Data mining in medical databases

Anna Szymkowiak Have

LYNGBY 2003

IMM-PHD-2003-121

IMM

Data mining in medical databases

Anna Szymkowiak Have

LYNGBY 2003

IMM-PHD-2003-121

TECHNICAL UNIVERSITY OF DENMARK
Informatics and Mathematical Modelling
Richard Petersens Plads, Building 321,
DK-2800 Kongens Lyngby, Denmark

Ph.D. Thesis
In partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Title: Data mining in medical databases

Author: Anna Szymkowiak Have

TECHNICAL UNIVERSITY OF DENMARK
Informatics and Mathematical Modelling
Richard Petersens Plads, Building 321,
DK-2800 Kongens Lyngby, Denmark
IMM-PHD-2003-121

Lyngby 2003-07-31

Printed by IMM, Technical University of Denmark

ISSN 0909–3192

Abstract

This Ph.D. thesis focuses on clustering techniques for Knowledge Discovery in Databases. Various data mining tasks relevant for medical applications are described and discussed. A general framework which combines data projection and data mining and interpretation is presented. An overview of various data projection techniques is offered with the main stress on applied Principal Component Analysis. For clustering purposes, various Generalized Gaussian Mixture models are presented. Further the aggregated Markov model, which provides the cluster structure via the probabilistic decomposition of the Gram matrix, is proposed. Other data mining tasks, described in this thesis are outlier detection and the imputation of the missing data. The thesis presents two outlier detection methods based on the cumulative distribution and a special designated outlier cluster in connection with the Generalized Gaussian Mixture model. Two models for imputation of the missing data, namely the K-nearest neighbor and a Gaussian model are suggested. With the purpose of interpreting a cluster structure two techniques are developed. If cluster labels are available then the cluster understanding via the confusion matrix is available. If data is unlabeled, then it is possible to generate keywords (in case of textual data) or key-patterns, as an informative representation of the obtained clusters. The methods are applied on simple artificial data sets, as well as collections of textual and medical data.

Resumé

Denne ph.d.-afhandling fokuserer på klyngeanalyseteknikker til ekstraktion af viden fra databaser. Afhandling præsenterer og diskuterer forskellige datamining problemstillinger med relevans for medicinske applikationer. Specielt præsenteres en generel struktur der kombinerer data-projektion, datamining og automatisk fortolkning. Indenfor data-projektion gennemgås en række teknikker med speciel vægt på anvendt Principal Komponent Analyse. En række generaliserede Gaussisk miksturmodeller foreslås til klyngeanalyse. Desuden foreslås en aggregatet Markov model, som estimerer klyngestrukturen via dekomposition af en sandsynlighedsbaseret Gram-matrix. Herudover beskriver afhandlingen to andre datamining problemstillinger nemlig "outlier" detektion og imputering af manglende data. Afhandlinger præsenterer "outlier" detektionsmetoder. Dels baseret på kumulerede fordelinger, dels baseret på introduktion af en speciel "outlier" klynge i forbindelse med den generaliserede Gaussisk miksturmodel. Med hensyn til imputation af manglende data præsenteres to metoder baseret på K-nærmeste-nabo eller en Gaussisk model antagelse. Der er udviklet to metoder til automatisk fortolkning af klyngestrukturen. Når klynge annoteringer ("labels") er tilgængelige vil konfusionsmatricen danne grundlaget for fortolkningen. Hvis sådanne annoteringer ikke er tilgængelige, er det muligt at generere nøgleord (i tilfælde af tekst data) eller generelt nøgle-mønstre, som således bibringer til fortolkning af klyngerne. De foreslåede metoder er testet på simple kunstige datasæt så vel som kollektioner af tekst og medicinske data.

Preface

This thesis was prepared at the Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU), in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering. The work was funded by the Danish Research Councils through SITE, signal and image processing for telemedicine. The project commenced in July 2000 and was completed in August 2003. Throughout the period the project was supervised by professor Jan Larsen and professor Lars Kai Hansen from IMM, DTU. The thesis reflects the studies being done during the Ph.D. project which relates to clustering data originating from different sources located in e.g. various databases.

The thesis is printed by IMM, Technical University of Denmark and available as softcopy from http://www.imm.dtu.dk

Publication Note

Parts of the work presented in this thesis have previously been published journals and at conference proceedings. Following are the paper contributions in the context of this thesis.

A. Szymkowiak, P.A. Philipsen, J. Larsen, L.K. Hansen, E. Thieden and H.C. Wulf, *Imputating missing values in diary records of sun-exposure study*, in Proceedings of IEEE Workshop on Neural Networks for Signal Processing XI, editor: D. Miller, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, pages: 489–498. Own contribution estimated to 50%. Included in appendix B.

A. Szymkowiak, J. Larsen, L.K. Hansen, *Hierarchical Clustering for Datamining*, In Proceedings of KES-2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, pages: 261–265. Own contribution approximated to 80%. Included in appendix C.

J. Larsen, A. Szymkowiak-Have, L.K. Hansen, *Probabilistic Hierarchical Clustering with Labeled and Unlabeled Data*, in International Journal of Knowledge-Based Intelligent Engineering Systems, volume: 6(1), pages: 56-62. Contribution approximated to 30%. Included in appendix D.

A. Szymkowiak-Have, J. Larsen, L.K. Hansen, P.A. Philipsen, E. Thieden and H.C. Wulf, *Clustering of Sun Exposure Measurements*, In Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII, editor: D. Miller and T. Adali and J. Larsen and M. Van Hulle and S. Douglas, pages: 727–735. Contribution approximated to 50%. Included in appendix E.

J. Larsen, L.K. Hansen, A. Szymkowiak-Have, T. Christiansen and T. Kolenda, *Webmining: Learning from the World Wide Web*, in special issue of Computational Statistics and Data Analysis, volume: 38, pages: 517–532. Contribution approximated to 10%. Included in appendix F.

Anna Szymkowiak-Have and Mark Girolami and Jan Larsen, *Clustering via Kernel Decomposition* submitted to IEEE Transactions in Neural Networks, yet unpublished. Contribution approximated to 45%.

Nomenclature

Standard symbols and operators are used consistently throughout the thesis. Symbols and operators are introduced as they are needed. In general matrices are presented in uppercase bold letters e.g. $\mathbf{X}$, while vectors are shown in lowercase bold letters, e.g. $\mathbf{x}$. Letters not bolded are scalars, e.g. x.

Acknowledgement

I would like to thank Jan Larsen and Lars Kai Hansen for their supervision, support and answers to all my questions. Also, I would like to thank the rest of the people at the Intelligent Signal Processing group, especially the department secretary Ulla Nørhave for looking after me. Furthermore, I would like to thank the people I got to know during my stay at Paisley University, especially Mark Girolami, Jackie Royal and Leif Azzopardi, for all the discussions and fun both during work time and at the pubs.

Finally, I want to thank my husband Kim for his support and patience he has given me during my studies, and my family and my friends for their support and encouragement and for the carrot in front of my donkey.

Contents

Notation and Symbols

$\mathbf{X}_{D\times N}$	observation data matrix, where N is denoting number of examples, $n = 1, 2, \ldots, N$, and D is a dimensionality of the feature space, $d = 1, 2, \ldots, D$
x_{dn}	the d'th, n'th element of the data matrix $\mathbf{X}_{N\times D}$
$\mathbf{x}_n$	n'th data vector
$\widehat{\mathbf{X}}_{K\times N}$	projected matrix, where K, $k = 1, \ldots, K$ is dimensionality of the projected feature space
$\mathbf{z}_n$	n'th D-dimensional test data vector, where $n = 1, 2, \ldots, N_z$
$\vert \cdot \vert$	determinant
$\vert\vert \cdot \vert\vert_2$	$\mathcal{L}_2$ norm, vector length, $\vert\vert \mathbf{x}_n \vert\vert_2 = \sqrt{\sum_d \mathbf{x}_{nd}^2}$
./	matrix element-wise division
$\mathbf{1}$	matrix with all elements 1
$p(\mathbf{x})$	Probability Density Function of the vector $\mathbf{x}$
$p(k\vert\mathbf{x})$	Class Posterior Probability of the class k conditioned on vector $\mathbf{x}$
$p(\mathbf{x}\vert k)$	Likelihood of vector $\mathbf{x}$
μ	mean vector $\mu = \frac{1}{N}\sum_n x_n$
$\mathbf{\Sigma}$	covariance matrix $\mathbf{\Sigma} = \frac{1}{N}\sum_n (x_n - \mu)(x_n - \mu)^T$
θ	vector collecting the parameters of the model
EM	Expectation-Maximization algorithm
FA	Factor Analysis
GGM	Generalizable Gaussian Mixture model
I	Identity matrix
ICA	Independent Component Analysis
$i.i.d.$	Independent and Identically Distributed

KDD	Knowledge Discovery in Databases
KL	Kullback-Leibler divergence
KNN	K-Nearest Neighbor
KPCA	Kernel Principal Component Analysis
LSA	Latent Semantic Analysis
MAR	data missing at random
MCAR	data missing completely at random
ML	Maximum Likelihood methods
NMAR	data not missing at random
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
p.d.f.	Probability Density Function
PLSA	Probabilistic Latent Semantic Analysis
RP	Random Projection
SVD	Singular Value Decomposition
UGGM	unsupervised Generalizable Gaussian Mixture model
USGGM	unsupervised/supervised Generalizable Gaussian Mixture model

CHAPTER 1

Introduction

Recent progress in data storage and acquisition has resulted in a growing number of enormous databases. The information contained in these databases can be extremely interesting and useful, however the amount is too large for humans to process manually. These databases store information covering every aspect of human activity. In the business world, the data bases are created for marketing, investment, manufacturing, customers, products or transactions, and such information can be stored both for accounting and analysis purposes. The domain of scientific research is another area which makes heavy use of databases to store information about, for example patients, patient histories, surveys, medical investigations. It is within this domain that the research for this thesis has been performed. Medical databases contain data in a variety of formats: images in the form of X-rays or scans, textual information to describe details of diseases, medical histories, psychology reports, medical articles or various signals like EKG, ECG, EEG, etc. This data does not need to be located on the same system. It may be distributed amongst various disparate computers depending on the source of the data. Such heterogeneous data make the process of information retrieval an even more complex process.

There is a substantial need for signal/image processing and data mining tools in medical information processing systems and in telemedicine that are flexible, effective and used friendly. Telemedicine is defined as: *The investigation, monitoring and management of patients and education of patients and staff using systems which allow ready access to expert advice and patient information no matter of where the patient or relevant information is located.* Such systems serve a dual role: they can assist medical professionals in medical research and in the clinic medical diagnosis and secondly, they can be used as information sources for patients. Telemedicine faces a number of basic problems concerning document retrieval, data mining in the databases, and visualization of high-dimensional data.

This project focuses on the fact finding in the distributed databases by use of advanced signal processing and machine learning methods which can be applied to extract the important for the medical scientists information. Such data mining can also be used by patients to search for analogies or similar diseases or any relevant information.

Data mining is a new discipline within the field of information retrieval that extracts interesting and useful material from large data sets. In the broader sense, data mining is defined as part of *knowledge discovery in databases* (KDD) [25, 26, 36] and draws on the fields of statistics, machine learning, pattern recognition and database management. The general approach of KDD process is presented on figure 1.1

The KDD process involves the following steps:

1. *Selecting the target data* (focusing on the part of the data on which to make discoveries). The data may consist from different type of information.

2. *Data cleaning and preprocessing* includes removal of noise, outliers, missing data (if they are not an object of the discovery). See Section 3.6.

3. *Transforming data* includes data reduction and projection in order to find and investigate useful features and reduce the data feature space dimensionality. See Chapter 2.

4. *Performing the data mining task*, e.g., classification, regression, clustering and as well the outlier detection and imputation of missing data, if they were not removed in data cleaning process. Data mining is performed in order to extract patterns and relationships in the data. See Chapter 3, Chapter 5 and Chapter 6.

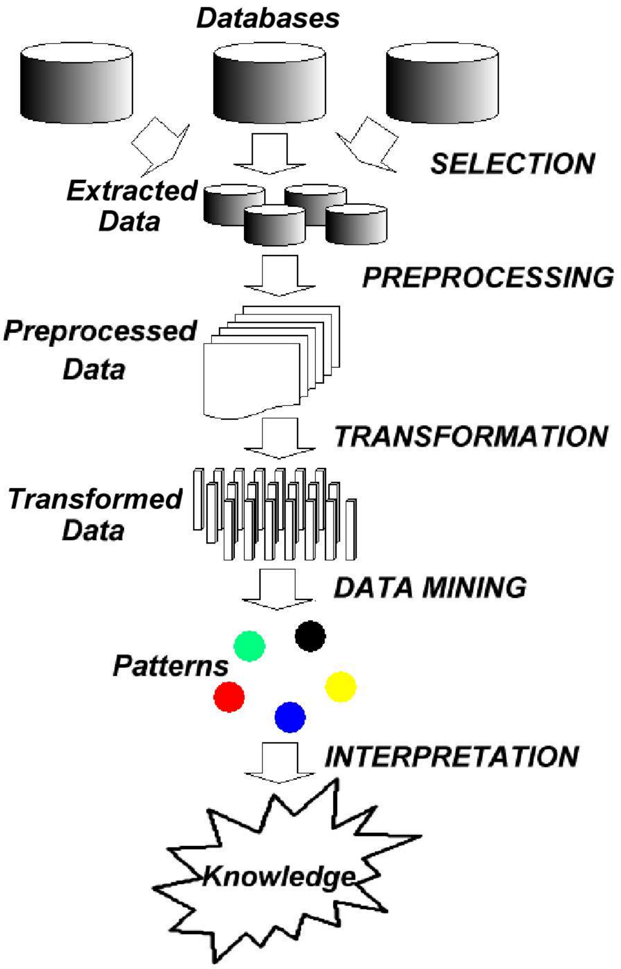


Figure 1.1 The steps of the KDD process. In the first step the data is extracted from the source location, then the preprocessing is performed and the data is transformed to the form suitable for further processing. In the next step the data mining is applied, for example clustering, classification, regression etc. The final part of KDD process is a meaningful interpretation is the obtained results.

5. *Interpretation and visualization* includes interpretation of mined patterns, visualization of models, patterns and data. See Chapter 4.

Typically, data mining algorithms assume that the data is already stored in main memory, therefore no database extraction methods are investigated in this work. However, in order to facilitate the extraction of the required information from the databases, the issue of large data sets and the preprocessing and required transformation algorithms is addressed in Chapter 2. This is necessary as the data we are dealing with is observational and has usually been collected for a purpose other than data mining. Some of the preprocessing steps are uniquely connected with the data and as such, they are described with the experiments in Chapter 7. The relationships found in data are presented using models for clustering, classification and outlier detection and imputation, in Chapters 3, 6 and 5, respectively. Finally, techniques for the useful summarization, visualization and interpretation of the discovered structures are given in Chapter 4.

The contents of this thesis are organized in the following way:

Chapter 2 gives an overview of the several projection methods implemented: random projection, principal component analysis and non-negative matrix factorization. A literature overview of the reduction techniques for huge data sets is also included, as is a discussion of the various ways to select the optimum number of components in the projected data. The chapter concludes with a evaluation of the presented projection methods applied to the artificial data sets.

Chapter 3 is dedicated to the various Gaussian Mixture models that were used for classification and clustering the data focusing on the possible medical databases. The algorithms of unsupervised, supervised and unsupervised/supervised Generalizable Gaussian Mixture models are described. Two techniques are presented for outlier and novelty detection; the outlier detection methods are compared through a simple example.

Chapter 4 describes the different similarity measures and methods that can be used with agglomerative hierarchical clustering. In order to interpret the resultant clusters, a method for obtaining keywords and prototypes is also explained.

Chapter 5 explains two methods used for the imputation of missing values in the study, namely the Gaussian and K-Nearest Neighbor models.

Chapter 6 defines a different approach for unsupervised clustering by means of Gram matrix decomposition with the aggregated Markov model.

Chapter 7 collates the experimental results on the observational data sets. The first section describes the data sets that have been used. The collection of emails and newsgroups is used as an example of textual data. The next data set originates from a survey undertaken by the Department of Dermatology, Bispebjerg Hospital, University of Copenhagen, Denmark. This study examined the connection between the level of sun exposure to the skin and the risk of developing cancer disease. The third data set is a dermatological collection of six diagnosed erythemato-squamous diseases. Preprocessing is carried out separately for each of the data sets. Data projection is then performed, except for the aggregated Markov model, where the features space reduction is not required. Since, data mining is the next step of the KDD process, the clustering of the text data sets and the sun-exposure study is implemented using the various models. For visualization and interpretation, hierarchical clustering is applied and keywords are generated. With respect to the data form sun-exposure study, an investigation of the performance of the imputation models is conducted.

Appendix A contains the definitions of the Jensen's and Minkowski's inequalities and the derivation of the Kullback-Leibler similarity measure for Gaussian density functions that is used in agglomerative hierarchical clustering.

Appendix B–F contain reprints of the papers authored or co-authored during the Ph.D. study.

CHAPTER 2

Dimensionality reduction and feature selection

Data transformation is a first discussed here step of the KDD process which is shown on figure 1.1.

When processing large databases, one faces two major obstacles: numerous samples and high dimensionality of the feature space. For example, the documents are represented by several thousands of words, images are composed of millions of pixels, where each word or pixel is here understood as a feature. Currently, processing abilities are often not able to handle such high dimensional data, mostly due to numerical difficulties in processing, requirements in storage and transmission within a reasonable time. To reduce the computational time, it is common practice to project the data onto a smaller, latent space. Moreover, such a space is often beneficial for further investigation due to noise reduction and desired feature extraction properties. Smaller dimensions are also advantageous when visualizing and analyzing the data. Thus, in order to extract desirable information, dimensionality reduction methods are often applied.

This task is non-trivial. The goal is to determine the coordinate system where the mapping will create low-dimensional compact representation of the data whilst maximizing the information contained within.

There are many solutions to this problem. Several techniques for dimensionality reduction have been developed which use both linear and non-linear mappings. Among them are, for example, low-dimensional projections of the data [10, 45, 40, 49], neural networks [3, 12, 74], self-organizing maps [48]. One can apply second order methods which use the covariance structure in determining directions. To this family belongs the popular Principal Component Analysis [10, 23, 44, 45] that restricts directions to those that are orthogonal; Factor Analysis [40, 44, 45] which additionally allows the noise level to differ along the directions and Independent Component Analysis [49] for which the directions are independent but not necessarily orthogonal. Description of some of the afore-mentioned methods can be found in [7, 10, 14, 40, 42, 49, 55, 66, 84].

Since the projection is not the main topic of this thesis, but the preprocessing step in KDD process, only a few techniques, that were investigated alongside the performed research, are presented in detail. Three algorithms are described below, namely Random Projection [4, 9, 19], Principal Component Analysis [10, 14, 23, 44, 45, 66] and Non-negative Matrix Factorization [55]. The modified version of Non-negative Matrix Factorization, is applied in the decomposition of the Gram matrix in the aggregated Markov model, described in Chapter 6.

2.1 Dimensionality reduction methods

2.1.1 Random Projection

A lot of research studies have focused on an investigation of the Random Projection method. Many of the details and experiments not included in this work, can be found in [4, 9, 19].

The method is simple and it does not require the presence of the data whilst creating the projection matrix. Many find this a significant advantage. Random projection is based on the set of vectors which are orthogonalized from the normal distributed random matrix. The orthogonal projection vectors[1] ($\mathbf{R} \cdot \mathbf{R}^T = I$) preserves the distances among the data points. In this way it is possible to

[1] Two vectors are said to be orthogonal if the cosine of its angle is equal 0, i.e. $\mathbf{u} \cdot \mathbf{v}^T = 0$, where $\mathbf{v} \cdot \mathbf{v}^T = 1$ and $\mathbf{u} \cdot \mathbf{u}^T = 1$.

project the data to the smaller dimensional space while preserving fair separation of the clusters. Note however, that $\mathbf{R}$ is orthogonal only column-wise, i.e. the identity outcome of the dot product of the projection vectors holds only for quadratic $\mathbf{R}$ (projecting onto the space of the same dimension as the original). In the case of projecting onto smaller dimensions, the distance is no longer preserved in an ideal way.

Let the D-dimensional data matrix be defined as $\mathbf{X}_{D\times N}$, where N is the number of samples. Data is projected with the help of the projection matrix $\mathbf{R}$ to K dimensions ($K \leq D$) and the projected data matrix is denoted as $\widehat{\mathbf{X}}_{K\times N}$. Figure 2.1 presents the algorithm of the Random Projection method.

The Random Projection Algorithm

1. Generate random matrix $\mathbf{R}_{K\times D} = \{r_{kd}, k = 1\dots K, d = 1\dots D\}$ drawn form normal distribution of zero mean and spherical covariance structure $r_{kd} \in \mathcal{N}(0,1)$. $K \leq D$.

2. Orthogonalize each of the projection vectors (use for example Gram-Schmidt algorithm (see [56] or most linear algebra books for reference)).

3. Project data on the new set of directions $\widehat{\mathbf{X}}_{K\times N} = \mathbf{R}_{K\times D} \cdot \mathbf{X}_{D\times N}$

Figure 2.1 The Random Projection algorithm.

In order to check the success of the random projection method, the separability is investigated with the following toy example. In the example, 200 data points are generated from 2 Gaussian distributions with spherical covariance structure $\mathbf{\Sigma} = I$ and different mode locations, where distance between the data means $||\mu_1 - \mu_2||_2$ approximately equals 5. Only one dimension is enough to separate clusters. Originally, data spans 500 dimensions. The clusters are almost linearly separable, i.e. the cluster overlap is negligible. Also by using Principal Component Analysis, described later in section 2.1.2 and presented here as reference, the distance between the clusters is fully preserved.

Figure 2.2 presents the results of the experiment. Left plot shows the distance between the cluster centers. As was suspected, the distance decays, and in the small dimensional space it can be expected that the data is poorly separated. This artifact is more significant for large dimensional and more complex data sets, such as sparse vector space representation of textual information. The

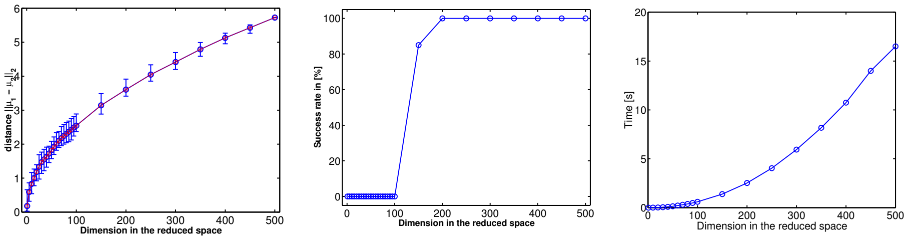


Figure 2.2 Illustration of the Random Projection method with example of 2 Gaussian distributed clusters. Originally the measured distance between the clusters is close to 6. The experiments are repeated 20 times and the average performance is shown. Left plot describes the distance $||\mu_1 - \mu_2||_2$ between the cluster centers as a function of dimensionality. It decays with the projected dimension. Error-bars show maximum and minimum measured distance in the trials. They increase with the dimension. Middle plot shows success rate of the projection (the trials for which the distance is larger than 3, for smaller numbers clusters are assumed not to be separable). Also the success rate decays with the reduced dimension. That means that for the large reductions there is a considerable chance that the projected data will lose separability. Note, that one dimension is enough to linearly separate the data. The time needed to produce the projection matrix grows exponentially with the size of the matrix $\mathbf{R}$ (right figure).

smaller the original dimensionality, the longer the separation is kept, when the projection matrix is increased. The success rate[2] (middle plot) is decaying significantly with the dimension. The projection is not in all cases satisfactory, the separability in some *trials*[3] is lower due to the random origin of the projection vectors. The number of “unlucky” projections grows significantly with reduced dimension. The time needed to produce the projection matrix grows significantly with the size of the projection matrix $\mathbf{R}$ (right plot).

When the projected dimension is large enough, the distortion of the distance is not significant but then the time spent for creating and diagonalizing projection matrix is considerable. To summarize, for large dimensional data sets it is not necessarily efficient and satisfying enough[4] to use random projection without significant separability loss, especially when the dimensionality the data is

[2] Success rate is the percentage of the number of projection runs in which the distance is preserved (in this case the clusters are assumed to lose separability when the distance between them is smaller than 3 (originally the distance is equal to 5)). The cluster distance is calculated from projected known cluster centers.

[3] One particular run of the experiment is understood to be a trial.

[4] The benefits of the random projection may vary accordingly to the application.

projected onto is high.

Random Projection is not applied in further experiments due to the fact that the dimensionality of the data used in the experiments allowed us to use Principal Component Analysis (section 2.1.2). However, when facing larger databases it could be advisable to use RP as the preprocessing step before applying PCA. In such a case, creating the random projection matrix is less expensive than PCA and with the projected space large enough, the distances in the data is not be distorted significantly.

2.1.2 Principal Component Analysis

Principal Component Analysis (PCA) is probably the most widely used technique for dimensionality reduction. It is similar to random projection through performing a linear mapping that uses the orthogonal projection vectors, but in case of PCA the projections are found by diagonalizing the data covariance structure, i.e. the variance along new directions is maximized. Thus, the projection of vector $\mathbf{x}$ on the new latent space appointed by orthogonal projection vectors $\mathbf{u}$ can be written as $\widehat{\mathbf{x}}_k = \mathbf{u}_k^T\mathbf{x}$, where $k = 1 \dots K$ and $K \leq D$.

It is proofed, that the minimum projection error (in LS sense) is achieved when the basis vectors satisfy the eigen-equations $\mathbf{\Sigma} \cdot \mathbf{u}_k = \lambda_k \mathbf{u}_k$ [10]. Therefore, the singular value decomposition (SVD) is often performed to find the orthogonal projections. The algorithm for PCA is shown in figure 2.3. Such a method for determining the latent space is applied by Latent Semantic Analysis (LSA), described in section 7.2.2 and introduced in [20]. For further and more comprehensive analysis for PCA refer to [10, 23, 44, 45].

The most significant disadvantages of the PCA technique are complexity and high memory usage. However, a great number of alternative algorithms have been developed that reduce this problem, for example using networks to estimate the few first eigenvalues and corresponding principal directions [3, 12].

Principal Component Analysis Algorithm

1. Create the data matrix $\mathbf{X}$
2. Determine covariance $\mathbf{\Sigma} = \frac{1}{N}\sum_n(\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T$
3. Determine eigenvalues λ_k and eigenvectors $\mathbf{u}_k$ of the covariance structure. Since $\mathbf{\Sigma}$ is positive and symmetric, λ is positive and real satisfying $\mathbf{\Sigma} \cdot \mathbf{u}_k = \lambda_k \mathbf{u}_k$. λ is found in the optimization process of the characteristic equation $|\mathbf{\Sigma} - \lambda I| = 0$.
4. Sort eigenvalues and corresponding eigenvectors in the descending order.
5. Select $K < D$ and project the data on selected directions: $\widehat{\mathbf{X}}_{K\times N} = \mathbf{U}^T_{D\times K} \cdot \mathbf{X}_{D\times N}$

Figure 2.3 The Principal Component Analysis algorithm.

2.1.3 Non-negative Matrix Factorization

Some medical data such as images or texts contain only positive values. This information can be utilized by adding the positivity constraint in the optimization process for finding the projection vectors. One possible choice is the Non-Negative Matrix Factorization (NMF). This approach is proposed by Lee & Seung in [55]. The technique is closely related with proposed by Hofmann in [42] Probabilistic Latent semantic Analysis (PLSA) and by Saul and Peveira in [76] aggregated Markov model. NMF is based on the decomposition of the data matrix $\mathbf{X}$ into two matrix factors $\mathbf{W}$ and $\mathbf{H}$ so that $\mathbf{X}_{D\times N} \approx \mathbf{W}_{D\times K}\mathbf{H}_{K\times N}$. Both $\mathbf{W}$ and $\mathbf{H}$ are constrained to be positive, i.e. $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$. The proposed optimization is done in an iterative way minimizing one of the two suggested objective functions: Euclidean distance [34]:

$$||\mathbf{X} - \mathbf{WH}||_2^2 = \sum_d \sum_n (x_{dn} - \sum_k w_{dk}h_{kn})^2 \tag{2.1}$$

or Kullback-Leibler (KL) divergence [69]:

$$\mathcal{D}(\mathbf{X}||\mathbf{WH}) = \sum_d \sum_n (x_{dn} \cdot \log \frac{x_{dn}}{\sum_k w_{dk}h_{kn}} - x_{dn} + \sum_k w_{dk}h_{kn}) \tag{2.2}$$

between $\mathbf{X}$ and $\mathbf{WH}$. The update rules in case of the Euclidean distance are shown in matrix notation by the equations 2.3.

$$\mathbf{H}^{new} = \mathbf{H}^{old}\frac{\mathbf{W}^T\mathbf{X}}{\mathbf{W}^T\mathbf{WH}} \qquad \mathbf{W}^{new} = \mathbf{W}^{old}\frac{\mathbf{XH}^T}{\mathbf{WHH}^T} \tag{2.3}$$

Regarding KL divergence, the update rules are taking the form:

$$\mathbf{H}^{new} = \mathbf{H}^{old}\frac{\mathbf{W}^T \cdot \mathbf{X}./\mathbf{W}\mathbf{H}}{\mathbf{1}_{N\times K} \cdot \mathbf{W}^T} \qquad \mathbf{W}^{new} = \mathbf{W}^{old}\frac{\mathbf{X}./\mathbf{W}\mathbf{H} \cdot \mathbf{H}^T}{\mathbf{1}_{D\times N} \cdot \mathbf{H}^T}. \tag{2.4}$$

The algorithm is proofed to converge, see [55] for further details. The modified NMF updates rules are used in segmentation by the aggregated Markov model described in detail in Chapter 6.

2.1.4 Evaluation of dimensionality reduction methods

Several methods have been described in this chapter. Visual comparison of the techniques is usually useful for understanding differences between them. Here, the comparison of those techniques is presented for artificially created toy data. Three data sets are generated, namely:

3 Gaussians - Linear structure in the form of three Gaussian ideally separated clusters with elliptical covariance structures, see figure 2.4. 1200 data points are generated, 400 points for each cluster. The signal space has 2 dimensions, but the noisy data exists in 20 dimensions.

Rings - Non-linear manifold structure in the form of three uniformly distributed circles centered at the origin of the coordinate system. The problem is separable but a nonlinear decision boundary is needed (figure 2.5). 1200 data points are generated, 400 points for each cluster. The signal space has 2 dimensions, but the data with the noise exists in 20 dimensions.

Text - The discrete data artificially created that resemble the discrete text in the form of preprocessed and normalized term-document matrix[5]. The data is modeled in latent space found in LSI framework (Latent Semantic Indexing [20]. The data is multi-dimensional (60 features) and forms two perfectly separable clusters. Features are selected so that a part is shared and some of the features are unique for each of the clusters, see figure 2.6. Data vectors are normalized to the $\mathcal{L}_2$ norm unity length.

Figure 2.7 shows the results of the projection of the 3 Gaussian clusters. The

[5] For explanation of text processing method refer to section 7.2.1

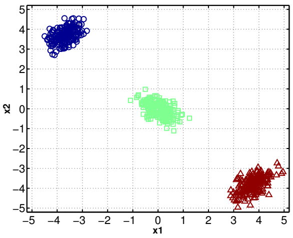


Figure 2.4 Scatter plot of the 3 Gaussian clusters. 2 first dimensions (out of 20) are shown. The clusters are linearly separable.

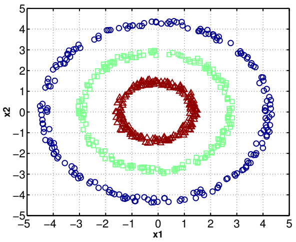


Figure 2.5 Scatter plot of the 3 rings. 2 first dimensions are presented. The clusters can be separated by the non-linear decision surface.

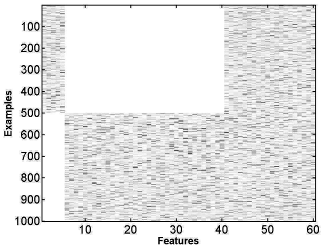


Figure 2.6 Feature plot of the 2 text clusters. On the *y*-axis are the 1000 examples (500 in each class) and on the *x*-axis are the 60 features.

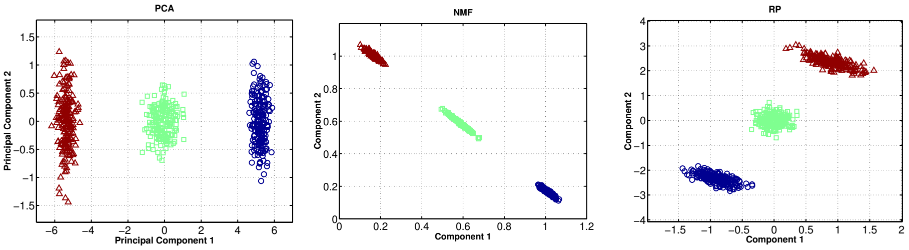


Figure 2.7 Results of the projection for the 3 Gaussian clusters (shown on figure 2.4). Three techniques for creating the projection vectors are compared here: Principal component Analysis (PCA), Non-Negative Matrix Factorization (NMF) and Random Projection (RP). In all the cases the separation is preserved.

problem is simple and linearly separable. Three different techniques for creating the projection vectors are compared here, Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF) and Random Projection (RP). In this case, all the presented methods preserve the linear separability in the data.

Figure 2.8 shows the results of the projection of the rings structure of 3 clusters. As mentioned previously, the PCA, NMF and RP are compared. Only the Principal Component Analysis preserves the distances amongst the data points well, while the other methods have lost the separability. RP preserves the separation

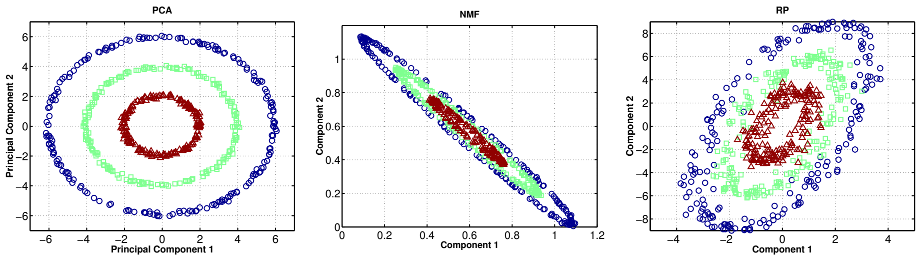


Figure 2.8 Results of the projection of the structure of 3 clusters in the shape of rings (shown on figure 2.5). Three techniques for creating the projection vectors are compared here: Principal component Analysis, Non-Negative Matrix Factorization and Random Projection. The original data is separable but a non-linear separation function is needed. PCA does not change the data shape since the data variance in all signal directions is identical. NMF returns a highly skewed result, difficult to separate. A similar result is achieved with RP. However, some of the other trials returned both better and worse results with respect to the possibility of separation.

in some of the trials while in others the separation is lost. This happens due to the random origin of the projection matrix.

