile,"w")
        iteration=1
        clusterlist=[]
        oldclusterlist=[0]

        binproxmatrix,binproxmean,binproxvar=quality_measure_help
           (pats,distmeasure)

        while(oldclusterlist!=clusterlist):
                if 1<iteration<40:
                        means=calculateMeans(pats,clusters)
                clusters={}
                for i,object1 in enumerate(pats):
                                mindist=999
                                mean=-1
                                for j,object2 in enumerate(means)
                                   :
                                        dist=calculateDistance(
                                           object1,object2,
                                           distmeasure)
                                        if dist<mindist:
                                                mindist=dist
                                                mean=j
                                if clusters.has_key(mean):
                                        clusters[mean].append(i)
                                else:
                                        clusters[mean]=[i]

                oldclusterlist=clusterlist[:]
                clusterlist=clusters.values()
                clusterlist.sort()
                if clusterlist!=oldclusterlist:
                        writeResult(clusterlist,iteration,ooutput
                           ,freqcodes,binpats,codes,binproxmatrix
                           ,binproxmean,binproxvar,distmeasure,
                           attrtype)
                iteration+=1
```

```
def hierarchical(pats,strategy,distmeasure,outputfile,freqcodes,
   binpats,codes,attrtype):
        '''
        implementation of pure hierarchical clustering
        '''
        if attrtype=="binary_codes" or attrtype=="ICPCcluster":
                binproxmatrix,binproxmean,binproxvar=
                   quality_measure_help(binpats,distmeasure)

        elif attrtype=="illform":
                attrtypes=pats[0]
                pats=pats[1:]

        #calculates proximity matrix
        pmatrix=[0.0]
        pmatrix=pmatrix*len(pats)
        for i,item in enumerate(pmatrix):
                pmatrix[i]=[0.0]
                pmatrix[i]=pmatrix[i]*len(pats)


        for i,pat1 in enumerate(pats):
                for j,pat2 in enumerate(pats[i+1:]):
                        if attrtype=="illform":
                                pat1and2=[pat1,pat2]
                                pmatrix[i][j+i+1]=
                                   calculateDistance(pat1and2,
                                   attrtypes,distmeasure)

                        else:
                                pmatrix[i][j+i+1]=
                                   calculateDistance(pat1,pat2,
                                   distmeasure)

        if attrtype=="illform":
                binproxmatrix=[]
                for item in pmatrix:
                        binproxmatrix.append(item[:])


        clusters=[]
        for i,line in enumerate(pats):
                clusters.append([i])
        ofile=open(outputfile,"w")

        if strategy=="single":
                clusters=single_link(clusters,pmatrix,ofile,
                   freqcodes,binpats,codes,binproxmatrix,
                   binproxmean,binproxvar,distmeasure,attrtype)
        if strategy=="max":
                clusters=max_link(clusters,pmatrix,ofile,
                   freqcodes,binpats,codes,binproxmatrix,
                   binproxmean,binproxvar,distmeasure,attrtype)
        if strategy=="avg":
                clusters=avg_link(clusters,pmatrix,ofile,
                   freqcodes,binpats,codes,binproxmatrix,
                   binproxmean,binproxvar,distmeasure,attrtype)


def single_link(clusters,pmatrix,ofile,freqcodes,binpats,codes,
   binproxmatrix,binproxmean,binproxvar,distmeasure,attrtype):
        '''
        implementation of the minimum distance strategy
        '''
        dunnfile=open("clusters/ICPCclustering/euclid_7000/dunn/
           euc_dunn_hiersingle_7000.txt","w")
        hubertfile=open("clusters/ICPCclustering/euclid_7000/
           hubert/euc_hub_hiersingle_7000.txt","w")
        dbfile=open("clusters/ICPCclustering/euclid_7000/db/
           euc_db_hiersingle_7000.txt","w")
        print "inside single link"
        ofile.write("******************************SINGLE LINK
           ********************************\n")
        while(len(clusters)>1):
                #seek for minimum
                mindist=sys.maxint
                for i,item in enumerate(pmatrix[:-1]):
                        localmin=min(item[i+1:])
                        if localmin<mindist:
                                mindist=localmin
                                ipos=i
                jpos=pmatrix[ipos][ipos+1:].index(mindist)
                jpos=jpos+ipos+1

                #calculate new min
                k=0
                l=len(pmatrix)
                pmatrix[ipos][jpos]=0.0
                while k<l:
                        pmatrix[min(k,ipos)][max(k,ipos)]=min(
                           pmatrix[min(k,ipos)][max(k,ipos)],
                           pmatrix[min(k,jpos)][max(k,jpos)])
                        k+=1

                #pop and append
                pmatrix.pop(jpos)
                for i,item in enumerate(pmatrix):
                        pmatrix[i].pop(jpos)


                for item in clusters[jpos]:
                        clusters[ipos].append(item)
                clusters.pop(jpos)
                if len(clusters)<50:
                        dunnfile.write(str(dunn_index(clusters,
                           binproxmatrix))+"\n")
                        hubertfile.write(str(mod_Hubert_stat(
                           clusters,binpats,binproxmatrix,
                           binproxmean,binproxvar,distmeasure))
                           +"\n")
                        dbfile.write(str(davies_Bouldin(clusters,
                           binpats,distmeasure))+"\n")

                if len(clusters)<11: #write to file only the last
                    iterations
                        writeResult(clusters,len(clusters),ofile,
                           freqcodes,binpats,codes,binproxmatrix,
                           binproxmean,binproxvar,distmeasure,
                           attrtype)


def max_link(clusters,pmatrix,ofile,freqcodes,binpats,codes,
   binproxmatrix,binproxmean,binproxvar,distmeasure,attrtype):
        '''
        implementation of the maximum distance strategy
        '''
        dunnfile=open("clusters/ICPCclustering/euclid_7000/dunn/
           euc_dunn_hiermax_7000.txt","w")
        hubertfile=open("clusters/ICPCclustering/euclid_7000/
           hubert/euc_hub_hiermax_7000.txt","w")
        dbfile=open("clusters/ICPCclustering/euclid_7000/db/
           euc_db_hiermax_7000.txt","w")
        print "inside max link"
        ofile.write("******************************MAX LINK
           ************************************'\n")
        while(len(clusters)>1):
                #seek for minimum
                if distmeasure=="Jaccard":
                        mindist=1.0
                else:
                        mindist=sys.maxint
                for i,item in enumerate(pmatrix[:-1]):
                        localmin=min(item[i+1:])
                        if localmin<mindist:
                                mindist=localmin
                                ipos=i
                jpos=pmatrix[ipos][ipos+1:].index(mindist)
                jpos=jpos+ipos+1

                #calculates new max
                k=0
                l=len(pmatrix)
                pmatrix[ipos][jpos]=0.0
                while k<l:
                        pmatrix[min(k,ipos)][max(k,ipos)]=max(
                           pmatrix[min(k,ipos)][max(k,ipos)],
                           pmatrix[min(k,jpos)][max(k,jpos)])
                        k+=1

                #pop and append
                pmatrix.pop(jpos)
                for i,item in enumerate(pmatrix):
                        pmatrix[i].pop(jpos)

                for item in clusters[jpos]:
                        clusters[ipos].append(item)
                clusters.pop(jpos)

                if len(clusters)<50:
                        dunnfile.write(str(dunn_index(clusters,
                           binproxmatrix))+"\n")
                        hubertfile.write(str(mod_Hubert_stat(
                           clusters,binpats,binproxmatrix,
                           binproxmean,binproxvar,distmeasure))
                           +"\n")
                        dbfile.write(str(davies_Bouldin(clusters,
                           binpats,distmeasure))+"\n")

                if len(clusters)<11: #write to file only the last
                    iterations
                        writeResult(clusters,len(clusters),ofile,
                           freqcodes,binpats,codes,binproxmatrix,
                           binproxmean,binproxvar,distmeasure,
                           attrtype)


def avg_link(clusters,pmatrix,ofile,freqcodes,binpats,codes,
   binproxmatrix,binproxmean,binproxvar,distmeasure,attrtype):
        '''
        implementation of the average distance strategy
        '''
        dunnfile=open("clusters/ICPCclustering/euclid_7000/dunn/
           euc_dunn_hieravg_7000.txt","w")
        hubertfile=open("clusters/ICPCclustering/euclid_7000/
           hubert/euc_hub_hieravg_7000.txt","w")
        dbfile=open("clusters/ICPCclustering/euclid_7000/db/
           euc_db_hieravg_7000.txt","w")
        print "inside avg link"
        ofile.write("*******************************************
           AVG LINK**************************\n")
        while(len(clusters)>1):
                #seek for minimum
                mindist=sys.maxint
                for i,item in enumerate(pmatrix[:-1]):
                        localmin=min(item[i+1:])
                        if localmin<mindist:
                                mindist=localmin
                                ipos=i
                jpos=pmatrix[ipos][ipos+1:].index(mindist)
                jpos=jpos+ipos+1

                #calculate new averages
                k=0
                l=len(pmatrix)
                pmatrix[ipos][jpos]=0.0
                while k<l:
                        pmatrix[min(k,ipos)][max(k,ipos)]=(
                            pmatrix[min(k,ipos)][max(k,ipos)]*len(
                            clusters[ipos])+pmatrix[min(k,jpos)][
                            max(k,jpos)]*len(clusters[jpos]))/(len
                            (clusters[ipos])+len(clusters[jpos]))
                        k+=1

                #pop and append
                pmatrix.pop(jpos)
                for i,item in enumerate(pmatrix):
                        pmatrix[i].pop(jpos)

                for item in clusters[jpos]:
                        clusters[ipos].append(item)
                clusters.pop(jpos)

                if len(clusters)<50:
                        dunnfile.write(str(dunn_index(clusters,
                            binproxmatrix))+"\n")
                        hubertfile.write(str(mod_Hubert_stat(
                            clusters,binpats,binproxmatrix,
                            binproxmean,binproxvar,distmeasure))
                            +"\n")
                        dbfile.write(str(davies_Bouldin(clusters,
                            binpats,distmeasure))+"\n")

                if len(clusters)<11: #write to file only the last
                     iterations
                        writeResult(clusters,len(clusters),ofile,
                            freqcodes,binpats,codes,binproxmatrix,
                            binproxmean,binproxvar,distmeasure,
                            attrtype)


def calculateDistance(pat1,pat2,distmeasure):
        '''
        controls the distance calculation
        '''
        if distmeasure=="Jaccard":
                dissim=calculateJaccard(pat1,pat2)
        elif distmeasure=="Euclid":
                dissim=calculateEuclid(pat1,pat2)
        elif distmeasure=="Manhattan":
                dissim=calculateManhattan(pat1,pat2)
        elif distmeasure=="Mixed":
                attrtypes=pat2
                pat_1=pat1[0]
                pat_2=pat1[1]
                dissim=0.0
                for i,item in enumerate(attrtypes):
                        if item=="I":
                                dissim+=abs(float(pat_1[i])-float
                                   (pat_2[i]))
                        elif item=="N":
                                if pat_1[i]!=0 and pat_2[i]!=0:
                                        if pat_1[i]!=pat_2[i]:
                                                dissim+=1
                        elif item=="B":
                                if pat_1[i]!=pat_2[i]:
                                        dissim+=1
                dissim/=len(attrtypes)
        return dissim


def calculateJaccard(pat1,pat2):
        '''
        calculation of Jaccard coefficient
        '''
        pat1count=0.0
        pat2count=0.0
        common=0.0
        for k,item1 in enumerate(pat1):
                if item1==True:
                        pat1count+=1
                if pat2[k]==True:
                        pat2count+=1
                if item1 and pat2[k]==True:
                        common+=1
        if pat1count==pat2count==0.0:
                dissim=0
        else:
                dissim=(pat1count+pat2count-2*common)/(pat1count+
                   pat2count-common)
        return dissim


def calculateEuclid(pat1,pat2):
        '''
        calculation of Euclidian distance
        '''
        dissim=0
        for k,item in enumerate(pat1):
                localdissim=(item-pat2[k])**2
                dissim+=localdissim
        dissim=math.sqrt(dissim)
        return dissim


def calculateMeans(pats,clusters):
        '''
        calculation of means during k-means
        '''
        keys = clusters.keys()
        means=[]
        for key in keys:
                cluster=clusters.get(key)
                localmean=[0.0]
                localmean=localmean*len(pats[0])
                for item in cluster:
                        pat=pats[item]
                        for i,item in enumerate(pat):
                                localmean[i]+=item
                for i,item in enumerate(localmean):
                        localmean[i]=item/len(cluster)
                means.append(localmean)
        return means


def writeResult(clusters,iteration,ofile,freqcodes,binpats,codes,
   binpmatrix,binpmean,binpvar,distmeasure,attrtype):
        '''
        writes results from the clustering procedure to file
        '''
        ofile.write("***********Iteration "+str(iteration)
           +"***********\n\n")
        ofile.write("NUM CLUSTERS: "+str(len(clusters))+"\n\n")
        for list in clusters:
                ofile.write("Size of cluster: ")
                ofile.write(str(len(list))+"\n")

                if attrtype=="binary_codes":
                        for patid in list:
                                strpatid=str(patid)
                                ofile.write(strpatid+" ")
                        ofile.write("\n\n")
                        topfreqcodes=findCodes(list,freqcodes,
                           binpats,codes)
                        for code,freq,desc in topfreqcodes:
                                ofile.write(str(code)+" "+str(
                                   freq)+" ")
                                for item in desc:
                                        ofile.write(item+" ")
                                ofile.write("\n")
                        ofile.write("\n")

                elif attrtype=="ICPCcluster":
                        icpcfile=open("dbfiles/icpc2codes.txt","r
                           ")
                        for id in list:
                                code=codes[id]
                                icpcfile.seek(0)
                                for line in icpcfile:
                                        desccode=line.strip().
                                           split()[0]
                                        if code==desccode:
                                                ofile.write(line.
                                                   strip()+"\n")
                                                break
                        ofile.write("\n")
                        icpcfile.close()

                elif attrtype=="illform":
                        for patid in list:
                                strpatid=str(patid)
                                ofile.write(strpatid+" ")
                        ofile.write("\n\n")
                        avg_age,stdev,mar_stat,occ_stat,sex,codes
                           ,code_groups=findIllformInfo(list,
                           binpats)#binpats=pats because PCA is
                           not used
                        ofile.write("Avg age "+str(avg_age)+"\n
                           "+"Age st.dev "+str(stdev)+"\n")
                        for item in mar_stat:
                                ofile.write(item+" ")
                        ofile.write("\n")
                        for item in occ_stat:
                                ofile.write(item+" ")
                        ofile.write("\nSex ")
                        for item in sex:
                                ofile.write(item+" ")
                        ofile.write("\nCodes ")
                        for item in codes:
                                ofile.write(item+" ")
                        ofile.write("\nCodegroups ")
                        for item in code_groups:
                                ofile.write(item+" ")
                        ofile.write("\n\n")

        dunn=dunn_index(clusters,binpmatrix)
        hyb=mod_Hubert_stat(clusters,binpats,binpmatrix,binpmean,
           binpvar,distmeasure)
```