In figure 2.9, the results are presented of the projection of the toy example: the vector space representation of two text clusters. The data is linearly separable and this characteristic is preserved by using any of the presented methods

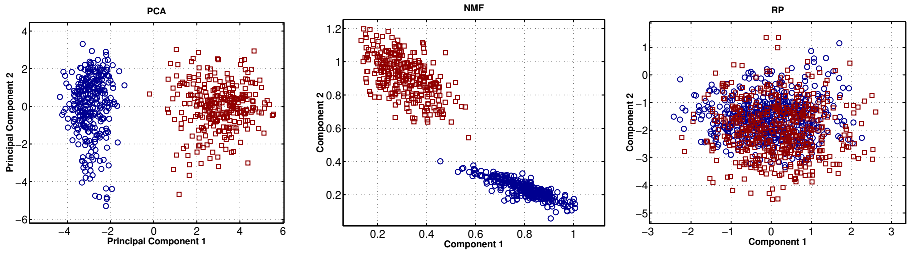


Figure 2.9 Results of the projection of the 2 cluster artificial text example (shown on figure 2.6). Three techniques for creating the projection vectors are compared here, Principal Component Analysis, Non-Negative Matrix Factorization and Random Projection. In this case RP is unable to separate the data, while the other methods perform reasonably well.

with the exception of Random Projection, which completely blends these two classes. In this case Random Projection confirms its inability to extract the separating dimensions, when processing large dimensional sparse data.

Out of three presented methods only Principal Component Analysis have shown robustness for all the applied data sets. In conclusion, PCA is selected for dimensionality reduction in later experiments due to being the simplest method and the best performer out of all the researched techniques.

2.2 High dimensional data

The methods described earlier can operate on medium sized databases. However, when the dimensionality of the data is much larger other techniques have to be implemented. In this thesis, only a short review of the projection techniques for huge dimensional databases is presented, since no such databases were available for the experiments presented in Chapter 7.

The simplest, most intuitive way to determine the low-dimensional projections is to reduce the number of points in the data set. Random sampling without replacement creates a subset expected to be a good representation of the data distribution. PCA or any other classical method can then be computed from this subset [64, 88]. This technique is obviously introducing an error so the projection vectors will not be optimal. However, one can hope that given the sufficient number of data points the error introduced by this reduction will be negligible.

When the class structure is available a priori some sort of simple clustering technique can be performed on high dimensional data (e.g., K-nearest neighbor [36]). Another approach can be used, namely local PCA [29, 46]. With this technique only the eigenvectors are extracted that correspond to the largest eigenvalues for each cluster. This set of projections defines a new space for the data.

The other way to reduce the complexity is to use recursive or iterative methods. In the case of dimensionality reduction that means computing the most important components incrementally [22, 41, 65, 66].

It is also possible to create a hybrid structure by performing a two step projection as is suggested in [19]. In order to find the high dimensional basis vectors Random Projection could be used. The goal is to project the data to lower dimensional space without causing a significant change in the distance. For the second step, classic methods like Principal Component Analysis, Factor Analysis [40, 44, 45] or Independent Component Analysis [49] can be used so the important directions are extracted.

2.3 Selection of number of principal components

All the projection techniques presented in this chapter estimate the set of projection directions but also leave an open question as to how the optimum dimension of the space will be chosen. It is extremely important to notice that too small a dimension may result in serious information loss, whilst one that is too large will often introduce noise to the projected data. In both cases the consequences are significant in the form of large errors in the information extraction, clustering, density estimation, prediction etc.

Thus, several methods have been developed for selecting the optimum dimension. An overview is presented below of some of the techniques for selecting the number of principal components in the case of PCA (described in section 2.1.2).

The simplest technique to find the effective dimension K is to calculate the eigenvalue curve and base the decision on its shape. If the effective dimensionality of the data is smaller than the actual data size D, the eigenvalue curve will have the shape of the letter "L". Various, mainly ad hoc rules, have been proposed for estimating the turning point of that curve[6], which will minimize the projection error [10, 45]. One way is to use the cumulative percentage of the total variation. Here, it is desired that the selected components contribute significantly, say $80\% - 90\%$ of the total variation. A similar way is to select the threshold value for which the components with smaller variance/eigenvalues are discarded. Thus, only the largest/dominant components contribute in the final result. In this approach, typically the eigenvalues are rejected that are smaller than $75 - 90\%$ of the maximum.

[6] The turning point that determines the optimal K is usually ambiguous, due to the fact that the L-curve often has a smooth characteristic.

Another approach is proposed by Girolami in [32]. It suggests using both the ordering information included in the eigenvalues and in the variation of the eigenvectors. Thus, instead of looking at the eigenvalue curve, we now investigate the transformed curve of the form $\lambda_k(\frac{1}{N}\sum_j u_{kj}^2)$. Both the thresholding and the turning point search can be applied. This transformation usually makes the slope of the L-curve steeper which facilitates the decision.

Cattell in [15] suggests finding the largest eigenvalue beyond the point where the curve becomes a straight line as close to horizontal as possible. A similar technique can be found in [18, 45], but here, the results are based on the log-eigenvalue curve. Craddock and Flood in [18] claim that the eigenvalues corresponding to the noise value should decay in geometric progression and that will produce a straight horizontal curve in the logarithmic space.

The success of the methods mentioned above is subjective and dependent on the choice of cut-off level. Nevertheless, they are often used due to their simplicity and fairly reliable outcomes.

Additionally, objective rules have been developed for choosing the number of principal components [37, 45]. Usually in such cases the decision is based on the cross-validation error.

For example the cross-validated choice is proposed by [24, 45, 90]. For an increasing number of retained components, the Least Square error is estimated between the original data and the leave-one-out estimate of the data in the reduced space. Thus, the so-called Prediction Sum of Squares is calculated based on the following equation

$$PRESS(K) = \sum_{d=1}^{D}\sum_{n=1}^{N}(\hat{x}_{dn} - x_{dn})^2, \tag{2.5}$$

where $\hat{x}_{dn} = \sum_{k=1}^{K}\hat{u}_{dk}\cdot\hat{s}_k\cdot\hat{v}_{nk}^T$ is the set diminished with the x_{dn} sample.

The next approach is presented in line with Hansen [37] and Minka [61]. The D-dimensional observation vectors $\mathbf{x}$ are assumed here to be a composition of the signal and the white noise: $\mathbf{x} = \mathbf{s} + \mathbf{v}$. If we further assume, that both the signal and the noise are normally distributed, then the observed N point data sample $\mathbf{x}$ will have as well normal characteristics of the form $\mathcal{N}(\mu_s, \mathbf{\Sigma}_s + \mathbf{\Sigma}_v)$. Notice that the noise has zero mean: $\mu_v = 0$.

Thus, the density function of the data can be expressed by:

$$p(\mathbf{x}|\mu_s, \mathbf{\Sigma}_s, \mathbf{\Sigma}_v) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}_s + \mathbf{\Sigma}_v|}} exp(-\frac{1}{2}(\mathbf{x} - \mu_s)^T(\mathbf{\Sigma}_s + \mathbf{\Sigma}_v)^{-1}(\mathbf{x} - \mu_s)). \tag{2.6}$$

The $\mathbf{\Sigma}_s$ is further assumed to be singular, i.e. the rank $K \leq D$ and $\mathbf{\Sigma}_v = \sigma^2 I_D$, where I_D is $D \times D$ dimensional identity matrix. Then, $\mathbf{\Sigma} = \mathbf{\Sigma}_s + \sigma_v^2 I_D$ and it can be estimated from the data set:

$$\mu_s = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n, \quad \widehat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \mu_s)(\mathbf{x}_n - \mu_s)^T. \tag{2.7}$$

Further, $\widehat{\mathbf{\Sigma}}$ can be decomposed by singular value decomposition $\widehat{\mathbf{\Sigma}} = \mathbf{S\Lambda S}^T$, where $\mathbf{S} = \{s_d, \ d = 1 \ldots D\}$ collects the eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues λ_k ordered in descending order. Since the variation of the noise is assumed to be significantly lower than the signal, the first eigenvalues will represent the signal components while the last ones are responsible for the noise. Thus, by truncation in the eigenvalue space, the noise level can be estimated:

$$\widehat{\mathbf{\Sigma}}_K = \mathbf{S} \cdot diag([\lambda_1, \lambda_2, \cdots, \lambda_K, 0, \cdots, 0]) \cdot \mathbf{S}^T, \tag{2.8}$$

$$\widehat{\sigma}^2 = \frac{1}{D-K} Trace(\widehat{\mathbf{\Sigma}} - \widehat{\mathbf{\Sigma}}_K,) \tag{2.9}$$

$$\widehat{\mathbf{\Sigma}}_v = \widehat{\sigma}^2 I_D, \tag{2.10}$$

and then the signal covariance is described by:

$$\widehat{\mathbf{\Sigma}}_s = \mathbf{S} \cdot diag([\lambda_1 - \widehat{\sigma}^2, \lambda_2 - \widehat{\sigma}^2, \cdots, \lambda_K - \widehat{\sigma}^2, 0, \cdots, 0]) \cdot \mathbf{S}^T \tag{2.11}$$

If, in addition to the sample $\mathbf{x}$, the N_z points are observed $\mathbf{z}_n, n = \{1, 2, \ldots N_z\}$ forming the test set then the generalization error[7] can be estimated in cross-

[7] The generalization error is defined as the expected loss on the future independent sample.

validation scheme as the negative log-likelihood over this test set[8]

$$\begin{aligned}
\widehat{\mathcal{G}}_z &= -\frac{1}{N_z}\sum_{i=1}^{N_z}\log(p(\mathbf{z}_i|\mu_s,\mathbf{\Sigma}_s,\mathbf{\Sigma}_v)) \\
&= -\frac{1}{N_z}\sum_{i=1}^{N_z}\log\Big(\frac{exp(-\frac{1}{2}(\mathbf{z}_i-\mu_s)^T(\mathbf{\Sigma}_s+\mathbf{\Sigma}_v)^{-1}(\mathbf{z}_i-\mu_s))}{\sqrt{|2\pi(\mathbf{\Sigma}_s+\mathbf{\Sigma}_v)|}}\Big) \\
&= \frac{1}{2}\log|2\pi(\widehat{\mathbf{\Sigma}}_s+\widehat{\mathbf{\Sigma}}_v)|+\frac{1}{2}\mathrm{Trace}[(\widehat{\mathbf{\Sigma}}_s+\widehat{\mathbf{\Sigma}}_v)^{-1}\widehat{\mathbf{\Sigma}}_z], \qquad (2.12)
\end{aligned}$$

where $\widehat{\mathbf{\Sigma}}_s$ and $\widehat{\mathbf{\Sigma}}_v$ are estimated covariance structures of the signal and the noise, respectively, for the K largest eigenvalues. And the covariance of the test set $\widehat{\mathbf{\Sigma}}_z = \frac{1}{N_z}\sum_{n=1}^{N_z}(\mathbf{z}_n-\mu_s)^T(\mathbf{z}_n-\mu_s)$.

By minimizing the approximated generalization error the optimum signal space is found.

Some of the presented approaches are illustrated with an example. For this purpose a Gaussian cluster is generated, where the effective dimension is equal to 3, but the signal, due to the added noise, existed in 10-dimensional space. In order to determine the effective dimension of the data, the approximated generalization error is calculated according to equation 2.12. Results are presented on figure 2.10 (lower right plot). The optimum dimension that gives a minimum generalization error is equal to 3. Additionally, the Prediction Sum of Squares (equation 2.5) is computed. Here, the curve flattens at the third component suggesting that dimension for the signal space. For comparison the eigenvalue curves are also shown. Both the eigenvalues of the data and the scaled version, proposed in [32] indicate 3 components.

2.4 Discussion

Choosing the most suitable procedure for the projection should take into consideration the performance of the particular method as well as the dimensionality of the data and the application. For example, on-line learning can require a

[8] It is also possible to use for example the leave-one-out estimate or predicted generalization error base on the sum of training error and penalized term. Penalization is typically dependent on the model complexity eg. AIC or BIC criterion see [1, 69, 79].

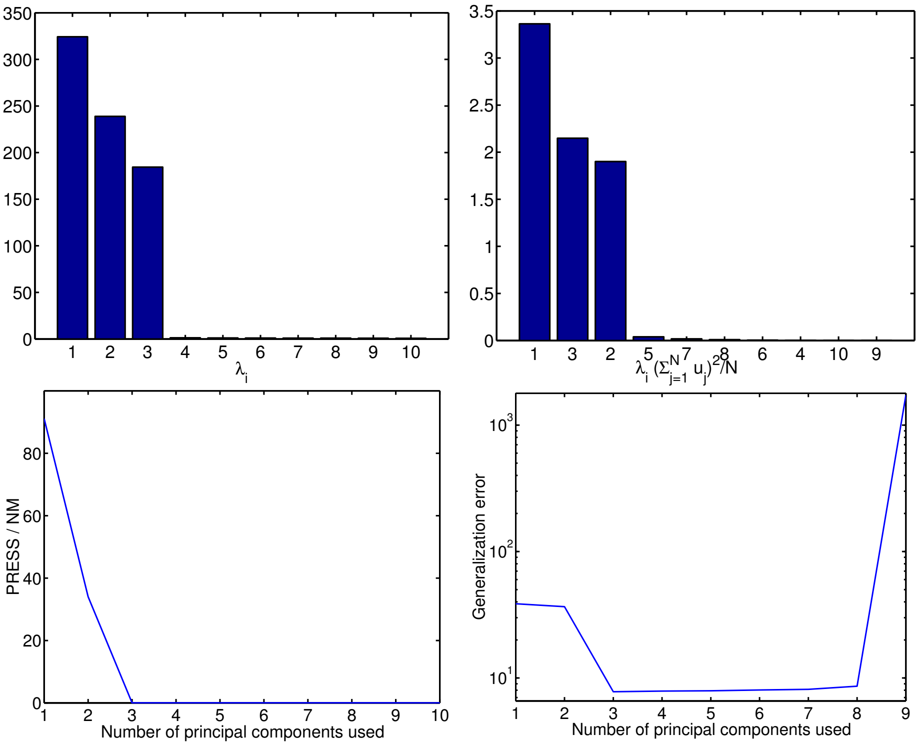


Figure 2.10 Illustration selecting the number of components. For the example, one Gaussian cluster is generated where the signal space had 3 components but the noisy data existed in 10 dimensions. Upper left plot present the eigenvalues λ_k. Here, the first 3 eigenvalues are dominant. A similar result is obtained for a scaled eigenvalue curve [32] (upper right plot). Lower panel presents the Predicted Sum of Squares (Eq. 2.5)) and the generalization error is calculated based on Eq. 2.12. In this simple case all methods find the correct space of the signal.

fast dimensionality reduction method whilst batch learning can afford a slower but more accurate technique. Principal Component Analysis is both simple and well suited to the applications presented in this work, where mostly medium-sized databases are available and batch learning is performed. Thus, in further investigations, PCA is used.

CHAPTER 3

Generalizable Gaussian Mixture models

Following the steps of the KDD process presented on figure 1.1, in the first step the transformation task is performed. Therefore in the preceding Chapter 2 the dimensionality reduction and feature selection techniques are presented. On such transformed data the essential part of KDD process may be applied, i.e. data mining. In the next Chapter 4 the methods for data mining results interpretation are introduced. The results of the full process based on the realistic data are presented in the last Chapter 7.

Data mining general task is to discover relationships among the data samples. Since, this work focuses mostly on segmentation one way to investigate these connections is to determine data underlying distribution. The mixture models are supplying with that cluster structure through the determined multi-modal density.

A mixture model is a particular, very flexible form of the density function. It can be written as a linear combination of the component densities $p(\mathbf{x}|j)$ as:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}|k)P(k). \tag{3.1}$$

Coefficients $P(k)$ are called *mixing parameters* or *mixing proportions* and they are in fact the prior probability of the data being generated by the component k. $P(k)$ satisfies the following conditions: $\sum_{k=1}^{K} P(k) = 1$ and, $0 \leq P(k) \leq 1$. Component densities are, naturally, normalized so that they satisfy the conditions of the probability density function, i.e. $\int p(\mathbf{x}|k)d\mathbf{x} = 1$. By Bayes optimal decision rule (assuming 0/1 loss function [69]) a new point $\mathbf{z}$ is assigned to cluster k if

$$k = \underset{k}{argmax}\ p(k|\mathbf{z}) = \underset{k}{argmax}\ \frac{p(\mathbf{z}|k)P(k)}{\sum_k p(\mathbf{z}|k)P(k)} \tag{3.2}$$

where $\sum_{k=1}^{K} p(k|\mathbf{z}) = 1$.

Mixture models are widely used, mostly due to their flexibility in modeling the unknown distribution shapes. In the data mining area, they are useful in estimation of density and what follows in cluster analysis, i.e. the investigation of group structure in the data (which is the main thread of this work). They can be used also in many other applications like for example: neural networks, soft weight sharing or mixtures of experts, description of which can be found in [10].

Provided with sufficient number of components and correct selected parameters the accuracy of the mixture models is very high. The estimate is accurate even when the data was generated from a distribution which was different than the chosen component densities. In such case, typically, only larger number of components are needed in the approximation. As the component densities $p(\mathbf{x}|k)$ all valid density functions can be used. However, exponential density family [70] and especially the Gaussian p.d.f. is a good choice due to simple analytical expressions.

The learning method for the mixture model is based on the maximum likelihood (section 3.1) which leads to the Expectation-Maximization (EM) algorithm (section 3.2).

3.1 Maximum likelihood

When modeling the data distribution with the parametric form of the density function (e.g., mixture model) the set of optimum parameters has to be determined. The *maximum likelihood* (ML) estimation is a solid tool for learning those parameters. In that technique, the optimum values of the parameters are found by maximizing the likelihood derived from the training data set [10, 69].

In the parametric form of the density function $p(\mathbf{x}|\theta)$ the density depends on the set of parameters θ.

The training data set consist of N samples $\mathcal{D} = \{\mathbf{x}_n, n = 1, 2, \dots N\}$. The data likelihood $\mathcal{L}$, assuming $i.i.d.$ data points distribution, can be written in the following way:

$$\mathcal{L}(\theta) = p(\mathcal{D}|\theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|\theta)). \tag{3.3}$$

Since we are looking for the set of parameters θ that maximize the likelihood $\mathcal{L}(\theta)$, the solution can be found from the differential equations of the form $\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = 0$. Then, the optimum θ^* is found that maximizes the likelihood $\mathcal{L}$,

$$\theta^* = \underset{\theta^*}{argmax} \;\; \mathcal{L}(\theta). \tag{3.4}$$

It is often advantageous (for the computational reasons) to use the log-likelihood instead of the regular likelihood (equation 3.3) or, what is equivalent, to minimize the negative log-likelihood.

3.2 Expectation-Maximization algorithm

Even though, the maximum likelihood approach supplies with the solution to the optimization problem it does not provide the method for calculating the parameters. Sets of differential equations are highly nonlinear and therefore difficult to solve. However, the ML approach gives the opportunity to derive the iterative algorithm that will allow to find the approximation to the correct solution and at the same time will make the optimization process much simpler. In order to find the maximum likelihood estimate of the parametric models the Expectation-Maximization algorithm (EM) can be applied, which was introduced in [21]. The other alternative technique, however not addressed here, for determining parameters of the probability density function is a Variational Bayes approach [2, 5, 16].

The EM term was first introduced and formalized by Dempster, Laird and Rubin in 1977 [21]. It is an iterative schema which involves the non-linear function optimization and re-estimation of the parameters what leads to the maximization of the data likelihood.

Let us write the log-likelihood as a function of some visible v and hidden h variables

$$\mathcal{L}(\theta) = \log(p(v|\theta)) = \log \int p(h, v|\theta) dh \tag{3.5}$$

Now, by introducing the additional function – the distribution over the hidden states $q(h)$ and by using Jensen's inequality (appendix A), the likelihood given in equation 3.5 can be rewritten in the following way:

$$\mathcal{L}(\theta) = \log \int q(h) \frac{p(h, v|\theta)}{q(h)} dh \geq \int q(h) \log \frac{p(h, v|\theta)}{q(h)} dh = \mathcal{F}(q(h)\theta), \tag{3.6}$$

where

$$\mathcal{F}(q(h), \theta) = \int q(h) \log(p(h, v|\theta)) dh - \int q(h) \log(q(h)) dh \tag{3.7}$$

The last term $\int q(h) \log(q(h)) dh$ is, in fact, an entropy of q. The lower bound on the likelihood $\mathcal{F}(q(h), \theta)$ is introduced in equation 3.7. It can be proofed [21] that the difference between the likelihood $\mathcal{L}$ and the new objective function $\mathcal{F}$ is the nonnegative Kullback-Leibler (KL) divergence between the distribution over hidden variables and the probability of the hidden states. Note also, that maximizing $\mathcal{F}$ maximizes $\mathcal{L}$, what is our objective.

Finally, the EM optimization algorithm can be introduced.

- **E step:**
  In the E step the distribution over the hidden variables $\mathcal{F}(q(h), \theta)$ is estimated given the data and the current parameters i.e.

  $$q^k(h) = \underset{q(h)}{argmax} \ \ \mathcal{F}(q(h), \theta^{k-1}) \tag{3.8}$$

- **M step:**
  The M step is responsible for modifying the parameters in order to maximize the joint distribution over the data and hidden variables i.e.

  $$\theta^k = \underset{\theta}{argmax} \ \ \mathcal{F}(q^k(h), \theta) \tag{3.9}$$

Figure 3.1 Expectation-Maximization algorithm

When the exact EM steps are applied[1] the algorithm is maximizing the data likelihood at each step. Even though, usually, only approximate EM steps can be applied, the algorithm converges and still maximizes the data likelihood. The convergence proof is provided by Dempster et al. [21], and it can as well be found in [10, 92].

3.3 Unsupervised Generalizable Gaussian Mixture model

Gaussian probability density function is the most often choice for the density components. This assures the ability of deriving analytically the expressions for parameter updates in the EM algorithm. It also gives a flexible and accurate estimate of the true density, which can be used in discovering the hidden cluster structure in the data. The Gaussian mixture model is also applied for example in [39, 52, 53, 80].

Let define the density of the D-dimensional data vector $\mathbf{x}$ under the model assumptions by the following equations:

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} p(\mathbf{x}|\theta_{\mathbf{k}}) \cdot P(k), \tag{3.10}$$

where

$$p(\mathbf{x}|\theta_{\mathbf{k}}) = (2\pi)^{-\frac{D}{2}} |\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x} - \mu_k)) \tag{3.11}$$

The Gaussian components $p(\mathbf{x}|\theta_{\mathbf{k}})$ in equation 3.10 are mixed with proportions $P(k)$, which satisfy earlier mentioned conditions. θ_k represents the collection of parameters for component k so that $\theta_k \equiv \{\mu_k, \mathbf{\Sigma}_k, P(k)\}$, where K is the total number of components. The parameters are estimated from the set of observations $\mathcal{D} = \{\mathbf{x}_n, n = 1, 2, \ldots N\}$ by the Expectation-Maximization algorithm described in section 3.2. The objective of the iterative EM is to minimize

[1] The exact form of the hidden states distribution is known and the parameters minimize the joint distribution.

the cost function – the negative log-likelihood of the form:

$$\begin{aligned}
\mathcal{L}(\theta|\mathcal{D}) &= -\log(p(\mathcal{D}|\theta)) \\
&= -\log(\prod_{n}^{N} p(\mathbf{x}_n|\theta)) \\
&= -\sum_{n=1}^{N} \log(p(\mathbf{x}_n|\theta))) \\
&= -\sum_{n=1}^{N} \log\Big(\sum_{k=1}^{K} p(\mathbf{x}_n|\theta_k)\cdot P(k)\Big). \qquad (3.12)
\end{aligned}$$

In the E-step the cluster posterior $P(k|\mathbf{x}_n)$ is estimated for the fixed set of the Gaussian parameters θ_k. The optimum parameters that minimize the cost function are found in the M-step.

Naturally, the cost function is monotonically decreasing, what is ensured by EM therefore in order to find the optimum model complexity, a generalization error must be determined. By the complexity of the model the model order is understood, i.e. in the case of the discussed GGM model the number of clusters or what is strictly connected - number of parameters in the mixture. The model has optimal complexity if its generalization error is minimal. By definition *the generalization error* is *a measure of how well a model can respond to new data on which it has not been trained but which are related in some way to the training patterns. An ability to generalize is crucial to the decision making ability of the model.* For example, the generalization error can be simply calculated from the validation set or approximated from the training set. Then the rule is to add to the cost function (equation 3.12) the penalty value that is proportional to the total number of parameters[2] (eg. Akaike Information Criterion [1], Bayesian Information Criterion [79]). The minimum point of this composition (the cost function plus the number of parameters in the system) defines the optimum model complexity. Figure 3.2 illustrates this basic idea.

Two techniques are presented in the implementation of the EM algorithm for Gaussian mixture model. The EM algorithm for Gaussian Mixture model can be implemented in two different ways. So called *hard assignment* algorithm bases on the 0/1 decision function, i.e. the points are uniquely assigned to one

[2] The simple models that generalize well are preferred.

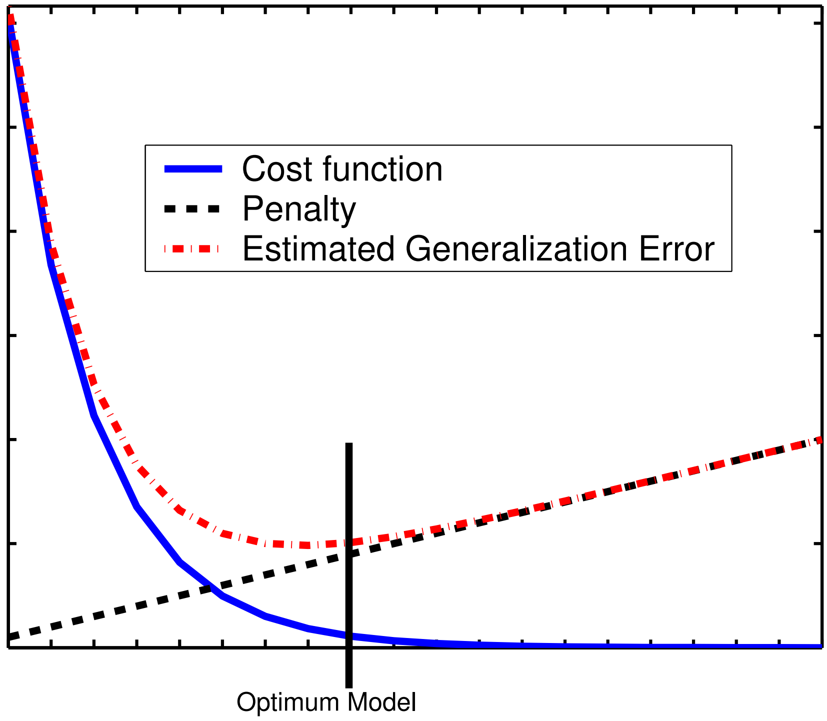


Figure 3.2 The illustration of the selection of model complexity. The optimum model is minimum in the estimated generalization error, which is determined as a sum of the cost function and the penalty value, defines to optimum model.

cluster with maximum cluster posterior $p(k|\mathbf{x})$. Then the cluster parameters are estimated from the subsets of the data assigned to this cluster. The other method is to use in the computations the precise cluster posterior probability (*soft assignment*). Such a method certainly offers better density estimation. This technique, however, returns often high number of components with few clusters without assigned members form the actual data. Therefore, for clustering purposes, hard assignment algorithm is often more useful.

Figure 3.3 presents the algorithm for unsupervised Generalizable Gaussian Mixture model with soft assignment. The parameters are initialized with the data variance and randomly selected data points are used as cluster centers.

```
Calculate mean and the covariance of the data:
$\mu_0 = N^{-1}\sum_n \mathbf{x}_n$,
$\mathbf{\Sigma}_0 = N^{-1}\sum_n (\mathbf{x}_n - \mu_0)(\mathbf{x}_n - \mu_0)^T$

**Initialization**
1. Initialize $\mu_k$ as the random point drown from the data set.
2. Initialize $\mathbf{\Sigma}_k = \mathbf{\Sigma}_0$
3. Initialize $P(k) = \frac{1}{K}$
**Optimization**
    • **E-step:**
        1. Calculate the likelihood: $p(\mathbf{x}|\theta)$
        2. Calculate the cost function: $\mathcal{L}(\theta)$
        3. Compute posterior: $p(k|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|k)P(k)}{p(\mathbf{x}_n)}$

        Split the data set $\mathcal{D}$ into two parts:
        $-\mathcal{D}_\mu$ (for mean estimation)
        $-\mathcal{D}_\Sigma$ (for covariance estimation).

    • **M-step:**
        1. Estimate $\mu_k = \frac{\sum_{n\in D_\mu} p(k|\mathbf{x}_n)\cdot\mathbf{x}_n}{\sum_{n\in D_\mu} p(k|\mathbf{x}_n)}$
        2. Estimate $\mathbf{\Sigma}_k = \frac{\sum_{n\in D_\Sigma} (\mathbf{x}_n-\mu_k)(\mathbf{x}_n-\mu_k)^T\cdot p(k|\mathbf{x}_n)}{\sum_{n\in D_\Sigma} p(k|\mathbf{x}_n)}$
        3. Estimate $P(k) = \sum_n p(k|\mathbf{x}_n)$
```

Figure 3.3 The unsupervised Gaussian Mixture algorithm.

To obtain hard assignement algoritm the cluster parameters μ_k, $\mathbf{\Sigma}_k$ and $P(k)$ are estimated in the M-step only based on the points assigned to the particular cluster, the cluster posterior $p(k|\mathbf{x})$ is quantized to binary (0/1) representation.

It is well known that the classic EM algorithm for Gaussian mixture model overfits easily, i.e., the optimum parameters maximizing the likelihood tend towards the Gaussian densities of zero covariance placed over each of the data points[3]. In order to avoid this artifact the generalization is introduced by divid-

[3] It can be easily shown that for $\mu_i = \mathbf{x}_i$ and $\sigma \rightarrow 0$ the negative log-likelihood $\mathcal{L} \rightarrow -\infty$.

ing the data set into two parts, and estimating mean and the covariance from this disjoint sets. The alternative possibility is the Variational Bayes [2] methods, however they are not addressed in this work. The EM update steps are performed until the algorithm converge what in practice means that the early stopping criteria is used. The algorithm has converged when no changes in the cluster assignments are observed.

In order to achieve good estimation of the parameters, the number of data samples must outnumber the number of parameters in the model. In case of unsupervised Gaussian Mixture model the mean, covariance and the mixing proportions are estimated, what means that for each cluster there are $D + \frac{D\cdot(D+1)}{2} + 1$ parameters to estimate, where D is the data dimension. For example, for 2-dimensional data there are $2 + \frac{2\cdot(2+1)}{2} + 1 = 6$ parameters to estimates per cluster while in the case of 10 dimensions this number grows to 66 parameters per cluster. In result, if the data consist from, e.g. 10 clusters, only 60 points are needed in estimation of 2-dimensional case but for 10-dimensional data no less than 660 points are required to ensure correct estimate of the parameters. So, the number of data points needed in estimation grows quadratically with the feature space dimensionality.

3.4 Supervised Generalizable Gaussian Mixture model

Supervised Gaussian Mixture model is used in the classification schema when not only data points are provided, but as well the corresponding labels, i.e. the data set consist of couples: $\mathcal{D} = \{\{\mathbf{x}_n, y_n\}, n = 1, 2, \ldots N\}$, where $\mathbf{x}_n$ is a D-dimensional data sample and y_n is the corresponding class label, $y_n \in \{c = 1 \ldots c = C\}$, where C is a total number of observed classes.

The idea is to adapt the separate Gaussian mixture for each of the observed classes. Thus, the unsupervised mixture algorithm presented on figure 3.3 is used for each of the class-separated subsets of the data.

The density of the data point $\mathbf{x}$ belonging to the class c is then expressed by joint probability:

$$p(\mathbf{x}, c|\theta) \quad = \quad \sum_{k=1}^{K} p(\mathbf{x}|\theta_k) \cdot P(k|c) \cdot P(c). \tag{3.13}$$

Similarly to unsupervised Gaussian mixture the $p(\mathbf{x}|\theta_\mathbf{k})$ states for Gaussian density components (equation 3.10), $P(c)$ is the probability of the class c, $\sum_{c=1}^{C} P(c) = 1$ and it is calculated from the data simply by counting members of each class. $P(k|c)$ gives the proportions of the clusters in the class c.

3.5 Unsupervised/supervised Generalizable Gaussian Mixture model

For classification purposes both unlabeled and labeled data may be useful. Castelli & Cover in [17] proofs that labeled samples have exponential contribution in reducing the probability of error, since they carry important information about decision boundaries. Thus, unlabeled data alone, are insufficient in classification task due to the lack of this information. In real world applications, however, labels are usually difficult to obtain. For example, in medical diagnoses or the categories of the web-pages it is often not only expensive but also time consuming and in some applications unrealistic (e.g., labeling of the continuously growing world wide web). Therefore, an excellent idea is to use available labeled examples with collected unlabeled data as, for example, results of medical tests or web-pages. Thus, the unsupervised/supervised Generalizable Gaussian Mixture (USGGM) is proposed.

In USGGM the joint input/output density is modeled, as previously, as the Gaussian Mixture [53, 60, 63] in the following manner:

$$p(y, \mathbf{x}|\theta) = \sum_{k=1}^{K} P(y|\theta_k) p(\mathbf{x}|\theta_k) P(k). \tag{3.14}$$

$p(\mathbf{x}|\theta_k)$ are the Gaussian density components defined earlier by equation 3.10, which are mixed with the non-negative proportions $P(k)$. An additional set of parameters, in the case of this model, the class cluster posteriors $P(y|k)$ are also non-negative and $\sum_{y=1}^{C} P(y|k) = 1$. θ collects the parameters of the model what in the case of USGGM model for the k-th component is described as $\theta_k \equiv \{P(y|k), \mu_k, \mathbf{\Sigma}_k, P(k)\}$. It is assumed that the joint input/output density factorize so $p(y, \mathbf{x}) = P(y|k)p(\mathbf{x}|k)$ and the data points are assigned to the class when $\widehat{y} = \underset{y}{argmax}\ P(y|\mathbf{x})$, where

$$p(y|\mathbf{x}) = \frac{p(y,\mathbf{x})}{p(\mathbf{x})} = \sum_{k=1}^{K} P(y|k)p(\theta_k|\mathbf{x}) = \sum_{k=1}^{K} P(y|k)\frac{p(\mathbf{x}|\theta_k)P(k)}{p(\mathbf{x})}. \quad (3.15)$$

The entire data set consists of the unlabeled $\mathcal{D}_u = \{\mathbf{x}_n; n = 1 \ldots N_u\}$ and labeled examples $\mathcal{D}_l = \{\mathbf{x}_n, y_n; n = 1 \ldots N_l\}$. The objective is to estimate joint density parameters θ based on the data set $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_l$ that will ensure generalizability. The cost function (the negative log-likelihood) for this model is given by:

$$\begin{aligned} \mathcal{L} &= -\log(p(\mathcal{D}|\theta)) \quad (3.16) \\ &= -\sum_{n \in \mathcal{D}_l} \log \sum_{k=1}^{K} P(y_n|k)p(\mathbf{x}_n|k)P(k) - \lambda \sum_{n \in \mathcal{D}_u} \log \sum_{k=1}^{K} p(\mathbf{x}_n|k)P(k), \end{aligned}$$

where $0 \leq \lambda \leq 1$ is called a *discount factor*. The discount factor is introduced to the cost function in order to control the influence of the unlabeled samples, and that affects also the EM updates. Its value is estimated in a cross-validation scheme. For a small number of labeled examples, EM will converge almost to the unsupervised algorithm where the labeled points will effect only the initialization part and the identification of the components with the class labels. With growing size of $\mathcal{D}_l$ the influence of the unlabeled data set starts to decrease since the labeled data set is providing enough information both for classification and for parameter estimation. Therefore, it is generally expected that λ is close to one with only few labeled data points and decreases towards zero with the increasing size of labeled samples.

The unsupervised/supevised Gaussian Mixture algorithm is presented in figure 3.4. Similarly to previously described algorithms, the parameters for USGGM are optimized by the EM algorithm. To ensure generalization the means and covariances are estimated from a disjoint data set and $P(y|k)$ and $P(k)$ from the whole set. As previously, the cluster covariances $\mathbf{\Sigma}_k$ are initialized with the data covariance, means μ_k are chosen randomly from the samples, component proportions $P(k)$ take uniform values and the class posterior probabilities $P(y|k)$ are forming the table computed from the available labels.

Initialization

1. Choose values for K and $0 \leq \lambda \leq 1$.

2. Let $\mathbf{i}$ be K different randomly selected indices from $\{1, 2, \cdots, N\}$, and set $\mu_k = \mathbf{x}_{\mathbf{i}_k}$.

3. Let $\boldsymbol{\Sigma}_0 = N^{-1} \sum_{n \in \mathcal{D}} (\mathbf{x}_n - \mu_0)(\mathbf{x}_n - \mu_0)^\top$,where $\mu_0 = N^{-1} \sum_{n \in \mathcal{D}} \mathbf{x}_n$, and set $\forall k : \; \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_0$.

4. Set $\forall k : \; P(k) = 1/K$.

5. Compute class prior probabilities: $P(y) = N_l^{-1} \sum_{n \in \mathcal{D}_l} \delta(y_n - y)$, where $\delta(z) = 1$ if $z = 0$, and zero otherwise. Set $\forall k : \; P(y|k) = P(y)$.

6. Select a split ratio $0 < \gamma < 1$. Split the unlabeled data set into disjoint sets as $\mathcal{D}_u = \mathcal{D}_{u,1} \cup \mathcal{D}_{u,2}$, with $|\mathcal{D}_{u,1}| = [\gamma N_u]$ and $|\mathcal{D}_{u,2}| = N_u - |\mathcal{D}_{u,1}|$. Do similar splitting for the labeled data set $\mathcal{D}_l = \mathcal{D}_{l,1} \cup \mathcal{D}_{l,2}$.

Repeat until convergence

1. Compute posterior component probabilities:
$p(k|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|k)P(k)}{\sum_k p(\mathbf{x}_n|k)P(k)}$, for all $n \in \mathcal{D}_u$,
and for all $n \in \mathcal{D}_l$, $p(k|y_n, \mathbf{x}_n) = \frac{P(y_n|k)p(\mathbf{x}_n|k)P(k)}{\sum_k P(y_n|k)p(\mathbf{x}_n|k)P(k)}$.

2. For all k, update means

$$\mu_k = \frac{\sum\limits_{n \in \mathcal{D}_{l,1}} \mathbf{x}_n P(k|y_n, \mathbf{x}_n) + \lambda \sum\limits_{n \in \mathcal{D}_{u,1}} \mathbf{x}_n P(k|\mathbf{x}_n)}{\sum\limits_{n \in \mathcal{D}_{l,1}} P(k|y_n, \mathbf{x}_n) + \lambda \sum\limits_{n \in \mathcal{D}_{u,1}} P(k|\mathbf{x}_n)}$$

3. For all k, update covariance matrices

$$\boldsymbol{\Sigma}_k = \frac{\sum\limits_{n \in \mathcal{D}_{l,2}} \mathbf{S}_{kn} P(k|y_n, \mathbf{x}_n) + \lambda \sum\limits_{n \in \mathcal{D}_{u,2}} \mathbf{S}_{kn} P(k|\mathbf{x}_n)}{\sum\limits_{n \in \mathcal{D}_{l,2}} P(k|y_n, \mathbf{x}_n) + \lambda \sum\limits_{n \in \mathcal{D}_{u,2}} P(k|\mathbf{x}_n)}$$

where $\mathbf{S}_{kn} = (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^\top$. Perform a regularization of $\boldsymbol{\Sigma}_k$.

4. For all k, update cluster priors

$$P(k) = \frac{\sum\limits_{n \in \mathcal{D}_l} P(k|y_n, \mathbf{x}_n) + \lambda \sum\limits_{n \in \mathcal{D}_u} P(k|\mathbf{x}_n)}{N_l + \lambda N_u}$$

5. For all k, update class cluster posteriors

$$P(y|k) = \frac{\sum\limits_{n \in \mathcal{D}_l} \delta(y - y_n) P(k|y_n, \mathbf{x}_n)}{\sum\limits_{n \in \mathcal{D}_l} P(k|y_n, \mathbf{x}_n)}$$

Figure 3.4 The USGGM algorithm.

3.6 Outlier & Novelty detection

The outlier detection can be a part of the data cleaning process (see figure 1.1) as well as data mining. In the first case the outlier problem is addressed in order to enhance the data model. In the second case the outlier detection is itself considered as a data mining task where outliers are the final outcomes of the processing. The following outlier detection techniques base on the probabilistic outcome of the GGM model, and therefore they are introduced in this chapter. However, this work do address the outlier detection only as a data cleaning problem performed prior to the data mining task.

Even though, methods presented later in this chapter are developed for the outlier detection they can be also used in novelty detection.

There are various origins of outliers. When the outlier sample is a result of an error in the system, it may greatly influence the learning and estimation process. Therefore, it is usually advised to exclude it from the data set, before estimation, unless the applied model is designed to deal with outliers. On the other hand, when the outlier is in fact an unusual example, it may itself be a subject to the study. Thus, for example in medical systems, the outlier detection techniques can be used to detect, e.g. incorrect diagnoses, abnormal changes in the performed tests, scans or other signals. A novel sample is closely related to an outlier. The difference is that novelty comes in fact from the new data generation mechanism appearing in the observed data after the model was trained and the parameters were obtained. The novel points have low likelihood values in the current model.

Thus, the ability to spot the outlier and novel samples is an important task in processing data and in cluster analysis. Many definitions of an outlier can be formulated. Here, *an outlier* is referred as *an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism* [47]. Based on that definition it is assumed that the outlier observations are not produced by the model, or the survey is possibly harmed by an error. Such an unlikely sample should have low probability value in that model. In the case of Gaussian mixture it means that the probability of the outlier data point for any cluster is low. Figure 3.5 illustrates this problem. The outlier point is placed far away from the rest of the data samples. Therefore, it is unlikely for it to be generated by any of the presented densities. Such a situation may be indication of abnormal behavior, error or novel (unobserved

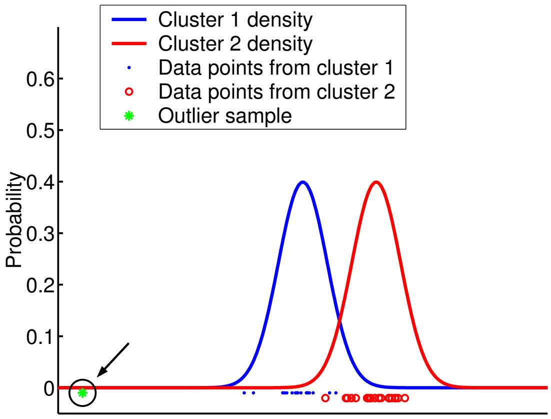


Figure 3.5 The illustration of the outlier sample. The outlier observation point (green star) can not be unambiguously assigned to any of the presented densities since the probability for each of them is very low.

before) data point appearing in the data set. Any of those scenarios is important in data analysis.

If the existence of the outliers is not taken into consideration during optimization process, the final result may be affected greatly. For example, figure 3.6 illustrate such an influence in case of simple linear model. Including the outlier point in the parameter estimation (blue dashed line) causes substantial deviation from the correct result (red solid line). Removing this unlikely data point from the training set will produce significantly better estimate (green dashed line).

Similar techniques are used in the *novelty* detection. While in case of the outliers, the unlikely data points were removed from the data set to improve generalization, the novelty samples should be included and the model parameters re-estimated.

3.6.1 Cumulative distribution

A simple outlier detection technique is proposed in line with [39]. The method bases on the estimate of the input density $p(\mathbf{x})$ which is available through eg. mixture model.

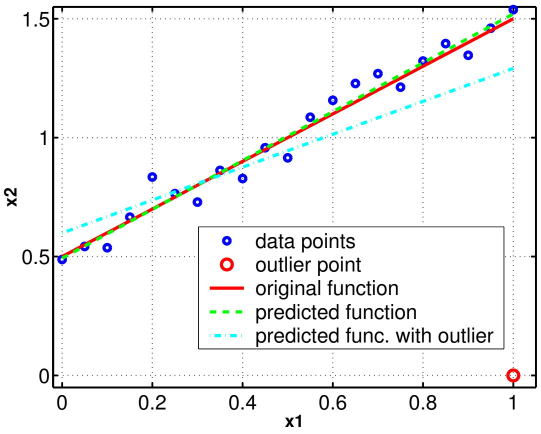


Figure 3.6 The influence of the one outlier sample on the estimation of the function parameters from the data. A simple linear function is used as an example. The linear function calculated from the data set with one outlier sample produces strongly skewed incorrect result.

The cumulative distribution $Q(t)$ is defined as the probability of the data that has likelihood below some threshold t, calculated for all the thresholds, i.e. $Q(t) = P(\mathbf{x} \in \mathcal{R})$ where $\mathcal{R} = \{\mathbf{x}_n : p(\mathbf{x}_n) < t\}$. Now, the threshold in set on the distribution $Q(t)$, which allows to reject the low probable events. Typically, low value is chosen, for example 5%, what means that the rejected data has the likelihood lower than 5% of the maximum likelihood value observed in the data.

Again, the same technique can be used in detection of the novel events in the new observed data.

3.6.2 Outlier cluster

The other method [54] is working as an extension to the earlier presented Generalizable Gaussian Mixture model. However similar techniques can be developed for arbitrary mixture model. It simultaneously detects the outliers and estimates the data density in the mixture model framework.

The method involves the additional, especially created, wide, outlier cluster, which is placed in the center of the data. Figure 3.7 illustrates this idea. Due

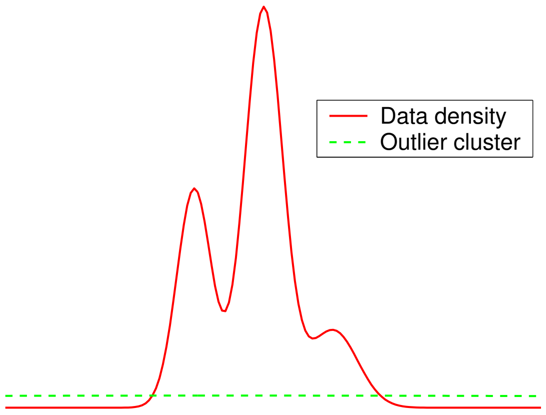


Figure 3.7 The outlier cluster (dashed line) has much larger covariance than the data (solid line). Therefore the likelihood shown on the figure in the center is dominated by data and on tails by outlier density.

to the high variance, the density of the outlier cluster takes much lower values than the data density in the center of the distribution but higher on the tails. Therefore, the unlikely samples fall into the outlier cluster rather than contribute to the process of estimating the parameters of the data density.

The method uses modified unsupervised GGM in estimation. The model is a linear combination of $K+1$ component densities and the proportions:

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K+1} p(\mathbf{x}|\theta_k)P(k). \tag{3.17}$$

In the formula the Gaussians (given by equation 3.11) are used as component densities $p(\mathbf{x}|\theta_k)$. The parameters are estimated, as in earlier presented models, via EM algorithm that minimizes the negative log-likelihood given in equation 3.12.

This modification of the UGGM algorithm, presented in figure 3.3, uses one additional Gaussian cluster with fixed parameters: $\mu_{K+1} = \mu_0 = N^{-1}\sum_n \mathbf{x}_n$ and $\mathbf{\Sigma}_{K+1} = c\mathbf{\Sigma}_0 = cN^{-1}\sum_n(\mathbf{x}_n - \mu_0)(\mathbf{x}_n - \mu_0)^T$, where c is a multiplying constant which decides about the wideness of the outlier cluster and typically takes large values[4]. The EM updates for μ_k and $\mathbf{\Sigma}_k$ concern only the K clusters describing the data density while the parameters of the outlier cluster $K + 1$ remain the same. Mixing proportions $P(k)$ that are indication of the cluster membership are updated for all components.

Even though, the presented technique is developed for unsupervised GGM it can easily be extended to other discussed models like SGGM and USGGM.

3.6.3 Evaluation of outlier detection methods

A 2-dimensional toy data set was generated for illustration and comparison of the outlier detection techniques presented in sections 3.6.1 and 3.6.2. Results are shown on figure 3.8.

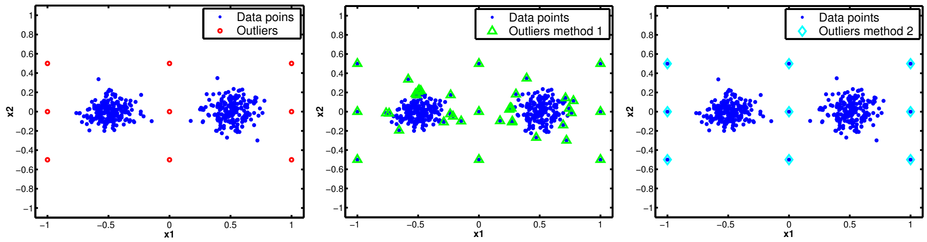


Figure 3.8 Two Gaussian clusters with the outliers are used as the data set (left panel). The outliers are marked in red. Method 1 (middle plot) is corresponding to the outcome of the cumulative distribution. The samples classified as outliers are marked with the green triangle. Here, more samples than expected is detected as outliers. Right panel is presenting the results of the clustering with the outlier cluster (method 2) . The examples that were classified as outliers are marked with the cyan diamond. Precisely the same points that were originally outliers are predicted as outliers.

Two Gaussian distributed spherical clusters were generated. Then, additional 9 outlier samples formed in regular grid were included in the data set. Left plot of figure 3.8 shows the created data. Blue points are generated by Gaussians and red circles are the outliers. In the middle plot the result of the cumulative distribution is presented and in right plot the outcome of outlier cluster technique is shown. The cumulative distribution was used with the threshold of 0.5% but

[4] Value of $c = 10$ is used in the experiments

still many low probable samples from the Gaussian clusters were classified as outliers. In case of outlier cluster method the result is accurate.

3.7 Discussion

Summarizing, the mixture models are universal and very flexible approximations for the data densities. The significant drawback of that model is the computational complexity. The number of parameters in the model grows quadratically with the data feature dimension what next requires a large number of data points needed for good parameter estimation. For high dimensional data it is moreover a time consuming process, since the inversion of the $D \times D$ covariance matrix is needed in each training iteration. Therefore, dimensionality reduction techniques are often used in the preprocessing step.

In many applications, finding outliers, i.e. rare events, is more interesting than finding the common cases. They indicate either some unusual behavior or conditions or give information about new introduced cases or simply about errors existing in the system. Therefore, ability of detecting outliers is an useful feature for the model.

The outlier cluster was found superior over the cumulative distribution. It does not detect any outlier samples in the data where in fact there is no outliers. If the density is estimated accurately then the outcome of this method is also precise. The outlier detection technique is however, based on the mixture model and as such inherit all its drawbacks.

CHAPTER 4

Cluster visualization & interpretation

Visualization and the result interpretation, the next step in the KDD process (see figure 1.1), follows the transformation part that is described in Chapter 2 and the modeling part presented in Chapter 3. The following chapter discovers the visualization and interpretation possibilities based on the outcome of the Generalizable Gaussian Mixture model. The modeling results of the realistic data sets together with theirs interpretation are visualized in the Chapter 7.

4.1 Hierarchical clustering

Hierarchical methods for unsupervised and supervised data mining provide a multilevel description of data. It is relevant for many applications related to information extraction, retrieval, navigation and organization, for further reading in this topic refer to, e.g. [13, 28]. Many different approaches have been suggested to hierarchical analysis from divisive to agglomerative clustering and recent developments include [11, 27, 58, 69, 87, 89]. In this section, *the agglomerative probabilistic clustering* method is investigated [52, 53, 80] as an additional level based on the outcome of the Generalizable Gaussian Mixture model.

The agglomerative hierarchical clustering algorithms start with each object as a separate element. These elements are successively combined based on *a similarity measure* until only one group remains.

Let us start from the outcome of the unsupervised GGM model (described in section 3.3), the probability density components $p(\mathbf{x}|k)$ (equation 3.11), where k is an index of the cluster with parameters collected in θ_k, and treat it as first level $j = 1$ of the new developed hierarchy. On this level, K clusters are observed that are described by means μ_k, covariances $\mathbf{\Sigma}_k$ and the proportion values $P(k)$, $k = 1, 2, \ldots, K$. The goal is to group together the clusters that are similar to each other. This similarity can be understood, for example, as the distance between the clusters in the Euclidean space, or the similar characteristics in the probability space. Therefore, various similarity measures can be applied what often leads to different clustering results. A few of the possible choices for distance measures between the clusters are presented further in this chapter.

The simplest technique for combining the clusters is to merge only two clusters at each level of the hierarchy. Two clusters are merged, when the distance between them is the smallest. The procedure is repeated until one cluster at the top level is reached containing all the elements. That is, at level $j = 1$ there are K clusters and there is one cluster at the final level, $j = 2K - 1$.

Let $p_j(\mathbf{x}|k)$ be the density for the k'th cluster at level j and $P_j(k)$ the corresponding mixing proportion. Then the density model at level j is defined by:

$$p(\mathbf{x}) = \sum_{k=1}^{K-j+1} P_j(k) p_j(\mathbf{x}|k). \tag{4.1}$$

If cluster r and m at level j are merged into ℓ at level $j + 1$ then

$$p_{j+1}(\mathbf{x}|\ell) = \frac{p_j(\mathbf{x}|r) \cdot P_j(r) + p_j(\mathbf{x}|m) \cdot P_j(m)}{P_j(r) + P_j(m)}, \tag{4.2}$$

and

$$P_{j+1}(\ell) = P_j(r) + P_j(m). \tag{4.3}$$

The class posterior

$$P_j(k|\mathbf{x}_n) = \frac{p_j(\mathbf{x}|k)P_j(k)}{p(\mathbf{x})} \tag{4.4}$$

propagates also in the hierarchy so that

$$P_{j+1}(\ell|\mathbf{x}_n) = P_j(r|\mathbf{x}_n) + P_j(m|\mathbf{x}_n). \tag{4.5}$$

Once, the hierarchy is created it is possible to determine the class/level membership for new samples. Thus, $\mathbf{x}_n$ is assigned to cluster k at level j if

$$P_j(k|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|k)P(k)}{p(\mathbf{x}_n)} > \rho, \tag{4.6}$$

where ρ is a threshold value, typically close to 1. That ensures high confidence in the assignment. The data sample "climbing up" the tree structure gains confidence with each additional level. That means, that if the sample at level j can not be assigned unambiguously to any of the clusters it will surely be possible at one of the higher levels of the hierarchy.

4.1.1 Similarity measures

4.1.1.1 Kullback-Leibler similarity measure

The natural similarity measure between the cluster densities is the symmetric Kullback-Leibler (KL) divergence [10, 69], since it reflects dissimilarity between the densities in the probabilistic space. Symmetric KL for agglomerative hierarchical clustering was introduced and investigated in [52]. It is defined as

$$\mathcal{D}(r,m) = \frac{1}{2}\int p(\mathbf{x}|r)\log\frac{p(\mathbf{x}|r)}{p(\mathbf{x}|m)}d\mathbf{x} + \frac{1}{2}\int p(\mathbf{x}|m)\log\frac{p(\mathbf{x}|m)}{p(\mathbf{x}|r)}d\mathbf{x}. \tag{4.7}$$

On level $j = 1$, KL divergence for the Gaussian clusters can be expressed by the simplified form:

$$\begin{aligned} \mathcal{D}_1(r,m) &= -\frac{D}{2} + \frac{1}{4}(\mathrm{Trace}[\mathbf{\Sigma}_r^{-1}\mathbf{\Sigma}_m] + \mathrm{Trace}[\mathbf{\Sigma}_m^{-1}\mathbf{\Sigma}_r]) \\ &+ \frac{1}{4}(\mu_r - \mu_m)^T(\mathbf{\Sigma}_r^{-1} + \mathbf{\Sigma}_m^{-1})(\mu_r - \mu_m) \end{aligned} \tag{4.8}$$

The derivation of this equation is provided in appendix A.

Unfortunately, a significant drawback of KL is that an exact analytical expression can be obtained only for the first level of the hierarchy, while distances for the next levels have to be approximated [52, 53, 80]. For that approximation simple combination rule can be used in which, the distances to a new cluster are calculated from the original distances weighted by the mixing proportions $P(k)$. Thus, the distance between the merged cluster $\ell\{r,m\}$ and the other cluster k is defined by the following recursive formula:

$$\mathcal{D}_{j+1}(k,\ell) = \frac{P_j(r)D_j(r,k) + P_j(m)D_j(m,k)}{P_j(r) + P_j(m)} \tag{4.9}$$

4.1.1.2 $\mathcal{L}_2$ similarity measure

The $\mathcal{L}_2$ distance was presented in [53] as a similarity measure in connection with agglomerative hierarchical clustering .

The $\mathcal{L}_2$ distance in case of density functions is defined as

$$\mathcal{D}_{j+1}(r,m) = \int (p_j(\mathbf{x}|r) - p_j(\mathbf{x}|m))^2 \, d\mathbf{x} \tag{4.10}$$

where r and m index two different clusters. Due to Minkowski's inequality (appendix A), $\mathcal{D}(r,m)$ is a distance measure.

Let define the set of cluster indices $\mathcal{I} = \{1, 2, \cdots, K\}$ and $\mathcal{I}_\alpha$ and $\mathcal{I}_\beta$ are disjoint subsets of $\mathcal{I}$ such that $\mathcal{I}_\alpha \cap \mathcal{I}_\beta = \emptyset$, $\mathcal{I}_\alpha \subset \mathcal{I}$ and $\mathcal{I}_\beta \subset \mathcal{I}$. $\mathcal{I}_\alpha$, $\mathcal{I}_\beta$ contain the indices of clusters, such that they include clusters r and m at level j, respectively. Then, the density of cluster r is given by:

$$p_j(\mathbf{x}|r) = \sum_{i \in \mathcal{I}_\alpha} \alpha_i p(\mathbf{x}|i), \;\; \alpha_i = \frac{P(i)}{\sum_{i \in \mathcal{I}_\alpha} P(i)} \tag{4.11}$$

for $i \in \mathcal{I}_\alpha$, and zero otherwise. The density of cluster m with β_i is defined in a similar way:

$$p_j(\mathbf{x}|m) = \sum_{i \in \mathcal{I}_\beta} \beta_i p(\mathbf{x}|i), \;\; \beta_i = \frac{P(i)}{\sum_{i \in \mathcal{I}_\beta} P(i)}. \tag{4.12}$$

The Gaussian integral is given by [91]:

$$\int p(\mathbf{x}|a)p(\mathbf{x}|b) \, dx = \mathcal{N}(\mu_a - \mu_b, \mathbf{\Sigma}_a + \mathbf{\Sigma}_b), \tag{4.13}$$

where $\mathcal{N}(\mu, \mathbf{\Sigma}) = (2\pi)^{-D/2} \cdot |\mathbf{\Sigma}|^{1/2} \cdot \exp(-\frac{1}{2}\mu^{\top}\mathbf{\Sigma}^{-1}\mu)$.

If we further define the K–dimensional vectors $\alpha = \{\alpha_i\}$, $\beta = \{\beta_i\}$ and the $K \times K$ symmetric matrix $\mathbf{G} = \{G_{ab}\}$, where $G_{ab} = \mathcal{N}(\mu_a - \mu_b, \mathbf{\Sigma}_a + \mathbf{\Sigma}_b)$, then the distance $\mathcal{D}$ can be rewritten in the simplified notation as:

$$\mathcal{D}(r, m) = (\alpha - \beta)^{\top}\mathbf{G}(\alpha - \beta). \tag{4.14}$$

It is also important to include the prior of the component in the distance measure. Thus, the modified $\mathcal{L}_2$ is then given by:

$$\widetilde{\mathcal{D}}_{j+1}(r, m) = \int (p_j(\mathbf{x}|r)P_j(r) - p_j(\mathbf{x}|m)P_j(m))^2 \, d\mathbf{x} \tag{4.15}$$

which easily can be expressed by a modified matrix $\widetilde{G}_{ab} = P(a)P(b)G_{ab}$.

4.1.1.3 Cluster Confusion similarity measure

Another possibility is to use as a similarity measure the confusion between the clusters [53]. When merging two clusters, the similarity E is defined as a probability of misassignment when drawing the samples from two clusters separately.

$$E(r, m) = \int_{\mathcal{R}_m} p(\mathbf{x}|r)P(r)d\mathbf{x} + \int_{\mathcal{R}_r} p(\mathbf{x}|m)P(m)d\mathbf{x}, \tag{4.16}$$

where $\mathcal{R}_m$ is a set that contains vectors $\mathbf{x}$ classified as belonging to cluster m i.e. $\mathcal{R}_m = \{\mathbf{x} : m = \underset{j}{argmax}\ p(j|\mathbf{x})\}$ and $\mathcal{R}_r = \{\mathbf{x} : r = \underset{j}{argmax}\ p(j|\mathbf{x})\}$. In general, $E(r, m)$ can not be computed analytically, but it can be approximated with arbitrarily accuracy by using an auxiliary set of data samples drawn from the estimated model. The cluster confusion similarity measure can be computed using the following algorithm:

1. Randomly select a cluster i with probability $P(i)$
2. Draw a sample $\mathbf{x}_n$ from $p(\mathbf{x}_n|i)$
3. Determine the estimated cluster $j = \underset{k}{argmax}\ \ p(k|\mathbf{x}_n)$
4. Estimate $E(r, m)$ as the fraction of samples where $(i = r \wedge j = m)$ or $(j = r \wedge i = m)$

Figure 4.1 The Cluster Confusion similarity measure algorithm.

4.1.1.4 Sample Dependent similarity measure

Instead of constructing a fixed hierarchy, a sample dependent hierarchy can be obtained by merging a number of clusters relevant for a new data sample $\mathbf{x}_n$. This similarity measure was proposed in [53].

Let $p(k|\mathbf{x}_n)$, $k = 1, \ldots, K$ be the computed posteriors that are ranked in descending order and the accumulated posterior is stored in $A(i) = \sum_{k=1}^{i} p(k|\mathbf{x}_n)$. The sample dependent cluster is then formed by merging the fundamental, for this sample, components $k = 1, 2, \cdots, M$, where $M = \underset{i}{argmin} \ \ A(i) > \rho$, with for example, $\rho = 0.9$.

4.1.2 Evaluation of similarity measures in hierarchical clustering

To compare the presented similarity measures a 6 cluster toy example was created. All clusters are drawn from normal distributions with spherical covariance structures. In four clusters (1, 2, 3, 4) the covariance is set to identity matrix $\mathbf{\Sigma} = I$, and for two clusters (5, 6), $\mathbf{\Sigma} = 0.1 \cdot I$. The scatter plot of the data is shown on figure 4.2.

The graphs visualizing hierarchical clustering are called *dendrograms*. Dendrograms for different similarity measures are shown on figure 4.3. All the discussed measures recognize the dissimilarity between the narrow and the wide variance clusters. As mentioned earlier, the hierarchy produced by different similarity measures vary.

For the selected test samples shown also on figure 4.2 ordered class posterior values are shown in table 4.1. Five test points were especially selected for good illustration of the Sample Dependent similarity measure. The sample coordinates are shown on top of the table 4.1 and refer to figure 4.2. The threshold value ρ is assumed to equal $\rho = 0.9$. For example, sample $\mathbf{x}_1$ gains full confidence after second hierarchy level, thus the fundamental clusters for that example ware clusters number 1 and 4. A similar situation is in the case of sample $\mathbf{x}_5$. Regarding point $\mathbf{x}_2$, the single cluster no. 6 provides enough of the confidence. Sample $\mathbf{x}_3$ builds two level hierarchy $\{1, 3\}$ and sample $\mathbf{x}_4$ three level $\{\{6, 3\}, 1\}$. In that way the hierarchy is independently created for each of the samples.

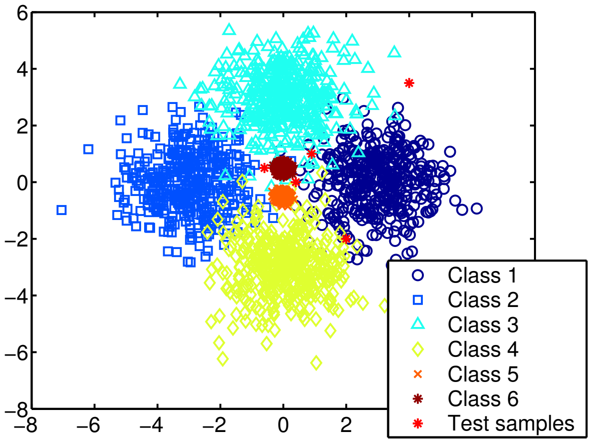


Figure 4.2 The scatter plot of the artificial data generated for comparison between the similarity measures presented in Chapter 4. The dendrogram for the presented data build with KL, $\mathcal{L}_2$, modified $\mathcal{L}_2$ and Cluster Confusion similarity measures are presented on figure 4.3. 6 test samples that are marked with red stars were selected for illustration of Sample Dependent similarity measure that is presented in table 4.1.

$\mathbf{x}_1 = (2, -2)$		$\mathbf{x}_2 = (-0.6, 0.5)$		$\mathbf{x}_3 = (4, 3.5)$		$\mathbf{x}_4 = (0.9, 1)$		$\mathbf{x}_5 = (0.4, 0)$	
$p(k\vert\mathbf{x}_i)$	k	$p(k\vert\mathbf{x}_i)$	k	$p(k\vert\mathbf{x}_i)$	k	$p(k\vert\mathbf{x}_i)$	k	$p(k\vert\mathbf{x}_i)$	k
0.5	1	0.967	6	0.82	1	0.388	6	0.495	5
0.5	4	0.015	2	0.18	3	0.351	3	0.495	6
0.0	2	0.011	3	0.00	4	0.259	1	0.006	1
0.0	3	0.007	5	0.00	2	0.001	2	0.002	3
0.0	5	0.000	4	0.00	5	0.001	4	0.002	4
0.0	6	0.000	1	0.00	6	0.000	5	0.000	2

Table 4.1 The Sample Dependent similarity measure. The test samples, which coordinates and are referring to figure 4.2. The ordered class posterior and the corresponding cluster numbers are shown in columns. The assumed threshold value is $\rho = 0.9$. For example, sample $\mathbf{x}_1$ gains full confidence already after first hierarchy level, thus the fundamental clusters for that example are a composition of clusters number 1 and 4. For point $\mathbf{x}_2$, the single cluster no. 6 provides enough of the confidence. In case of sample $\mathbf{x}_4$ 3 level hierarchy is proposed namely, $\{\{6, 3\}, 1\}$.

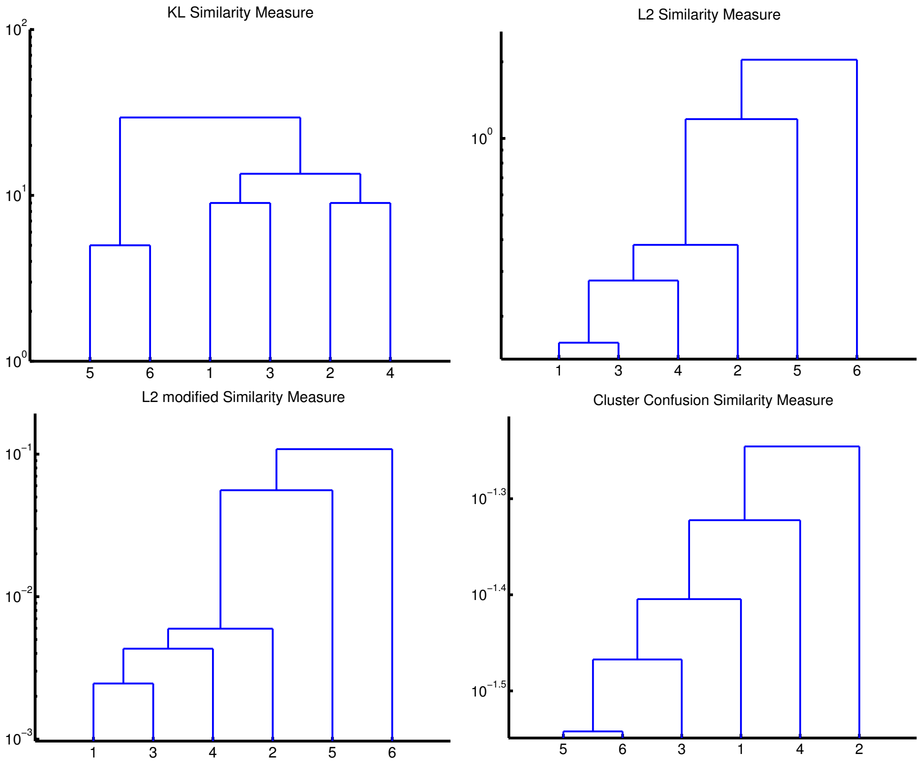


Figure 4.3 The presented dendrograms are corresponding to hierarchical clustering created with different similarity measures for a 6 cluster toy example shown on figure 4.2. The type of the measure is shown above each figure. By referring to figure 4.2 the differences, among the methods, in the ordering of clusters can be seen.

4.2 Confusion matrix

The important final step of the KDD process (figure 1.1) is an understandable presentation of the obtained results. In case of supervised learning *the confusion matrix* is often used as a cluster interpretation. The confusion matrix, by definition, contains information about actual and predicted classifications done by a classification system. An illustration of that idea is presented in table 4.2. $\mathbf{a}$ represents a vector generated by class A, and $\mathbf{b}$ by class B. The estimated corresponding clusters are marked by $\mathbb{A}$ and $\mathbb{B}$, respectively. $P(\mathbf{a}|\mathbb{A})$ is a probability of the assignment of the vector that was generated from class A to the

	Class A	Class B
Cluster $\mathbb{A}$	$P(\mathbf{a}\vert\mathbb{A})$	$P(\mathbf{b}\vert\mathbb{A})$
Cluster $\mathbb{B}$	$P(\mathbf{a}\vert\mathbb{B})$	$P(\mathbf{b}\vert\mathbb{B})$

Table 4.2 A simple illustration of the confusion matrix idea. Vector $\mathbf{a}$ is generated by class A, while vector $\mathbf{b}$ by class B. The estimated corresponding clusters are marked by $\mathbb{A}$ and $\mathbb{B}$, respectively. $P(\mathbf{a}|\mathbb{A})$ is the probability of the assignment of the vector that was generated from class A to the corresponding cluster $\mathbb{A}$. $P(\mathbf{a}|\mathbb{B})$ denotes probability of misassignment of the vector $\mathbf{a}$. When confusion matrix equals identity matrix ,there is no cluster confusion observed.

corresponding cluster $\mathbb{A}$. $P(\mathbf{a}|\mathbb{B})$ denotes probability of misassignment of the vector $\mathbf{a}$. When confusion matrix equals identity matrix, there is no cluster confusion observed.