```
        db=davies_Bouldin(clusters,binpats,distmeasure)
        ofile.write("Dunn-index: "+str(dunn)+"\n\n")
        ofile.write("Modified Hubert statistic: "+str(hyb)+"\n\n
           ")
        ofile.write("Davies-Bouldin-index: "+str(db)+"\n\n")


def findIllformInfo(list,binpats):
        '''
        returns information about the medical sertificates
        '''
        pats=binpats[1:]
        sum_age=0.0
        ages=[]
        mar_dict={}
        occ_dict={}
        sex_dict={}
        codes_dict={}
        codegroup_dict={}
        for item in list:
                attr=pats[item]
                age=attr[0]
                ages.append(age)
                marstat=attr[1]
                occstat=attr[3]
                sex=attr[2]
                code=attr[4]
                codegroup=attr[5]
                sum_age+=float(age)
                if mar_dict.has_key(marstat):
                        mar_dict[marstat]+=1
                else:
                        mar_dict[marstat]=1
                if occ_dict.has_key(occstat):
                        occ_dict[occstat]+=1
                else:
                        occ_dict[occstat]=1
                if sex_dict.has_key(sex):
                        sex_dict[sex]+=1
                else:
                        sex_dict[sex]=1
                if codes_dict.has_key(code):
                        codes_dict[code]+=1
                else:
                        codes_dict[code]=1
                if codegroup_dict.has_key(codegroup):
                        codegroup_dict[codegroup]+=1
                else:
                        codegroup_dict[codegroup]=1
        marstatlist=[(mar_dict[key],key) for key in mar_dict.keys
           ()]
```

```
marstatlist.sort()
marstatlist.reverse()
occstatlist=[(occ_dict[key],key) for key in occ_dict.keys
   ()]
occstatlist.sort()
occstatlist.reverse()
sexlist=[(sex_dict[key],key) for key in sex_dict.keys()]
sexlist.sort()
sexlist.reverse()
codelist=[(codes_dict[key],key) for key in codes_dict.
   keys()]
codelist.sort()
codelist.reverse()
cgrouplist=[(codegroup_dict[key],key) for key in
   codegroup_dict.keys()]
cgrouplist.sort()
cgrouplist.reverse()

avgage=sum_age/len(list)
#avgage=avgage*(102-17)+17#illform_1 linear max/min
   normalization
#avgage=avgage*(99-18)+18#illform_2 linear max/min
   normalization
avgage=avgage/0.7*12.142177554+44.23242176#z-score
   normalization#illform_1
#avgage=avgage/0.7*12.436339892+45.89035#z-score
   normalization#illform_2
for i,age in enumerate(ages):
#       ages[i]=float(ages[i])*(102-17)+17#illform_1
   linear
#       ages[i]=float(ages[i])*(99-18)+18#illform_2
   linear
        ages[i]=float(ages[i])
           /0.7*12.142177554+44.23242176#illform_1 z-
           score
#       ages[i]=float(ages[i])/0.7*12.436339892+45.89035#
   illform_2 z-score
var=0.0
for age in ages:
        var+=(float(age)-avgage)**2
var/=len(ages)
stdev=sqrt(var)
marstatfile=open("dbfiles/marstatus.txt","r")
marstat=[]
for i,item in enumerate(marstatlist):
        marstatfile.seek(0)
        for line in marstatfile:
                linelist=line.strip().split()
                if linelist[0]==marstatlist[i][1]:
                        mardesc=linelist[1]
                        break
                marstat.append(mardesc+" "+str(float(marstatlist[
                   i][0])/len(list)))
        occstatfile=open("dbfiles/occstatus.txt","r")
        occstat=[]
        for k,item in enumerate(occstatlist):
                occstatfile.seek(0)
                for line in occstatfile:
                        linelist=line.strip().split()
                        if linelist[0]==occstatlist[k][1]:
                                occdesc=linelist[1]
                                break
                occstat.append(occdesc+" "+str(float(occstatlist[
                   k][0])/len(list)))
        marstatfile.close()
        occstatfile.close()
        sex=[]
        for i,item in enumerate(sexlist):
                sex.append(sexlist[i][1]+" "+str(float(sexlist[i
                   ][0])/len(list)))
        codes=[]
        i=0
        while i<5 and i<len(codelist):
                codes.append(codelist[i][1]+" "+str(float(
                   codelist[i][0])/len(list)))
                i+=1
        cgroups=[]
        j=0
        while j<5 and j<len(cgrouplist):
                cgroups.append(cgrouplist[j][1]+" "+str(float(
                   cgrouplist[j][0])/len(list)))
                j+=1
        return avgage,stdev,marstat,occstat,sex,codes,cgroups


def findCodes(patlist,freqcodes,binpats,codes):
¨       '''
        returns code and textual description for the most
           frequent codes.
        '''
        sumCodes=[0.0]
        sumCodes=sumCodes*len(freqcodes)
        for patid in patlist:
                for i,item in enumerate(binpats[patid]):
                        if item==1:
                                sumCodes[i]+=1
        for i,item in enumerate(sumCodes):
                sumCodes[i]/=len(patlist)

        odescrfile=open("dbfiles/icpc2codes.txt","r")
        top8=[]
        while len(top8)<8:
                maxfreq=0.0
                maxpos=0
                for i,item in enumerate(sumCodes):
                        if sumCodes[i]>maxfreq:
                                maxfreq=sumCodes[i]
                                maxpos=i
                odescrfile.seek(0)
                for line in odescrfile:
                        desccode=line.strip().split()[0]
                        if codes[maxpos]==desccode:
                                desc=line.strip().split()[1:]
                                break
                l=[codes[maxpos],maxfreq,desc]
                top8.append(l)
                sumCodes[maxpos]=False
        odescrfile.close()
        return top8


def dunn_index(clusters,binpmatrix):
        '''
        calculates the Dunn index for a clustering
        '''
        dunn_index=0.0
        minsinglelink=sys.maxint
        for i,cluster1 in enumerate(clusters):
                for cluster2 in clusters[i+1:]:
                        for item1 in cluster1:
                                for item2 in cluster2:
                                        distance=binpmatrix[min(
                                           item1,item2)][max(
                                           item1,item2)]
                                        if distance<minsinglelink
                                           :
                                                minsinglelink=
                                                   distance
        maxdiameter=0.0
        for cluster in clusters:
                for i,item1 in enumerate(cluster):
                        for item2 in cluster[i+1:]:
                                diameter=binpmatrix[min(item1,
                                   item2)][max(item1,item2)]
                                if diameter>maxdiameter:
                                        maxdiameter=diameter
        if maxdiameter==0 or minsinglelink==sys.maxint:
                dunn_index=None
        else:
                dunn_index=minsinglelink/maxdiameter
        return dunn_index
```

```
def quality_measure_help(binpats,distmeasure):
        '''
        calculates the proximity matrix and its mean and variance
            for the original binary matrix.
        Used for calculating Hubert statistic when the proximity
           matrix are changed due to PCA.
        '''
        pmatrix=[0.0]
        pmatrix=pmatrix*len(binpats)
        for i,item in enumerate(pmatrix):
                pmatrix[i]=[0.0]
                pmatrix[i]=pmatrix[i]*len(binpats)

        for i,pat1 in enumerate(binpats):
                for j,pat2 in enumerate(binpats):
                        pmatrix[i][j]=calculateDistance(pat1,pat2
                           ,distmeasure)

        meanpmatrix=mean(mean(pmatrix))

        varpmatrix=0
        for i,item1 in enumerate(pmatrix):
                for j,item2 in enumerate(item1):
                        varpmatrix+=pmatrix[i][j]**2-meanpmatrix
                           **2
        varpmatrix/=len(pmatrix)**2
        varpmatrix=sqrt(varpmatrix)
        return pmatrix,meanpmatrix,varpmatrix


def mod_Hubert_stat(clusters,binpats,p,pmean,pvar,distmeasure):
        '''
        calculates the modified Hubert gamma statistic for a
           clustering
        '''
        if len(clusters)>1:
                l=[0]
                l=l*len(binpats)
                clusterdict={}
                for i,cluster in enumerate(clusters):
                        clusterdict[i]=cluster
                        for patid in cluster:
                                l[patid]=i

                #calculate mean for each cluster
                means=calculateMeans(binpats,clusterdict)

                #calculate distance between means for each pair
                   of clusters
                meandist=calcDistBetweenClusters(means,
                   distmeasure)
                #calculate distance between means representing
                   the clusters in which the objects i,j belong
                q=[0.0]#patient*patient matrix
                q=q*len(binpats)
                for i,item in enumerate(q):
                        q[i]=[0.0]
                        q[i]=q[i]*len(binpats)

                for i,item1 in enumerate(q):
                        for j,item2 in enumerate(q):
                                q[i][j]=meandist[l[i]][l[j]]

                qmean=mean(mean(q))
                qvar=0
                for i,item1 in enumerate(q):
                        for j,item2 in enumerate(item1):
                                qvar+=q[i][j]**2-qmean**2
                qvar/=len(q)**2
                qvar=sqrt(qvar)

                modhyb=0
                for i,item1 in enumerate(p):
                        for j,item2 in enumerate(item1):
                                modhyb+=(p[i][j]-pmean)*(q[i][j]-
                                   qmean)

                modhyb/=len(p)**2*pvar*qvar
                return modhyb


def davies_Bouldin(clusters,binpats,distmeasure):
        '''
        calculates the Davies-Bouldin index for a clustering
        '''
        #calculate mean for each cluster
        clusterdict={}
        for i,cluster in enumerate(clusters):
                clusterdict[i]=cluster

        means=calculateMeans(binpats,clusterdict)

        #calculate dispersion for each cluster
        dispersion=[]
        for i,cluster in enumerate(clusters):
                clusterdisp=0
                for pat in cluster:
                        disp=calculateDistance(binpats[pat],means
                           [i],distmeasure)
                        clusterdisp+=disp**2
                clusterdisp/=len(cluster)
                clusterdisp=sqrt(clusterdisp)
                dispersion.append(clusterdisp)

        #calculate distance between means for each pair of
           clusters
        meandist=calcDistBetweenClusters(means,distmeasure)

        #calculate similarity index
        r=[0.0]#cluster*cluster matrix
        r=r*len(clusters)
        for i,item in enumerate(r):
                r[i]=[0.0]
                r[i]=r[i]*len(clusters)

        for i,item1 in enumerate(r):
                for j,item2 in enumerate(item1):
                        if meandist[i][j]!=0.0:
                                r[i][j]=(dispersion[i]+dispersion
                                   [j])/meandist[i][j]
                        else:
                                r[i][j]=0.0

        db=0
        for item in r:
                db+=max(item)
        db/=len(clusters)
        return db


def calcDistBetweenClusters(means,distmeasure):
        '''
        calculates the distance between the means of each cluster
        '''
        meandist=[0.0]#cluster*cluster matrix
        meandist=meandist*len(means)
        for i,item in enumerate(meandist):
                meandist[i]=[0.0]
                meandist[i]=meandist[i]*len(means)

        for i, item1 in enumerate(means):
                for j,item2 in enumerate(means):
                        meandist[i][j]=calculateDistance(item1,
                           item2,distmeasure)
        return meandist


def PCA(inputarray,outputnum):
        '''
        performs principal component analysis of an array
        '''
        o=array(inputarray,Float)
        meanarray=sum(o)/float(len(o))

        #subtracts mean
        for i,item1 in enumerate(o):
                for j,item2 in enumerate(item1):
                        o[i][j]=o[i][j]-meanarray[j]