The confusion matrix can be also useful in the interpretation when unsupervised learning is used but all the labels are available. In such case not a class but a cluster structure of the data is the matter of the interest. In such case, number of clusters is typically larger than number of class labels. Thus, the confusion matrix provides the “supervised” labeling for the clusters.

4.3 Keywords and prototypes

As another method of interpretation, *keywords* and *prototypes* are proposed. Here, no class labels are required. The term *Keywords* originate from textual data[1] but it can be easily extended to other data types. For example, as keyword a typical image can be understood or typical values of the observed features. As prototype the most representative cluster example, is considered, which is selected from the original documents, what in GGM model corresponds to the example with the highest density value.

The simplest approach for keywords generation is to use the location parameter (mean) as a representative for each cluster. A more general and accurate technique requires generating a set of the most probable vectors $\widehat{\mathbf{y}}_i$ from each of the clusters found by, e.g., Monte Carlo sampling of the probability density

[1] Detailed description of the text data type can be found in section 7.1.

function $p(\mathbf{x})$. Vectors $\widehat{\mathbf{y}}_i$ are called *typical features*. Since, in most of the cases the computations are not hold in the original space[2], the clusters are described in the latent space, where processing was also performed. Therefore, the typical features need to be back-projected to the original space, where an interpretation is possible, i.e. $\mathbf{y}_i = \mathbf{U} \cdot \widehat{\mathbf{y}}_i$, where $\mathbf{U}$ is a projection matrix. Then, keywords correspond to the most fundamental features of the back-projected vectors may be generated:

$$\mathrm{Keywords}_k = \mathrm{Feature}(\mathbf{y}_{di} > \rho) \tag{4.17}$$

Since the density values for particular clusters may vary greatly it is unpractical to use directly equation 4.17. Instead, the back-projected vectors are normalized so that maximum value is equal 1 or they sum to 1. Then, ρ is typically selected as a value close to 1.

In some cases it is more useful to represent clusters by one common example. Thus, the prototype of a given cluster can be found by selecting the example, that has the highest density value $p(\mathbf{x}_n|k)$. For example, for text data the prototypical documents can be found. It is also possible to generate keywords directly from such prototype.

The interpretation with keywords and prototypes is possible on each level of the hierarchy. The typical features are drawn from mixture distribution of the merged clusters and back-projected to the original term space, where the common keywords representation is found. The prototypes must be found, separately, for each of the component clusters.

4.4 Discussion

The hierarchical clustering supplies with the multilevel description of the data. In that way, the data points, which can not be unambiguously assigned to any of the clusters found by GGM model can gain the confidence on higher levels in the hierarchy. Several similarity measures may be used, which result with different structures. However, no method significantly outperformed the other , despite their drawbacks. Therefore, the results from all the similarity measures are presented later in the experimental Chapter 7. It should be noted that for-

[2] One step of the KDD process is projection of the data, which is carried out whenever the original data dimensionality is significantly large. The Gaussian Mixture models considered in Chapter 3 require small (from numerical reasons) dimensionality of the data.

mula for Kullback-Leibler similarity measure is the approximate and Cluster Confusion similarity measure is computationally expensive.

Interpretation of the clusters on all hierarchy levels is provided by keywords and prototypes. Additionally, if the data labels are available it is possible to compute the confusion matrix between the obtained and original labeling.

CHAPTER 5

Imputating missing values

The missing data imputation task can be both a subject of data cleaning process as well as the data mining itself, see figure 1.1. In this work the problem is addressed in connection to the particular data set that was collected during the survey and therefore the data records are partially missing. One of the tasks in connection with this data set was to create models that will allow for accurate prediction of the lacking values.

The sun-exposure data was used in connection with missing values imputation. The experiments can be found in [81]. Since originally, the diary records are categorical, both nominal and ordinal, coding technique is proposed that converts the data to binary vectors. *1-out-of-c* coding was used for this purpose. It represent c level categorical variable with a binary c bits vector. For example, if the variable takes one of the following three states $\{1, 2, 3\}$, ($c = 3$) then the states are coded as shown on the figure 5.1 Missing data appears in

various applications both in the statistical and in the real life problems. Data may be missing from various reasons. For example, subjects to studies, medical patients, often drop out from different reasons before the study is finished.

state	binary representation
1	100
2	010
3	001

Table 5.1 An example of 3 state categorical variable coded into binary representation using *1-out-of-c* coding.

The questionnaires are not filled out properly because they were forgotten, the questions were skipped accidentally or they were not understood or simply the subject did not know the answers. The error can occur in the data storage, the samples may get contaminated, etc. The cause may also be the weather conditions, which did not allow the samples to be collected or the investigation to be performed. Missing data can be detected in the original space for example by finding the outlier samples by use of methods described in Section 3.6.

A lot of research effort has been spend in this subject. One could refer, for example, to the books written by Little & Rubin [57] or by Schafer [77], some theory can as well be found in the articles by Ghahramani [30, 31] or Rubin [73].

In order to define a model for missing data, lets, in line with Rubin [72] decompose the data matrix $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ to the observed and the missing part as follows: $\mathbf{X} = [\mathbf{X}^o, \mathbf{X}^m]$. One could now introduce the matrix $\mathbf{R} = \{\mathbf{r}_n\}_{n=1}^N$ which is an indicator of the missingness:

$$\mathbf{R} = \begin{cases} 1, & x_{dn} \text{ observed} \\ 0, & x_{dn} \text{ missing} \end{cases}$$

The joint probability distribution of the data generation process and the missingness mechanism can be then decomposed:

$$p(\mathbf{R}, \mathbf{X}|\xi, \theta) = p(\mathbf{R}|\mathbf{X}, \xi)p(\mathbf{X}|\theta), \tag{5.1}$$

where θ and ξ are the parameters for the data generation process and missing data mechanism, respectively.

The mechanism for generation of missing data is divided into the following three categories [30, 72]:

MAR (data Missing At Random). The probability of generating the missing value may depend on observation but not on missing value itself, i.e. $p(\mathbf{R}|\mathbf{X}^o, \mathbf{X}^m, \xi) = p(\mathbf{R}|\mathbf{X}^o, \xi)$, see figure 5.1 for illustration.

MCAR (data Missing Completely At Random) is a special case of MAR. By that definition the missing data is generated independently from $\mathbf{X}^o$ and $\mathbf{X}^m$, i.e. $p(\mathbf{R}|\mathbf{X}^o, \mathbf{X}^m, \xi) = p(\mathbf{R}|\xi)$, see figure 5.2.

NMAR (data Not Missing at Random). The missing values generation mechanism is depending both on observed and missing part, i.e. $p(\mathbf{R}|\mathbf{X}^o, \mathbf{X}^m, \xi)$. Then the data is said to be censored.

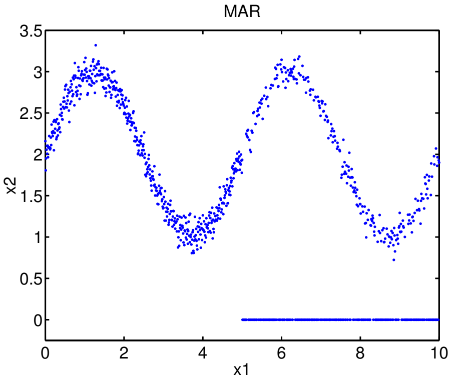


Figure 5.1 An example for data missing at random (MAR) generating mechanism.

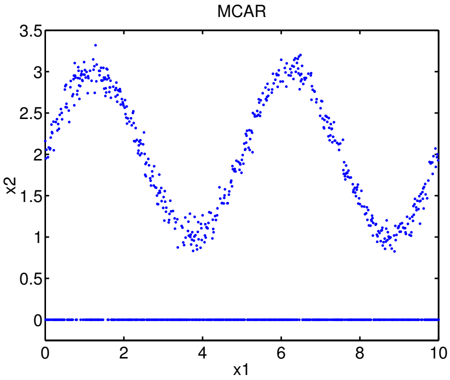


Figure 5.2 An example for data missing completely at random (MCAR) generating mechanism.

Majority of the research covers the cases, where missing data is of the MCAR or the MAR type. There is no general approaches for NMAR type.

Let us imagine, a following example: the sensor that is unreadable outside a certain range. In such case, learning the data distribution where the missing values were omitted will cause severe error (NMAR case). But if the same sensor fails only occasionally (MAR case), due to other reasons than measured temperature, the data, however harmed, often supplies enough information to create the imputation system and achieve a good estimate of the data distribution.

Most of the issues in statistical literature [10, 30, 57, 77] concerns two tasks, namely *the imputation*[1] of the missing values and *the estimation* of the model parameters. In the estimation, it is a common practice to use *the complete case analysis* (CCA) or *available-case analysis* (ACA). Case deletion is often used in case of both tasks in order to force the incomplete data to the complete data format. Omitting the patterns with unknown features (CCA) can be justified, whenever the large quantity of the data is available[2] and the variables are missing completely at random or just at random as shown in figures 5.1 and figure 5.2. Then the accuracy of the estimation of the model parameters will not suffer severely due to the deletion process. However, the estimate may be biased if the missingness process depends on some latent variable. The advantage of this technique is the possibility of using the standard methods for statistical analysis developed for fully observed data. The significant drawback is the loss of information that occurs in the deletion process. When there is not enough data samples, the recommended technique is to use all the available for the modeling data (ACA). It is well-founded since the missing data vectors also carry information which, may turn out crucial for the modeling. Thus, all the cases, where the variable of interest occurs are included in the estimation.

Once the parameters are estimated, the imputation can be performed. Naturally with the proper model selected, the missing values are re-inserted with a minimum error.

There has been suggested many different methods for completing the missing values (e.g., [57]). One way for the multivariate data is replacing the missing value with a predicted plausible value. The following list presents the most often used techniques in the literature:

Unconditional Mean Imputation replaces the missing value with the mean of all observed cases. It assumes MCAR generation mechanism and both the mean and the variance of the completed data is underestimated.

Cold deck uses values from a previous study to replace missing values. It assumes MCAR case. Variance is underestimated.

[1] The imputation is a general term for reconstruction of the missing data by plausible values.

[2] Only c.a. 5% of the missing data is an acceptable amount to delete from the data set.

Hot deck replaces the missing value with the value from a similar, fully observed case. It assumes MAR case and it gives better variance estimate than the mean and cold deck imputation.

Regression replaces the missing value with a value predicted from the regression of observed variables. If the regression is linear the MAR assumption is done. The variance however, is underestimated, but less than in the mean estimation.

Substitution replaces the missing study with another fully observed study not included earlier in the data set.

For example, one could use the conditional Mean Imputation. Here, both the mean and the variance are calculated based on the complete cases and the missing values are filled in by means of linear regression conditioned on the observed variables. An example of this technique as a method of estimating missing values in multivariate data can be found in [57].

There are also alternative methods that maintain the full sample size and result in unbiased estimates of parameters, namely *multiple imputation* [73, 71] and *maximum likelihood estimation based approaches* [30, 85, 86]. In the multiple imputation approach, as the name indicates unlike earlier mentioned single imputation techniques, the value of the missing variable is re-inserted several times. That technique returns together with the plausible value also the uncertainty over the missing variable. In the maximum likelihood based approach the missing values are treated as hidden variables and they are estimated via EM (algorithm 3.1).

Below, two models for estimating missing data are presented. The first method is a simple non-parametric K-Nearest Neighbor (KNN). In the second model the imputation is based on the assumption that the data vectors are coming from the Gaussian distribution. Both KNN and the Gaussian imputation analysis are based on the article [81] enclosed in appendix B. The experimental results are found in section 7.3.2.

5.1 K-Nearest Neighbor model

In order to determine nearest neighbors of the investigated data point, a distance measure must be proposed. In case of binary data *the Hamming distance measure* is used, which is given by the following formula:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d=1}^{D} |x_{di} - x_{dj}|, \tag{5.2}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two binary vectors, d is a bit index and D is a number of bits in the vector. Thus, for example, the distance between two vectors: $\mathbf{x}_1 = [100]$ and $\mathbf{x}_2 = [001]$ is calculated in the following way: $\mathcal{D}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{d=1}^{3} |x_{d1} - x_{d2}| = |1 - 0| + |0 - 0| + |0 - 1| = 2.$

When the data has real values, for example $\mathcal{L}_n$-norm can be applied as a distance measure. $\mathcal{L}_2$-norm between D-dimensional vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{d=1}^{D} (x_{di} - x_{dj})^2}. \tag{5.3}$$

Naturally, any other valid[3] distance measure may be used, that is suitable for the application, data type, or domain the calculations are hold. For example, with the discrete data often the linear cosine inner-product is used to measure vector dissimilarities or in the probability domain the KL divergence [69] may be applied.

As the optimum number of nearest neighbors K, for the investigated data set, the number is selected, for which the minimum imputation error is observed, in the experiment performed on fully observed data vectors.

The algorithm for the non-parametric K-Nearest Neighbor Model is presented on figure 5.3.

[3] The distance metric must satisfy the positivity, symmetry and triangle inequality conditions.

KNN Algorithm:

1. Divide the data set $\mathcal{D}$ into two parts. Let the first set contain data vectors in which at least one of the features is missing, $\mathcal{D}_m$. The remaining part where all the vectors are complete is called $\mathcal{D}_o$.

2. For each vector $\mathbf{x} \in \mathcal{D}_m$:

- Divide the vector into observed and missing parts as $\mathbf{x} = [\mathbf{x}^o, \mathbf{x}^m]$.
- Calculate the distance (Eq. 5.2 or Eq. 5.3) between the $\mathbf{x}^o$ and all the vectors from the set $\mathcal{D}_o$. Use only those features, which are observed in $\mathbf{x}$.
- Use the K closest vectors (K-nearest neighbors) and perform a majority voting (in the discrete case) or mean value (for real data) estimate of the missing values.

Figure 5.3 The algorithm for KNN model for imputation.

5.2 Gaussian model

Let now make an assumption that $\mathbf{x}$ comes form Gaussian distribution with mean μ and covariance $\mathbf{\Sigma}$. The feature vector $\mathbf{x}$, is divided into an observed and missing part: $\mathbf{x} = [\mathbf{x}^o, \mathbf{x}^m]$, as it was done in previous sections. Under the Gaussian model assumption, the optimal inference of the missing part is given as the expected value of the missing part given the observed part. That is given by the least-squares linear regression between $\mathbf{x}^m$ and $\mathbf{x}^o$ predicted by a Gaussian (for reference see [30]) i.e.,,

$$E(\mathbf{x}^m|\mathbf{x}^o) = \mu_m + \mathbf{\Sigma}_{mo}\mathbf{\Sigma}_{oo}^{-1} \cdot (\mathbf{x}^o - \mu_o) \tag{5.4}$$

where

$$\mu = [\mu_o, \mu_m] \quad \text{and} \quad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{oo} & \mathbf{\Sigma}_{om} \\ \mathbf{\Sigma}_{om}^\top & \mathbf{\Sigma}_{mm} \end{bmatrix} \tag{5.5}$$

The Gaussian imputation algorithm is presented in figure 5.4.

GM Algorithm:

1. Divide the data set $\mathcal{D}$ into two parts. Let the first set contain data vectors in which at least one of the features is missing, call it $\mathcal{D}_m$. Then the remaining part, where all the vectors are complete is called $\mathcal{D}_o$.

2. Estimate mean μ and the covariance matrix $\mathbf{\Sigma}$ from $\mathcal{D}_o$, i.e.

$$\widehat{\mu} = \frac{1}{N_o} \sum_{n \in \mathcal{D}_o} \mathbf{x}_n \tag{5.6}$$

$$\widehat{\mathbf{\Sigma}} = \frac{1}{N_o - 1} \sum_{n \in \mathcal{D}_o} (\mathbf{x}_n - \widehat{\mu})(\mathbf{x}_n - \widehat{\mu})^\top \tag{5.7}$$

where $N_o = |\mathcal{D}_o|$ is the number of complete vectors.

3. For each vector $\mathbf{x} \in \mathcal{D}_m$

- Divide the vector into two parts $\mathbf{x} = [\mathbf{x}^o, \mathbf{x}^m]$, where $\mathbf{x}^o$ is the observed vector features and $\mathbf{x}^m$ the missing.
- Estimate the missing part of the vector using Eq. 5.4

$$\widehat{\mathbf{x}}^m = E(\mathbf{x}^m|\mathbf{x}^o) = \mu_m + \mathbf{\Sigma}_{mo}\mathbf{\Sigma}_{oo}^{-1} \cdot (\mathbf{x}^o - \mu_o) \tag{5.8}$$

In case of binary vectors, sign of that estimate is used i.e., $\widehat{\mathbf{x}}^m = \operatorname{sign}[E(\mathbf{x}^m|\mathbf{x}^o)]$. For discrete numbers the estimate may be adequately quantized.

Figure 5.4 The algorithm for Gaussian model for imputation.

5.3 Discussion

The missing data problem occurs in many medical related databases. One of the data mining task concerns the imputation of such lost data. In this chapter two methods for imputation are presented. The experimental results and analysis of the performance for both methods are presented in section 7.3.2. It is shown in this section that the Gaussian imputation, for the applied data set, performs slightly better than KNN model.

CHAPTER 6

Probabilistic approach to Kernel Principal Component Analysis

Spectral clustering methods, one of which is presented in the following chapter, are the another techniques used in data mining task, see figure 1.1. In this case the presented in Chapter 2 feature dimensionality reduction methods are not necessary even when the data dimensionality is large. It can be, however, used prior to the performed clustering.

The kernel principal component analysis (KPCA) [78] in decomposition of a Gram matrix has been shown to be a particularly elegant method for extracting nonlinear features from multivariate data. It has been shown to be a discrete analogue of the Nyström approximation to obtaining the eigenfunctions of a process from a finite sample [88]. This relationship between KPCA and non-parametric orthogonal series density estimation was highlighted in [33], and the relation with spectral clustering has recently been investigated in [8]. The basis functions obtained from KPCA can be viewed as the finite sample estimates of the truncated orthogonal series [33]. However, a problem common to orthogonal series density estimation is that the strict non-negativity required from a probability density is not guaranteed when employing these finite order sequences to make point estimates [43]. This is also the case of the KPCA decomposition [33].

The following chapter considers the non-parametric estimation of a probability density from a finite sample [43] and relates this to the identification of class structure. This approach is presented in yet unpublished article [83].

6.1 Density Estimation and Decomposition of the Gram Matrix

Lets consider the estimation of an unknown probability density function $p(\mathbf{x})$ from a finite sample of N points $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \sim p(\mathbf{x})$ where the feature space of $\mathbf{x}$ is D-dimensional. Such sample can be employed to estimate the density in a non-parametric form by using, for example, a Parzen window estimator (refer to [43] for a review). Then the obtained density estimate is given by

$$p(\mathbf{x}) = N^{-1} \sum_{n=1}^{N} \mathcal{K}_h(\mathbf{x}, \mathbf{x}_n) \tag{6.1}$$

where $\mathcal{K}_h(\mathbf{x}_i, \mathbf{x}_j)$ denotes the window (or kernel function) of width h, between the points $\mathbf{x}_i$ and $\mathbf{x}_j$, which itself satisfies the requirements of a density function [43]. It is important to note that the pairwise kernel function values $\mathcal{K}_h(\mathbf{x}_i, \mathbf{x}_j)$ or Gram matrix[1] provides the necessary information regarding the sample estimate of the underlying probability density function $p(\mathbf{x})$.

For applications of unsupervised kernel methods such as KPCA the selection of the kernel parameter, in case of the Gaussian kernel h, is often problematic. However, since the kernel matrix can be viewed as the sample density estimate, methods such as leave-one-out cross-validation can be employed in obtaining an appropriate value of the kernel width parameter.

[1] $N \times N$ Gram matrix $\mathbf{G} = \{g_{ij}\}$ of the data matrix $\mathbf{X} = \{x_{dn}\}$ is defined as the matrix of the dot products $g_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ or equivalently: $\mathbf{G} = \mathbf{X}^T \mathbf{X}$.

6.1.1 Kernel Decomposition

The density estimate can be decomposed in the following probabilistic manner:

$$\begin{aligned} p(\mathbf{x}) &= \sum_{n=1}^{N} p(\mathbf{x}, \mathbf{x}_n) && (6.2)\\ &= \sum_{n=1}^{N} p(\mathbf{x}|\mathbf{x}_n) P(\mathbf{x}_n) && (6.3)\\ &= N^{-1} \sum_{n=1}^{N} p(\mathbf{x}|\mathbf{x}_n) && (6.4) \end{aligned}$$

It is assumed that each sample point is equally probable *a priori*, i.e. $P(\mathbf{x}_n) = N^{-1}$, which is an outcome of the *independent and identically distributed (i.i.d.)* assumption. The kernel operation can then be seen (compare equations 6.1 and 6.4) as the conditional density $p(\mathbf{x}|\mathbf{x}_n) = \mathcal{K}_h(\mathbf{x}, \mathbf{x}_n)$.

By Bayes' theorem [10], a discrete posterior probability can be defined for any point $\mathbf{x}$ given each of the N sample points

$$p(\mathbf{x}_n|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{x}_n)P(\mathbf{x}_n)}{\sum_{n'=1}^{N} p(\mathbf{x}|\mathbf{x}_{n'})P(\mathbf{x}_{n'})} = \frac{\mathcal{K}(\mathbf{x}, \mathbf{x}_n)}{\sum_{n'=1}^{N} \mathcal{K}(\mathbf{x}, \mathbf{x}_{n'})} \equiv \breve{\mathcal{K}}(\mathbf{x}, \mathbf{x}_n) \quad (6.5)$$

such that $p(\mathbf{x}_n|\mathbf{x}) \geq 0 \;\; \forall \; n$. It is easy to note that $\sum_{n=1}^{N} p(\mathbf{x}_n|\mathbf{x}) = 1$.

If there is the underlying class structure in the density then the sample posterior probability can be decomposed by introducing a discrete class variable

$$p(\mathbf{x}_n|\mathbf{x}) = \sum_{c=1}^{C} p(\mathbf{x}_n, c|\mathbf{x}) = \sum_{c=1}^{C} p(\mathbf{x}_n|c, \mathbf{x}) p(c|\mathbf{x}). \quad (6.6)$$

If the points have been drawn $i.i.d.$ from the respective C classes forming the distribution such that $\mathbf{x}_n \perp \mathbf{x} \mid c$, then

$$p(\mathbf{x}_n|\mathbf{x}) = \sum_{c=1}^{C} p(\mathbf{x}_n, c|\mathbf{x}) = \sum_{c=1}^{C} p(\mathbf{x}_n|c) p(c|\mathbf{x}), \quad (6.7)$$

where the stochastic constraints $\sum_{n=1}^{N} p(\mathbf{x}_n|c) = 1$ and $\sum_{c=1}^{C} p(c|\mathbf{x}) = 1$ are satisfied.

The decomposition of the posterior probabilities for each point in the available set $p(\mathbf{x}_i|\mathbf{x}_j) = \sum_{c=1}^{C} p(\mathbf{x}_i|c)p(c|\mathbf{x}_j)$, for all $i, j = 1, \cdots, N$ is identical to the aggregate Markov model originally proposed by Saul and Periera [76]. The matrix of posteriors (elements of the normalized kernel matrix) can be viewed as an estimated state transition matrix for a first order Markov process. This decomposition then provides class posterior probabilities $p(c|\mathbf{x}_n)$ which can be employed for clustering purposes.

A divergence based criterion such as cross-entropy

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \breve{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j) \log \left\{ \sum_{c=1}^{C} p(\mathbf{x}_i|c)p(c|\mathbf{x}_j) \right\} \tag{6.8}$$

or distance based criterion such as squared error

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \left\{ \breve{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j) - \left\{ \sum_{c=1}^{C} p(\mathbf{x}_i|c)p(c|\mathbf{x}_j) \right\} \right\}^2 \tag{6.9}$$

can be locally optimized by employing the standard non-negative matrix multiplicative update equations (NMF) [55] (also described in section 2.1.3) or equivalently the iterative algorithm which performs *Probabilistic Latent Semantic Analysis* (PLSA) [42]. If the normalized Gram matrix is defined as $G_{N\times N} = p(\mathbf{x}_i|\mathbf{x}_j) = \breve{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j)$ then the decomposition of that matrix with NMF or PLSA algorithms yields $G_{N\times N} = WH$ such that $W = p(\mathbf{x}_i|c)$ and $H = p(c|\mathbf{x}_j)$ are understood as the required probabilities which satisfy the previously defined stochastic constraints.

If a new point $\mathbf{z}$ is observed the estimated decomposition components can, in conjunction with the kernel, provide the required class posterior $p(c|\mathbf{z})$,

$$\begin{aligned} p(c|\mathbf{z}) &= \sum_{n=1}^{N} p(c|\mathbf{x}_n)\hat{p}(\mathbf{x}_n|\mathbf{z}) && (6.10)\\ &= \sum_{n=1}^{N} p(c|\mathbf{x}_n)\breve{\mathcal{K}}(\mathbf{z}, \mathbf{x}_n) && (6.11)\\ &= \sum_{n=1}^{N} p(c|\mathbf{x}_n)\frac{\mathcal{K}(\mathbf{z}, \mathbf{x}_n)}{\sum_{n'=1}^{N}\mathcal{K}(\mathbf{z}, \mathbf{x}_{n'})}. && (6.12) \end{aligned}$$

This can be viewed as a form of “kernel” based non-negative matrix factorization, where the ’basis’ functions $p(c|\mathbf{x}_n)$ define the class structure of the estimated density.

In attempting to identify the *model order* a generalization error based on the test sample predictive negative log-likelihood can be employed

$$\mathcal{G}_z = -N_z^{-1} \sum_{n=1}^{N_z} \log \left\{ p(\mathbf{z}_n) \right\}, \tag{6.13}$$

where N_z denotes the number of test points. In the presented model, the likelihood of the test sample $p(\mathbf{z})$ is derived from in the following manner:

$$\begin{aligned} p(\mathbf{z}) &= \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{z}|\mathbf{x}_n) & (6.14) \\ &= \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} p(\mathbf{z}|c) p(c|\mathbf{x}_n) & (6.15) \end{aligned}$$

The $p(\mathbf{z}|c)$ can be decomposed given the finite set such that

$$p(\mathbf{z}|c) = \sum\nolimits_{l=1}^{N} p(\mathbf{z}|\mathbf{x}_l) p(\mathbf{x}_l|c), \tag{6.16}$$

where $p(\mathbf{z}|\mathbf{x}_l) = \mathcal{K}(\mathbf{z}|\mathbf{x}_l)$. So the unconditional density estimate of an test point given the current kernel decomposition which assumes a specific class structure in the data can be computed as follows

$$p(\mathbf{z}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{N} \sum_{c=1}^{C} \mathcal{K}(\mathbf{z}|\mathbf{x}_l) p(\mathbf{x}_l|c) p(c|\mathbf{x}_n). \tag{6.17}$$

6.1.2 Examples of kernels

For continuous D-dimensional data a common choice of kernel, for both kernel PCA and density estimation, is the isotropic Gaussian kernel of the form

$$\mathcal{K}_h(\mathbf{x}, \mathbf{x}_n) = (2\pi)^{-\frac{D}{2}} h^{-D} exp \left\{ -\frac{1}{2h^2} ||\mathbf{x} - \mathbf{x}_n||^2 \right\} \tag{6.18}$$

Of course many other forms of kernel can be employed, though they may not themselves satisfy the requirements of being a density. If we consider for example the case of the linear cosine inner-product in case of discrete data such as vector space representations of text, the kernel is defined as

$$\mathcal{K}(\mathbf{x}, \mathbf{x}_n) = \frac{\mathbf{x}^T \mathbf{x}_n}{||\mathbf{x}|| \cdot ||\mathbf{x}_n||}. \tag{6.19}$$

The decomposition of such cosine based matrix directly yields the required probabilities.

This interpretation provides a means of spectral clustering which, in case of continuous data, is linked directly to non-parametric density estimation and extends easily to discrete data such as for example text. We should also note that the aggregate Markov perspective allows us to take the random walk viewpoint as elaborated in [59] and so a K-connected graph may be employed in defining the kernel similarity $\mathcal{K}_K(\mathbf{x}, \mathbf{x}_n)$. Similarly to the smoothing parameter and the number of clusters, the number of connected points in the graph can be also estimated from the generalization error.

6.2 Discussion

For the case of a Gaussian kernel this interpretation of a kernel based clustering enables estimating the kernel width parameter by means of test predictive likelihood and as such cross-validation can be employed. In addition, the problem of choosing the number of possible classes, a problem common to all non-parametric clustering methods such as spectral clustering [59, 62], can now be addressed. This overcomes the lack of an objective means of selecting the smoothing parameter in most other forms of spectral clustering models. The proposed method first defines a non-parametric density estimate, and then the inherent class structure is identified by the basis decomposition of the normalized kernel in the form of class conditional posterior probabilities $P(\mathbf{x}_n|c)$ and $P(c|\mathbf{x}_n)$. Since the projection coefficients are provided a new, previously unobserved point can be allocated in the structure. Thus, projection of the normalized kernel function of a new point onto the class-conditional basis functions yields the posterior probability of class membership for the new point. Something which cannot be achieved by partitioning based methods such as those found in [59] and [62].

A number of points arise from the presented exposition. Firstly, in case of continuous data, it can be noted that the quality of the clustering is directly related to the quality of the density estimate. Once a density has been estimated the proposed clustering method attempts to find modes in the density. Also if the density is poorly estimated perhaps due to a window smoothing parameter which is too large then class structure may be over-smoothed and so modes may be lost. In other words essential class structure may not be identified by the clustering. The same argument applies to a smoothing parameter which is too small thus causing non-existent structure to be discovered and to the connectedness of the underlying graph connecting the points under consideration.

CHAPTER 7

Segmentation of textual and medical databases

In the previous chapters some of the results were presented, and there the discussed techniques were compared on the simple data sets. Carefully selected auxiliary data sets were used in order to obtain good illustration. It allowed in some of the cases to decide which technique to apply in further investigations. This chapter focuses on implementation of the earlier described models on observational data.

7.1 Data description

In this section the applied data sets are presented together with short description of the accompanying them preprocessing procedure. The detailed report of the preprocessing steps is given in sections 7.2.1 and 7.3.1. Four data sets were used in the further investigations.

Email data: is a collection[1] of private emails categorized into three groups namely: *conference*, *job*, and *spam*. The documents are hand-labeled. Since emails were collected by an university employee the categories are university related. The collection was preprocessed, details of the procedure are presented in section 7.2.1, so the final data matrix consists of 1405 documents each described by 7798 terms. The clustering of the data was performed in latent space found in Latent Semantic Indexing framework [20]. This data set is used due to its simplicity and fairly good separation.

Newsgroups: is collection of 4 selected newsgroups[2]. The data is used in many publications, starting from the data collector Ken Lang published in [51] and also for example in [6, 9, 68]. The original collection consists of 20 different newsgroup categories each containing around 1000 records. In the experiments only four newsgroups were selected, namely *computer graphics, motorcycles, baseball* and *Christian religion* each of 200 records. The preprocessing steps are identical to those performed on the Email collection which are described in detail in section 7.2.1. In preprocessing 2 documents were removed[3] resulting with the final number of 798 records and 1367 terms. Also in case of this data set labels are available.

Sun-exposure study: The data was collected by Department of Dermatology, Bispebjerg Hospital University of Copenhagen, Denmark. It concerns a cancer risk study. The data set used in the experiments represents one year study or in fact 138 days, collected during spring, summer, and autumn period. The survey was performed on a group of 196 volunteers resulting in a total number of 24212 collected records. The experiments concern only the one year fraction of diary database. However, full study was performed throughout 3 years survey. This extended data set consist of diary records, detailed UV measurements, questionnaire of the past sun habits and the measurements of the skin type. This data is available for future study.

For the survey purpose, a special device was constructed for measuring sun exposure[4] of the subjects. The picture of the device is shown on figure 7.1. Additionally, subjects were asked to fill out daily diary records

[1] The Email database can be obtained at following location: http://isp.imm.dtu.dk/staff/anna

[2] The full data collection consisting of 20 categories is available at e.g., http://kdd.ics.uci.edu

[3] See section 7.2.1 for details.

[4] UVA and UVB radiation dose was measured every 10 minutes. In the performed experi-

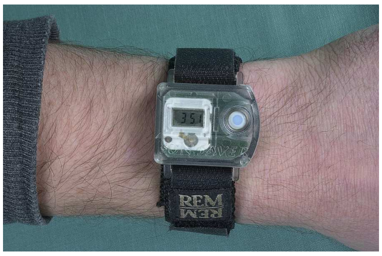


Figure 7.1 The devise measuring sun radiation the skin is exposed on.

about their sun behavior by answering 10 questions which are listed in table 7.1. Since it is a common knowledge that high sun exposure leads to the increase of cancer risk, it was expected that a link between these two events can be established.

No.	Question	Answer
1.	Using measuring device	yes/no
2.	Holiday	yes/no
3.	Abroad	yes/no
4.	Sun Bathing	yes/yes-solarium/no
5.	Naked Shoulders	yes/no
6.	On the Beach/Water	yes/no
7.	Using Sun Screen	yes/no
8.	Sun Screen Factor Number	yes-number (26 values)/no
9.	Sunburned	no/red/hurts/blisters
10.	Size of Sunburn Area	no/little/medium/large

Table 7.1 Questions concerning the daily sun habits in the sun-exposure study.

In the questionnaire, question number seven contains redundant information, for the investigation, since it is already included in question eight, and therefore it was removed from the data set. Also question number one, which is in fact an indicator of missingness in the data, was not included in the cluster structure investigation. In result, the records are described by 8 questions – 8 dimensional, categorical vectors. As expected, in the survey, a lot of data is missing. Therefore, the techniques are investigated to imputate missing values in these records. For other experiments, concerning data segmentation, the missing records were re-

ments only daily dose was available.

moved. In the diary collection there are 1073 incomplete records, 4171 are missing in UVA/UVB measurements and when combining these two data sets 5041 records have missing values. Therefore, only 19171 complete records from 187 subjects were used in the investigation.

Dermatological collection: is the collection[5] of erythemato-squamous diseases. Six classes (the diagnosis of erythemato-squamous diseases) are observed: *psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, pityriasis rubra pilaris*. This database contains 358 instances described by 34 attributes, 33 of which are nominal (values 0, 1, 2, 3) and one of them (age) is ordinal. Original data set contains few missing values that were removed for this investigation. The data set was previously used in [35]. The set is segmented by the aggregated Markov model, that is described in Chapter 6.

7.2 Segmentation of textual databases

7.2.1 Preprocessing

In order to obtain the vector space representation of the the textual data [75] the documents are transformed into word-frequencies. Then, in order to reduce, often large, dimension of the feature space certain preprocessing steps are performed that are described below. Such processing the textual databases is common in literature and can as well be found for example, in [39, 50, 52].

Each of the text documents is represented by the unique *histogram* over the *word collection*. The word collection is the list of all the words that occur in any of the observed documents. For each document, the number of occurrences of each word is recorded and that creates a unique fingerprint (histogram, feature vector) of that document. The relationships among the words are neglected. Typically, the feature space has extremely high dimension. One could easily imagine vectors of the size of a full dictionary, what is several thousands of terms. In every language the sentences are build from verbs, substantives, adjectives, adverbs and also from the conjunctions, pronouns, prepositions etc. The last, do not carry any crucial information, for clustering, but they occur in the sentences significantly more often than the other words. Therefore, it is an

[5] Available at http://ftp.ics.uci.edu/pub/machine-learning-databases

important preprocessing step to remove them. In the information technology area they are called *stopwords*. A list of 585 stopwords has been used in the experiments. Also words with very high[6] or very low[7] frequency of occurrence are erased. In this work the terms that occur less than 2 times and the documents containing less than 2 words are removed from the data set. In order to reduce dimensionality and compress the information a stem merging algorithm is applied. In this case it means that the frequencies of the words that have the same stem but different suffixes are merged together[8]. For example, if words like **working**, **worked**, **works** occur in the collection, they are all represented by single stem word **work** and the frequencies of the sub-terms are accumulated in the main term. That technique require lookup in the dictionary and therefore it is time expensive. Summarizing, each text document is converted and represented by a histogram over the list of selected words. The collection of such histograms is referred to as *term-document matrix* (TD).

Since the documents have various lengths the term-document matrix is normalized. Basically, any normalization method can be applied here, however, in the experiments the normalization to the unit sphere was used (unit $\mathcal{L}_2$-norm length), i.e. $\widehat{\mathbf{x}}_i = \frac{\mathbf{x}_i}{||\mathbf{x}_i||_2}$, where $\mathbf{x}$ is not normalized term-document matrix.

For the cross-validation purposes data was randomly divided to the training and the test set. In the Email data, training set contains 702 vectors leaving for test 703 examples. For newsgroups the sets had the same size of 399 samples each.

From both sets, the mean vector calculated from the training set $\mu = N^{-1}\sum_n \mathbf{x}_n$ was subtracted form the data vectors.

In the following sections, steps of the KDD process, presented in Chapter 1, are applied on the observational data sets. The data is preprocessed and then projected to low-dimensional latent space using the Principal Component Analysis. That framework was introduced in [20]). In that space, the density is modeled with Generalizable Gaussian Mixture model, from which the cluster structure is obtained. The interpretation of clusters is provided, in the form of keywords. Hierarchical clustering is also applied enabling multiple level classification and interpretation. Some of the results can also be found in the following contribu-

[6] Those words are not unique for any of the documents.

[7] It is generally not recommended to force small clusters.

[8] Also another technique is very popular in literature, so called *suffix stripping*, where the recognized word endings are deleted and the remaining identical stems are merged, e.g. Porter Stemming Algorithm [67].

tions [52, 53, 80].

7.2.2 Projection to the latent space

All the textual data sets are high dimensional. The email collection after preprocessing is described by 7798 terms and the newsgroups by 1217. In each case the dimension is too large to be able to use effectively any of the GGM models[9] discussed in chapter 3. Therefore, the representation of the data in the reduced space must be obtained. Selected projection methods are described in detail in chapter 2. In the experiments Principal Component Analysis is used. This general technique in context of the text data was originally proposed by Deerwester in [20] and named *Latent Semantic Indexing (LSI)*, where as the projection vectors the eigenvectors are used that are obtained by singular value decomposition. That defines a new latent space where the semantic similarities of the documents can be discovered.

On figure 7.2 the latent space for the investigated collections is shown. In each case (for visualization purpose), 2 suitable principal components are selected. Based on the presented scatter plots of both of the data sets, it is easy to see in case of both sets the existing structure in the data.

In order to determine the optimal number of principal components for each collection the eigenvalue curves are investigated[10] as are shown on figure 7.3. For Email collection 5 principal components were selected and for Newsgroups 4 components are used. In both cases, this decision is arguable, since no dominant group of components is observed.

7.2.3 Clustering and clusters interpretation

[9] In high dimensions the number of parameters to estimate is also high so large number of data points is needed. Additionally, the covariance matrix is also large and it needs to be inverted many times in the learning process, which is time consuming. This drawbacks of the GGM models are explained in section 3.3.

[10] The attempt of calculating the generalization error (Eq. 2.12) was unsuccessful, since the error curve was monotonically decreasing. The reason may be due to the shape of the eigenvalue curve (see figure 7.3), where for high principal components large values, in comparison to the first eigenvalues, are still observed.

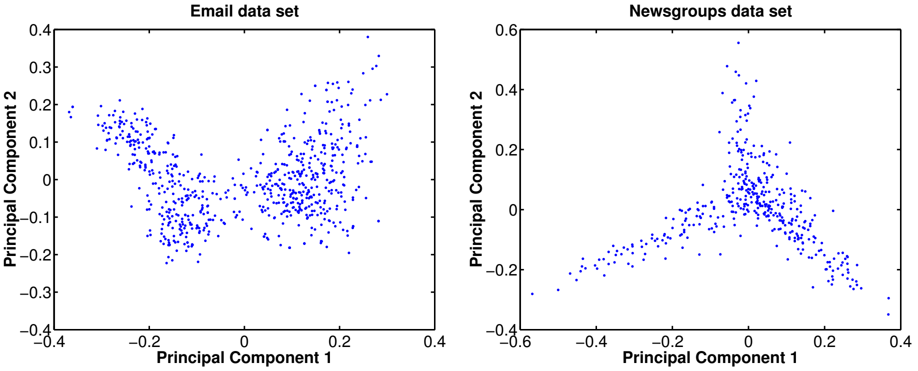


Figure 7.2 Scatter plots of the training data sets in the latent space for the investigated text collections, Email and Newsgroups data sets, respectively. For good visualization 2 principal components are carefully selected. It is easy to see in both cases that there exists a cluster structure in the data.

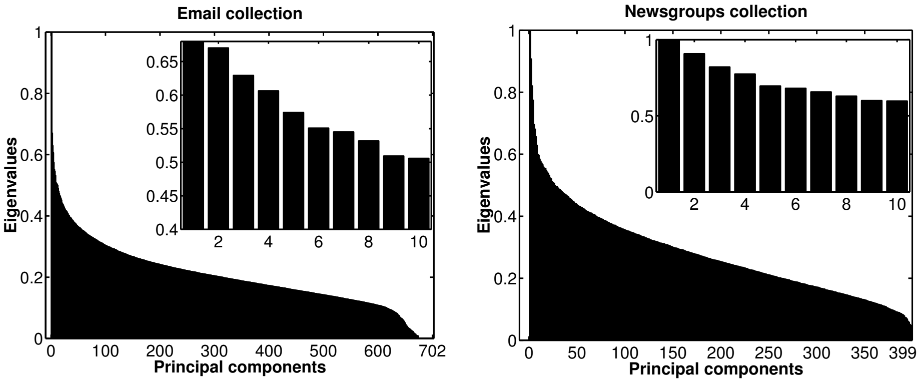


Figure 7.3 Eigenvalue curves for the investigated textual collections, Email and Newsgroups data sets. Additionally, the largest eigenvalues are shown in close-up. Based on these plots the decision is made about the number of principal components. In case of Email collection 5 largest principal components are used and for Newsgroups 4 are selected.

7.2.3.1 Email collection

In the performed experiment for the Email collection the latent space of 5 components was used. At first, the collection was scanned against outliers. For detection, the UGGM model with an outlier cluster was applied, for details see section 3.6.2. No outlier samples were found in the training set. Thus, the full collection of 702 samples was used in segmentation process.

Clustering

For clustering the unsupervised Generalizable Gaussian Mixture model hard assignment (0/1 decision function) was applied, details of which can be found in section 3.3. In the particular result, presented later in this section, seven clusters[11] were obtained. The cluster assignment for selected components is shown on figure 7.4.

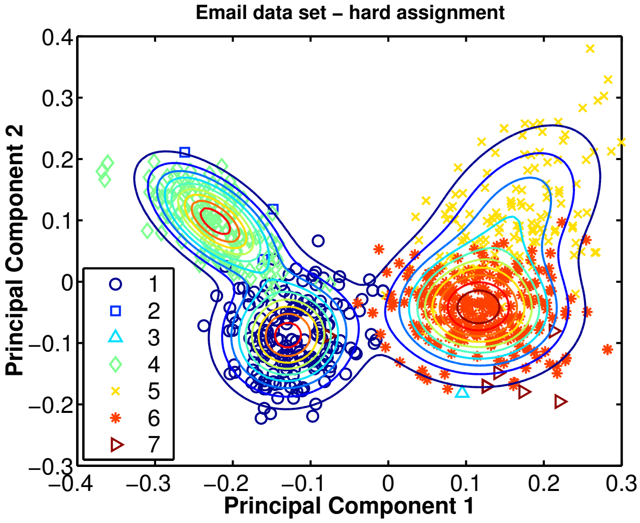


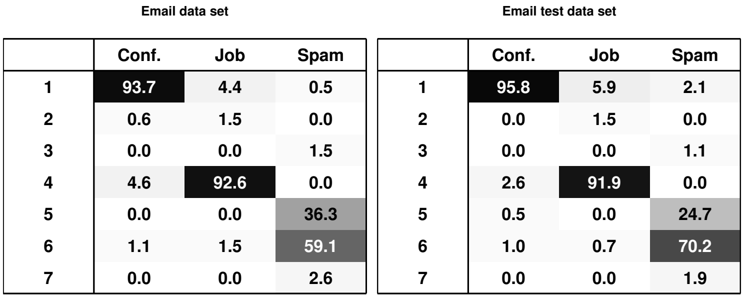
Figure 7.4 Scatter plot of the Email data with indicated data points assignment to the clusters and shown density contours. The dependency between principal components 1 and 2 are shown. The density was estimated by the UGGM model with hard cluster assignment. 7 clusters, indicated in the legend bar, are observed.

For illustration, the same 2 principal components are displayed as were shown on figure 7.2.

Since labels are available, the confusion matrix, described in section 4.2, was

[11] Number of clusters was decided based on the training error with AIC penalty [1]. Typically, the number of clusters in this model, for Email collection varies in the range between 6 and 11.

also calculated and presented on figure 7.5. It can be used in further analysis as an supervised indication of cluster contents. Based on figure 7.5 it may



Email data set

	Conf.	Job	Spam
1	93.7	4.4	0.5
2	0.6	1.5	0.0
3	0.0	0.0	1.5
4	4.6	92.6	0.0
5	0.0	0.0	36.3
6	1.1	1.5	59.1
7	0.0	0.0	2.6

Email test data set

	Conf.	Job	Spam
1	95.8	5.9	2.1
2	0.0	1.5	0.0
3	0.0	0.0	1.1
4	2.6	91.9	0.0
5	0.5	0.0	24.7
6	1.0	0.7	70.2
7	0.0	0.0	1.9

Figure 7.5 Confusion matrix for Email data set. Left table shows the confusion in the training set and right, in the test set. From the presented figures, it may be concluded that most of the *conference* emails are accumulated in cluster number 1. Cluster 4 takes majority of *job* emails and clusters 5 and 6 of *spam*. The rest of the clusters (2,3 and 7) contain only small fractions of data points.

be concluded that most of the *conference* emails are accumulated in cluster number 1 (94% of the training data points). Cluster 4 takes majority of *job* emails (93%) and clusters 5 and 6 of *spam* (36% and 59%, respectively). The rest of the clusters contain only small fractions of data points.

A soft version of the GGM model was also applied for comparison. As expected, the outcome is much more complex (model uses 21 clusters to describe the density $p(\mathbf{x})$). The scatter plot of the data with the surface plot of the density function is presented on figure 7.6. Density structure between the soft and the hard assignment does not differ significantly (compare figures 7.4 and 7.6). In the soft assignment algorithm the density is described by 21 components while, in case of the hard assignment only 7 are needed. In the hard GGM algorithm, in the optimization process, the clusters that do not have members assigned, are re-initialized or deleted, while in soft version they are preserved, even when the mixing proportions $P(k)$ are smaller than N^{-1}, what corresponds to one member. In that way, more complex models are possible, leading to a more accurate density estimate, but at the same time allowing clusters without members among the training set. Therefore, for clustering, hard assignment is usually more useful.

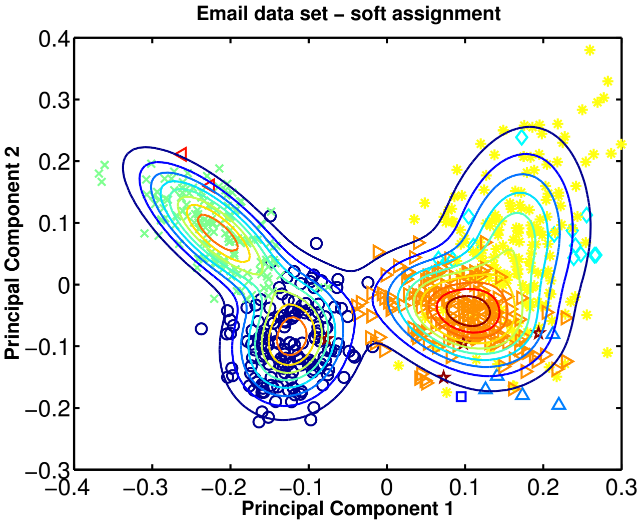


Figure 7.6 Scatter plot of the Email data with indicated cluster structure and data probability density contours. The dependency between principal components 1 and 2 are shown. The density is found by the UGGM model with soft cluster assignment. 21 clusters is observed. The observed density structure does not differ significantly for the hard assignment outcome (see figure 7.4).

Hierarchical clustering

All similarity measures are applied here that are presented in section 4.1. At the first hierarchy level ($j = 1$) 7 clusters are used, obtained from hard UGGM model. The dendrograms for KL, Cluster Confusion, $\mathcal{L}_2$ and modified $\mathcal{L}_2$ similarity measures are presented on figure 7.7. For the Sample Dependent similarity measure the frequencies with which certain cluster combinations occur in the test set, are shown with the bar plot. The most often used combinations are labeled over the corresponding bars. Dendrogram structures describe the consecutive clusters that are merged in the process. Even though, the structures vary significantly, it is difficult to select a superior method, since the quality of hierarchical clustering is a subjective issue. Note, however, that even though KL often provides interesting results it is an approximate measure and the Cluster Confusion similarity measure require density sampling and therefore is computationally expensive.

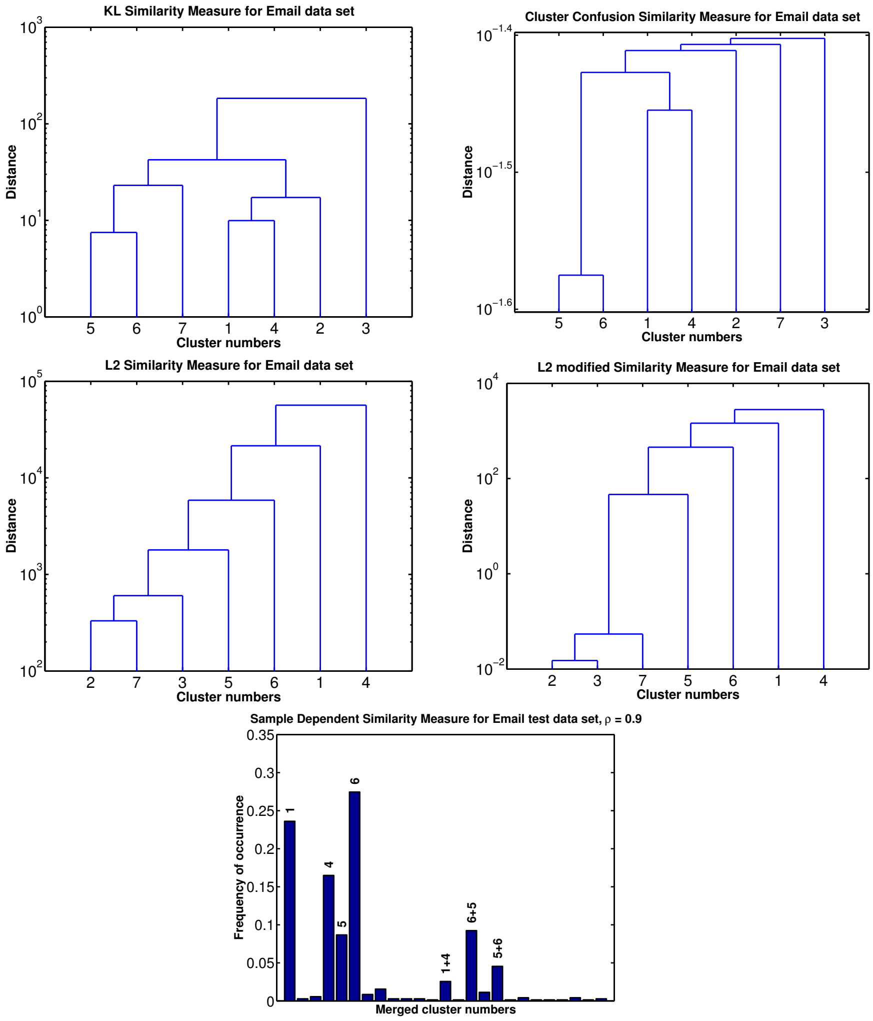


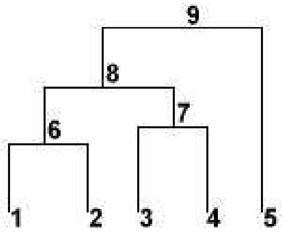
Figure 7.7 The dendrograms for Email data set. UGGM with hard cluster assignment resulted with 7 clusters that are used at the first level $j = 1$. Based on that outcome the presented hierarchies are build with five similarity measures are applied which are described in section 4.1. In case of sample depended similarity measure the most often combinations are presented above the frequency bars. The tree structures describe the consecutive clusters that are merged in the process. The closest clusters include: clusters 5 and 6 (*spam* emails) merged firstly by all the measures, and 1 and 4 which are found to be closely related in semantic domain (both are university related.)

Once the hierarchy is build it is possible to determine the cluster assignment of new points. For the majority of the test samples, the first hierarchy level gives enough confidence in assignment to the cluster. Approximately 76% of the data points can be uniquely assigned to one of the seven basic clusters (majority is assigned to clusters 1, 4, 5 and 6) with a confidence larger than 90% ($\rho = 0.9$). Those values are shown on figure 7.7 for Sample Depended similarity measure, where the cluster posterior $p(k|\mathbf{x})$ is is observed for the first hierarchy level (clusters from 1 to 7). The rest of the samples (24%) needs at least 2 clusters to provide sufficient description. A big fraction of these data points fall into the composition of *spam* emails (clusters 5 and 6). Such union is suggested by KL and Cluster Confusion and Sample depended similarity measures. $\mathcal{L}_2$ and modified $\mathcal{L}_2$ first combines all the minor importance clusters and then adds to this mixture two *spam* clusters. Another likely formation provides mixture of *job* and *conference* emails (cluster 1 and 4).

Another method for interpretation is to obtain the cluster representation as for example, to generate keywords for each cluster or cluster union. Such a technique is fully unsupervised but it requires understanding of the context through a provided set of related words.

Keywords assignment

For the Email collection, clustered with hard UGGM and with hierarchies shown on figure 7.7, keywords are generated and presented in this section. First, let us introduce the numbering schema of clusters in the dendrogram structure. The clusters are marked with consecutive numbers. In case of the presented Email example there are 7 clusters (numbered: $1, 2, \ldots, 7$) at the first hierarchy level $j = 1$. At each next level new number is assigned to the new composition. Thus, at level $j = 2$, the two closest clusters create cluster number 8, at the level $j = 3$ the next 2 clusters result with composition number 9, etc.



The idea is illustrated on the figure to the right. In this simple example 5 clusters are observed at the basic level $j = 1$. At second level in the hierarchy $j = 2$ clusters 1 and 2 are merged, resulting with the composition number 6. On the next level, clusters 2 and 3 create cluster 7. Next, 2 new clusters are merged together (6 and 7) and to the new composition, number 8 is assigned. Finally, cluster 9 is the composition of 8 and 5, collecting also all the basic level clusters: 1, 2, 3, 4 and 5. Totally, with K clusters on the basic level $j = 1$ there are $K - 1$ hierarchy levels, thus, total number of clusters in the hierarchy equals $2K - 1$.

In order to assign keywords, the set of typical features is generated. Thus, based on the outcome of the UGGM algorithm, i.e. cluster means, covariances and mixing proportions, a new set of 5000 points are randomly drawn form the modeled data distribution. From this sample as typical features the vectors are selected for which density value is in top 20%. By lowering this threshold, in case of multi-modal densities, it is possible to include to keywords representation, also contributions from weaker clusters, i.e. those with wider variance. The typical features were back-projected to the original space are the data mean is added. Reconstructed in that way histograms are no longer positive due to the restricted number of principal components used in projection. Thus, the negative values are neglected. As keywords, words, form the histograms assigned to the clusters or cluster unions, are accepted with frequency higher than, e.g. 10% of the total weight, i.e. the word is accepted if $w_{di}/\sum_{di} w_{di} > 0.9$, where w_{di} is the frequency of a d'th word from the i'th typical feature vector (reconstructed histogram).

Keywords for the first level in the hierarchy with corresponding mixing proportions $P(k)$ are presented in table 7.2.

Cluster number 1 inherit clearly the *conference* keywords like: *information, university, conference, call, workshop*, etc. Cluster number 2 and number 4 are university *job* related, thus, *position, university, science, work*, are appearing here. Clusters 3, 5, 6 and 7 are easily recognized with often used words in *spam* emails thus, such terms like: *list, address, call, remove, profit* and *free* are observed. One can refer now to the confusion matrix presented on figure 7.5 and find similarities in the cluster interpretation.

For each dendrogram shown on figure 7.7 the keywords corresponding to higher hierarchy levels are given in tables 7.3, 7.4, 7.5, and 7.6. When merging the clusters together, the density of the union is expressed by the following equation: $p(\mathbf{x}) = \sum_{k \in \mathcal{I}_\alpha} p(\mathbf{x}|k)P(k)$, where $\mathcal{I}_\alpha$ collects the indexes of the merged clusters. If one cluster in the union is dominant (has narrow variance) main keywords will most likely come from this cluster, since majority of typical features will be sampled from its center (higher probability region). The more comparable the densities are, the more likely is that the union keywords are a mixture from both component clusters. For example, in table 7.3 the keywords corre-

Cluster	*P(k)*	Keywords
1	.245	information, conference, university, paper, call, neural, research, application, fax, contact, science, topic, workshop, system, computer, invite, internation, network, submission, interest
2	.004	research, university, site, information, science, web, application, position, computer, computation, work, brain, candidate, neural, analysis, interest, year, network
3	.009	succeed, secret, profit, open, recession, million, careful, produce, trump, success, address, letter, small, administration, question
4	.190	university, research, science, position, computation, brain, application, candidate, year, computer, send, department, information, analysis, neuroscience, interest, cognitive
5	.202	free, site, web, remove, click, information, visit, subject, service, adult, internet, list, sit, offer, business, line, time
6	.335	call, remove, free, address, card, list, order
7	.014	call, remove, free, address, list, order, card, day, send, service, information, offer, business, money, mail, succeed, make, company, time, line

Table 7.2 Keywords for 7 clusters obtained from UGGM model. Keywords provide the interpretation of the clusters. Cluster number 1 inherit clearly the *conference* keywords like: *information, university, conference, call, workshop*, etc. Cluster number 2 and number 4 are university *job* related, thus, *position, university, science, work*, are appearing here. Clusters 3, 5, 6 and 7 are easily recognized with often used words in *spam* emails thus, such terms like: *list, address, call, remove, profit* and *free* are observed.

sponding to the KL similarity measure are presented. There, cluster number 8 is a composition of clusters 5 and 6 and as such inherit the keywords from both clusters, which densities are comparable. Cluster number 9, however, which is composed from clusters 1 and 4 is represented by *job* keywords, so keywords are coming only from cluster number 4, which is dominant. It is possible to include keywords form other clusters by selecting larger number of typical features and by that sampling the larger area of the probability space.

Cluster	Keywords
8	free, call, remove, information, site, subject, list, web, business, offer, message, rep, time, mail, visit, line, day, address, send, year
9	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
10	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
11	free, call, remove, information, site, subject, list, web, business, offer, message, rep, time, mail, visit, line, day, address, send, year
12	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
13	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network

Table 7.3 Keywords for hierarchy build with KL similarity measure. Cluster number 4 containing *job* emails has the narrower variance and therefore higher density values. When selecting typical features, most or all of the points are generated by this particular cluster. Therefore, its keywords are dominating keywords from other clusters in hierarchy.

Similar interpretation obtain the hierarchy based on Cluster confusion similarity measure, presented on figure 7.4. Also here cluster number 4 is merged at the beginning, thus it dominates in the selected typical features.

In case of $\mathcal{L}_2$ and modified $\mathcal{L}_2$ similarity measures, shown in tables 7.5 and 7.6, the clusters are merged successively, first the minor clusters, with few members, and then the clusters containing the *spam* emails and at last the clusters with *conference* and *job* emails. That structure, taking into consideration the known labeling, seems to give the best results for Email collection, both in hierarchy and the corresponding keywords. The major difference, between $\mathcal{L}_2$ and modified $\mathcal{L}_2$ similarity measures, is the increased distance from small clusters to the rest arising form included mixing proportion values in the distance measure.

Cluster	Keywords
8	free, call, remove, information, site, subject, list, web, business, offer, message, rep, time, mail, visit, line, day, address, send, year
9	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
10	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
11	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
12	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network
13	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network

Table 7.4 Keywords for hierarchy build with Cluster Confusion similarity measure. Similar to KL measure, here cluster number 4 is merged on the beginning of the hierarchy and since its density values are dominant it control the keywords generation process i.e., all the higher hierarchy level keywords are *job* related.

Cluster	Keywords
8	call, remove, free, list, message, information, rep
9	call, remove, free, list, message, information, rep
10	call, succeed, address, free, secret, information, day, card, site, make, money, web, offer, number, profit, time, order, business, remove, company
11	free, call, remove, information, site, subject, list, web, business, offer, message, rep, time, mail, visit, line, day, address, send, year
12	call, information, university, conference, neural, fax, application, research, address, model, science, internation, computer, includ, paper, workshop, program, network, work, phone
13	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network

Table 7.5 Keywords for hierarchy build with $\mathcal{L}_2$ similarity measure. The clusters are merged successively first the minor clusters, with few members (clusters 8 and 9) and then the clusters containing the *spam* emails (clusters 10 and 11) and at last the clusters with *conference* (12) and *job* emails (13).

Cluster	Keywords
8	free, work, year, position, card, program, research, interest, business, address, offer, computation, send, time, start, candidate, order, service, application, subject
9	call, remove, free, list, message, information, rep
10	call, succeed, address, free, secret, information, day, card, site, make, money, web, offer, number, profit, time, order, business, remove, company
11	free, call, remove, information, site, subject, list, web, business, offer, message, rep, time, mail, visit, line, day, address, send, year
12	call, information, university, conference, neural, fax, application, research, address, model, science, internation, computer, includ, paper, workshop, program, network, work, phone
13	research, university, computation, science, application, succeed, position, interest, information, neural, work, cognitive, open, brain, year, neuroscience, model, fax, secret, network

Table 7.6 Keywords for hierarchy build with modified $\mathcal{L}_2$ similarity measure. The clusters are merged successively first the minor clusters, with few members and then the clusters containing the *spam* emails and at last the clusters with *conference* and *job* emails. The major difference, between this similarity measure and $\mathcal{L}_2$, is the increased distance from small clusters to the rest what is due to including the mixing proportion value in the distance measure. With respect to known labeling modified $\mathcal{L}_2$ similarity measure provides the best results.

7.2.3.2 Newsgroups collection

In the experiments, the term-document matrix of the Newsgroups collection is projected on the latent space defined by 4 left eigenvectors associated with largest eigenvalues obtained from singular value decomposition of that matrix. The collection is first screened for outliers. As before, the UGGM algorithm with outlier cluster was applied (for details refer to section 3.6.2). Since no outliers are detected in the training set, the full set of 399 samples is used in segmentation.

Clustering

In one particular trial of the hard assignment unsupervised Generalizable Gaussian Mixture model, described in section 3.3, 6 clusters were obtained. The scatter plot of the assigned to the clusters data and the corresponding density contours $p(\mathbf{x})$ is presented on figure 7.8. Clusters are marked as indicated on the legend bar. The same 2 principal components are shown as also are displayed on figure 7.2.

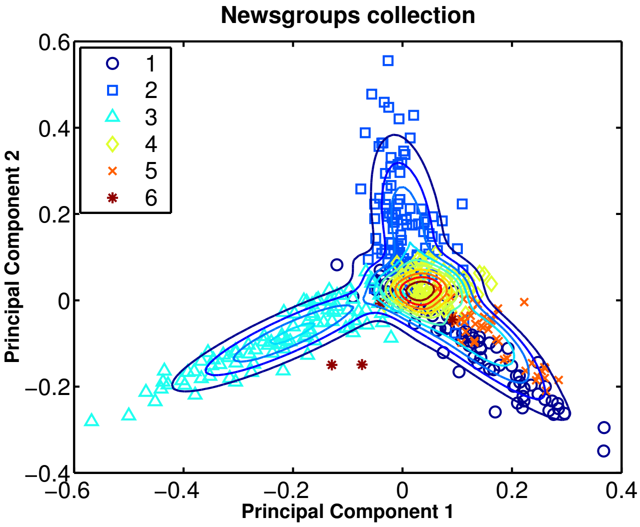
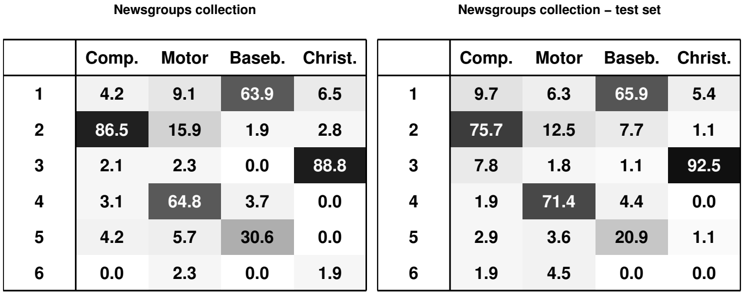


Figure 7.8 Scatter plot of the data clustered with UGGM model with hard assignment for Newsgroups collection. The estimated probability density function $p(\mathbf{x})$ is shown with the counter plot. 6 clusters were found in this particular experiment. The legend bar provides cluster labeling.

Since data points are labeled, the confusion matrices for training and test ensemble are calculated and they are presented on figure 7.9. From that matrices it is possible to determine what kind of documents to expect in the clusters. Thus, for example, cluster 3 contain the majority of newsgroup documents concerning *Christian religion*, clusters number 1 and 5 – *Baseball* and cluster number 2 – *Computer Graphics*. Some of the clusters, for example number 1, 3 or 5, are mixed with respect to the original labels.

Soft assignment UGGM model was performed for comparison. On average, the number of clusters obtained in the algorithm was larger than that obtained from hard assignment algorithm. In one particular experiment 13 clusters were acquired. The scatter plot of the labeled data and the obtained probability density function $p(\mathbf{x})$ is presented on figure 7.10. As in case of Email collection, the density does not differ significantly from the hard version algorithm. However, it is described with twice as many components, what suggests better fit to the underlying data distribution.



Newsgroups collection

	Comp.	Motor	Baseb.	Christ.
1	4.2	9.1	63.9	6.5
2	86.5	15.9	1.9	2.8
3	2.1	2.3	0.0	88.8
4	3.1	64.8	3.7	0.0
5	4.2	5.7	30.6	0.0
6	0.0	2.3	0.0	1.9

Newsgroups collection – test set

	Comp.	Motor	Baseb.	Christ.
1	9.7	6.3	65.9	5.4
2	75.7	12.5	7.7	1.1
3	7.8	1.8	1.1	92.5
4	1.9	71.4	4.4	0.0
5	2.9	3.6	20.9	1.1
6	1.9	4.5	0.0	0.0

Figure 7.9 The confusion matrix of the Newsgroups collection calculated based on the available data labels. Left figure present the numbers for the training set and right figure for the test set. The data is originally labeled in four groups: *Computer Graphics*, *Motorcycles*, *Baseball* and *Christian Religion*. The matrices in supervised way supply with the cluster explanation. For example, cluster 4 collects documents concerning motorcycles, cluster number 3 in 89% is composed from *Christian Religion* newsgroups and Baseball newsgroups are divided between two clusters number 1 and 5.

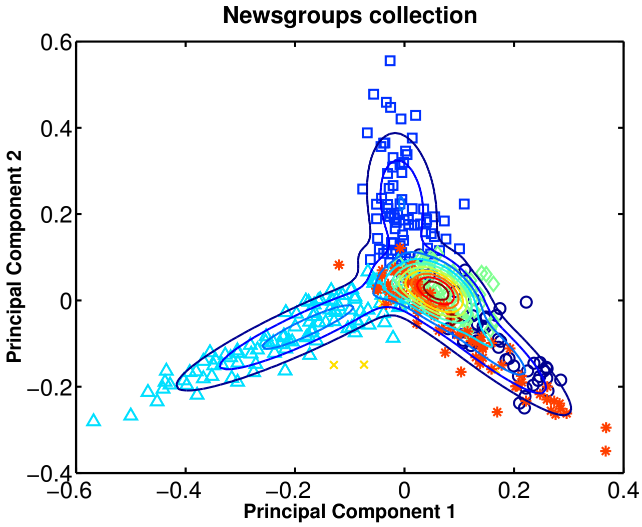


Figure 7.10 Scatter plot of the Newsgroups collection clustered with soft assignment UGGM model. The estimated probability density function $p(\mathbf{x})$ is shown with the counter lines. 13 components are acquired as the setting that provides minimum generalization error. Visually, the density structure is similar to the one obtained by the hard assignment algorithm (figure 7.8).