        #make covariance matrix etc
        c=r.cov(o)
        e=r.eigen(c)
        eigenvalues=e["values"]

        #write eigenvalues to file
        oeigenval=open("pcafiles/eigenvalues.txt","w")
        for i, item in enumerate(eigenvalues):
                oeigenval.write(str(i)+" "+str(item)+"\n")

        eigenvectors=e["vectors"]
        evaldict={}
        for i,item in enumerate(eigenvalues):
                evaldict[i]=item

        evallist=[(evaldict[key],key) for key in evaldict.keys()]
        evallist.sort()
        evallist.reverse()

        o=transpose(o)
        evectors=transpose(evectors)
        outputdata=matrixmultiply(evectors,o)
        outputtrans=transpose(outputdata)
        return outputtrans.tolist()


def writeGraphViz(pmatrix,codes):
        '''
        writes information about clustering on a valid GraphViz
           input format
        '''
        ograph=open("GraphViz/icpc/top300_euclid_len2.dot.txt","w
           ")
        ograph.write("graph icpc_graph {\nstart = rand\nroot =
           contact\nrankdir = LR\nsize = \"7.5,10\"\nnode[style=
           filled,height=0.1,width=0.1]\n")

        #write all codes with weights

        for i,row in enumerate(pmatrix):
                for j,item in enumerate(row[i+1:]):
                        weight=int(pmatrix[i][i+j+1]*10)
                        if weight==0:
                                weight=1
                        ograph.write(str(codes[i])+" -- "+str(
                           codes[j+i+1])+" [len="+str(weight)+"]\
                           n")
        ograph.write("}")

        #write X smallest distances
        dist=[]
        for i,row in enumerate(pmatrix):
                for j,item in enumerate(row[i+1:]):
                        dist.append(pmatrix[i][i+j+1])
        dist.sort()
        maxdist=dist[300]
        for i,row in enumerate(pmatrix):
                for j,item in enumerate(row[i+1:]):
                        if pmatrix[i][i+j+1]<=maxdist:
                                weight=int(pmatrix[i][i+j+1]*10)
                                if weight==0:
                                        weight=1
                                ograph.write(str(codes[i])+" --
                                   "+str(codes[j+i+1])+" [len
                                   =2"+"]\n")
        ograph.write("}")

        #write minimum distance for each node
        minp=sys.maxint
        maxp=0
        for i,list in enumerate(pmatrix):
                for j,item in enumerate(list):
                        if pmatrix[i][j]!=0:
                                ograph.write(str(codes[i])+" --
                                   "+str(codes[j])+" "+str(
                                   pmatrix[i][j])+"\n")
                                if pmatrix[i][j]<minp:
                                        minp=pmatrix[i][j]
                                if pmatrix[i][j]>maxp:
                                        maxp=pmatrix[i][j]
        ograph.write("minimum distance "+str(minp)+"\nmaximum
           distance "+str(maxp))


def preprocessIllform(inputfile,outputfile):
        '''
        performs preprocessing of medical certificates. Inserts
           default values, normalises age attributt,
        deletes fields that lacks crucial information
        '''
        infile=open(inputfile,"r")
        helpfile=open("soppel\helpfile.txt","w")
        exception=0
        minage=100.0
        maxage=0.0
```

```
sumage=0.0
varage=0.0
nomar=0
noocc=0
agelist=[]
for i,line in enumerate(infile):
        lineiter=iter(line.strip().split())
        lineiter.next()
        dob=(lineiter.next())[0:4]
        lineiter.next()
        doi=(lineiter.next())[0:4]
        age=int(doi)-int(dob)
        agelist.append(age)
        if age<minage:
                minage=age
        if age>maxage:
                maxage=age
        sumage+=age
meanage=sumage/(i+1)
for age in agelist:
        varage+=(age-meanage)**2
varage/=len(agelist)
stdevage=math.sqrt(varage)
infile.seek(0)
for line in infile:
        lineiter=iter(line.strip().split())
        patid=lineiter.next()
        dob=(lineiter.next())[0:4]
        lineiter.next()
        doi=(lineiter.next())[0:4]
        age=float(doi)-float(dob)
        linearnormage=(age-minage)/(maxage-minage)
        zscorenormage=((age-meanage)/stdevage)*0.7
        lineiter.next()
        mar_or_sex=lineiter.next()
        if mar_or_sex.isdigit():
                mar=mar_or_sex
                sex=lineiter.next()
        else:
                mar="0"
                nomar+=1
                sex=mar_or_sex
        try:
                occ_or_code=lineiter.next()
                if len(occ_or_code)==1:
                        try:
                                code=lineiter.next()
                                helpfile.write(patid+" "+
                                   str(zscorenormage)+"
                                   "+mar+" "+sex+" ")
```

```
                                        helpfile.write(
                                           occ_or_code+" "+code.
                                           capitalize()+" "+code
                                           [0].capitalize()+"\n")

                                except StopIteration:exception+=1
                        elif len(occ_or_code)>1:
                                noocc+=1
                                helpfile.write(patid+" "+str(
                                   zscorenormage)+" "+mar+" "+sex
                                   +" ")
                                helpfile.write("0 "+occ_or_code.
                                   capitalize()+" "+occ_or_code
                                   [0].capitalize()+"\n")

                except StopIteration:exception+=1

        infile.close()
        helpfile.close()
        newhelp=open("soppel\helpfile.txt","r")
        outfile=open(outputfile,"w")
        prevline=[]
        counter=0
        wrongcodecounter=0
        for line in newhelp:#ordne saa line ikke skrives hvis
           patid og diagnose er lik
                linelist=[]
                for item in line.strip().split():
                        linelist.append(item)
                if len(prevline)==0:
                        outfile.write(line)
                        prevline=linelist
                elif linelist[6].isdigit():
                        wrongcodecounter+=1
                elif linelist[0]==prevline[0] and linelist[5]==
                   prevline[5]:
                        counter+=1
                else:
                        outfile.write(line)
                        prevline=linelist


def comparePat(file1,file2):
        '''
        removes patients that do not occur in both temporary and
           long-term group
        '''
        oill_1=open(file1,"r")
        oill_2=open(file2,"r")
        pats_1=[]
        pats_2=[]
```

```
prevpatid=0
for line in oill_1:
        linelist=line.strip().split()
        patid=linelist[0]
        if patid!=prevpatid:
                pats_1.append(patid)
                prevpatid=patid

for line in oill_2:
        linelist=line.strip().split()
        patid=linelist[0]
        if patid!=prevpatid:
                pats_2.append(patid)
                prevpatid=patid
oill_1.seek(0)
oill_2.seek(0)

#make list of codes that both files have in common
both=[]
for item1 in pats_1:
        for item2 in pats_2:
                if item1==item2:
                        both.append(item1)
print "patients in common ",len(both)
attrtypes="PID I N B N N N"

#patients in commonlist is written to file
ofile_1=open("dbfiles\illform1_sel_def0.txt","w")
ofile_1.write(attrtypes+"\n")
for line in oill_1:
        linelist=line.strip().split()
        for patid in both:
                if patid==linelist[0]:
                        ofile_1.write(line)
                        break

ofile_2=open("dbfiles\illform2_sel_def0.txt","w")
ofile_2.write(attrtypes+"\n")
for line in oill_2:
        linelist=line.strip().split()
        for patid in both:
                if patid==linelist[0]:
                        ofile_2.write(line)
                        break
```

B Clustering results

This appendix lists the clusterings that were visualised in this work. Section B.1 lists the results from the ICPC clustering explored in Section 5.1.4, while Section B.2 lists the results from the medical certificate clusterings explored in Section 5.2.2.

B.1 ICPCclustering

```
Cluster  0
A97 NO DISEASE
U06 HAEMATURIA
U95 URINARY CALCULUS

Cluster  1
A03 FEVER
A72 CHICKENPOX
A76 VIRAL EXANTHEM OTHER
A77 VIRAL DISEASE OTHER/NOS
H01 EAR PAIN/EARACHE
H71 ACUTE OTITIS MEDIA/MYRINGITIS
H72 SEROUS OTITIS MEDIA
R77 LARYNGITIS/TRACHEITIS ACUTE
S07 RASH GENERALIZED
S84 IMPETIGO
S87 DERMATITIS/ATOPIC ECZEMA
S98 URTICARIA
Y75 BALANITIS

Cluster  2
A04 WEAKNESS/TIREDNESS GENERAL
A13 CONCERN ABOUT/FEAR OF MEDICAL
    TREATMENT
B80 IRON DEFICIENCY ANAEMIA
B82 ANAEMIA OTHER/UNSPECIFIED
D01 ABDOMINAL PAIN/CRAMPS GENERAL
D06 ABDOMINAL PAIN LOCALIZED OTHER
D09 NAUSEA
D87 STOMACH FUNCTION DISORDER
D98 CHOLECYSTITIS/CHOLELITHIASIS
U29 URINARY SYMPTOM/COMPLAINT OTHER
U70 PYELONEPHRITIS/PYELITIS
U71 CYSTITIS/URINARY INFECTION OTHER

Cluster  3
A06 FAINTING/SYNCOPE
A96 DEATH
H82 VERTIGINOUS SYNDROME
K01 HEART PAIN
K74 ISCHAEMIC HEART DISEASE WITH ANGINA
K76 ISCHAEMIC HEART DISEASE WITHOUT
    ANGINA
K77 HEART FAILURE
K78 ATRIAL FIBRILLATION/FLUTTER
K85 ELEVATED BLOOD PRESSURE
K86 HYPERTENSION UNCOMPLICATED
K89 TRANSIENT CEREBRAL ISCHAEMIA
K90 STROKE/CEREBROVASCULAR ACCIDENT
L75 FRACTURE: FEMUR
N17 VERTIGO/DIZZINESS
R06 NOSE BLEED/EPISTAXIS
R81 PNEUMONIA
S70 HERPES ZOSTER
U05 URINATION PROBLEMS OTHER
U99 URINARY DISEASE OTHER
Y85 BENIGN PROSTATIC HYPERTROPHY

Cluster  4
A80 TRAUMA/INJURY NOS
L73 FRACTURE: TIBIA/FIBULA
L76 FRACTURE: OTHER
L77 SPRAIN/STRAIN OF ANKLE
L79 SPRAIN/STRAIN OF JOINT NOS
N79 CONCUSSION
N80 HEAD INJURY OTHER

Cluster  5
P78 NEURAESTHENIA/SURMENAGE
R72 STREP THROAT
R74 UPPER RESPIRATORY INFECTION ACUTE
R75 SINUSITIS ACUTE/CHRONIC
R76 TONSILLITIS ACUTE
W01 QUESTION OF PREGNANCY
W03 ANTEPARTUM BLEEDING
W10 CONTRACEPTION POSTCOITAL
W11 CONTRACEPTION ORAL
W12 CONTRACEPTION INTRAUTERINE
W14 CONTRACEPTION OTHER
W78 PREGNANCY
W84 PREGNANCY HIGH RISK
W94 PUERPERAL MASTITIS
X01 GENITAL PAIN FEMALE
X02 MENSTRUAL PAIN
X06 MENSTRUATION EXCESSIVE
X07 MENSTRUATION IRREGULAR/FREQUENT
X08 INTERMENSTRUAL BLEEDING
X14 VAGINAL DISCHARGE
X17 PELVIS SYMPTOM/COMPLAINT FEMALE
X72 GENITAL CANDIDIASIS FEMALE
X74 PELVIC INFLAMMATORY DISEASE
X84 VAGINITIS/VULVITIS NOS

Cluster  6
A85 ADVERSE EFFECT MEDICAL AGENT
D02 ABDOMINAL PAIN EPIGASTRIC
D16 RECTAL BLEEDING
H03 TINNITUS, RINGING/BUZZING EAR
K96 HAEMORRHOIDS
L18 MUSCLE PAIN
L19 MUSCLE SYMPTOM/COMPLAINT NOS
L29 SYMPTOM/COMPLAINT MUSCULOSKELETAL
    OTHER
P06 SLEEP DISTURBANCE
S10 BOIL/CARBUNCLE
S11 SKIN INFECTION POST-TRAUMATIC
S29 SKIN SYMPTOM/COMPLAINT OTHER
S80 SOLAR KERATOSIS/SUNBURN

Cluster  7
D20 MOUTH/TONGUE/LIP SYMPTOM/COMPLAINT
F72 BLEPHARITIS/STYE/CHALAZION
H29 EAR SYMPTOM/COMPLAINT OTHER
H70 OTITIS EXTERNA
S12 INSECT BITE/STING
S86 DERMATITIS SEBORRHOEIC
S88 DERMATITIS CONTACT/ALLERGIC
S91 PSORIASIS

Cluster  8
D10 VOMITING
D11 DIARRHOEA
D12 CONSTIPATION
D93 IRRITABLE BOWEL SYNDROME
F03 EYE DISCHARGE
F70 CONJUNCTIVITIS INFECTIOUS
K04 PALPITATIONS/AWARENESS OF HEART
K05 IRREGULAR HEARTBEAT OTHER
K27 FEAR OF CARDIOVASCULAR DISEASE
    OTHER
```

```
K79 PAROXYSMAL TACHYCARDIA
L20 JOINT SYMPTOM/COMPLAINT NOS
L88 RHEUMATOID/SEROPOSITIVE ARTHRITIS

Cluster  9
F02 RED EYE
F73 EYE INFECTION/INFLAMMATION OTHER
F93 GLAUCOMA
F99 EYE/ADNEXA DISEASE OTHER
L13 HIP SYMPTOM/COMPLAINT
L89 OSTEOARTHROSIS OF HIP
S75 MONILIASIS/CANDIDIASIS SKIN
S99 SKIN DISEASE OTHER
T86 HYPOTHYROIDISM/MYXOEDEMA
U04 INCONTINENCE URINE
X87 UTEROVAGINAL PROLAPSE

Cluster  10
A12 transferred to A92
B02 LYMPH GLAND(S) ENLARGED/PAINFUL
D19 TEETH/GUM SYMPTOM/COMPLAINT
D73 GASTROENTERITIS PRESUMED INFECTION
D82 TEETH/GUM DISEASE
F71 CONJUNCTIVITIS ALLERGIC
L01 NECK SYMPTOM/COMPLAINT
L81 INJURY MUSCULOSKELETAL NOS
L83 NECK SYNDROME
R05 COUGH
R07 SNEEZING/NASAL CONGESTION
R29 RESPIRATORY SYMPTOM/COMPLAINT OTHER
R80 INFLUENZA
R83 RESPIRATORY INFECTION OTHER
R97 ALLERGIC RHINITIS
S96 ACNE

Cluster  11
A27 FEAR OF OTHER DISEASE NOS
L17 FOOT/TOE SYMPTOM/COMPLAINT
N05 TINGLING FINGERS/FEET/TOES
R02 SHORTNESS OF BREATH/DYSPNOEA
R04 BREATHING PROBLEM OTHER
R08 NOSE SYMPTOM/COMPLAINT OTHER
R78 ACUTE BRONCHITIS/BRONCHIOLITIS
R90 HYPERTROPHY TONSILS/ADENOIDS
R96 ASTHMA
R99 RESPIRATORY DISEASE OTHER
S02 PRURITUS
S03 WARTS

Cluster  12
D83 MOUTH/TONGUE/LIP DISEASE
F75 CONTUSION/HAEMORRHAGE EYE
F76 FOREIGN BODY IN EYE
L10 ELBOW SYMPTOM/COMPLAINT
L11 WRIST SYMPTOM/COMPLAINT
L16 ANKLE SYMPTOM/COMPLAINT
L72 FRACTURE: RADIUS/ULNA
S13 ANIMAL/HUMAN BITE
S15 FOREIGN BODY IN SKIN
S18 LACERATION/CUT

Cluster  13
D03 HEARTBURN
F13 EYE SENSATION ABNORMAL
F29 EYE SYMPTOM/COMPLAINT OTHER
L02 BACK SYMPTOM/COMPLAINT
L03 LOW BACK SYMPTOM/COMPLAINT
L04 CHEST SYMPTOM/COMPLAINT
L84 BACK SYNDROME WITHOUT RADIATING
    PAIN
L86 BACK SYNDROME WITH RADIATING PAIN
L92 SHOULDER SYNDROME
L93 TENNIS ELBOW
L99 MUSCULOSKELETAL DISEASE OTHER
N01 HEADACHE
N02 transferred to N95
N89 MIGRAINE
P01 FEELING ANXIOUS/NERVOUS/TENSE
P02 ACUTE STRESS REACTION
P03 FEELING DEPRESSED
P74 ANXIETY DISORDER/ANXIETY STATE
P76 DEPRESSIVE DISORDER
R21 THROAT SYMPTOM/COMPLAINT
S04 LUMP/SWELLING LOCALIZED
S78 LIPOMA
S79 NEOPLASM SKIN BENIGN/UNSPECIFIED
S82 NAEVUS/MOLE
S93 SEBACEOUS CYST
T93 LIPID DISORDER
X11 MENOPAUSAL SYMPTOM/COMPLAINT
X12 POSTMENOPAUSAL BLEEDING
X19 BREAST LUMP/MASS FEMALE

Cluster  14
A78 INFECTIOUS DISEASE OTHER/NOS
L08 SHOULDER SYMPTOM/COMPLAINT
L09 ARM SYMPTOM/COMPLAINT
L12 HAND/FINGER SYMPTOM/COMPLAINT
L15 KNEE SYMPTOM/COMPLAINT
L74 FRACTURE: HAND/FOOT BONE
L78 SPRAIN/STRAIN OF KNEE
L90 OSTEOARTHROSIS OF KNEE
L96 ACUTE INTERNAL DAMAGE KNEE
L97 NEOPLASM BENIGN/UNSPECIFIED
    MUSCULOSKELETAL
S06 RASH LOCALIZED
S09 INFECTED FINGER/TOE
S14 BURN/SCALD
S16 BRUISE/CONTUSION
S17 ABRASION/SCRATCH/BLISTER
S22 NAIL SYMPTOM/COMPLAINT
S94 INGROWING NAIL

Cluster  15
B85 included with A91
D89 INGUINAL HERNIA
F05 VISUAL DISTURBANCE OTHER
F92 CATARACT
H02 HEARING COMPLAINT
H81 EXCESSIVE EAR WAX
H84 PRESBYACUSIS
K07 SWOLLEN ANKLES/OEDEMA
K92 ATHEROSCLEROSIS/PERIPHERAL VASCULAR
     DISEASE
K94 PHLEBITIS/THROMBOPHLEBITIS
K95 VARICOSE VEINS OF LEG
L14 LEG/THIGH SYMPTOM/COMPLAINT
R91 moved to R79
S97 CHRONIC ULCER SKIN
T90 DIABETES NON-INSULIN DEPENDENT
```

```
U02 URINARY FREQUENCY/URGENCY
%\end{verbatim}
%\relsize{+2}
```

B.2 Medical certificate clusterings

Temporary medical certificates with basis weighting

```
Size of cluster: 1675
Avg age 44.4676129543
Age st.dev 3.73302852441
Gift 0.594626865672 Samboende 0.271641791045 Uregistrert 0.0811940298507
Enslig 0.044776119403 Skilt 0.00537313432836 Enke_enkemann 0.00238805970149
Yrkesaktiv 0.928358208955 Trygdet 0.0405970149254 Student 0.0191044776119
Pensjonert 0.0107462686567 Uregistrert 0.00119402985075
Sex M 1.0
Codes L84 0.105671641791 L81 0.0811940298507 L92 0.070447761194 L93 0.0644776119403
L86 0.0405970149254
Codegroups L 0.709253731343 N 0.0680597014925 R 0.0668656716418 A 0.054328358209
P 0.035223880597 S 0.0208955223881 H 0.014328358209 U 0.00776119402985 K 0.00776119402985
Z 0.00417910447761 Y 0.00358208955224 T 0.00238805970149 D 0.00238805970149
B 0.00179104477612 F 0.00119402985075

Size of cluster: 2844
Avg age 44.1569786204
Age st.dev 3.57911677503
Gift 0.8435302391 Uregistrert 0.0559071729958 Samboende 0.0450070323488
Enslig 0.0393811533052 Skilt 0.0133614627286 Enke_enkemann 0.0028129395218
Yrkesaktiv 0.788326300985 Uregistrert 0.118846694796 Hjemmevarende 0.0523909985935
Trygdet 0.0246132208158 Student 0.0070323488045 Pensjonert 0.00632911392405
Arbeidsledig 0.00246132208158
Sex F 0.99964838256 M 0.000351617440225
Codes L81 0.0625879043601 L84 0.059423347398 R801 0.0541490857947 L93 0.0534458509142
L92 0.0397327707454
Codegroups L 0.478551336146 R 0.161040787623 P 0.0879043600563 D 0.0629395218003
N 0.0460618846695 A 0.0432489451477 W 0.0376230661041 K 0.0260196905767
H 0.0203938115331 F 0.0116033755274 T 0.010900140647 S 0.00597749648383
X 0.0042194092827 U 0.00210970464135 B 0.0014064697609

Size of cluster: 672
Avg age 46.0701904
Age st.dev 3.39320692027
Gift 0.546130952381 Skilt 0.254464285714 Enke_enkemann 0.0818452380952
Samboende 0.0431547619048 Uregistrert 0.0401785714286 Enslig 0.0342261904762
Yrkesaktiv 0.986607142857 Pensjonert 0.00744047619048 Hjemmevarende 0.00595238095238
Sex M 0.745535714286 F 0.254464285714
Codes P76 0.0580357142857 K74 0.0416666666667 P02 0.0386904761905 D73 0.0297619047619
D02 0.0282738095238
Codegroups K 0.227678571429 P 0.183035714286 L 0.169642857143 D 0.168154761905
R 0.0892857142857 F 0.0610119047619 S 0.0208333333333 B 0.0208333333333
Y 0.014880952381 T 0.0133928571429 N 0.0133928571429 A 0.0119047619048
U 0.00297619047619 H 0.00297619047619

Size of cluster: 1437
Avg age 42.0551803678
Age st.dev 3.42358569283
Samboende 0.611691022965 Gift 0.157967988866 Uregistrert 0.107167710508
Enslig 0.0869867780097 Skilt 0.0187891440501 Enke_enkemann 0.0173973556019
Yrkesaktiv 0.939457202505 Uregistrert 0.0375782881002 Hjemmevarende 0.0125260960334
Student 0.00556715379262 Arbeidsledig 0.00347947112039 Trygdet 0.00139178844816
Sex F 0.918580375783 M 0.0814196242171
Codes R801 0.044537230341 S18 0.0396659707724 L84 0.0396659707724 P76 0.0313152400835
L99 0.027139874739
Codegroups L 0.2832289492 R 0.153792623521 S 0.130132219903 P 0.107863604732
D 0.07167710508 W 0.0695894224078 X 0.0598469032707 N 0.0403618649965 A 0.0389700765484
U 0.0250521920668 Z 0.0062630480167 K 0.0062630480167 H 0.00487125956855 B 0.00208768267223
```

```
Size of cluster: 533
Avg age 43.7448713579
Age st.dev 3.85989421106
Gift 0.600375234522 Uregistrert 0.215759849906 Enslig 0.157598499062
Samboende 0.0168855534709 Enke_enkemann 0.00562851782364 Skilt 0.00375234521576
Yrkesaktiv 0.932457786116 Uregistrert 0.0675422138837
Sex M 0.701688555347 F 0.298311444653
Codes R801 0.262664165103 R75 0.155722326454 R74 0.114446529081 R83 0.0994371482176
R78 0.0994371482176
Codegroups R 1.0

Size of cluster: 640
Avg age 41.5114596941
Age st.dev 4.04269225701
Enslig 0.7671875 Uregistrert 0.1890625 Skilt 0.0171875 Samboende 0.015625 Gift 0.0109375
Yrkesaktiv 0.621875 Uregistrert 0.3296875 Arbeidsledig 0.021875 Trygdet 0.01875
Student 0.0046875 Pensjonert 0.003125
Sex M 0.990625 F 0.009375
Codes L84 0.096875 L81 0.0640625 L93 0.053125 L92 0.053125 L76 0.0390625
Codegroups L 0.65 R 0.08125 P 0.078125 D 0.078125 S 0.06875 K 0.0171875 T 0.00625
N 0.00625 Y 0.0046875 U 0.003125 H 0.003125 B 0.0015625 A 0.0015625

Size of cluster: 165
Avg age 40.8175985687
Age st.dev 3.20842819062
Uregistrert 0.939393939394 Samboende 0.0242424242424 Skilt 0.0181818181818
Enslig 0.0121212121212 Gift 0.00606060606061
Uregistrert 0.666666666667 Yrkesaktiv 0.321212121212 Hjemmevarende 0.00606060606061
Arbeidsledig 0.00606060606061
Sex M 0.6 F 0.4
Codes P76 0.133333333333 P02 0.121212121212 P78 0.0969696969697 P03 0.0848484848485
D73 0.0606060606061
Codegroups P 0.618181818182 D 0.139393939394 A 0.0848484848485 S 0.0787878787879
N 0.0545454545455 F 0.0121212121212 Z 0.00606060606061 H 0.00606060606061
```

Long-term medical certificates with basis weighting

```
Size of cluster: 396
Avg age 46.5431918801
Age st.dev 10.974055071
Gift 0.724747474747 Samboende 0.136363636364 Uregistrert 0.0681818181818
Enslig 0.0429292929293 Skilt 0.0277777777778
Yrkesaktiv 0.861111111111 Uregistrert 0.136363636364 Hjemmevarende 0.00252525252525
Sex M 0.888888888889 F 0.111111111111
Codes L84 0.164141414141 L92 0.136363636364 L86 0.108585858586 L93 0.0833333333333
L83 0.0656565656566
Codegroups L 0.997474747475 N 0.00252525252525

Size of cluster: 563
Avg age 51.7251505154
Age st.dev 8.48059576763
Gift 0.806394316163 Samboende 0.0550621669627 Uregistrert 0.0426287744227
Enke_enkemann 0.0355239786856 Enslig 0.0319715808171 Skilt 0.0284191829485
Yrkesaktiv 0.836589698046 Uregistrert 0.056838365897 Hjemmevarende 0.0550621669627
Trygdet 0.0284191829485 Pensjonert 0.0230905861456
Sex F 0.959147424512 M 0.0408525754885
Codes L93 0.103019538188 L84 0.0817051509769 L92 0.0763765541741 L86 0.056838365897
L99 0.0373001776199
Codegroups L 0.721136767318 X 0.056838365897 D 0.0550621669627 K 0.0497335701599
A 0.0390763765542 P 0.0301953818828 T 0.0213143872114 S 0.0142095914742 U 0.00355239786856
```