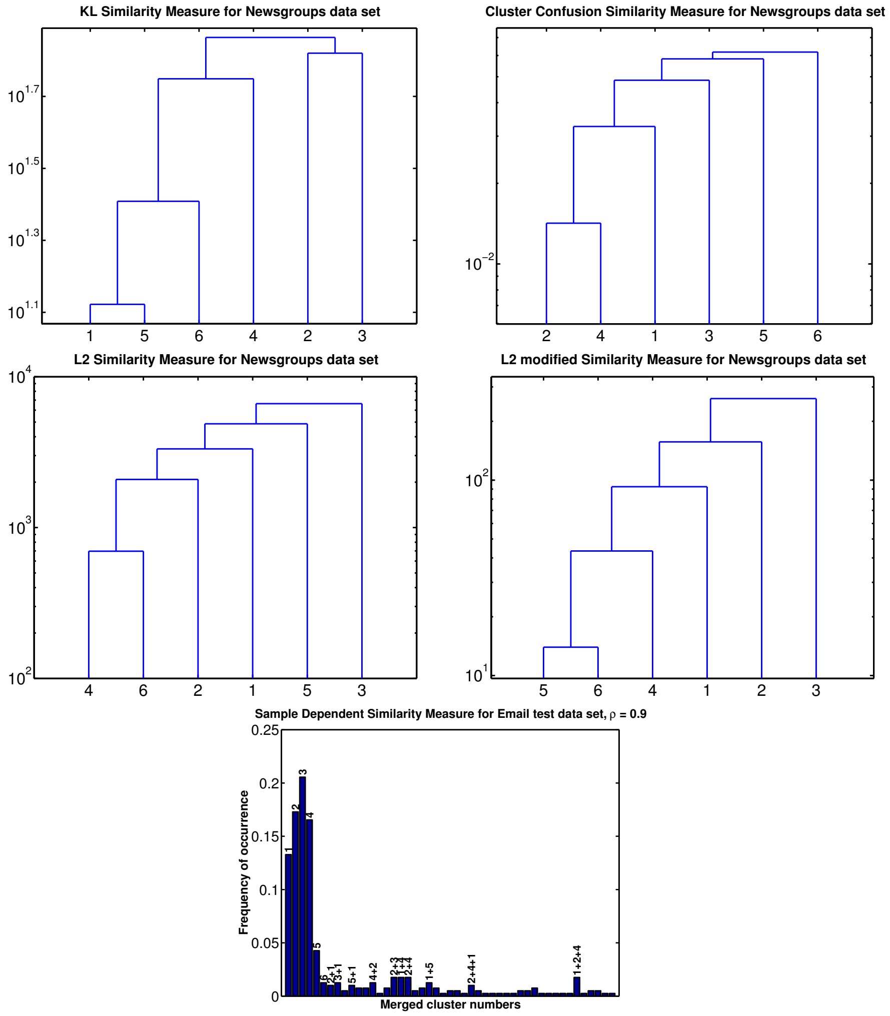
Hierarchical clustering

The hierarchical structure is build based in the 6 cluster outcome of the hard UGGM model. Five similarity measures, described in section 4.1.1 are applied and they are presented on figure 7.11. KL, Cluster Confusion, $\mathcal{L}_2$ and modified $\mathcal{L}_2$ similarity measures are visualized with the dendrograms and for Sample Depended similarity measure the most frequently occurring combinations are shown. Approximately 70% of the data points are assigned to the first level clusters that are coming directly from the UGGM model. The confidence level used in assignment is larger than 90%, i.e. the cluster posterior of the data points ia larger than 0.9, $p(k|\mathbf{x}_n) > 0.9$. Remaining 30% of the data points are assigned in the hierarchy. Most often combinations include following cluster unions: $\{1,4\}$, $\{2,4\}$, $\{1,2,4\}$, $\{2,3\}$, $\{1,5\}$, $\{5,6\}$, $\{4,6\}$. Clusters 1, 2 and 4 are likely merged, since their overlap is significant. This overlap is concluded based on the confusion matrices on figure 7.9 and the Cluster Confusion similarity measure (figure 7.11) which combines these clusters first. With respect to the original labeling the KL similarity measure performs the best in case of this collection. It preserved the original classes division on the third hierarchy level, where 4 clusters corresponding to 4 classes are remained.

Keywords assignment

Keywords corresponding to the dendrogram structures are presented below on figures 7.7, 7.8, 7.9, 7.10 and 7.11.

On figure 7.7 keywords for the first hierarchy level are presented, for the clusters obtained with hard assignment UGGM model. Thus, 6 clusters are observed, scatter plot of which is given on figure 7.8. In the table also corresponding mixture proportions are shown.


KL Similarity Measure for Newsgroups data set

Cluster Confusion Similarity Measure for Newsgroups data set

L2 Similarity Measure for Newsgroups data set

L2 modified Similarity Measure for Newsgroups data set

Sample Dependent Similarity Measure for Email test data set, ρ = 0.9

Figure 7.11 Hierarchical structures for Newsgroups collection. UGGM with hard cluster assignment resulted with 7 clusters at the first level $j = 1$. Four similarity measures described in the section 4.1 are presented on figures. The confusion matrix and the scatter plots of the data were shown previously on figures 7.5 and 7.4, respectively. All similarity measures differ significantly.

Cluster	*P(k)*	Keywords
1	.22	team, game, year, win, run, pitch, hit, write, article, score
2	.26	file, write, article, image, graphic, format, window, program, color, gif, ftp, convert, package, work, read
3	.25	god, write, christian, people, jesus, article, question, faith, truth, life, christ, time, thing, bible, church
4	.16	bike, write, article, motorcycle, dod, dog, good, road
5	.11	write, article, year, game
6	.01	write, article, bike, year, game, team, god, win, good, hit, run, pitch, people, make, time, dod

Table 7.7 Keywords for the 6 clusters obtained from hard assignment UGGM model. Keywords provide the fully unsupervised interpretation of the clusters. For example, cluster number 1 is represented by such keywords as: *team, game, win, pitch, hit* which certainly connect with a team game topic. Keywords for cluster number 2 represent computer graphics related topics with words like: *file, image, graphics, format*, etc. Cluster number 3 with words like: *god, christian, people, jesus, faith* belongs to *Christian religion* newsgroup. Keywords for cluster number 4 (*bike, motorcycle, dod, road*) correspond to the *Motorcycle* newsgroup.

Cluster number 1 is represented by such keywords as: *team, game, win, pitch, hit* which certainly connect with a team game topic. Keywords for cluster number 2 represent computer graphics related topics with words like: *file, image, graphics, format*, etc. Cluster number 3 with words like: *god, christian, people, jesus, faith* belongs to *Christian religion* newsgroup. Keywords for cluster number 4 (*bike, motorcycle, dod, road*) correspond to the *Motorcycle* newsgroup. Cluster 5 relates to writing articles about games and cluster 6 contains mixture of keywords from different subjects like: *bike, game, people*. In table 7.8 keywords for hierarchy build based on KL similarity measure are presented. For this outcome of UGGM model cluster number 3 is dominant. It can be seen in all the following tables containing hierarchy keywords, where the top

Cluster	Keywords
7	team, game, year, win, run, pitch, hit, write, article, score
8	team, game, year, win, run, pitch, hit, write, article, score
9	team, game, year, win, run, pitch, hit, write, article, score
10	god, write, christian, people, jesus, article, question, faith, truth, life, christ, time, thing, bible, church
11	god, write, christian, people, jesus, article, question, faith, truth, life, christ, time, thing, bible, church

Table 7.8 Keywords for the clusters in the hierarchy build with KL similarity measure. The first levels are dominated by cluster 1, the top clusters (10 and 11) inherit keywords from the structure most dominant cluster – number 3.

level cluster is represented by keywords coming from this particular cluster. In the superior for this collection KL similarity measure the following clusters are observed on the level with remained 4 clusters: $8\{1, 5, 6\}$, 4, 2 and 3.

Cluster	Keywords
7	bike, write, article, motorcycle, dod, dog, good, road
8	file, write, article, image, graphic, format, window, program, color, gif, ftp, convert, package, work, read
9	file, write, team, game, article, image, bike, year, win, graphic, run, format, pitch, hit, window, program, color, gif, ftp, convert
10	file, write, team, game, article, image, bike, year, win, graphic, run, format, pitch, hit, window, program, color, gif, ftp, convert
11	god, write, christian, people, jesus, article, question, faith, truth, life, christ, time, thing, bible, church

Table 7.9 Keywords for the clusters in the hierarchy build with Cluster Confusion similarity measure.

Cluster	Keywords
7	bike, write, article, motorcycle, dod, dog, good, road
8	file, write, article, image, graphic, format, window, program, color, gif, ftp, convert, package, work, read
9	file, write, team, game, article, image, bike, year, win, graphic, run, format, pitch, hit, window, program, color, gif, ftp, convert
10	file, write, team, game, article, image, bike, year, win, graphic, run, format, pitch, hit, window, program, color, gif, ftp, convert
11	god, write, christian, people, jesus, article, question, faith, truth, life, christ, time, thing, bible, church

Table 7.10 Keywords for the clusters in the hierarchy build with $\mathcal{L}_2$ similarity measure.

Cluster	Keywords
7	write, article, year, game
8	write, article, bike, year, game, motorcycle, dod, dog, team, baseball
9	team, game, year, win, run, pitch, hit, write, article, score
10	file, write, team, game, article, image, bike, year, win, graphic, run, format, pitch, hit, window,program, color, gif, ftp, convert
11	god, write, christian, people, jesus, article, question, faith, truth, life, christ, time, thing, bible, church

Table 7.11 Keywords for the clusters in the hierarchy build with modified $\mathcal{L}_2$ similarity measure.

7.2.4 Clustering of Email collection with unsupervised/supervised Generalizable Gaussian Mixture model

Email collection is applied in illustration of the USGGM model, described in section 3.5. In these experiments slightly different, preprocessing steps are performed, than reported earlier, resulting with total number of 1280 email documents (640 for both in training and test set), and the term-vector consists of 1652 words. The difference is in a another threshold value which is set on term frequencies.Words that occur less than 40 times[12] are removed. what reduces significantly the length of the term vector. In result, also some of the atypical[13] documents are removed that contained less than 2 words. The commonly used framework Latent Semantic Indexing (LSI) [20] is employed, which operates using a latent space of feature vectors. These are found by projecting term-vectors into a subspace spanned by the left eigenvectors associated with the largest eigenvalues found by a singular value decomposition of the term-document matrix, for reference see chapter 2 with description of the projection methods. 5-dimensional subspace is used. The data in reduced space do not differ significantly from the previous experiments.

Figure 7.12 shows the average performance (over 1000 runs) of the USGGM algorithm.

The algorithm parameter is set to $\gamma = 0.5$. The algorithm is performed with $N_u = 200$ unlabeled examples and a variable number of labeled samples. As expected, with few labeled examples available, $N_l = 10, 20$, the optimal λ is close to one, where all unlabeled data are fully used. The minimums in the classification error curves (upper left plot on figure 7.12) are marked with the black triangles ▲. As N_l increases, λ decreases and for $N_l = 200$ equals 0.3, indicating the reduced utility of unlabeled examples. The classification error is reduced, approximately 26% using unlabeled data for $N_l = 10$ and gradually decreasing to 1% when $N_l = 200$. Thus, with large set of labeled examples the importance of unlabeled samples is negligible, therefore the value of the discount factor λ is not crucial for the level of the error. The classification error for optimal λ as a function of the size of the labeled data set N_l is shown in the right upper plot of figure 7.12. The number of optimal components, selected by

[12] The threshold value was investigated and, in conclusion, the classification error remains roughly the same for any values below approximately 100 occurrences. In order to reduce dimensionality of the term vector, the threshold is set to 40 occurrences.

[13] Atypical documents contain many words, which are rare in the database.

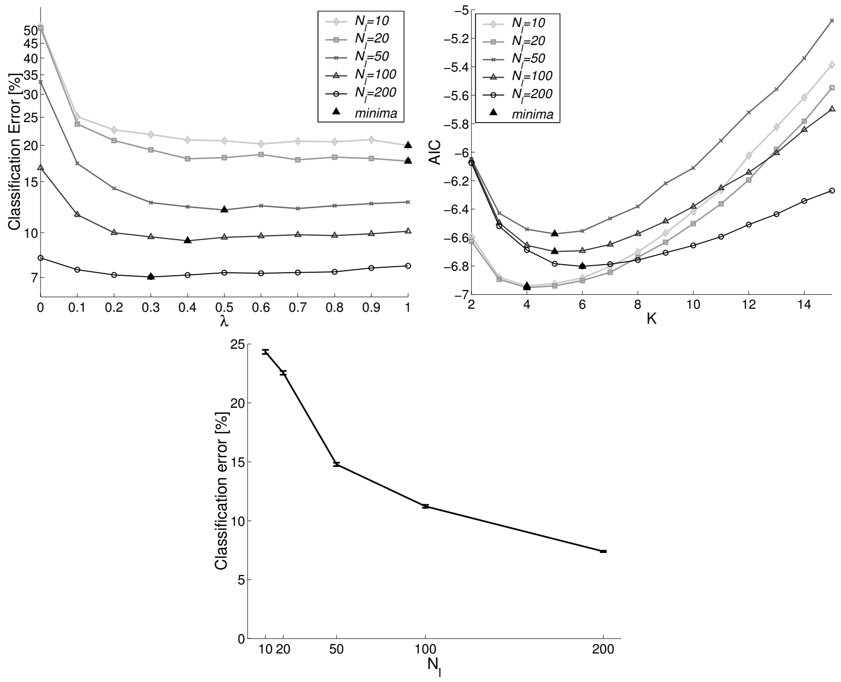


Figure 7.12 Average performance of the USGGM algorithm over 1000 repeated runs using $N_u = 200$ unlabeled examples and a variable number of labeled examples N_l. Upper left plot shows the performance as a function of the discount factor λ for unlabeled examples ($\lambda = 0$ corresponds to no unlabeled data). The upper right plot shows number of components selected by the AIC criterion for optimal λ as described in section 3.5.

the mixture model[14], grows with the number of labeled examples N_l. Naturally, the classification error decreases when enlarging the size of the labeled data set (lower plot of figure 7.12).

In figure 7.13, the hierarchies of individual class dependent densities $p(\mathbf{x}|y)$ are presented. In this experiment only the modified $\mathcal{L}_2$ dissimilarity is used, since the KL similarity measure is approximate and the Cluster Confusion measure is computational expensive if little overlap exist as many ancillary data are required. The modified $\mathcal{L}_2$ is computational inexpensive and basically treat dissimilarity as the cluster confusion, while the standard $\mathcal{L}_2$ do not incorporate

[14] The selection was based on the training error with AIC criterion [1].

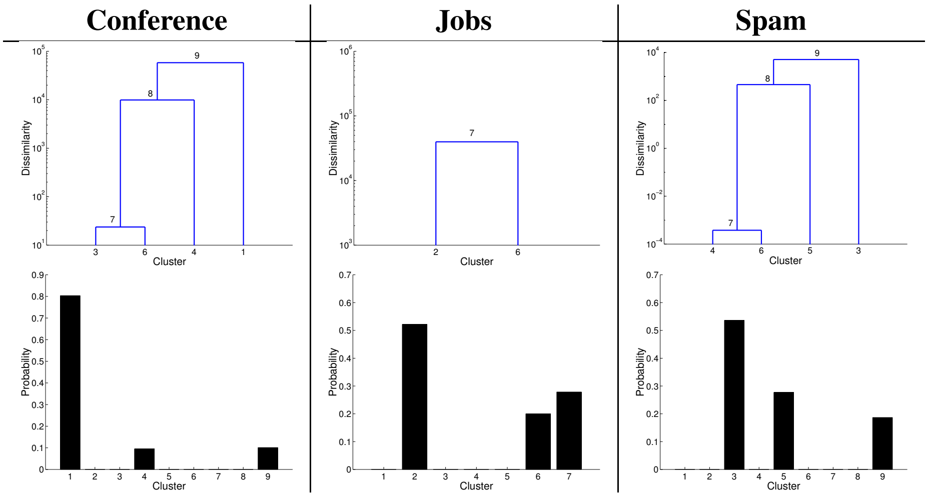


Figure 7.13 Hierarchical clustering using the USGGM model. Left column is class $y = 1$ *conference*, middle column $y = 2$ *jobs*, and right column is for $y = 3$ *spam*. Upper rows show the dendrogram using the modified $\mathcal{L}_2$ dissimilarity for each class, and the lower row the histogram of cluster level assignments for test data.

priors. The lower panel on figure 7.13 shows the cluster level assignment distributions of test samples. In case of *conference* class 90% of the data points falls into the first level clusters, obtained directly form the USGGM model. For *job* 74% are classified in the first hierarchy level and for *spam* emails 83%. Rest of the data points is described by the higher hierarchy levels, i.e. by the combined cluster representations.

Typical features, as described in section 4.3, are selected from the high density region and back-projected into original term-space providing keywords for each cluster. Keywords are given in table 7.12. The *conference* class is dominated by cluster 1 and cluster 4. This has keywords listed in table 7.12, which are in accordance with the meaning of conference. The *job* class split between 2 clusters, namely 2 and 6 and *spam* emails are divided between clusters 3 and 5.

y	k	$P(k\|y)$	Keywords
1	1	.7354	information, conference, call, workshop, university
	3	.0167	remove, address, call, free, business
	4	.2297	call, conference, workshop, information, submission, paper, web
	6	.0181	research, position, university, interest, computation, science
2	2	.6078	research, university, position, interest, science, computation, application, information
	6	.3922	research, position, university, interest
3	3	.6301	remove, call, address, free, day, business
	5	.3698	free, remove, call

Table 7.12 Keywords for the USGGM model. $y = 1$ is *conference*, $y = 2$ is *jobs* and $y = 3$ is *spam*. The *conference* class is dominated by cluster 1 and cluster 4. This has keywords which are in accordance with the meaning of conference. Similarly for the *job* class which is split between 2 clusters, namely 2 and 6 and for *spam* emails that are divided between 3 and 5.

7.3 Segmentation of medical database

7.3.1 Segmentation of the data from sun-exposure study

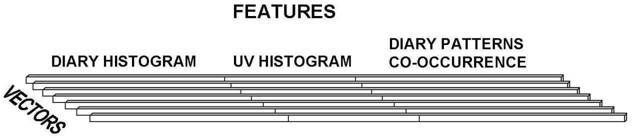
The hard assignment UGGM model, described in section 3.5, is used in clustering the data of the sun-exposure study. The *behavioral patterns* are investigated, i.e. as data samples the several consecutive diary records are understood. In that way, a similar approach to the one used in processing textual databases, may be used. The results of the presented below experiments can be found in [82].

Preprocessing

Since the technique was developed for textual data, it is necessary to redefine some of the variables. As features, unique records are found in the data and the histograms over these records are build. The total number of observed patterns for 8 questions, selected from table 7.1, is 20736. This number is obtained by multiplying all possible answers, i.e. $20736 = 2 \cdot 2 \cdot 3 \cdot 2 \cdot 2 \cdot 27 \cdot 4 \cdot 4$. However, only a small fraction of 423 patterns exists actually in the data. As documents, the sliding time window is used which is calculated separately for each of the survey participants. The optimal size of the window is an important issue. For example, taking the full set of records belonging to each person will produce a set of points in the space that will not form any particular structure, since each of them will contain most of the observed patterns. On the other hand, if observing only one diary record at the time, all the different vectors will be equal distant so the cluster structure in such space will be lost as well. In the experiments a window of size 7 is used. This was decided after performing several experiments, taking into account stationarity of the obtained clustering and the level of the computational complexity, which is large for small window sizes. Attempts were made to apply the generalization error (equation 2.12) in window selection, however, without success. The histograms are, at the same time, characteristic for the subjects and the time windows. Similar to the text data, the histograms create the pattern-window matrix which, for simplicity, is referred to as the term-document matrix. This matrix is normalized to the unit $\mathcal{L}_2$-norm length.

The term-document matrix is formed from the histogram vectors that are obtained by counting occurrences of every pattern in the window. However, the histograms does not convey time ordering information. Thus, it is possible to include additionally time information by considering the co-occurrence matrix of joint occurrences of neighbor patterns in the window. There are 20736^2 possible co-occurrences (or in the case of this particular collection 423^2) but only a small fraction of 1509 combinations is present in the actual data set.

An attempt was made to combine the different types of data. While the diary entries are of categorical type, the measured UV radiation is continuous. The UV measurements, in order to match the presented framework, are quantized and they are expressed by 4-valued representation. In order to incorporate the time relation between the consecutive records the co-occurrence of the diary records is introduced. Those three data sources can be combined in the presented framework simply representing each document[15] by the histogram of the feature vector from the combined sources. This idea is shown on the figure below.



[15] The document in this case corresponds to a window of n consecutive days of the survey belonging to one subject.

The set of 19171 diary records with corresponding UV measurements is selected for the clustering experiments. Data is complete, i.e. there is no missing records or UV measurements. From this collection, records belonging to 10 selected subjects are hold out for testing. The sliding window of size 7 resulted with 2738 data vectors from which 158 is selected as test set and 2064, 80% of remaining data, as training set and 516 as validation set. Each feature vector consists of the diary histogram, the co-occurrence matrix and the UV histogram.

In line with the KDD process, presented on figure 1.1, this particular collection is processed as shown on figure 7.14. In the first step, data is windowed creating vectors that contain data from consecutive days. Both diary histograms, the co-occurrence matrix and UV radiation histograms are screened against rare patterns by removing patterns. The diary histogram is reduced from 423 to 97 patterns[16] and in a similar way, the co-occurrence matrix is reduced from

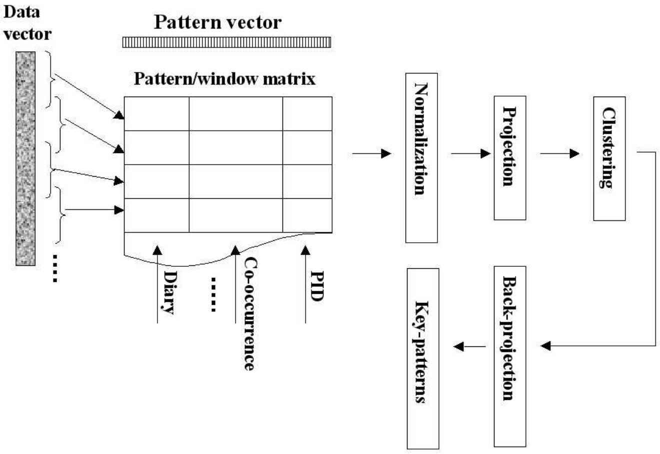


Figure 7.14 Framework for data clustering: 1) The data is windowed into several histogram vectors and together with the co-occurrence matrix and the UV histograms form a term-document matrix. 2) Data is normalized and projected into the latent space found by singular value decomposition. 3) The Gaussian mixture model is used to cluster the data. 4) For interpretation the typical features are drawn from data distribution and back-projected to the original space where key-patterns are found.

[16] The patterns larger than .1% maximum value was preserved. It equals 96% of total mass ($\sum_{dn} x_{dn} = 100\%$).

1509 to the 80 most often occurring pairs of patterns. The next step involves normalization of the term-document matrix. Two types of normalization are performed. First, each window histogram is scaled to unity $\mathcal{L}_2$-norm length, and then, pattern vectors are scaled to zero mean and unit variance over training samples.

Projection to the latent space

Each type of the data[17] is projected separately on the orthogonal directions found by singular value decomposition (the PCA projection method which is described in section 2.1.2). The scatter plots of the data in selected principal component space are presented on figure 7.15.

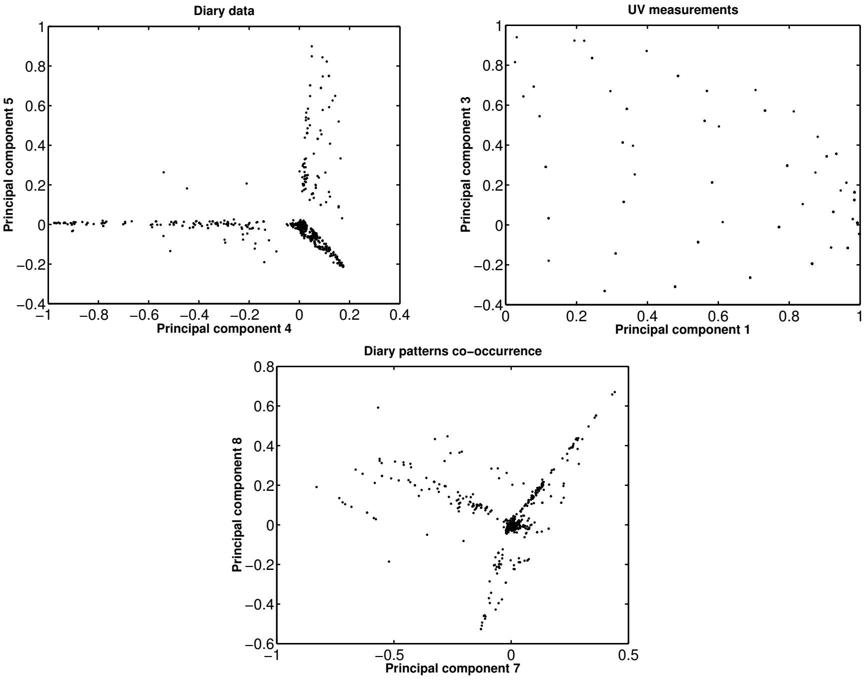


Figure 7.15 Scatter plots of the training data sets in the latent space for the sun-exposure collection. For good visualization 2 principal components were carefully selected. The top panel shows the diary data and UV measurements. The lower plot presents the scatter plot of the projected diary patterns co-occurrence matrix. It is possible to see the existing structure in the data.

[17] Diary and UV histograms and co-occurrence matrix.

The corresponding eigenvalues are shown on figure 7.16. For both diary and

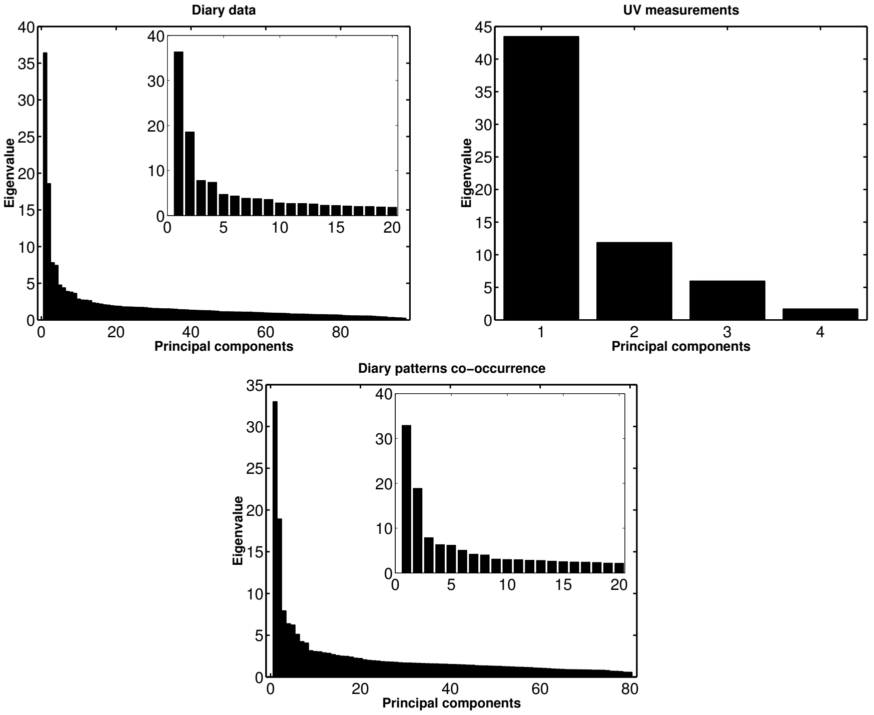


Figure 7.16 Eigenvalue curves for the sun-exposure collection. Additionally, the largest eigenvalues are shown in close-up. Based on these plots the decision of number of principal components is made. 9 eigenvalues were chosen in case of Diary data and Diary patterns co-occurrence. 3-dimensional space was selected for UV measurements.

co-occurrence 9 largest eigenvalues are selected. In case if UV measurements 3 eigenvalues are chosen.

Clustering and cluster interpretation

The clustering is performed of term-document matrix build from diary records. Additionally, the cluster structure is investigated of combined diary records with UV measurements and the co-occurrence matrix. The segmentation was performed by the unsupervised Gaussian Mixture model with hard cluster assignment.

Since the data is not labeled only unsupervised cluster interpretation can be applied. Based on estimated density the typical features are selected and back-projected to the histogram space where the mean is added. As previously, in the case of textual collections, the negative values in back-projected vectors are neglected. As keywords in case of this data set key-patterns are selected which corresponds to the most probable diary records, UV measurements and co-occurrence couplings.

Furthermore, the used framework makes it possible to describe the behavior of every new person in the experiment in terms of cluster assignment and associated keywords. The confidence of assigning the person into a given cluster k can be expressed by the posterior probability:

$$p(k|Per) = \frac{1}{N} \sum_{n=1}^{N} p(k|Per, \mathbf{x}_n) \cdot p(\mathbf{x}_n), \qquad (7.1)$$

where $\mathbf{x}_n$ is a feature vector of the size d and $i = 1, 2, \ldots, N$. The number of feature vectors N is different for every person and depends on the number of returned diary records and the window size. In this experiment the selected test set with records from 10 subjects are used.

The investigation was performed of the importance of the co-occurrence matrix and the UV histograms for the clustering. The results of the experiments are collected in the tables 7.13, 7.14, 7.15 and 7.16 where the key-patterns, associated probabilities and description of the clusters are provided. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The third gives the probabilities for the key-patterns and the fourth column presents a general interpretation of the cluster based on the observed key-patterns.

In table 7.13 the results are shown of clustering of the diary histograms. The presented patterns are equivalent to the set of questions given in section 7.1. Note, that 2 questions (number 1 and 7) were a priori excluded for this experiments. For example: pattern 10111 describes the following set of answers: 1. holiday - yes, 2. abroad - no, 3. sun bathing - yes, 4. naked shoulders - yes, 5. on the beach - yes, remaining questions 6,7 and 8 - no, or pattern 0: all the questions where answered - no or pattern 1: 1. holiday - yes and the rest of the questions from 2 to 8 - no. This rule for describing patterns hold as well in case of tables 7.14, 7.15 and 7.16.

#.	Key-Pattern	Probability.	Description
1.	10001,11,10111	0.33,0.32,0.19	holiday, on the beach, sun bathing
2.	0	0.98	working - no sun
3.	1	0.9	on holiday - no sun
4.	0,0001,1	0.4,0.27,0.18	working naked shoulders - no sun
5.	1,1101	0.67,0.17	holiday, naked shoulders
6.	1011,1001,10011	0.47,0.17,0.16	holiday , sun bathing
7.	11	0.5	holiday abroad - no sun
8.	10111,0001,1001	0.45,0.17,0.13	holiday, sun bathing, naked shoulders
9.	0000001	0.05	no sun, sunburned - red
10.	0	0.99	working - no sun

Table 7.13 Key-patterns for clustering diary histograms. In the first column the cluster number is shown. Second column contains the most probable patterns for the cluster. The presented pattern numbers are equivalent to the set of questions given in section 7.1. For example: pattern 10111 gives the following set of answers: holiday - yes, abroad - no, sun bathing - yes, naked shoulders - yes, on the beach - yes, remaining questions 6,7 and 8 - no, or pattern 0 means that all the questions where answered - no. Third column gives the probabilities for the key-patterns, and fourth column presents a general description of cluster.

#	Key-Pattern	Probability.	Description
1.	1001,1000, 1_{UV},10011	0.31,0.26, 0.16,0.11	holiday, naked shoulders, small UV radiation
2.	11,0001,0, 2_{UV},0_{UV}	0.29,0.2,0.17, 0.16,0.15	holiday abroad, working
3.	1,11	0.39,0.12,	holiday
4.	1011,2_{UV}, 3_{UV},0001	0.0.31,0.25, 0.14,0.13	naked shoulders, high sun radiation
5.	1,2_{UV},3_{UV},10001	0.2,0.17,0.16,0.14,0.12,0.1	holidays, high UV
6.	1_{UV},0_{UV}	0.14,0.13	low UV
7.	3_{UV},1001, 2_{UV},10011	0.22,0.22, 0.16,0.15	holiday, naked, shoulders, high UV
8.	0,0_{UV}	0.6,0.4	no sun

Table 7.14 Key-patterns for clustering diary histograms combined with UV histograms. In the first column the cluster number is displayed. Second column contains the most probable patterns. The rule for interpreting the diary key-patterns is given in table 7.13. Patterns corresponding to the UV histograms are marked with the subscript "UV". Four different values of UV are observed: 0 corresponds to very low sun radiation and 3 describes very high one. Third column gives the probabilities for the key-patterns and fourth column presents general description of cluster based on the key-patterns.

Table 7.14 presents key-patterns for clustering diary histograms combined with UV histograms. Eight clusters were found. Diary key-patterns are explained in table 7.13. Patterns corresponding to the UV histograms are marked with the subscript "UV". Four different values of UV from 0 to 3 are observed: 0 corresponds to the very low sun radiation and 3 describes very high one. This rule for describing UV-patterns hold as well in case of table 7.16.

#	Pattern	Probability.	Description
1.	1001,1101-1101	0.27,0.13	holiday,naked sholders
2.	1001,1101-1101,1	0.26,0.21,0.1	holiday,naked sholders
3.	0001,1001-0, 10111,0-1001	0.17,0.12, 0.11,0.1	working, naked shoulders
4.	11,11-11	0.14,0.11	holiday, abroad
5.	0001,1001-0, 1001,0-1001	0.27,0.14, 0.13,0.1	holiday or working, naked shoulders
6.	1001,0,0-1,1-0,1,1-1	0.29,0.19,0.16,0.14,0.1,0.09	work - holiday, no sun
7.	10011	0.19	holiday, on the beach
8.	10001,1-10011	0.21,0.12	holiday, on the beach
9.	0-0,0,0-1,1-0	0.36,0.35,0.12,0.12	working - no sun
10.	1001,1-1101,1101-1, 1101-1101,1011	0.26,0.16,0.12, 0.12,0.11	holiday, naked shoulders

Table 7.15 Key-patterns for clustering diary histograms combined with co-occurrence matrix. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The rule for interpreting the diary key-patterns is given in table 7.13. The co-occurring patterns are shown with the dash between them, e.g. "0-1" means that a pattern working is followed by pattern holiday. Third column gives the probabilities for the key-patterns and fourth column presents general description of cluster based on the key-patterns.

In table 7.15 the key-patterns for clustering diary histograms combined with the co-occurrence matrix are presented. The diary key-patterns are explained in table 7.13. The co-occurring patterns are shown with the dash between them e.g., "0-1" means that a pattern working is followed by pattern holiday, pattern "1-10011" means that holiday without sun was followed by holiday spent on the beach. This rule for describing co-occurrence patterns hold as well in case of table 7.16.

#	Pattern	Probability	Description
1.	1001,0001,1101-1101	0.15,0.1,0.09	naked shoulders
2.	0,1_{UV},0-0,0001	0.17,0.16,0.14,0.13	working, low sun radiation
3.	1001-0,0-1001, 0,1-0,0-1,0_{UV}	0.17,0.14 ,0.14,0.12,0.11,0.1	no sun radiation, holiday-work
4.	1_{UV},2_{UV}	0.12,0.1	medium sun exposure
5.	3_{UV},11,11-11	0.11,0.11,0.09	holiday, high sun radiation
6.	0-0,0,0_{UV}	0.29,0.27,0.23	working, no sun

Table 7.16 Key-patterns for clustering the diary histograms combined with the co-occurrence matrix and the UV histograms. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The diary key-patterns are explained in table 7.13. The co-occurring patterns are explained in table 7.15 and the UV patterns in table 7.14. Third column gives the probabilities for the key-patterns and fourth column presents general description of cluster based on the key-patterns. This clustering provides the most compact and explicit result. Cluster no. 5 collects very high sun radiation patterns while cluster no. 2 very low ones.

Table 7.16 shows the key-patterns for clustering diary histograms combined with co-occurrence matrix and UV histograms. The diary key-patterns are explained in table 7.13. The co-occurring patterns are explained in table 7.15 and the UV patterns in table 7.14. Both the UV values and the co-occurrence pairs are likely to appear as key-patterns. Cluster ni. 5 collects very high sun radiation patterns while cluster no. 2 very low ones and similarly clusters 3 and 1. Cluster 4 corresponds to medium sun radiation behavior. This suggests that joining time information and the sun exposure measurements are important for the clustering. In conclusion this clustering is the most compact and explicit in result.