```
R 0.00355239786856 N 0.00355239786856 F 0.00177619893428

Size of cluster: 358
Avg age 38.1909015039
Age st.dev 10.6228730557
Gift 0.40782122905 Samboende 0.346368715084 Enslig 0.145251396648 Uregistrert 0.0446927374302
Skilt 0.036312849162 Enke_enkemann 0.0195530726257
Yrkesaktiv 0.988826815642 Student 0.00558659217877 Arbeidsledig 0.00279329608939
Uregistrert 0.00279329608939
Sex F 0.991620111732 M 0.00837988826816
Codes P78 0.131284916201 P76 0.0810055865922 L99 0.0502793296089 W05 0.0391061452514
L84 0.0391061452514
Codegroups P 0.31843575419 L 0.265363128492 N 0.0893854748603 W 0.0698324022346
R 0.0586592178771 A 0.0558659217877 S 0.0335195530726 K 0.0279329608939 T 0.0167597765363
D 0.0167597765363 B 0.0167597765363 F 0.0139664804469 X 0.00837988826816 Z 0.00558659217877
H 0.00279329608939

Size of cluster: 425
Avg age 52.9321306737
Age st.dev 9.38503763352
Gift 0.562352941176 Enslig 0.155294117647 Samboende 0.115294117647
Uregistrert 0.0870588235294 Skilt 0.0611764705882 Enke_enkemann 0.0188235294118
Yrkesaktiv 0.849411764706 Trygdet 0.0729411764706 Uregistrert 0.0588235294118
Pensjonert 0.0141176470588 Student 0.00470588235294
Sex M 0.936470588235 F 0.0635294117647
Codes P76 0.176470588235 K74 0.0682352941176 P03 0.04 K76 0.0376470588235
K90 0.0305882352941
Codegroups P 0.327058823529 K 0.247058823529 L 0.143529411765 N 0.0870588235294
T 0.0376470588235 R 0.0305882352941 S 0.0282352941176 H 0.0211764705882 D 0.0188235294118
A 0.0188235294118 U 0.0117647058824 F 0.0117647058824 B 0.0117647058824 Y 0.00470588235294

Size of cluster: 216
Avg age 35.9813797788
Age st.dev 9.23679992828
Gift 0.509259259259 Samboende 0.365740740741 Uregistrert 0.0833333333333
Skilt 0.0277777777778 Enslig 0.0138888888889
Yrkesaktiv 0.62037037037 Hjemmevarende 0.152777777778 Uregistrert 0.106481481481
Trygdet 0.0648148148148 Student 0.037037037037 Arbeidsledig 0.0185185185185
Sex F 0.99537037037 M 0.00462962962963
Codes L99 0.125 L84 0.101851851852 W84 0.0925925925926 H82 0.0648148148148
P76 0.0555555555556
Codegroups L 0.384259259259 W 0.175925925926 P 0.143518518519 N 0.12037037037
H 0.0925925925926 T 0.0185185185185 D 0.0185185185185 A 0.0138888888889
R 0.00925925925926 K 0.00925925925926 kommet hit S 0.00462962962963 F 0.00462962962963
B 0.00462962962963

Size of cluster: 256
Avg age 30.3348011751
Age st.dev 7.38516377405
Uregistrert 0.5546875 Enslig 0.359375 Samboende 0.0546875 Gift 0.02734375 Skilt 0.00390625
Yrkesaktiv 0.5546875 Uregistrert 0.38671875 Student 0.03515625 Arbeidsledig 0.015625
Hjemmevarende 0.0078125
Sex M 0.73046875 F 0.26953125
Codes L84 0.0859375 L86 0.0546875 L92 0.05078125 L99 0.046875 L93 0.046875
Codegroups L 0.52734375 P 0.12890625 N 0.08984375 D 0.0625 S 0.05859375 A 0.0390625
R 0.03515625 F 0.015625 Z 0.01171875 W 0.0078125 T 0.0078125 K 0.0078125 X 0.00390625
U 0.00390625

Size of cluster: 8
Avg age 93.7945956617
Age st.dev 3.8586627109
Enslig 0.625 Uregistrert 0.375
Uregistrert 0.75 Pensjonert 0.125 Yrkesaktiv 0.125
Sex M 0.875 F 0.125
Codes R31 0.125 L92 0.125 L84 0.125 L76 0.125 K86 0.125
```

```
Codegroups L 0.375 - 0.25 R 0.125 K 0.125 D 0.125
```

Temporary medical certificates with 1.5 weighting of code/code-group

```
Size of cluster: 3465
Avg age 43.7596768396
Age st.dev 3.78991586967
Gift 0.564790764791 Samboende 0.17316017316 Enslig 0.122655122655
Uregistrert 0.0966810966811 Skilt 0.0305916305916 Enke_enkemann 0.0121212121212
Yrkesaktiv 0.854834054834 Uregistrert 0.0796536796537 Hjemmevarende 0.023088023088
Trygdet 0.023088023088 Student 0.00894660894661 Pensjonert 0.00663780663781
Arbeidsledig 0.0037518037518
Sex F 0.530735930736 M 0.469264069264
Codes L84 0.138816738817 L81 0.107070707071 L93 0.0989898989899 L92 0.0877344877345
L86 0.0551226551227
Codegroups L 1.0

Size of cluster: 1694
Avg age 43.443194177
Age st.dev 3.70171752482
Gift 0.512987012987 Samboende 0.217827626919 Uregistrert 0.126328217237
Enslig 0.095041322314 Skilt 0.0312868949233 Enke_enkemann 0.0165289256198
Yrkesaktiv 0.845926800472 Uregistrert 0.10507674144 Hjemmevarende 0.025974025974
Trygdet 0.0112160566706 Student 0.00590318772137 Pensjonert 0.0047225501771
Arbeidsledig 0.00118063754427
Sex F 0.691263282172 M 0.308736717828
Codes P76 0.116883116883 P02 0.0914994096812 P03 0.0714285714286 P78 0.0584415584416
N01 0.0489964580874
Codegroups P 0.436245572609 N 0.191853600945 W 0.122195985832 X 0.0578512396694
H 0.0554899645809 F 0.0460448642267 U 0.0348288075561 T 0.0277449822904 B 0.0147579693034
Z 0.0100354191263 K 0.00177095631641 R 0.00118063754427

Size of cluster: 1433
Avg age 43.4867363013
Age st.dev 3.76817562886
Gift 0.539427773901 Samboende 0.205861828332 Enslig 0.103279832519 Uregistrert 0.101884159107
Skilt 0.0411723656664 Enke_enkemann 0.00837404047453
Yrkesaktiv 0.857641311933 Uregistrert 0.0879274249826 Trygdet 0.0202372644801
Hjemmevarende 0.0195394277739 Student 0.00907187718074 Arbeidsledig 0.00418702023726
Pensjonert 0.00139567341242
Sex F 0.618981158409 M 0.381018841591
Codes R801 0.290300069784 R74 0.13258897418 R83 0.130495464061 R78 0.108164689463
R75 0.0921144452198
Codegroups R 1.0

Size of cluster: 448
Avg age 43.4733158295
Age st.dev 3.89252252206
Gift 0.542410714286 Samboende 0.21875 Enslig 0.149553571429 Uregistrert 0.0535714285714
Skilt 0.0290178571429 Enke_enkemann 0.00669642857143
Yrkesaktiv 0.879464285714 Uregistrert 0.0669642857143 Trygdet 0.0178571428571
Hjemmevarende 0.015625 Arbeidsledig 0.0111607142857 Student 0.00669642857143
Pensjonert 0.00223214285714
Sex F 0.520089285714 M 0.479910714286
Codes D73 0.258928571429 D02 0.151785714286 D06 0.0736607142857 D87 0.0602678571429
D01 0.0580357142857
Codegroups D 1.0

Size of cluster: 324
Avg age 46.2321985784
Age st.dev 4.31586019626
```

```
Gift 0.583333333333 Uregistrert 0.188271604938 Enslig 0.0864197530864
Samboende 0.0864197530864 Skilt 0.0401234567901 Enke_enkemann 0.0154320987654
Yrkesaktiv 0.737654320988 Uregistrert 0.206790123457 Trygdet 0.037037037037
Pensjonert 0.0154320987654 Student 0.00308641975309
Sex M 0.654320987654 F 0.345679012346
Codes K74 0.141975308642 K86 0.0956790123457 K01 0.0833333333333 K76 0.0679012345679
K90 0.0617283950617
Codegroups K 0.793209876543 D 0.0740740740741 L 0.0648148148148 Y 0.0586419753086
T 0.00308641975309 S 0.00308641975309 R 0.00308641975309

Size of cluster: 293
Avg age 43.3296222342
Age st.dev 4.0898516799
Gift 0.470989761092 Samboende 0.225255972696 Uregistrert 0.136518771331
Enslig 0.126279863481 Skilt 0.037542662116 Enke_enkemann 0.00341296928328
Yrkesaktiv 0.815699658703 Uregistrert 0.133105802048 Hjemmevarende 0.0307167235495
Student 0.00682593856655 Trygdet 0.00682593856655 Pensjonert 0.00682593856655
Sex F 0.631399317406 M 0.368600682594
Codes A04 0.327645051195 A77 0.174061433447 A80 0.122866894198 A03 0.0716723549488
A12 0.0648464163823
Codegroups A 1.0

Size of cluster: 309
Avg age 43.330142342
Age st.dev 3.90383996895
Gift 0.478964401294 Samboende 0.187702265372 Uregistrert 0.152103559871
Enslig 0.148867313916 Skilt 0.0194174757282 Enke_enkemann 0.0129449838188
Yrkesaktiv 0.847896440129 Uregistrert 0.113268608414 Hjemmevarende 0.0129449838188
Student 0.00970873786408 Trygdet 0.00647249190939 Pensjonert 0.00647249190939
Arbeidsledig 0.00323624595469
Sex M 0.553398058252 F 0.446601941748
Codes S18 0.323624595469 S10 0.0647249190939 S88 0.0614886731392 S11 0.0485436893204
S14 0.0453074433657
Codegroups S 1.0
```

Long-term medical certificates with 1.5 weighting of code/code-group

```
Size of cluster: 1170
Avg age 44.4103537806
Age st.dev 12.2647571204
Gift 0.581196581197 Samboende 0.151282051282 Enslig 0.113675213675
Uregistrert 0.112820512821 Skilt 0.0282051282051 Enke_enkemann 0.0128205128205
Yrkesaktiv 0.812820512821 Uregistrert 0.100854700855 Hjemmevarende 0.0324786324786
Trygdet 0.0324786324786 Student 0.00854700854701 Pensjonert 0.00854700854701
Arbeidsledig 0.0042735042735
Sex F 0.57264957265 M 0.42735042735
Codes L84 0.151282051282 L92 0.105128205128 L93 0.103418803419 L86 0.0897435897436
L99 0.0735042735043
Codegroups L 1.0

Size of cluster: 550
Avg age 49.2200250396
Age st.dev 11.0446126414
Gift 0.589090909091 Samboende 0.147272727273 Enslig 0.109090909091
Uregistrert 0.0745454545455 Skilt 0.0490909090909 Enke_enkemann 0.0309090909091
Yrkesaktiv 0.841818181818 Uregistrert 0.0618181818182 Hjemmevarende 0.0345454545455
Trygdet 0.0327272727273 Student 0.0127272727273 Pensjonert 0.0127272727273
Arbeidsledig 0.00363636363636
Sex F 0.518181818182 M 0.481818181818
Codes P76 0.22 P78 0.116363636364 P03 0.0727272727273 K74 0.06 K86 0.04
Codegroups P 0.529090909091 K 0.261818181818 R 0.0854545454545 H 0.0545454545455
```

```
F 0.0290909090909 U 0.0127272727273 B 0.0109090909091 L 0.00909090909091
Y 0.00363636363636 S 0.00181818181818 N 0.00181818181818

Size of cluster: 65
Avg age 45.6195413426
Age st.dev 13.2836334514
Gift 0.615384615385 Enslig 0.169230769231 Samboende 0.123076923077 Skilt 0.0461538461538
Uregistrert 0.0461538461538
Yrkesaktiv 0.876923076923 Uregistrert 0.0769230769231 Arbeidsledig 0.0307692307692
Hjemmevarende 0.0153846153846
Sex M 0.661538461538 F 0.338461538462
Codes D02 0.169230769231 D94 0.107692307692 D99 0.0923076923077 D01 0.0615384615385
D98 0.0461538461538
Codegroups D 1.0

Size of cluster: 203
Avg age 40.5733960182
Age st.dev 12.2220646093
Gift 0.532019704433 Samboende 0.231527093596 Uregistrert 0.142857142857
Enslig 0.0738916256158 Skilt 0.0147783251232 Enke_enkemann 0.00492610837438
Yrkesaktiv 0.847290640394 Uregistrert 0.118226600985 Hjemmevarende 0.0246305418719
Trygdet 0.00985221674877
Sex F 0.773399014778 M 0.226600985222
Codes W84 0.118226600985 W05 0.0837438423645 T90 0.0738916256158 S88 0.0541871921182
X76 0.0394088669951
Codegroups W 0.320197044335 S 0.231527093596 T 0.197044334975 X 0.177339901478
B 0.0295566502463 Z 0.0246305418719 K 0.0147783251232 U 0.00492610837438

Size of cluster: 180
Avg age 43.2488524887
Age st.dev 11.6579210411
Gift 0.505555555556 Samboende 0.183333333333 Uregistrert 0.155555555556
Enslig 0.105555555556 Skilt 0.0388888888889 Enke_enkemann 0.0111111111111
Yrkesaktiv 0.811111111111 Uregistrert 0.122222222222 Hjemmevarende 0.0222222222222
Student 0.0166666666667 Trygdet 0.0166666666667 Pensjonert 0.0111111111111
Sex F 0.561111111111 M 0.438888888889
Codes A04 0.172222222222 N01 0.155555555556 N17 0.122222222222 N02 0.0666666666667
N99 0.0444444444444
Codegroups N 0.666666666667 A 0.333333333333

Size of cluster: 46
Avg age 26.9773567503
Age st.dev 6.13951980843
Uregistrert 0.673913043478 Enslig 0.217391304348 Samboende 0.108695652174
Uregistrert 0.673913043478 Yrkesaktiv 0.304347826087 Student 0.0217391304348
Sex M 0.673913043478 F 0.326086956522
Codes P76 0.173913043478 P74 0.173913043478 P99 0.0652173913043
P79 0.0652173913043 P29 0.0652173913043
Codegroups P 0.934782608696 A 0.0652173913043

Size of cluster: 8
Avg age 93.7945956617
Age st.dev 3.8586627109
Enslig 0.625 Uregistrert 0.375
Uregistrert 0.75 Pensjonert 0.125 Yrkesaktiv 0.125
Sex M 0.875 F 0.125
Codes R31 0.125 L92 0.125 L84 0.125 L76 0.125 K86 0.125
Codegroups L 0.375 - 0.25 R 0.125 K 0.125 D 0.125
```

C ICPC code standard

International Classification of Primary Care (ICPC) is a code standard developed by the Wonca International Classification Committee. ICPC was published in 1987 and is now available in more than twenty languages. The second edition, ICPC-2, was published in April 98.

The code system has 17 main categories, which again are divided into three subgroups: symptoms or complaints, process and diagnoses. These groups are:

- Symptoms and complaints: 1-29
- Process: 30-69
- Diagnoses: 70-99

The ICPC code system are explored in (Com04). Some information can also be found on (ICP).

The main categories of ICPC codes and the specific codes for symptoms, complaints and diagnoses are given in the following.

A	General and unspecified
B	Blood, blood forming organs, lymphatics, spleen
D	Digestive
F	Eye
H	Ear
K	Circulatory
L	Musculosceletal
N	Neurological
P	Psychological
R	Respiratory
S	Skin
T	Endocrine, metabolic and nutritional
U	Urology
W	Pregnancy, childbirth, family planning
X	Female genital system and breast
Y	Male genital system
Z	Social problems

Table 7: ICPC code system

A01	PAIN GENERAL/MULTIPLE SITES
A02	CHILLS
A03	FEVER
A04	WEAKNESS/TIREDNESS GENERAL
A05	FEELING ILL
A06	FAINTING/SYNCOPE
A07	COMA
A08	SWELLING
A09	SWEATING PROBLEM
A10	BLEEDING/HAEMORRHAGE NOS
A11	CHEST PAIN NOS
A12	transferred to A92
A13	CONCERN ABOUT/FEAR OF MEDICAL TREATMENT
A14	included with D01
A15	included with A16
A16	IRRITABLE INFANT
A17	included with A16
A18	CONCERN ABOUT APPEARANCE
A20	EUTHANASIA REQUEST/DISCUSSION
A21	RISK FACTOR FOR MALIGNANCY
A23	RISK FACTOR NOS
A25	FEAR OF DEATH/DYING
A26	FEAR OF CANCER NOS
A27	FEAR OF OTHER DISEASE NOS
A28	LIMITED FUNCTION/DISABILITY NOS
A29	GENERAL SYMPTOM/COMPLAINT OTHER
A70	TUBERCULOSIS
A71	MEASLES
A72	CHICKENPOX
A73	MALARIA
A74	RUBELLA
A75	INFECTIOUS MONONUCLEOSIS
A76	VIRAL EXANTHEM OTHER
A77	VIRAL DISEASE OTHER/NOS
A78	INFECTIOUS DISEASE OTHER/NOS
A79	MALIGNANCY NOS
A80	TRAUMA/INJURY NOS
A81	MULTIPLE TRAUMA/INJURIES
A82	SECONDARY EFFECT OF TRAUMA
A84	POISONING BY MEDICAL AGENT
A85	ADVERSE EFFECT MEDICAL AGENT
A86	TOXIC EFFECT NON-MEDICINAL SUBSTANCE
A87	COMPLICATION OF MEDICAL TREATMENT
A88	ADVERSE EFFECT PHYSICAL FACTOR
A89	EFFECT PROSTHETIC DEVICE
A90	CONGENITAL ANOMALY NOS/MULTIPLE
A91	ABNORMAL RESULT INVESTIGATION NOS
A92	ALLERGY/ALLERGIC REACTION NOS
A93	PREMATURE NEWBORN
A94	PERINATAL MORBIDITY OTHER
A95	PERINATAL MORTALITY
A96	DEATH
A97	NO DISEASE
A98	HEALTH MAINTENANCE/PREVENTIVE MEDICINE
A99	DISEASE/CONDITION OF UNSPECIFIED NATURE/SITE
B02	LYMPH GLAND(S) ENLARGED/PAINFUL
B03	included with B02
B04	BLOOD SYMPTOM/COMPLAINT
B25	FEAR OF AIDS/HIV
B26	FEAR OF CANCER BLOOD/LYMPH
B27	FEAR OF BLOOD/LYMPH DISEASE OTHER
B28	LIMITED FUNCTION/DISABILITY (B)
B29	SYMPTOM/COMPLAINT LYMPH/IMMUNE MECHANISM OTHER
B70	LYMPHADENITIS ACUTE
B71	LYMPHADENITIS CHRONIC/NON-SPECIFIC
B72	HODGKIN'S DISEASE/LYMPHOMA
B73	LEUKAEMIA
B74	MALIGNANT NEOPLASM BLOOD OTHER
B75	BENIGN/UNSPECIFIED NEOPLASM BLOOD
B76	RUPTURED SPLEEN TRAUMATIC
B77	INJURY BLOOD/LYMPH/SPLEEN OTHER
B78	HEREDITARY HAEMOLYTIC ANAEMIA
B79	CONGENITAL ANOMALY BLOOD/LYMPH OTHER
B80	IRON DEFICIENCY ANAEMIA
B81	ANAEMIA VIT B12/FOLATE DEFICIENCY
B82	ANAEMIA OTHER/UNSPECIFIED
B83	PURPURA/COAGULATION DEFECT
B84	UNEXPLAINED ABNORMAL WHITE CELLS
B85	included with A91
B86	included with B99
B87	SPLENOMEGALY
B90	HIV-INFECTION/AIDS
B99	BLOOD/LYMPH/SPLEEN DISEASE OTHER
D01	ABDOMINAL PAIN/CRAMPS GENERAL
D02	ABDOMINAL PAIN EPIGASTRIC
D03	HEARTBURN
D04	RECTAL/ANAL PAIN
D05	PERIANAL ITCHING
D06	ABDOMINAL PAIN LOCALIZED OTHER
D07	DYSPEPSIA/INDIGESTION
D08	FLATULENCE/GAS/BELCHING
D09	NAUSEA
D10	VOMITING
D11	DIARRHOEA
D12	CONSTIPATION
D13	JAUNDICE
D14	HAEMATEMESIS/VOMITING BLOOD
D15	MELAENA
D16	RECTAL BLEEDING
D17	INCONTINENCE OF BOWEL
D18	CHANGE IN FAECES/BOWEL MOVEMENTS
D19	TEETH/GUM SYMPTOM/COMPLAINT
D20	MOUTH/TONGUE/LIP SYMPTOM/COMPLAINT
D21	SWALLOWING PROBLEM
D22	transferred to D96
D23	HEPATOMEGALY
D24	ABDOMINAL MASS NOS
D25	ABDOMINAL DISTENSION
D26	FEAR OF CANCER OF DIGESTIVE SYSTEM
D27	FEAR OF DIGESTIVE DISEASE OTHER
D28	LIMITED FUNCTION/DISABILITY (D)
D29	DIGESTIVE SYMPTOM/COMPLAINT OTHER
D70	GASTROINTESTINAL INFECTION
D71	MUMPS
D72	VIRAL HEPATITIS
D73	GASTROENTERITIS PRESUMED INFECTION
D74	MALIGNANT NEOPLASM STOMACH
D75	MALIGNANT NEOPLASM COLON/RECTUM
D76	MALIGNANT NEOPLASM PANCREAS
D77	MALIGNANT DIGESTIVE NEOPLASM OTHER/NOS
D78	NEOPLASM DIGESTIVE SYSTEM BENIGN/UNSPECIFIED
D79	FOREIGN BODY DIGESTIVE SYSTEM

D80	INJURY DIGESTIVE SYSTEM OTHER
D81	CONGENITAL ANOMALY DIGESTIVE SYSTEM
D82	TEETH/GUM DISEASE
D83	MOUTH/TONGUE/LIP DISEASE
D84	OESOPHAGUS DISEASE
D85	DUODENAL ULCER
D86	PEPTIC ULCER OTHER
D87	STOMACH FUNCTION DISORDER
D88	APPENDICITIS
D89	INGUINAL HERNIA
D90	HIATUS HERNIA
D91	ABDOMINAL HERNIA OTHER
D92	DIVERTICULAR DISEASE
D93	IRRITABLE BOWEL SYNDROME
D94	CHRONIC ENTERITIS/ULCERATIVE COLITIS
D95	ANAL FISSURE/PERIANAL ABSCESS
D96	WORMS/OTHER PARASITES
D97	LIVER DISEASE NOS
D98	CHOLECYSTITIS/CHOLELITHIASIS
D99	DISEASE DIGESTIVE SYSTEM OTHER