In figure 7.17 the probability is presented of observing certain groups of behaviors in the clusters together with registered sun exposure values are. Clustering was performed using full pattern/window matrix[18] for which keywords are displayed in table 7.16. Five behaviors are specified: *working - no sun exposure*, *holiday - no sun exposure*, *sun exposure* describes mild sun behaviors often on the beach or naked shoulders without sun-screen and without sunburns, and corresponding to high sun exposure behaviors: *using sun-block* and *sunburns*. In the lower panel the measurements are presented of the sun radiation. For example cluster number 6 groups behaviors marked as *working - no sun* and corresponding UV values are low. An opposite situation occurs for cluster no.

[18] With all the sources combined, i.e. diary and UV and co-occurrence histograms.

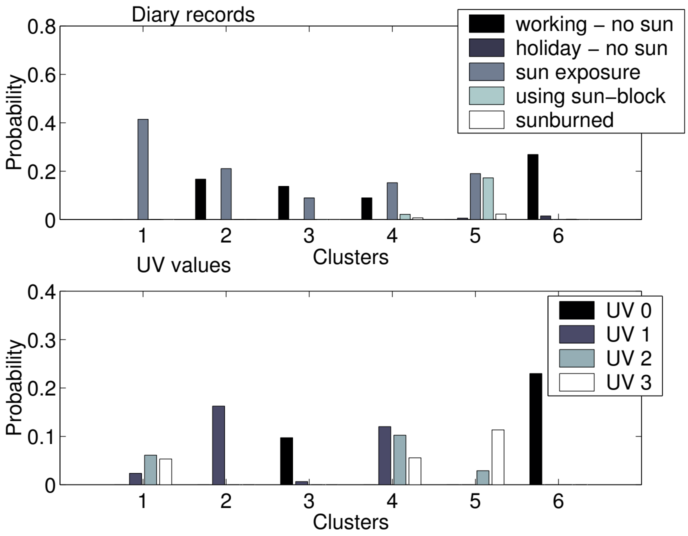


Figure 7.17 The probability of observing certain groups of behaviors in the clusters together with registered sun exposure values. Key-patterns for the clusters are presented in the table 7.16. For each cluster grouped behaviors from diary records are presented on the upper plot and corresponding UV radiations are shown on the lower figure. For example cluster number 6 groups behaviors marked as *working - no sun* and corresponding UV values are low. An opposite situation occurs for cluster no. 5 which contains records with reported sunburns, sun exposure and using sun-block and consequently the observed UV values are high.

5 which contains records with reported sunburns, sun exposure and using sun-block and consequently the observed UV values are high.

For the same clustering setting the cluster probabilities were calculated for 10 test subjects using equation 7.1. Together with key-patterns presented in table 7.16 description is possible of the behavior of the particular persons during the whole period of the survey. For all test persons there is a large probability of the cluster no. 6 that describes working and no sun exposure. However, some of the periods are described by other patterns. For example, for person no. 251 there is high probability component for cluster no. 5 describing holidays with high sun radiation. Persons no. 213 and 35 can be well described by clusters 6 (working, no sun) and 1 (naked shoulders) while person no. 23 by clusters 6, 1 and 4 (medium sun exposure).

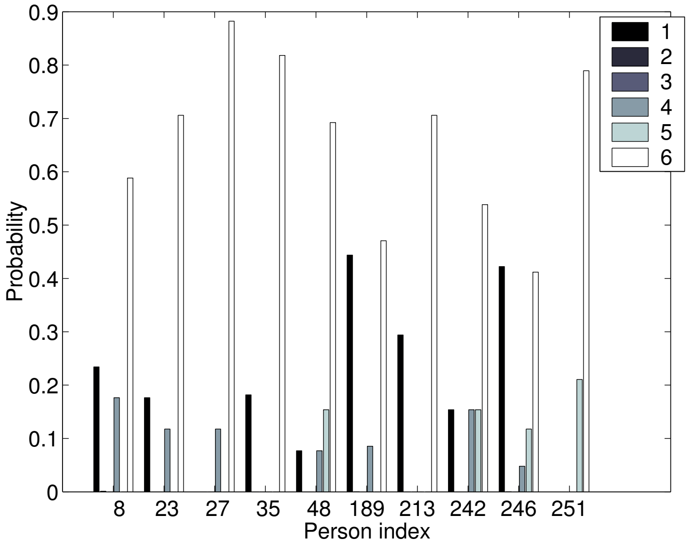


Figure 7.18 Cluster probabilities calculated for the 10 test persons equation (7.1). Person index is shown on the x-axes and different grey level colors corresponds to six clusters. The corresponding key-patterns are given in table 7.16. For all test persons there is a large probability of the cluster no. 6 that describes working and no sun exposure. However, some of the periods are described by other behavioral patterns. For example, for person no. 251 there is high probability component for cluster no. 5 describing holidays with high sun radiation. Persons no. 213 and 35 can be well described by clusters 6 (working, no sun) and 1 (naked shoulders) while person no. 23 by clusters 6, 1 and 4 (medium sun exposure).

7.3.2 Imputation missing values in Sun-Exposure study

Preprocessing

The sun-exposure data was used in connection with the missing values imputation in [81]. Since, the diary records, originally, are categorical, both nominal and ordinal, coding technique is proposed that converts the data to binary vectors. For this purpose *1-out-of-c* coding is used. It represents c level categorical variable with a binary c bits vector. The example of coding is presented in Chapter 5 in table 5.1.

From the questionnaire, given in table 7.1, one question is ordinal variable of

26 states, namely *Sun Screen Factor Number*. Therefore, for simplicity adn for dimensionality reduction, it is quantized, to 5 levels (no/1–7/8–16/17–35/> 35), before coding is applied. Each diary record, in the final form for missing data analysis, is described by 17 dimensional binary feature vector.

Due to the characteristics of data, three different profiles are taken into consideration. The first, which is called *Complete Diary Profile* (CDP), uses all records in the estimation process. The second, *Personal Profile* (PP), assumes that all questionnaires from one person have similar characteristics while the characteristics across the persons differ. This arise from the expectation that human behavior varies from person to person. Thus, the estimation is done from the other complete records of given person. The third profile is the *Day Profile* (DP), which assumes that data vectors for one day are similar or equivalently belong to one distribution while parameters of the distributions across the days vary. This is due to the fact that human behavior is influenced by weather, day of week, temperature, the season of the year, etc. The model using each of the described profiles is called a method. In addition, a *Voting* procedure is also considered. It compares proposals from all the above mentioned methods and takes the majority vote among the outcomes. This method is expected to give the best results, however, it is much more computationally expensive since it combines the other three methods.

In order to imputate missing values in the diary records, the models described in chapter 5 are implemented here. To check the performance, the techniques are tested on complete questionnaires. Diary records description is presented in section 7.1 and necessary preprocessing steps are given in section 7.3.1.

The leave-one-out permutation estimate of the generalization error is performed, where one validation sample is chosen randomly from the complete data set in 500 repeated permutations and then a number of training samples. The performance is then an average over the 500 permutations. As an example, if considering Day Profile, the day number of the validation sample specifies the day number of the training samples of which, there are at most 194 persons to choose from. When training set size, N, is smaller than 194 then N randomly chosen samples out of 194 are selected.

Errors of, so called, low concentration are investigated, i.e. only one binary block in the vector, that is corresponding to one question, is missing at the time. The final error rate is an average over such single errors made in all possible nine blocks.

In case of KNN model, the number of nearest neighbors is separately optimized, for each profile and for each block, using another set of 500 repeated permutation samples. The optimal (K in the range $1-30$) is then found by determining the number which leads to minimum leave-one-out error.

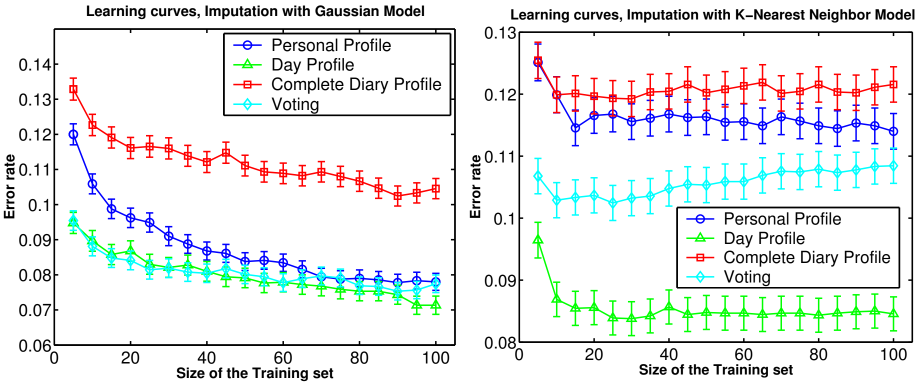
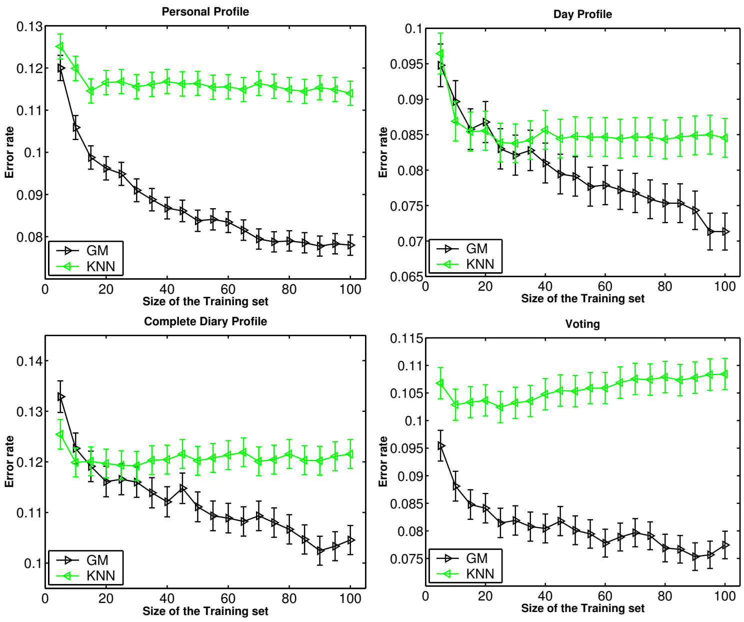


Figure 7.19 Learning curves for the Gaussian model (left plot) and the K-Nearest Neighbor model (right plot). Four different methods are presented here: Personal Profile, Day Profile, Complete Diary Profile and Voting. Error bars show standard error in 500 runs. The best results with respect to error rate are observed for Day Profile when the size of the training set is maximal. The results of the Personal Profile and Voting are similar in the Gaussian model case. The Complete Diary Profile performs significantly worst. For the K-Nearest Neighbor model the size of the training set is not so important for the level of error rate like it is in case of Gaussian model.

Figure 7.19 presents learning curves for the Gaussian model (left plot) and the K-Nearest Neighbor model (right plot), respectively. All four methods are shown, namely Personal Profile, Day Profile, Complete Diary Profile and Voting. As it was expected, Voting, gives very good results both for Gaussian and K-Nearest Neighbor model, however in both cases the Day Profile outperform the other methods. The Complete Diary Profile performs significantly worst. For the K-Nearest Neighbor model the size of the training set is not so important for the level of error rate like it is in case of Gaussian model.

The same results, but compared method-wise are presented in figure 7.20. Gaussian model performs at least as good and in many cases much better than K-Nearest Neighbor model. The difference is more significant for larger training data sets.



Personal Profile

Day Profile

Complete Diary Profile

Voting

Figure 7.20 Comparison between GM (light line) and KNN model (dark line) for all the profiles shown separately. Gaussian model performs at least as good and in many cases much better than K-Nearest Neighbor model. The difference is more significant for larger training data sets.

Figure 7.21 and 7.22 presents the performance of the imputation for each of the nine blocks separately. Every sub-figure corresponds to one question in the questionnaire. The highest error is made in imputation of the second block (question no. 2: Holiday). The error rate for this block is so significant that it basically creates the overall error rate of the validation sample. Not surprisingly, the value of this field is best predicted by Day Profile. Also therefore the observed on figure 7.19 average error rate is the smallest for Day Profile. For the rest of the blocks, imputation with Personal Profile performs the most successfully. The situation is similar for the KNN model (figure 7.22).

Table 7.17 presents error correlation matrices for Gaussian and K-Nearest Neighbor models, respectively, for three methods: Personal, Day and Complete Diary Profile. The E_{ij} entry of error correlation matrix, which was presented in [38] is defined as $E_{ij} = \mathrm{Prob}\{\text{error in method } i \wedge \text{error in method } j\}$. The error rates are shown for two extreme training set sizes, namely 5 and 100 samples.

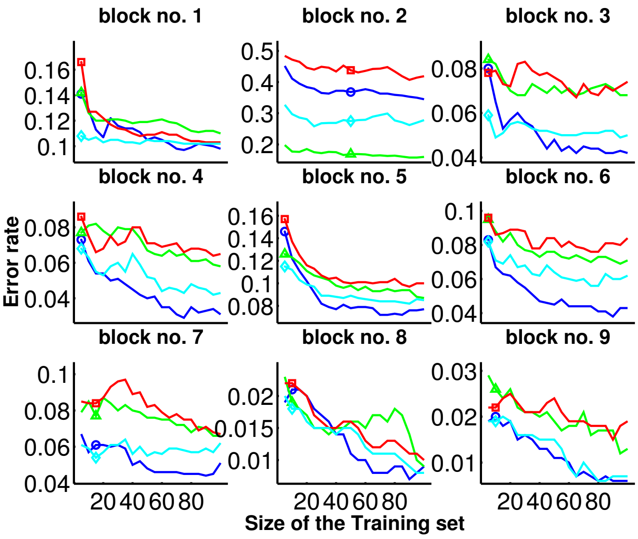


Figure 7.21 Learning curves for GM model shown separately for all 9 blocks. On x-axes the size of the training set is shown and on the y-axes the error rate. Learning curves for the block no. 2 present the highest rate. In this case, the Day Profile △ gives the best results in imputation. In the other cases, the Personal Profile ○ performs best. Voting is marked with ◊ and Complete Diary profile with □.

Ideally uncorrelated methods would return errors only on the main diagonal. In such case the Voting procedure would produce optimal results.

Figures 7.23 and 7.24 show the error rate for 30 validation samples as a function of training set size. All the methods share the same set of validation samples. It is interesting to see that for some of the validation samples, the error does not depend on which method or model is used or the size of the training set. This phenomenon is even stronger when using KNN model (for example, sample no. 2). In other cases, increased size of the training set reduces the error rate (sample no. 20 for the Gaussian model). It can also be seen that for other validation samples, the error rate varies from method to method (e.g., samples 12 and 20 in Gaussian model) and between the models (e.g., sample 2). In such cases, Voting returns the lowest error rate.

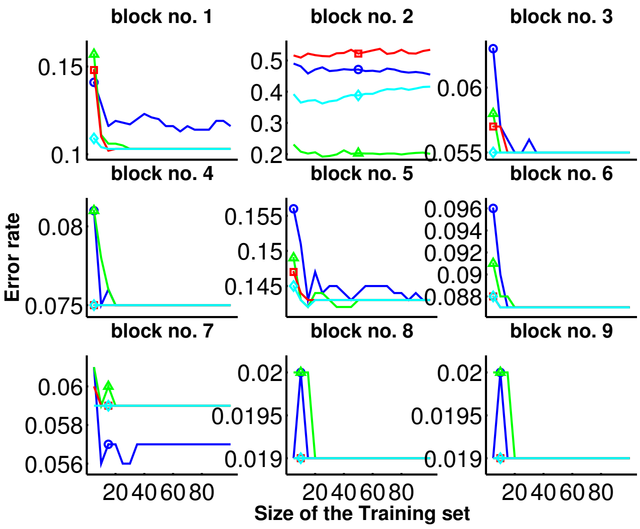


Figure 7.22 Learning curves for GM model shown separately for all 9 blocks. On x-axes the size of the training set is shown and on the y-axes the error rate. Similarly to the GM (figure 7.21), learning curves for the block no. 2 present the highest rate. In this case, the Day Profile △ gives the best results in imputation. In most of the other cases, the Voting ◊ performs best. Personal Profile is marked with ◯ and Complete Diary profile with □.

Gaussian model

5 samples				**30 samples**			
	PP	DP	CDP		PP	DP	CDP
PP	0.2481	0.0518	0.1701	PP	0.1695	0.0351	0.2096
DP	0.0518	0.1566	0.1100	DP	0.0351	0.1484	0.1705
CDP	0.1701	0.1100	0.2634	CDP	0.2096	0.1705	0.2668

K-Nearest Neighbor model

5 samples				**30 samples**			
	PP	DP	CDP		PP	DP	CDP
PP	0.1754	0.0255	0.4093	PP	0.2217	0.0405	0.2294
DP	0.0255	0.0870	0.1004	DP	0.0405	0.1359	0.1240
CDP	0.4093	0.1004	0.2024	CDP	0.2294	0.1240	0.2486

Table 7.17 Error correlation table for KNN model. Left and right tables present data for small training set (5 samples) and large training set (100 samples), respectively. Used abbreviations: PP - Personal Profile, DP - Day Profile, CDP - Complete Diary Profile. Ideally uncorrelated methods would return errors only on the main diagonal. In such case the Voting procedure would produce optimal results.

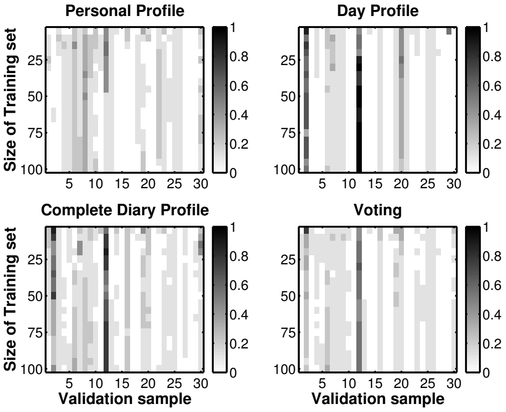


Figure 7.23 30 validation samples predicted with Gaussian models as a function of size of the training set. The error rate is an average over 9 observed blocks. 0 corresponds to no error made and 1 to the error made in each of the blocks. Increased size of the training set reduces the error rate as for example in case of sample no. 20 and for other validation samples, the error rate varies from method to method (e.g., samples 12 and 20).

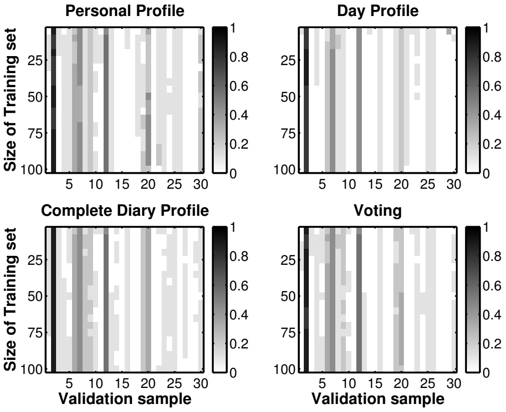


Figure 7.24 30 validation samples predicted with KNN model as a function of size of the training set. The error rate is an average over 9 observed blocks. 0 corresponds to no error made and 1 to the error made in each of the blocks. For some of the validation samples, the error does not depend on which method or model is used or the size of the training set, for example sample no. 2.

7.4 Aggregated Markov model for clustering and classification

In the experiments presented in this section the aggregated Markov model is applied. The model is described in detail in Chapter 6. Five data sets are used. The description of two of them, which were not stated earlier, is given below.

Linear structure 2-dimensional five Gaussian distributed clusters are created with the spherical covariance structure, as shown on figure 7.25 (left plot). The clusters are linearly separable. This artificially created data is used for illustration of the simple clustering problem.

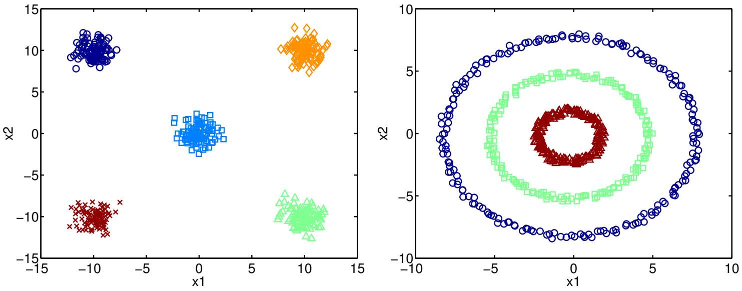


Figure 7.25 The scatter plots of the artificial data for 5 Gaussian distributed clusters (left figure) and 3 cluster ring formations (right panel).

Manifold structure Three clusters are created as shown on the right plot of figure 7.25. Clusters are formed in the shape of rings all centered at the origin with radiuses 2, 5 and 8, respectively. The data span 2 dimensions. This data is given as an example of complex nonlinear, yet separable, problem.

Additionally the performance of the algorithm is investigated on textual data: Email and Newsgroups collections and the small medical data set of six erythemato-squamous dermatological diseases. These databases are described in section 7.1.

Preprocessing

In case of continuous space collections (Gaussian and Rings clusters) data vectors are normalized with its maximum value so, they fall in the range between 0 and 1. This step is necessary whenever the features describing data points are significantly different in values or ranges. Such normalization is also performed in case of dermatological collection, in order to equalize the range of different attributes. The normalization to the unit $\mathcal{L}_2$-norm length is applied for discrete domain data sets, i.e. for Email, Newsgroup and Dermatological collection.

For Gaussian and Rings clusters the isotropic Gaussian kernel (equation 6.18) is used. With discrete data sets (Emails, Newsgroups and Dermatological collection) the cosine inner-product (equation 6.19) is applied.

The cluster structure in the data is investigated in original space, i.e. no projection is performed to the latent space and no dimensionality reduction is applied.

7.4.1 Clustering of the toy data sets

Linear structure

The Gaussian clusters example is a simple separation problem. The model is trained using 500 randomly generated samples and for generalization error 2500 validation samples are selected. The aggregated Markov model as, a probabilistic framework, allows the new data points, not included in the training set, to be uniquely mapped in the model. Therefore, it is possible to compute the generalization error and based on it to determine the model parameters: the kernel width h for the continuous low dimensional data or K in K-connected graph for discrete high dimensional values and the optimum number of classes c.

In 20 experiments, different training sets are generated, so the final error is an average over 20 outcomes of the algorithm on the same validation set. The right plot of figure 7.26 presents the dependency of the generalization error as a function of the kernel smoothing parameter h. For $h = 0.06$ the error is minimal. For this particular h the model complexity is then investigated (left plot of figure 7.26). Here, the minimal error is given for all 2, 3, 4 and 5 clusters. It can be shown, that the generalization error of the discussed model, in case of fully separable examples, is identical for the number of clusters lower or

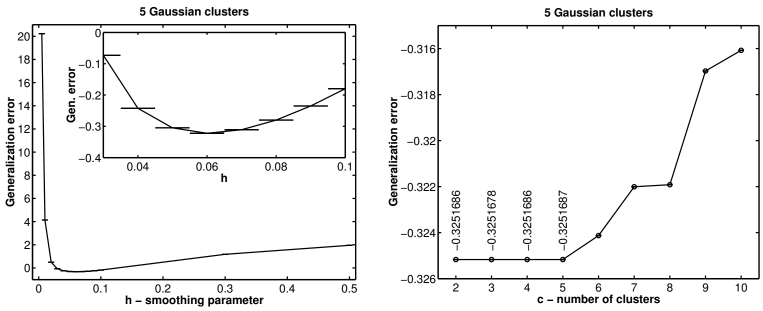


Figure 7.26 On both figures the generalization error is presented as a function of the kernel smoothing parameter h (left panel) for 5 Gaussian distributed clusters. The optimum choice is $h = 0.06$. The right figure presents, for optimum smoothing parameter, the generalization error as a function of number of clusters. Here, any cluster number below or equal 5 may give the minimum error for which the error values are shown above the points. The optimum choice is a maximum model, i.e., $c = 5$ (see the explanation of this choice in the text). The error bars show the standard error values.

equal the correct number. When perfect[19] cluster posterior probability $p(c|\mathbf{z}_i)$ is observed, the sample probability $p(\mathbf{z}_i)$ is the same for both smaller and larger models. It is true, as long as the natural cluster separations are not split, i.e. as long as the sample has large (close to 1) probability of belonging to one of the clusters $\underset{c}{max}\ p(c|\mathbf{z}_i) \approx 1$.

For Gaussian clusters the cluster posterior $p(c|\mathbf{z})$ is presented on figure 7.27. Naturally, the values are in the 0–1 range. Perfect decision surfaces can be observed. The probabilistic framework simplifies determination of the decision surfaces. For comparison, on figure 7.28, the components of the traditional kernel PCA are presented. Here, both the positive and the negative values are observed. That makes it difficult to determine the optimum decision surface.

Manifold structure

In case of Rings structure the clustering problem is not any longer simple. The clusters are, however, separable but a nonlinear decision boundary is needed. In the training, 600 examples are used and for generalization 3000 validation samples are generated. There are performed 40 experiments, where the different

[19] 0/1 valued

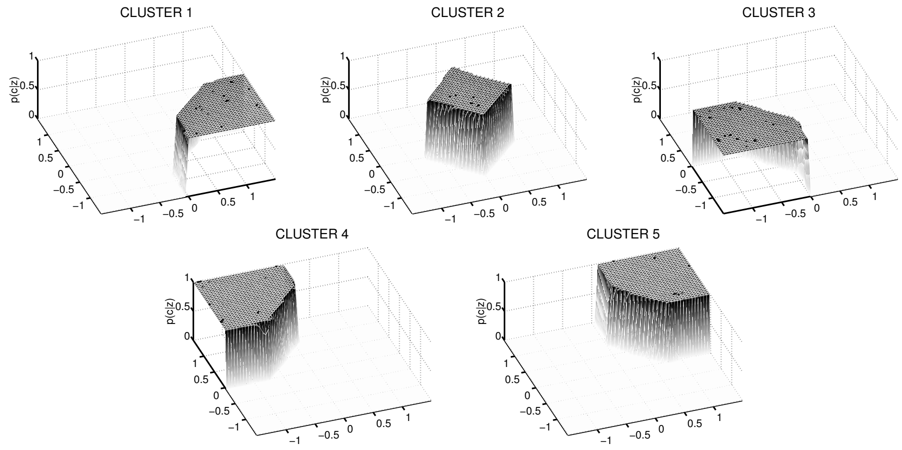


Figure 7.27 The cluster posterior values $p(c|z)$ obtained from the aggregate Markov model for five Gaussian clusters. The decision surfaces are nonnegative. In case of this simple example the separation is perfect, where 0/1 class posterior probability values are observed. It is simple to form the decision surfaces in case of the probabilistic outcome.

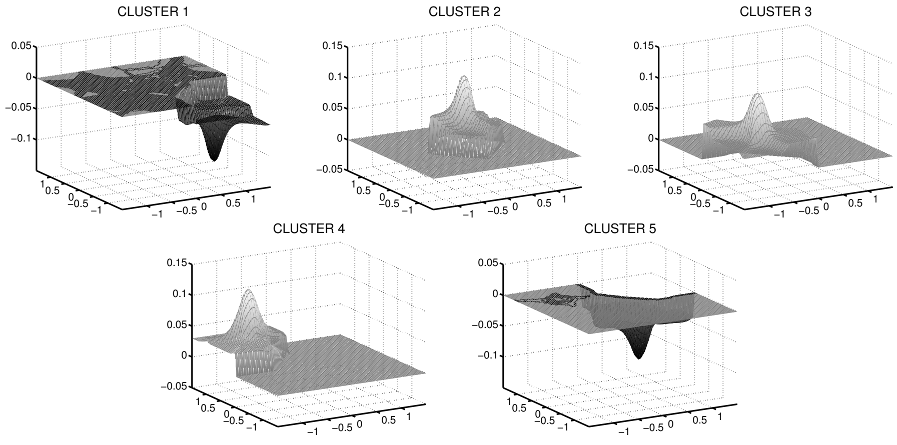


Figure 7.28 The components of the traditional kernel PCA model for five Gaussian clusters. The values are both positive and negative. That makes the determination of the decision surface more ambiguous.

training sets are selected. The generalization error, shown on figures 7.29, is an average over errors obtained in each of the 40 runs on the same validation set.

The optimum smoothing parameter (figure 7.29, left plot) equals $h = 0.065$

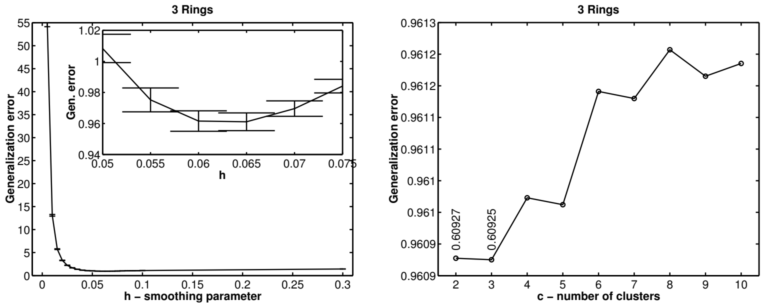


Figure 7.29 The generalization error as a function of kernel smoothing parameter h for 3 clusters formed in the shape of rings (left panel). The optimum choice is $h = 0.065$. On the right figure the generalization error as a function of number of classes is shown for the optimum choice of smoothing parameter. The error bars show the standard error. 2 and 3 clusters provide the minimal error values. As before the maximal model of 3 clusters is selected. The explanation of such selection can be found on page 116.

and the minimum generalization error is obtained for 3 classes. Similarly to the Gaussian clusters example, a smaller model of 2 classes is also probable.[20]

The class posterior for Rings data set and the kernel PCA components are presented on figures 7.30 and 7.31, respectively. Also in this case perfect (0/1)

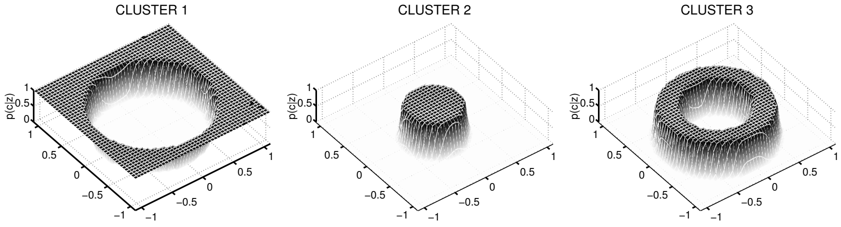


Figure 7.30 The cluster posterior values $p(c|\mathbf{z})$ obtained from the aggregate Markov model for the Rings structure of three clusters. The decision surfaces are nonnegative. The separation is perfect since the class posterior probability values are 0/1. It is easy to determine the decision surfaces.

cluster posterior is observed (figure 7.30), which is the outcome of the aggre-

[20] The generalization error is similar for both 2 and 3 numbers of classes.

gate Markov model.

The components of kernel PCA (figure 7.31) provide, as in case of cluster posterior, the separation but with more ambiguity in selection of the decision surface.

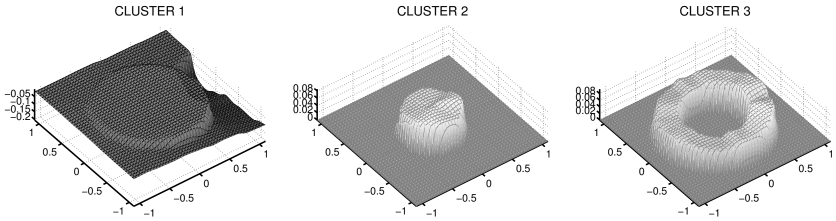


Figure 7.31 The components of the traditional kernel PCA model for Rings structure of three clusters. The components are both positive and negative. It is difficult to determine the appropriate decision surface.

7.4.2 Clustering of the textual data collections

The generalization error for Email collection is shown on figure 7.32. The mean

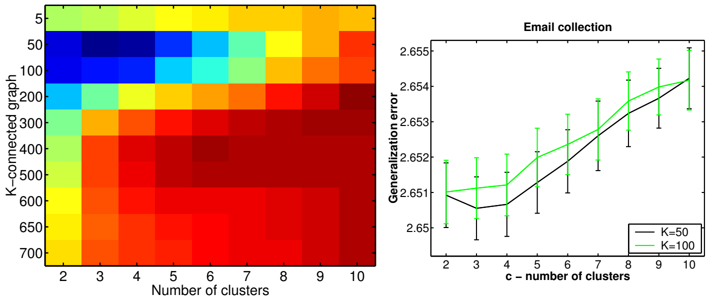
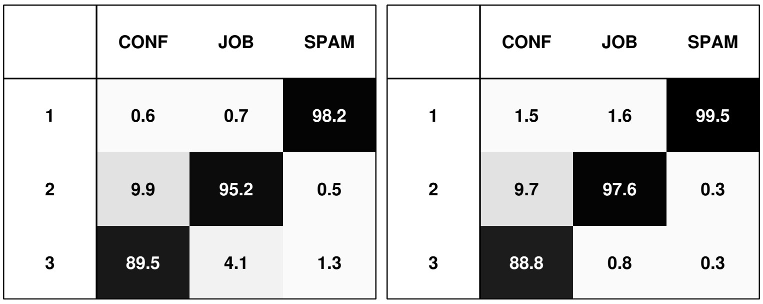


Figure 7.32 Left panel presents the mean generalization error as a function of both the class number and the K numbers of active neighbors in the K-connected graph for Emails collection. Since the differences around the minimum are small, the additional plot (left figure) is provided, which shows the minimal curves. Thus, as the optimal model $K = 50$ (50-connected graph) with 3 clusters is chosen.

values are presented averaged from 20 random choices of the training and the test set. For training 702 samples are reserved and the rest of 703 are used in calculation of the generalization error. Since, the used kernel is the cosine inner-product, the K-connected graph is applied to determine the number of active neighbors in the Gram matrix. For Email collection, the minimal generalization error is obtained when using 50-connected graph with the model complexity of 3 clusters. Since the data is not naturally separated, there exist small confusion among the clusters. The smaller models are not favored as it was in case of Rings and Gaussian data sets.



	CONF	JOB	SPAM
1	0.6	0.7	98.2
2	9.9	95.2	0.5
3	89.5	4.1	1.3

	CONF	JOB	SPAM
1	1.5	1.6	99.5
2	9.7	97.6	0.3
3	88.8	0.8	0.3

Figure 7.33 As interpretation the confusion matrix of the training (left plot) and of the test set (right plot) is provided. The separation is almost perfect, only small confusion especially between *conference* and *job* emails is observed.

Figure 7.33 presents the confusion matrices of the training (left plot) and the test set (right plot). The figures provide the supervised interpretation of the found clusters. Thus, *spam* emails fall into cluster 1 in almost 100%. Cluster number 2 covers *job* emails and cluster number 3 *conference* emails. There is slight confusion between cluster 2 and cluster 3. It can be expected, since both *job* and *conference* emails are university related. The results can be compared with similar experiment, as in outcome of UGGM model on figure 7.5.