F01	EYE PAIN
F02	RED EYE
F03	EYE DISCHARGE
F04	VISUAL FLOATERS/SPOTS
F05	VISUAL DISTURBANCE OTHER
F13	EYE SENSATION ABNORMAL
F14	EYE MOVEMENTS ABNORMAL
F15	EYE APPEARANCE ABNORMAL
F16	EYELID SYMPTOM/COMPLAINT
F17	GLASSES SYMPTOM/COMPLAINT
F18	CONTACT LENS SYMPTOM/COMPLAINT
F27	FEAR OF EYE DISEASE
F28	LIMITED FUNCTION/DISABILITY (F)
F29	EYE SYMPTOM/COMPLAINT OTHER
F70	CONJUNCTIVITIS INFECTIOUS
F71	CONJUNCTIVITIS ALLERGIC
F72	BLEPHARITIS/STYE/CHALAZION
F73	EYE INFECTION/INFLAMMATION OTHER
F74	NEOPLASM OF EYE/ADNEXA
F75	CONTUSION/HAEMORRHAGE EYE
F76	FOREIGN BODY IN EYE
F79	INJURY EYE OTHER
F80	BLOCKED LACRIMAL DUCT OF INFANT
F81	CONGENITAL ANOMALY EYE OTHER
F82	DETACHED RETINA
F83	RETINOPATHY
F84	MACULAR DEGENERATION
F85	CORNEAL ULCER
F86	TRACHOMA
F91	REFRACTIVE ERROR
F92	CATARACT
F93	GLAUCOMA
F94	BLINDNESS
F95	STRABISMUS
F99	EYE/ADNEXA DISEASE OTHER
H01	EAR PAIN/EARACHE
H02	HEARING COMPLAINT
H03	TINNITUS, RINGING/BUZZING EAR
H04	EAR DISCHARGE
H05	BLEEDING EAR
H13	PLUGGED FEELING EAR
H15	CONCERN WITH APPEARANCE OF EARS
H27	FEAR OF EAR DISEASE
H28	LIMITED FUNCTION/DISABILITY (H)
H29	EAR SYMPTOM/COMPLAINT OTHER
H70	OTITIS EXTERNA
H71	ACUTE OTITIS MEDIA/MYRINGITIS
H72	SEROUS OTITIS MEDIA
H73	EUSTACHIAN SALPINGITIS
H74	CHRONIC OTITIS MEDIA
H75	NEOPLASM OF EAR
H76	FOREIGN BODY IN EAR
H77	PERFORATION EAR DRUM
H78	SUPERFICIAL INJURY OF EAR
H79	EAR INJURY OTHER
H80	CONGENITAL ANOMALY OF EAR
H81	EXCESSIVE EAR WAX
H82	VERTIGINOUS SYNDROME
H83	OTOSCLEROSIS
H84	PRESBYACUSIS
H85	ACOUSTIC TRAUMA
H86	DEAFNESS
H99	EAR/MASTOID DISEASE OTHER
K01	HEART PAIN
K02	PRESSURE/TIGHTNESS OF HEART
K03	CARDIOVASCULAR PAIN NOS
K04	PALPITATIONS/AWARENESS OF HEART
K05	IRREGULAR HEARTBEAT OTHER
K06	PROMINENT VEINS
K07	SWOLLEN ANKLES/OEDEMA
K22	RISK FACTOR FOR CARDIOVASCULAR DISEASE
K24	FEAR OF HEART DISEASE
K25	FEAR OF HYPERTENSION
K27	FEAR OF CARDIOVASCULAR DISEASE OTHER
K28	LIMITED FUNCTION/DISABILITY (K)
K29	CARDIOVASCULAR SYMPTOM/COMPLAINT OTHER
K70	INFECTION OF CIRCULATORY SYSTEM
K71	RHEUMATIC FEVER/HEART DISEASE
K72	NEOPLASM CARDIOVASCULAR
K73	CONGENITAL ANOMALY CARDIOVASCULAR
K74	ISCHAEMIC HEART DISEASE WITH ANGINA
K75	ACUTE MYOCARDIAL INFARCTION
K76	ISCHAEMIC HEART DISEASE WITHOUT ANGINA
K77	HEART FAILURE
K78	ATRIAL FIBRILLATION/FLUTTER
K79	PAROXYSMAL TACHYCARDIA
K80	CARDIAC ARRHYTHMIA NOS
K81	HEART/ARTERIAL MURMUR NOS
K82	PULMONARY HEART DISEASE
K83	HEART VALVE DISEASE NOS
K84	HEART DISEASE OTHER
K85	ELEVATED BLOOD PRESSURE
K86	HYPERTENSION UNCOMPLICATED
K87	HYPERTENSION COMPLICATED
K88	POSTURAL HYPOTENSION
K89	TRANSIENT CEREBRAL ISCHAEMIA
K90	STROKE/CEREBROVASCULAR ACCIDENT
K91	CEREBROVASCULAR DISEASE
K92	ATHEROSCLEROSIS/PERIPHERAL VASCULAR DISEASE
K93	PULMONARY EMBOLISM
K94	PHLEBITIS/THROMBOPHLEBITIS
K95	VARICOSE VEINS OF LEG
K96	HAEMORRHOIDS
K99	CARDIOVASCULAR DISEASE OTHER

L01	NECK SYMPTOM/COMPLAINT
L02	BACK SYMPTOM/COMPLAINT
L03	LOW BACK SYMPTOM/COMPLAINT
L04	CHEST SYMPTOM/COMPLAINT
L05	FLANK/AXILLA SYMPTOM/COMPLAINT
L06	included with L05
L07	JAW SYMPTOM/COMPLAINT
L08	SHOULDER SYMPTOM/COMPLAINT
L09	ARM SYMPTOM/COMPLAINT
L10	ELBOW SYMPTOM/COMPLAINT
L11	WRIST SYMPTOM/COMPLAINT
L12	HAND/FINGER SYMPTOM/COMPLAINT
L13	HIP SYMPTOM/COMPLAINT
L14	LEG/THIGH SYMPTOM/COMPLAINT
L15	KNEE SYMPTOM/COMPLAINT
L16	ANKLE SYMPTOM/COMPLAINT
L17	FOOT/TOE SYMPTOM/COMPLAINT
L18	MUSCLE PAIN
L19	MUSCLE SYMPTOM/COMPLAINT NOS
L20	JOINT SYMPTOM/COMPLAINT NOS
L26	FEAR OF CANCER MUSCULOSKELETAL
L27	FEAR OF MUSCULOSKELETAL DISEASE OTHER
L28	LIMITED FUNCTION/DISABILITY (L)
L29	SYMPTOM/COMPLAINT MUSCULOSKELETAL OTHER
L70	INFECTIONS OF MUSCULOSKELETAL SYSTEM
L71	MALIGNANT NEOPLASM MUSCULOSKELETAL
L72	FRACTURE: RADIUS/ULNA
L73	FRACTURE: TIBIA/FIBULA
L74	FRACTURE: HAND/FOOT BONE
L75	FRACTURE: FEMUR
L76	FRACTURE: OTHER
L77	SPRAIN/STRAIN OF ANKLE
L78	SPRAIN/STRAIN OF KNEE
L79	SPRAIN/STRAIN OF JOINT NOS
L80	DISLOCATION/SUBLUXATION
L81	INJURY MUSCULOSKELETAL NOS
L82	CONGENITAL ANOMALY MUSCULOSKELETAL
L83	NECK SYNDROME
L84	BACK SYNDROME WITHOUT RADIATING PAIN
L85	ACQUIRED DEFORMITY OF SPINE
L86	BACK SYNDROME WITH RADIATING PAIN
L87	BURSITIS/TENDINITIS/SYNOVITIS NOS
L88	RHEUMATOID/SEROPOSITIVE ARTHRITIS
L89	OSTEOARTHROSIS OF HIP
L90	OSTEOARTHROSIS OF KNEE
L91	OSTEOARTHROSIS OTHER
L92	SHOULDER SYNDROME
L93	TENNIS ELBOW
L94	OSTEOCHONDROSIS
L95	OSTEOPOROSIS
L96	ACUTE INTERNAL DAMAGE KNEE
L97	NEOPLASM BENIGN/UNSPECIFIED MUSCULOSKELETAL
L98	ACQUIRED DEFORMITY OF LIMB
L99	MUSCULOSKELETAL DISEASE OTHER
N01	HEADACHE
N02	transferred to N95
N03	PAIN FACE
N04	RESTLESS LEGS
N05	TINGLING FINGERS/FEET/TOES
N06	SENSATION DISTURBANCE OTHER
N07	CONVULSION/SEIZURE
N08	ABNORMAL INVOLUNTARY MOVEMENTS
N16	DISTURBANCE OF SMELL/TASTE
N17	VERTIGO/DIZZINESS
N18	PARALYSIS/WEAKNESS
N19	SPEECH DISORDER
N26	FEAR OF CANCER OF NEUROLOGICAL SYSTEM
N27	FEAR OF NEUROLOGICAL DISEASE OTHER
N28	LIMITED FUNCTION/DISABILITY (N)
N29	NEUROLOGICAL SYMPTOM/COMPLAINT OTHER
N70	POLIOMYELITIS
N71	MENINGITIS/ENCEPHALITIS
N72	TETANUS
N73	NEUROLOGICAL INFECTION OTHER
N74	MALIGNANT NEOPLASM NERVOUS SYSTEM
N75	BENIGN NEOPLASM NERVOUS SYSTEM
N76	NEOPLASM NERVOUS SYSTEM UNSPECIFIED
N79	CONCUSSION
N80	HEAD INJURY OTHER
N81	INJURY NERVOUS SYSTEM OTHER
N85	CONGENITAL ANOMALY NEUROLOGICAL
N86	MULTIPLE SCLEROSIS
N87	PARKINSONISM
N88	EPILEPSY
N89	MIGRAINE
N90	CLUSTER HEADACHE
N91	FACIAL PARALYSIS/BELL'S PALSY
N92	TRIGEMINAL NEURALGIA
N93	CARPAL TUNNEL SYNDROME
N94	PERIPHERAL NEURITIS/NEUROPATHY
N95	TENSION HEADACHE
N99	NEUROLOGICAL DISEASE OTHER
P01	FEELING ANXIOUS/NERVOUS/TENSE
P02	ACUTE STRESS REACTION
P03	FEELING DEPRESSED
P04	FEELING/BEHAVING IRRITABLE/ANGRY
P05	SENILIITY, FEELING/BEHAVING OLD
P06	SLEEP DISTURBANCE
P07	SEXUAL DESIRE REDUCED
P08	SEXUAL FULFILMENT REDUCED
P09	SEXUAL PREFERENCE CONCERN
P10	STAMMERING/STUTTERING/TIC
P11	EATING PROBLEM IN CHILD
P12	BEDWETTING/ENURESIS
P13	ENCOPRESIS/BOWEL TRAINING PROBLEM
P15	CHRONIC ALCOHOL ABUSE
P16	ACUTE ALCOHOL ABUSE
P17	TOBACCO ABUSE
P18	MEDICATION ABUSE
P19	DRUG ABUSE
P20	MEMORY DISTURBANCE
P21	transferred to P81
P22	CHILD BEHAVIOUR SYMPTOM/COMPLAINT
P23	ADOLESCENT BEHAVIOUR SYMPTOM/COMPLAINT
P24	SPECIFIC LEARNING PROBLEM
P25	PHASE OF LIFE PROBLEM ADULT
P27	FEAR OF MENTAL DISORDER
P28	LIMITED FUNCTION/DISABILITY (P)
P29	PSYCHOLOGICAL SYMPTOM/COMPLAINT OTHER
P70	DEMENTIA
P71	ORGANIC PSYCHOSIS OTHER

```
P72 SCHIZOPHRENIA
P73 AFFECTIVE PSYCHOSIS
P74 ANXIETY DISORDER/ANXIETY STATE
P75 SOMATIZATION DISORDER
P76 DEPRESSIVE DISORDER
P77 SUICIDE/SUICIDE ATTEMPT
P78 NEURAESTHENIA/SURMENAGE
P79 PHOBIA/COMPULSIVE DISORDER
P80 PERSONALITY DISORDER
P81 HYPERKINETIC DISORDER
P82 POST-TRAUMATIC STRESS DISORDER
P85 MENTAL RETARDATION
P86 ANOREXIA NERVOSA/BULIMIA
P98 PSYCHOSIS NOS/OTHER
P99 PSYCHOLOGICAL DISORDERS OTHER
R01 PAIN RESPIRATORY SYSTEM
R02 SHORTNESS OF BREATH/DYSPNOEA
R03 WHEEZING
R04 BREATHING PROBLEM OTHER
R05 COUGH
R06 NOSE BLEED/EPISTAXIS
R07 SNEEZING/NASAL CONGESTION
R08 NOSE SYMPTOM/COMPLAINT OTHER
R09 SINUS SYMPTOM/COMPLAINT
R21 THROAT SYMPTOM/COMPLAINT
R22 deleted, amalgamated with R21
R23 VOICE SYMPTOM/COMPLAINT
R24 HAEMOPTYSIS
R25 SPUTUM/PHLEGM ABNORMAL
R26 FEAR OF CANCER OF RESPIRATORY
    SYSTEM
R27 FEAR OF RESPIRATORY DISEASE OTHER
R28 LIMITED FUNCTION/DISABILITY (R)
R29 RESPIRATORY SYMPTOM/COMPLAINT OTHER
R70 deleted, included with A70
R71 WHOOPING COUGH
R72 STREP THROAT
R73 BOIL/ABSCESS NOSE
R74 UPPER RESPIRATORY INFECTION ACUTE
R75 SINUSITIS ACUTE/CHRONIC
R76 TONSILLITIS ACUTE
R77 LARYNGITIS/TRACHEITIS ACUTE
R78 ACUTE BRONCHITIS/BRONCHIOLITIS
R79 CHRONIC BRONCHITIS
R80 INFLUENZA
R81 PNEUMONIA
R82 PLEURISY/PLEURAL EFFUSION
R83 RESPIRATORY INFECTION OTHER
R84 MALIGNANT NEOPLASM BRONCHUS/LUNG
R85 MALIGNANT NEOPLASM RESPIRATORY
    OTHER
R86 BENIGN NEOPLASM RESPIRATORY
R87 FOREIGN BODY NOSE/LARYNX/BRONCHUS
R88 INJURY RESPIRATORY OTHER
R89 CONGENITAL ANOMALY RESPIRATORY
R90 HYPERTROPHY TONSILS/ADENOIDS
R91 moved to R79
R92 NEOPLASM RESPIRATORY UNSPECIFIED
R93 deleted, included with R82
R95 CHRONIC OBSTRUCTIVE PULMONARY
    DISEASE
R96 ASTHMA
R97 ALLERGIC RHINITIS
R98 HYPERVENTILATION SYNDROME
R99 RESPIRATORY DISEASE OTHER
S01 PAIN/TENDERNESS OF SKIN
S02 PRURITUS
S03 WARTS
S04 LUMP/SWELLING LOCALIZED
S05 LUMPS/SWELLINGS GENERALIZED
S06 RASH LOCALIZED
S07 RASH GENERALIZED
S08 SKIN COLOUR CHANGE
S09 INFECTED FINGER/TOE
S10 BOIL/CARBUNCLE
S11 SKIN INFECTION POST-TRAUMATIC
S12 INSECT BITE/STING
S13 ANIMAL/HUMAN BITE
S14 BURN/SCALD
S15 FOREIGN BODY IN SKIN
S16 BRUISE/CONTUSION
S17 ABRASION/SCRATCH/BLISTER
S18 LACERATION/CUT
S19 SKIN INJURY OTHER
S20 CORN/CALLOSITY
S21 SKIN TEXTURE SYMPTOM/COMPLAINT
S22 NAIL SYMPTOM/COMPLAINT
S23 HAIR LOSS/BALDNESS
S24 HAIR/SCALP SYMPTOM/COMPLAINT
S26 FEAR OF CANCER OF SKIN
S27 FEAR OF SKIN DISEASE OTHER
S28 LIMITED FUNCTION/DISABILITY (S)
S29 SKIN SYMPTOM/COMPLAINT OTHER
S70 HERPES ZOSTER
S71 HERPES SIMPLEX
S72 SCABIES/OTHER ACARIASIS
S73 PEDICULOSIS/SKIN INFESTATION OTHER
S74 DERMATOPHYTOSIS
S75 MONILIASIS/CANDIDIASIS SKIN
S76 SKIN INFECTION OTHER
S77 MALIGNANT NEOPLASM OF SKIN
S78 LIPOMA
S79 NEOPLASM SKIN BENIGN/UNSPECIFIED
S80 SOLAR KERATOSIS/SUNBURN
S81 HAEMANGIOMA/LYMPHANGIOMA
S82 NAEVUS/MOLE
S83 CONGENITAL SKIN ANOMALY OTHER
S84 IMPETIGO
S85 PILONIDAL CYST/FISTULA
S86 DERMATITIS SEBORRHOEIC
S87 DERMATITIS/ATOPIC ECZEMA
S88 DERMATITIS CONTACT/ALLERGIC
S89 DIAPER RASH
S90 PITYRIASIS ROSEA
S91 PSORIASIS
S92 SWEAT GLAND DISEASE
S93 SEBACEOUS CYST
S94 INGROWING NAIL
S95 MOLLUSCUM CONTAGIOSUM
S96 ACNE
S97 CHRONIC ULCER SKIN
S98 URTICARIA
S99 SKIN DISEASE OTHER
T01 EXCESSIVE THIRST
T02 EXCESSIVE APPETITE
T03 LOSS OF APPETITE
T04 FEEDING PROBLEM OF INFANT/CHILD
T05 FEEDING PROBLEM OF ADULT
T06 deleted, transferred to P86
T07 WEIGHT GAIN
```

T08	WEIGHT LOSS
T10	GROWTH DELAY
T11	DEHYDRATION
T15	deleted, included with T81
T26	FEAR OF CANCER OF ENDOCRINE SYSTEM
T27	FEAR OF ENDOCRINE/METABOLIC DISEASE OTHER
T28	LIMITED FUNCTION/DISABILITY (T)
T29	ENDOCRINE/METABOLIC/NUTRITIONAL SYMPTOM/COMPLAINT OTHER
T70	ENDOCRINE INFECTION
T71	MALIGNANT NEOPLASM THYROID
T72	BENIGN NEOPLASM THYROID
T73	NEOPLASM ENDOCRINE OTHER/ UNSPECIFIED
T78	THYROGLOSSAL DUCT/CYST
T80	CONGENITAL ANOMALY ENDOCRINE/ METABOLIC
T81	GOITRE
T82	OBESITY
T83	OVERWEIGHT
T85	HYPERTHYROIDISM/THYROTOXICOSIS
T86	HYPOTHYROIDISM/MYXOEDEMA
T87	HYPOGLYCAEMIA
T88	deleted, included with T99
T89	DIABETES INSULIN DEPENDENT
T90	DIABETES NON-INSULIN DEPENDENT
T91	VITAMIN/NUTRITIONAL DEFICIENCY
T92	GOUT
T93	LIPID DISORDER
T99	ENDOCRINE/METABOLIC/NUTRITIONAL DISEASE OTHER
U01	DYSURIA/PAINFUL URINATION
U02	URINARY FREQUENCY/URGENCY
U04	INCONTINENCE URINE
U05	URINATION PROBLEMS OTHER
U06	HAEMATURIA
U07	URINE SYMPTOM/COMPLAINT OTHER
U08	URINARY RETENTION
U13	BLADDER SYMPTOM/COMPLAINT OTHER
U14	KIDNEY SYMPTOM/COMPLAINT
U26	FEAR OF CANCER OF URINARY SYSTEM
U27	FEAR OF URINARY DISEASE OTHER
U28	LIMITED FUNCTION/DISABILITY (U)
U29	URINARY SYMPTOM/COMPLAINT OTHER
U70	PYELONEPHRITIS/PYELITIS
U71	CYSTITIS/URINARY INFECTION OTHER
U72	URETHRITIS
U75	MALIGNANT NEOPLASM OF KIDNEY
U76	MALIGNANT NEOPLASM OF BLADDER
U77	MALIGNANT NEOPLASM URINARY OTHER
U78	BENIGN NEOPLASM URINARY TRACT
U79	NEOPLASM URINARY TRACT NOS
U80	INJURY URINARY TRACT
U85	CONGENITAL ANOMALY URINARY TRACT
U88	GLOMERULONEPHRITIS/NEPHROSIS
U90	ORTHOSTATIC ALBUMINURIA/PROTEINURIA
U95	URINARY CALCULUS
U98	ABNORMAL URINE TEST NOS
U99	URINARY DISEASE OTHER
W01	QUESTION OF PREGNANCY
W02	FEAR OF PREGNANCY
W03	ANTEPARTUM BLEEDING
W05	PREGNANCY VOMITING/NAUSEA
W10	CONTRACEPTION POSTCOITAL
W11	CONTRACEPTION ORAL
W12	CONTRACEPTION INTRAUTERINE
W13	STERILIZATION
W14	CONTRACEPTION OTHER
W15	INFERTILITY/SUBFERTILITY
W17	POST-PARTUM BLEEDING
W18	POST-PARTUM SYMPTOM/COMPLAINT OTHER
W19	BREAST/LACTATION SYMPTOM/COMPLAINT
W20	deleted, included with W19
W21	CONCERN ABOUT BODY IMAGE RELATED TO PREGNANCY
W27	FEAR OF COMPLICATIONS OF PREGNANCY
W28	LIMITED FUNCTION/DISABILITY (W)
W29	PREGNANCY SYMPTOM/COMPLAINT OTHER
W70	PUERPERAL INFECTION/SEPSIS
W71	INFECTION COMPLICATING PREGNANCY
W72	MALIGNANT NEOPLASM RELATED TO PREGNANCY
W73	BENIGN/UNSPECIFIED NEOPLASM RELATED TO PREGNANCY
W75	INJURY COMPLICATING PREGNANCY
W76	CONGENITAL ANOMALY COMPLICATING PREGNANCY
W77	deleted
W78	PREGNANCY
W79	UNWANTED PREGNANCY
W80	ECTOPIC PREGNANCY
W81	TOXAEMIA OF PREGNANCY
W82	ABORTION SPONTANEOUS
W83	ABORTION INDUCED
W84	PREGNANCY HIGH RISK
W85	GESTATIONAL DIABETES
W90	UNCOMPLICATED LABOUR/DELIVERY LIVEBIRTH
W91	UNCOMPLICATED LABOUR/DELIVERY STILLBIRTH
W92	COMPLICATED LABOUR/DELIVERY LIVEBIRTH
W93	COMPLICATED LABOUR/DELIVERY STILLBIRTH
W94	PUERPERAL MASTITIS
W95	BREAST DISORDER IN PREGNANCY/ PUERPERIUM OTHER
W96	COMPLICATIONS OF PUERPERIUM OTHER
W99	DISORDER OF PREGNANCY/DELIVERY OTHER
X01	GENITAL PAIN FEMALE
X02	MENSTRUAL PAIN
X03	INTERMENSTRUAL PAIN
X04	PAINFUL INTERCOURSE FEMALE
X05	MENSTRUATION ABSENT/SCANTY
X06	MENSTRUATION EXCESSIVE
X07	MENSTRUATION IRREGULAR/FREQUENT
X08	INTERMENSTRUAL BLEEDING
X09	PREMENSTRUAL SYMPTOM/COMPLAINT
X10	POSTPONEMENT OF MENSTRUATION
X11	MENOPAUSAL SYMPTOM/COMPLAINT
X12	POSTMENOPAUSAL BLEEDING
X13	POSTCOITAL BLEEDING
X14	VAGINAL DISCHARGE
X15	VAGINAL SYMPTOM/COMPLAINT OTHER
X16	VULVAL SYMPTOM/COMPLAINT
X17	PELVIS SYMPTOM/COMPLAINT FEMALE
X18	BREAST PAIN FEMALE
X19	BREAST LUMP/MASS FEMALE

X20	NIPPLE SYMPTOM/COMPLAINT FEMALE
X21	BREAST SYMPTOM/COMPLAINT FEMALE OTHER
X22	CONCERN ABOUT BREAST APPEARANCE FEMALE
X23	FEAR OF SEXUALLY TRANSMITTED DISEASE FEMALE
X24	FEAR OF SEXUAL DYSFUNCTION FEMALE
X25	FEAR OF GENITAL CANCER FEMALE
X26	FEAR OF BREAST CANCER FEMALE
X27	FEAR GENITAL/BREAST DISEASE FEMALE OTHER
X28	LIMITED FUNCTION/DISABILITY (X)
X29	GENITAL SYMPTOM/COMPLAINT FEMALE OTHER
X70	SYPHILIS FEMALE
X71	GONORRHOEA FEMALE
X72	GENITAL CANDIDIASIS FEMALE
X73	GENITAL TRICHOMONIASIS FEMALE
X74	PELVIC INFLAMMATORY DISEASE
X75	MALIGNANT NEOPLASM CERVIX
X76	MALIGNANT NEOPLASM BREAST FEMALE
X77	MALIGNANT NEOPLASM GENITAL FEMALE OTHER
X78	FIBROMYOMA UTERUS
X79	BENIGN NEOPLASM BREAST FEMALE
X80	BENIGN NEOPLASM FEMALE GENITAL
X81	GENITAL NEOPLASM FEMALE OTHER/ UNSPECIFIED
X82	INJURY GENITAL FEMALE
X83	CONGENITAL ANOMALY GENITAL FEMALE
X84	VAGINITIS/VULVITIS NOS
X85	CERVICAL DISEASE NOS
X86	ABNORMAL CERVIX SMEAR
X87	UTEROVAGINAL PROLAPSE
X88	FIBROCYSTIC DISEASE BREAST
X89	PREMENSTRUAL TENSION SYNDROME
X90	GENITAL HERPES FEMALE
X91	CONDYLOMATA ACUMINATA FEMALE
X92	CHLAMYDIA INFECTION GENITAL FEMALE
X99	GENITAL DISEASE FEMALE OTHER
Y01	PAIN IN PENIS
Y02	PAIN IN TESTIS/SCROTUM
Y03	URETHRAL DISCHARGE
Y04	PENIS SYMPTOM/COMPLAINT OTHER
Y05	SCROTUM/TESTIS SYMPTOM/COMPLAINT OTHER
Y06	PROSTATE SYMPTOM/COMPLAINT
Y07	IMPOTENCE NOS
Y08	SEXUAL FUNCTION SYMPTOM/COMPLAINT MALE
Y10	INFERTILITY/SUBFERTILITY MALE
Y13	STERILIZATION MALE
Y14	FAMILY PLANNING MALE OTHER
Y16	BREAST SYMPTOM/COMPLAINT MALE
Y24	FEAR OF SEXUAL DYSFUNCTION MALE
Y25	FEAR OF SEXUALLY TRANSMITTED DISEASE MALE
Y26	FEAR OF GENITAL CANCER MALE
Y27	FEAR OF GENITAL DISEASE MALE OTHER
Y28	LIMITED FUNCTION/DISABILITY (Y)
Y29	GENITAL SYMPTOM/COMPLAINT MALE OTHER
Y70	SYPHILIS MALE
Y71	GONORRHOEA MALE
Y72	GENITAL HERPES MALE
Y73	PROSTATITIS/SEMINAL VESICULITIS
Y74	ORCHITIS/EPIDIDYMITIS
Y75	BALANITIS
Y76	CONDYLOMATA ACUMINATA MALE
Y77	MALIGNANT NEOPLASM PROSTATE
Y78	MALIGNANT NEOPLASM MALE GENITAL OTHER
Y79	BENIGN/UNSPECIFIED NEOPLASM MALE GENITAL
Y80	INJURY MALE GENITAL
Y81	PHIMOSIS/REDUNDANT PREPUCE
Y82	HYPOSPADIAS
Y83	UNDESCENDED TESTICLE
Y84	CONGENITAL GENITAL ANOMALY MALE OTHER
Y85	BENIGN PROSTATIC HYPERTROPHY
Y86	HYDROCOELE
Y99	GENITAL DISEASE MALE OTHER
Z01	POVERTY/FINANCIAL PROBLEM
Z02	FOOD/WATER PROBLEM
Z03	HOUSING/NEIGHBOURHOOD PROBLEM
Z04	SOCIAL CULTURAL PROBLEM
Z05	WORK PROBLEM
Z06	UNEMPLOYMENT PROBLEM
Z07	EDUCATION PROBLEM
Z08	SOCIAL WELFARE PROBLEM
Z09	LEGAL PROBLEM
Z10	HEALTH CARE SYSTEM PROBLEM
Z11	COMPLIANCE/BEING ILL PROBLEM
Z12	RELATIONSHIP PROBLEM WITH PARTNER
Z13	PARTNER'S BEHAVIOUR PROBLEM
Z14	PARTNER ILLNESS PROBLEM
Z15	LOSS/DEATH OF PARTNER PROBLEM
Z16	RELATIONSHIP PROBLEM WITH CHILD
Z18	ILLNESS PROBLEM WITH CHILD
Z19	LOSS/DEATH OF CHILD PROBLEM
Z20	RELATIONSHIP PROBLEM PARENT/FAMILY
Z21	BEHAVIOUR PROBLEM PARENT/FAMILY
Z22	ILLNESS PROBLEM PARENT/FAMILY
Z23	LOSS/DEATH OF PARENT/FAMILY MEMBER PROBLEM
Z24	RELATIONSHIP PROBLEM FRIEND
Z25	ASSAULT/HARMFUL EVENT PROBLEM
Z27	FEAR OF A SOCIAL PROBLEM
Z28	LIMITED FUNCTION/DISABILITY (Z)
Z29	SOCIAL PROBLEM NOS