The generalization error for Newsgroups collection is shown on figure 7.34. For the training, 400 samples is used randomly selected from the set and the rest of the collection is designated for calculation of the generalization error. 40 experiments was performed and figure 7.34 displays the mean value of the generalization error. For clarity, since the differences at the minimum are small, four selected generalization error curves with the standard error on the error bars are shown on the right plot of figure 7.34. The optimum model has 4 classes

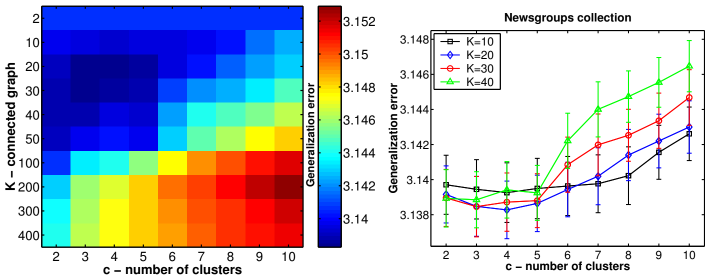


Figure 7.34 Left panel presents the mean generalization error as a function of both the cluster number c and K which is the number of active neighbors in the K-connected graph for Newsgroups collection. Since the differences at the minimum are small, it is difficult to see the placement of the minimum. Therefore, the curves for selected K (10 20 30 40) are provided on the right plot. The optimal model complexity is 4 clusters when using 20-connected graph.

where the connectivity among 20 closest neighbors are remained in the Gram matrix.

Figure 7.35 presents the confusion matrices of the training (left plot) and the test set (right plot).

	Comp.	Motor	Baseb.	Christ.
1	14.0	98.0	7.9	7.1
2	3.5	0.0	0.0	85.9
3	78.9	1.0	2.2	2.0
4	3.5	1.0	89.9	5.1

	Comp.	Motor	Baseb.	Christ.
1	12.9	95.1	6.4	4.0
2	1.2	2.0	0.9	91.1
3	83.5	1.0	3.6	5.0
4	2.4	2.0	89.1	0.0

Figure 7.35 The confusion matrix of the training (left plot) and the test set (right plot). There is a small confusion among the clusters. The data is well separated. Small confusion among the clusters is however observed.

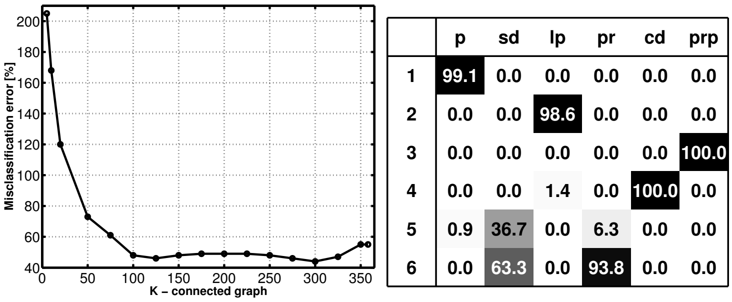
The figures provide the supervised interpretation of the discovered clusters. Thus, for example the *Motorcycles* newsgroups documents are mostly covered by cluster 1. Cluster 2 contains *Christian Religion*, cluster 3 *Computer Graphics* and cluster 4 *Baseball*. There is small confusion among the discovered clusters. The results can be compared with the similar outcome of the UGGM model on figure 7.9.

In order to compare the aggregated Markov model with the classical spectral clustering method, presented in [62], the following experiments are performed. For both, continuous and discrete data sets, using both the Gaussian kernel and inner-product the investigation of the overall performance in classification, measured by the miss-classification error, is made. It is found, that both the aggregated Markov model and the spectral clustering model for selected model parameters does comparably well in the sense of miss-classification error. However, the spectral clustering model is less sensitive to the choice of smoothing parameter h. Note, that the spectral clustering method does not provide the objective technique for selecting the optimal parameters. Thus, as the optimum, the parameters providing the minimum misclassification error are selected.

7.4.3 Classification of the medical data collections

The unsupervised classification task is performed on the dermatological collection. The data, described in detail in section 7.1 consists of 6 classes of erythemato-squamous diseases, namely *psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, pityriasis rubra pilaris*. In the experiments 358 examples are used that are described by 34 attributes. The minimum in the calculated generalization error is indicating two class structure, which is the minimal investigated complexity. In conclusion, the data set is too small to allow correct estimation of the model parameters. Therefore, the 6 class structure is optimized directly and the misclassification error is considered in determining the optimum number of connected neighbors. While the model complexity is determined in supervised way, the learning itself is performed in unsupervised manner.

The misclassification error and the confusion matrix is presented on figure 7.36 As the optimum number of active neighbors any number above $K = 100$ can be selected. For decomposition purposes $K = 300$ is chosen. Then the Gram matrix is decomposed, resulting with a data separation of 6 clusters. For interpretation purposes the confusion matrix is provided. Here, it can be seen, that there is a large confusion, between two clusters, namely *seboreic dermatitis* and *pityriasis rosea*. The rest of the clusters are separated.



	p	sd	lp	pr	cd	prp
1	99.1	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	98.6	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	100.0
4	0.0	0.0	1.4	0.0	100.0	0.0
5	0.9	36.7	0.0	6.3	0.0	0.0
6	0.0	63.3	0.0	93.8	0.0	0.0

Figure 7.36 Left panel presents the misclassification error as a function of the number of neighbors in K–connected graph for Dermatological collection. $K = 300$ is selected. The right plot presents the confusion matrix obtained from the model, where unsupervised learning is performed in 6 class structure. The classes are marked with abbreviations form original erythemato-squamous diseases. Almost perfect classification can be seen except of two classes of seboreic dermatitis and pityriasis rosea which are confused.

CHAPTER 8

Conclusion

The focus of this thesis has been in examining various principled approaches to data mining in order to discover inherent latent structure within the data and to provide and provide an intuitive interpretation of any such structure. The analysis was performed on medium size databases which came from disparate sources and were heterogeneous in nature.

The general framework, known as the Knowledge Discovery in Databases process was applied. The investigated methods include techniques for projection, clustering, structure interpretation, outlier detection and imputation missing values. In the first phase, projection techniques were investigated with the main focus on Principal Component Analysis. Though Random Projection and Non-negative Matrix Factorization were also examined as possible techniques for dimensionality reduction. The Random Projection method did not provide satisfactory results in the case of complex data types such as sparse textual data and when large dimensionality reduction is required. This is the case of Generalizable Gaussian Mixture models, which require a small number of dimensions in order to estimate correctly the parameters of the data density. Any increase in the dimensionality increases significantly the number of parameters to estimate which then requires more data points for an sensible estimate. The Non-negative Matrix Factorization provided the matrix decomposition which was subsequently applied to clustering in conjunction with the aggregated Markov model.

For clustering purposes, the unsupervised and unsupervised/supervised Generalizable Gaussian Mixture models were investigated and applied later on the observational data sets. These models return multi-modal density estimate which provides the cluster structure inherent in the data. In case of unsupervised/supervised GGM model the influence was investigated of the unlabeled samples used in learning process. It was found that unlabeled examples are important for the estimation whenever the labeled data set is small. This importance deteriorates as the size of labeled set increases.

Another important issue addressed in this thesis was the outlier detection methods. In the KDD process, outliers can have a dual role. They can be considered as noise. In which case, its removal improves the model. Alternatively, the outliers are the subject of interest and require careful analysis. In this work two outlier detection methods were compared. The first method was based on the cumulative distribution where the low probability samples were classified as outliers. The second method was based on the Generalizable Gaussian mixture model with one cluster was designated to collect outliers. The method based on cumulative distribution require manual tuning of the rejection threshold. The outlier cluster technique is fully automatic and provides better outcome whenever the data density is correct estimated.

The agglomerative hierarchical clustering was the next research area. In this work, hierarchical clustering was an extension to the aforementioned Gaussian Mixture model. Several similarity measures were applied. However, no method significantly outperformed the other, despite their noted drawbacks. For instances, only approximate formula is available for higher hierarchy levels with Kullback-Leibler similarity measure. The Cluster Confusion similarity measure is computational expensive, since it requires many ancillary data points to obtain a good dissimilarity estimate especially when there is small cluster overlap. The agglomerative hierarchical clustering can be employed as an additional clustering level. This was found to be most appropriate when visualizing the the data in order to present a multi-level understanding of the inherent clustered nature of the data.

Being able to intuitively understand the results is another important part of the KDD process. For labeled data sets, the confusion matrix between the class and the cluster structure was provided which enabled the interpretation of the discovered clusters in terms of known labels. Furthermore, the next proposed method for interpretation was to generate the meaningful representatives of clusters. In the case of textual databases keywords were provided and for other forms of data the model important features were selected.

An attempt was made to combine different data sources and various types of the data, in the clustering framework. It was shown the in case of the investigated sun-exposure collection, that clustering benefited from the additional sources of information. However, no objective measure of importance for the data sources was developed.

The issue of handling missing data was also addressed in this thesis. Similar to outliers, the missing data may be handled in two ways. First, if assumed as noise contribution, they can be removed from the data set as has been done in the majority of the performed experiments. In can be applies whenever the missingness process is believed to be completely at random. Second, the missing data can itself be the subject of the data mining task and as such imputation techniques can be applied. Two methods were collated in this work, namely K-Nearest Neighbors and Gaussian model. Even though the experiments were carried out on the binary data set of the sun-exposure study, the results obtained by the Gaussian model were found to be superior.

As an alternative to the Gaussian Mixture model in data segmentation the aggregated Markov model was proposed. This technique, originating from spectral clustering methods, performs the decomposition of the Gram matrix in a fully probabilistic framework. It provides the estimates of the class posterior probabilities of the data points and allows new data points to be assigned in the framework. Therefore cross-validation can be applied, to select the model parameters, which was not possible with classical spectral clustering techniques. Moreover, the data with the large dimensional feature space are easily handled by this method. Hence, no projection techniques are needed, even though they can be applied.

Future work

When working with the disparate and heterogeneous data it would be interesting to develop an objective measure to determine if the combination of sources provided more information than the individual sources.

It would be also interesting to develop a general model where both continuous and discrete data are combined in unsupervised and unsupervised/supervised clustering framework.

As a future research task also more complex imputation methods like, for example maximum likelihood and multiple imputation techniques can be interesting to develop.

APPENDIX A

Equations

Jensen's inequality:

If f is a convex function such that its hessian is positive semi-definite $H \geq 0$ for all $x \in \mathbb{R}$, then for random variable X holds

$$E[f(X)] \geq f(EX) \tag{A.1}$$

Minkowski's inequality:

If $p > 1$, then Minkowski's integral inequality states that

$$\Big[\int_a^b |f(x)+g(x)|^p dx\Big]^{\frac{1}{p}} \leq \Big[\int_a^b |f(x)|^p dx\Big]^{\frac{1}{p}} + \Big[\int_a^b |g(x)|^p dx\Big]^{\frac{1}{p}} \tag{A.2}$$

Similarly, if $p > 1$ and $a_k, b_k > 0$, then Minkowski's sum inequality states that

$$\Big[\sum_{k=1}^{n}(a_k+b_k)^p\Big]^{\frac{1}{p}} \leq \Big[\sum_{k=1}^{n} a_k^p\Big]^{\frac{1}{p}} + \Big[\sum_{k=1}^{n} b_k^p\Big]^{\frac{1}{p}} \tag{A.3}$$

Equality holds iff the sequences $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$ are proportional.

Kullback-Leibler divergence:

Kullback-Leibler divergence [69] is defined as follow:

$$\mathcal{D}iv(k,l) = \int p(\mathbf{x}|k) ln \frac{p(\mathbf{x}|k)}{p(\mathbf{x}|l)}. \tag{A.4}$$

A symmetric version is defined as $\mathcal{D}(k,l) = \frac{1}{2}(\mathcal{D}iv(k,l) + \mathcal{D}iv(l,k))$. Thus,

$$\mathcal{D}(k,l) = \frac{1}{2}\int p(\mathbf{x}|k) ln \frac{p(\mathbf{x}|k)}{p(\mathbf{x}|l)} + \frac{1}{2}\int p(\mathbf{x}|l) ln \frac{p(\mathbf{x}|l)}{p(\mathbf{x}|k)}. \tag{A.5}$$

where $\mathcal{D}(k,l) \geq 0$ and $\mathcal{D}(k,l) = \mathcal{D}(l,k)$.

If the density functions $p(\mathbf{x}|k)$ and $p(\mathbf{x}|l)$ are Gaussians,

$$p(\mathbf{x}|k) = (2\pi)^{-\frac{D}{2}}|\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_k)^T\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k)), \tag{A.6}$$

and

$$p(\mathbf{x}|l) = (2\pi)^{-\frac{D}{2}}|\mathbf{\Sigma}_l|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_l)^T\mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l)), \tag{A.7}$$

where D is dimension, then the KL divergence may be simplified in the following way.

$$\begin{aligned} ln\frac{p(\mathbf{x}|k)}{p(\mathbf{x}|l)} &= ln\frac{(2\pi)^{-\frac{D}{2}}|\mathbf{\Sigma}_l|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_k)^T\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k))}{(2\pi)^{-\frac{D}{2}}|\mathbf{\Sigma}_l|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_l)^T\mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l))} \\ &= ln\frac{|\mathbf{\Sigma}_l|^{\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_k)^T\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k))}{|\mathbf{\Sigma}_k|^{\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_l)^T\mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l))} \\ &= \frac{1}{2}ln(|\mathbf{\Sigma}_l|) - \frac{1}{2}ln(|\mathbf{\Sigma}_k|) - \frac{1}{2}(\mathbf{x}-\mu_k)^T\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k)) \\ &\quad +\frac{1}{2}(\mathbf{x}-\mu_l)^T\mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l) \end{aligned} \tag{A.8}$$

$$(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) = \text{Trace}(\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T). \tag{A.9}$$

and due to linearity of the integral:

$$\int \text{Trace}(af(x))g(x)dx = \text{Trace}(a\int f(x)g(x)dx) \tag{A.10}$$

$$\begin{aligned}\int p(\mathbf{x}|k) ln \frac{p(\mathbf{x}|k)}{p(\mathbf{x}|l)} d\mathbf{x} &= \frac{1}{2}\int p(\mathbf{x}|k) ln(|\mathbf{\Sigma}_l|) d\mathbf{x} - \frac{1}{2}\int p(\mathbf{x}|k) ln(|\mathbf{\Sigma}_k|) d\mathbf{x} \\ &- \frac{1}{2}\int p(\mathbf{x}|k)(\mathbf{x}-\mu_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k) d\mathbf{x} \\ &+ \frac{1}{2}\int p(\mathbf{x}|k)(\mathbf{x}-\mu_l)^T \mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l) d\mathbf{x} \qquad \text{(A.11)}\end{aligned}$$

$$\begin{aligned}-\frac{1}{2} \quad \int \quad & p(\mathbf{x}|k)(\mathbf{x}-\mu_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k) d\mathbf{x} = \\ &= -\int \frac{1}{2}\text{Trace}(\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mu_k)(\mathbf{x}-\mu_k)^T) p(\mathbf{x}|k) d\mathbf{x} \\ &= -\frac{1}{2}\text{Trace}(\mathbf{\Sigma}_k^{-1}\int(\mathbf{x}-\mu_k)(\mathbf{x}-\mu_k)^T p(\mathbf{x}|k) d\mathbf{x}) \\ &= -\frac{1}{2}\text{Trace}(\mathbf{\Sigma}_k^{-1}\mathbf{\Sigma}_k) \\ &= -\frac{D}{2} \qquad \text{(A.12)}\end{aligned}$$

$$\begin{aligned}\tfrac{1}{2}\int \quad & p(\mathbf{x}|k)(\mathbf{x}-\mu_l)^T \mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l) d\mathbf{x} = \\ = \quad & \int \frac{1}{2}\text{Trace}(\mathbf{\Sigma}_l^{-1}(\mathbf{x}-\mu_l)(\mathbf{x}-\mu_l)^T) p(\mathbf{x}|k) d\mathbf{x} \\ = \quad & \frac{1}{2}\text{Trace}(\mathbf{\Sigma}_l^{-1}\int(\mathbf{x}-\mu_l)(\mathbf{x}-\mu_l)^T p(\mathbf{x}|k) d\mathbf{x}) \\ = \quad & \frac{1}{2}\text{Trace}\Big(\mathbf{\Sigma}_l^{-1}\int p(\mathbf{x}|k)\big((\mathbf{x}-\mu_k)(\mu_k-\mu_l)\big)\big((\mathbf{x}-\mu_k)(\mu_k-\mu_l)\big)^T d\mathbf{x}\Big) \\ = \quad & \frac{1}{2}\text{Trace}\Big(\mathbf{\Sigma}_l^{-1}\int p(\mathbf{x}|k)\big((\mathbf{x}-\mu_k)(\mathbf{x}-\mu_k)^T + (\mathbf{x}-\mu_k)(\mu_k-\mu_l)^T + \\ & \qquad (\mu_k-\mu_l)(\mathbf{x}-\mu_k)^T + (\mu_k-\mu_l)(\mu_k-\mu_l)^T\big) d\mathbf{x}\Big) \\ = \quad & \frac{1}{2}\text{Trace}\Big(\mathbf{\Sigma}_l^{-1}\big(\mathbf{\Sigma}_k + \mathbf{0} + \mathbf{0} + (\mu_k-\mu_l)(\mu_k-\mu_l)^T\big)\Big) \\ = \quad & \frac{1}{2}\text{Trace}(\mathbf{\Sigma}_l^{-1}\mathbf{\Sigma}_k) + \frac{1}{2}(\mu_k-\mu_l)\mathbf{\Sigma}_l^{-1}(\mu_k-\mu_l)^T \qquad \text{(A.13)}\end{aligned}$$

$$\begin{aligned}\int p(\mathbf{x}|k) ln \frac{p(\mathbf{x}|k)}{p(\mathbf{x}|l)} d\mathbf{x} &= \frac{1}{2} ln(|\mathbf{\Sigma}_l|) - \frac{1}{2} ln(|\mathbf{\Sigma}_k|) - \frac{D}{2} + \frac{1}{2}\text{Trace}(\mathbf{\Sigma}_l^{-1}\mathbf{\Sigma}_k) \\ &+ \frac{1}{2}(\mu_k - \mu_l)\mathbf{\Sigma}_l^{-1}(\mu_k - \mu_l)^T \qquad \text{(A.14)}\end{aligned}$$

$$\begin{aligned}\int p(\mathbf{x}|l) ln \frac{p(\mathbf{x}|l)}{p(\mathbf{x}|k)} d\mathbf{x} &= \frac{1}{2} ln(|\mathbf{\Sigma}_k|) - \frac{1}{2} ln(|\mathbf{\Sigma}_l|) - \frac{D}{2} + \frac{1}{2}\text{Trace}(\mathbf{\Sigma}_k^{-1}\mathbf{\Sigma}_l) \\ &+ \frac{1}{2}(\mu_l - \mu_k)\mathbf{\Sigma}_k^{-1}(\mu_l - \mu_k)^T \qquad \text{(A.15)}\end{aligned}$$

$$\begin{aligned}\mathcal{D}(k,l) &= \frac{1}{2}\int p(\mathbf{x}|k) ln \frac{p(\mathbf{x}|k)}{p(\mathbf{x}|l)} + \frac{1}{2}\int p(\mathbf{x}|l) ln \frac{p(\mathbf{x}|l)}{p(\mathbf{x}|k)} \\ & -\frac{D}{2} + \frac{1}{4}(\text{Trace}(\mathbf{\Sigma}_k^{-1}\mathbf{\Sigma}_l) + \text{Trace}(\mathbf{\Sigma}_l^{-1}\mathbf{\Sigma}_k)) \\ &+ \frac{1}{4}(\mu_k - \mu_l)^T(\mathbf{\Sigma}_k^{-1} + \mathbf{\Sigma}_l^{-1})(\mu_k - \mu_l) \qquad \text{(A.16)}\end{aligned}$$

APPENDIX B

Contribution: NNSP 2001

This appendix contain the article *Imputating missing values in diary records of sun-exposure study.* in Proceedings of IEEE Workshop on Neural Networks for Signal Processing XI, editor: D. Miller, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, pages: 489–498. Author list: A. Szymkowiak, P.A. Philipsen, J. Larsen, L.K. Hansen, E. Thieden and H.C. Wulf. Own contribution estimated to 50%.

IMPUTATING MISSING VALUES IN DIARY RECORDS OF SUN-EXPOSURE STUDY

A. Szymkowiak[1], P.A. Philipsen[2], J. Larsen[1], L.K. Hansen[1],
E. Thieden[2], H.C. Wulf[2]

[1]Informatics and Mathematical Modelling, Building 321
Technical University of Denmark, DK-2800 Lyngby, Denmark
Phone: +45 4525 3900,3923,3889
Fax: +45 4587 2599
E-mail: asz,jl,lkh@imm.dtu.dk
Web: eivind.imm.dtu.dk

[2]Department of Dermatology, Bispebjerg Hospital
University of Copenhagen, Bispebjerg Bakke 23
DK-2400 Copenhagen, Denmark

Abstract. In a sun-exposure study, questionnaires concerning sun-habits were collected from 195 subjects. This paper focuses on the general problem of missing data values which occurs when some, or even all, the questions have not been answered in a questionnaire. Here only missing values of low concentration are investigated. We consider and compare two different models for imputating missing values: the Gaussian model and the non-parametric K-Nearest Neighbor model.

INTRODUCTION

The missing data problem occurs in virtually any application of statistics to real life problems. It is particularly important whenever statistical analysis is based on human responses. The severeness of missing data is aggravated if probability of data drop-out is a function of the missing value. Attempts to fill in missing data ranges from complex monte carlo procedures, like multiple imputation [5], over EM-based, deterministic, yet iterative, procedures [1, 2, 6, 7], to basic statistical methods based on simple multivariate parametric, typically Gaussian, density approximations [4].

In the sun-exposure experiment studied, questionnaires concerning sun-habits were collected from 195 subjects (the group of people involved in the experiment lasting 138 days). In addition, UV radiation were measured at a 10 minute sampling rate. While the ultimate objective is to relate sun-habits, UV dose, and risk of cancer, this work focuses on imputating missing questionnaire values. We present the analysis of two basic missing value approaches based on parametric and non-parametric representations, respectively. Rather than invoking complex statistical methods we concentrate on evaluating the two schemes using a modern learning theory tool, the "learning curve", which in the present context quantifies the fill-in error as function of training sample size. Such analysis is important for experimental design. Secondly, we investigate the utility of voting schemes for enhancing the performance of missing data mechanisms.

DESCRIPTION OF THE DIARY DATA

In the experiment two types of data was collected. The subjects wore a special designed watch called the "Sunsaver", which measures UVA and UVB radiation. In addition, the following questionnaire was also returned:

1. Using Sunsaver (yes/no)
2. Working (yes/no)
3. Abroad (yes/no)
4. Sun Bathing (yes/yes-solarium/no)
5. Naked Shoulders (yes/no)
6. On the Beach/On the water (yes/no)
7. Using Sun Screen (yes/no)
8. Sun Factor Number (no/1-7/8-16/17-35/>35)
9. Sunburned (no/red/hurts/blisters)
10. Size of Sunburn Area (no/little/medium/large)

Each questionnaire was stored along with date and subject identification number. Some of the answers are binary (yes/no) whereas others are coded using a 1-out-of-c binary representation. The 1-out-of-c coding ensures that the Hamming distance between any two data vectors equals one, thus preventing an arbitrary distance for categorical data such as Sunburned.

The sun factor number (question no. 8) has a larger range of values. In order to decrease the length of its binary representation, it was quantized into five levels (no/small (1-7)/medium (8-16)/large (17-35)/huge (>35)). Furthermore, it was combined with question no. 7 creating one binary vector block.

Eventually, for every person and every day, a 17-dimensional binary vector is created. It contains nine blocks from one to four bits each. There are 24212 data records in the diary, distributed among 195 persons and 138 days. There are at least one missing value in more than 1000 vectors due to partially unfilled questionnaires (i.e., in approx. 4% of the questionnaires).

MISSING DATA MODELS

The d-dimensional binary feature vector is defined as $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$. The data set is denoted as $\mathcal{D} = \{\boldsymbol{x}^{(n)}; n = 1, 2, \ldots, N\}$, where N is the number of examples.

Two models for filling in missing data are described here. The first method is based on the assumption that the diary data vectors are Gaussian distributed. The second is a non-parametric K-Nearest Neighbor model. Many different models can be proposed. This paper, however, focuses on comparing a the complicated stochastic model with a simpler non-parametric one for specific diary records.

Due to the characteristics of data, there are three different profiles taken into consideration. The first called the *Complete Diary Profile* uses the full data set in the estimation. The second called the *Personal Profile* assumes that questionnaires from one person have similar characteristics while the characteristics across the persons differ. This arise from the expectation that human behaviour varies from person to person. The third profile is called the *Day Profile* and assumes that data vectors for one day are similar or equivalently belong to one distribution, while parameters of the distributions across the days vary. This is due to the fact that human behavior is influenced by whether, temperature, the season of the year, etc. The model using each of the described profiles is called a method.

In addition, a *Voting* procedure is also considered. It compares proposals from all the above mentioned methods and takes the majority vote among the outcomes. This method is expected to give the best results, however, it is much more computationally expensive since it combines the other three methods.

Gaussian Model (GM)

Assume that $\boldsymbol{x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Further that the feature vector is divided into observed and missing parts, as $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$. Under the Gaussian model assumption, the optimal inference of the missing part is given as the condition expectation of the missing part given the observed part, i.e.,

$$E(\boldsymbol{x}_m | \boldsymbol{x}_o) = \boldsymbol{\mu}_m + \boldsymbol{\Sigma}_{mo} \boldsymbol{\Sigma}_{oo}^{-1} \cdot (\boldsymbol{x}_o - \boldsymbol{\mu}_o) \tag{1}$$

where

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_o, \boldsymbol{\mu}_m] \quad \text{and} \quad \boldsymbol{\Sigma} = \left[\begin{array}{cc} \boldsymbol{\Sigma}_{oo} & \boldsymbol{\Sigma}_{om} \\ \boldsymbol{\Sigma}_{om}^{\top} & \boldsymbol{\Sigma}_{mm} \end{array} \right] \tag{2}$$

The Gaussian imputation model is then given as:

GM Algorithm:

1. Divide the data set $\mathcal{D}$ into two parts. Let the first set contain data vectors in which at least one of the features is missing, call it $\mathcal{D}_m$. Then the remaining part, where all the vectors are complete is called $\mathcal{D}_c$.

2. Estimate mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ from $\mathcal{D}_c$, i.e.,

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N_c} \sum_{n \in \mathcal{D}_c} \boldsymbol{x}^{(n)}, \quad \widehat{\boldsymbol{\Sigma}} = \frac{1}{N_c - 1} \sum_{n \in \mathcal{D}_c} \left(\boldsymbol{x}^{(n)} - \widehat{\boldsymbol{\mu}}\right) \left(\boldsymbol{x}^{(n)} - \widehat{\boldsymbol{\mu}}\right)^{\top}, \quad (3)$$

   where $N_c = |\mathcal{D}_c|$ is the number of complete examples.

3. For each vector $\boldsymbol{x} \in \mathcal{D}_m$

   - Divide the vector into two parts $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$, where $\boldsymbol{x}_o$ is the observed vector features and $\boldsymbol{x}_m$ the missing.
   - Estimate the missing binary vector as the sign of the conditional-distribution mean for the missing part given the known features:

$$\widehat{\boldsymbol{x}}_m = \operatorname{sign}\left[\widehat{\boldsymbol{\mu}}_m + \widehat{\boldsymbol{\Sigma}}_{mo}\widehat{\boldsymbol{\Sigma}}_{oo}^{-1} \cdot (\boldsymbol{x}_o - \widehat{\boldsymbol{\mu}}_o)\right]$$

K-Nearest Neighbor Model (KNN)

The distance measure for binary vectors (Hamming distance) is defined as follows:

$$D(p,q) = \sum_{i=1}^{d} |x_i^{(p)} - x_i^{(q)}|, \quad (4)$$

where p and q are two binary vectors and i is a bit (dimension) index.

The algorithm for the non-parametric K-Nearest Neighbor Model is given as:

KNN Algorithm:

1. Divide the data set $\mathcal{D}$ into two parts. Let the first set contain data vectors in which at least one of the features is missing, $\mathcal{D}_m$. The remaining part where all the vectors are complete is called $\mathcal{D}_c$.

2. For each vector $\boldsymbol{x} \in \mathcal{D}_m$:

   - Divide the vector into observed and missing parts as $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$.
   - Calculate the distance Eq. (4) between the $\boldsymbol{x}_o$ and all of the vectors from the set $\mathcal{D}_c$. Substitute by non-missing parts in the vectors from the complete set $\mathcal{D}_c$.
   - Use the K closest vectors (K-nearest neighbors) and perform a majority voting estimate of the missing values.

EXPERIMENTS

In order to compare the performance of the models on the diary records, a validation set was taken out from the fully completed questionnaires. We perform a leave-one-out permutation estimate of the generalization error as in 1000 repeated permutations one validation sample is chosen randomly, then a number of training samples. The performance is then average over the 1000 permutations.

We are investigating errors of low concentration, i.e., only one block (question) in the vector is missing at the time. The final error rate is an average over such single errors made in all possible nine blocks.

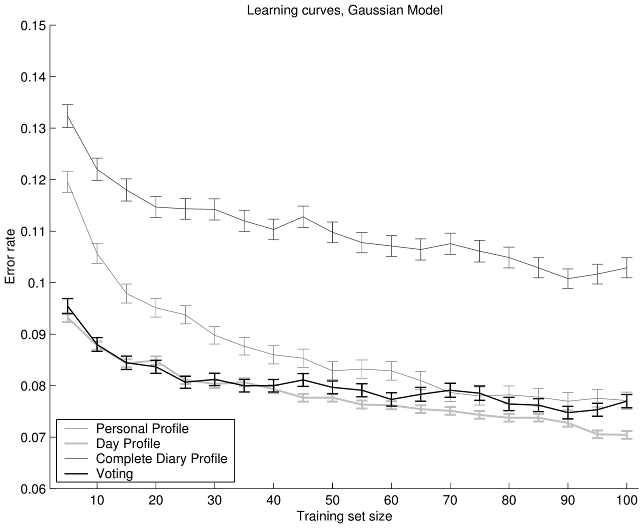


Figure 1: Learning curves for the Gaussian model. Four different methods are presented here: Personal Profile, Day Profile, Complete Diary Profile and Voting. Error bars show deviation from the mean curve over 1000 runs.

Figure 1 and figure 2 presents learning curves for the Gaussian model and the K-Nearest Neighbor model, respectively. The deviation from the mean is shown with the error bars. It decreases slightly with increasing training set size.

For the Gaussian model Voting, as it was expected, gives very good results, it performs better than all other methods for small training sets, however for larger training sets it performs slightly worse than Day Profile. Personal Profile and Complete Diary Profile does generally worse. In the case of KNN model, both Personal Profile and Complete Diary Profile return high error. In this case Day Profile clearly performed better than Voting for all the training sets.

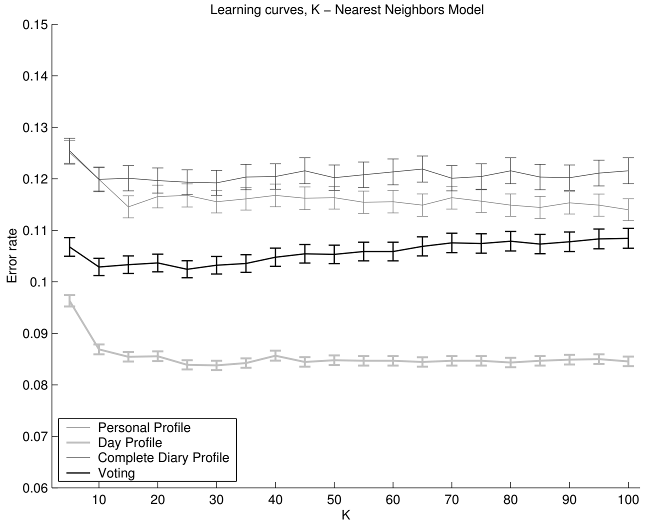


Figure 2: Learning curves calculated for the K-Nearest Neighbor model. Four different methods are presented here: Personal Profile, Day Profile, Complete Diary Profile and Voting. Error bars show deviation from the mean curve over 1000 runs.

In figure 3, the same set of curves is presented, as on the figures 1 and 2. However, here the comparison is made between the two discussed models.

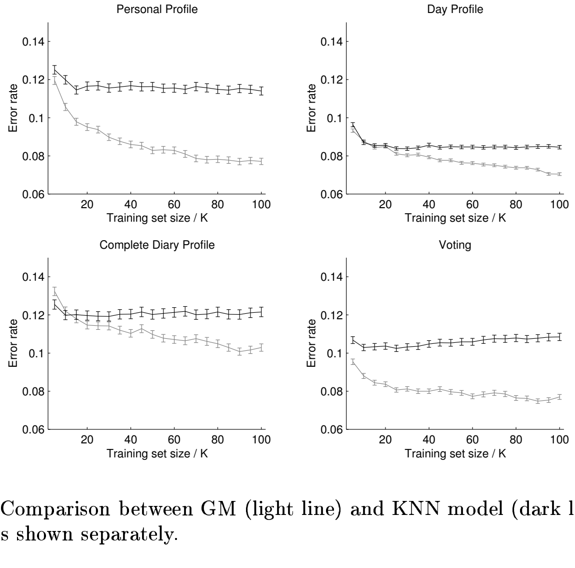


Figure 3: Comparison between GM (light line) and KNN model (dark line) for all the profiles shown separately.

It is clearly seen that for large training sets for every profile, the Gaussian model (light) performs better than KNN model (dark).

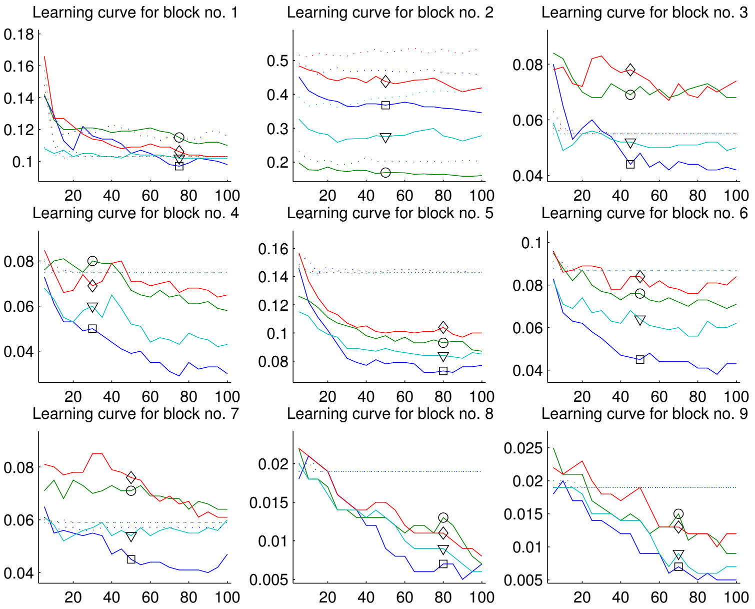


Figure 4: Learning curves for all the 9 blocks separately. GM is shown with solid lines and KNN with the dashed. x-axes are training set size while y-axes are error rates. Diamond marker is Complete Diary Profile, circle is Day Profile, square is Personal Profile, and triangle is Voting.

Figure 4 presents the error rate separately for each of the nine blocks. Every sub-figure corresponds to one question in the questionnaire. The learning curve for block no. 2 (middle-top sub-figure) presents the highest error rate. This block corresponds to question no. 2 (working). The error rate for this block basically creates the overall error rate for the validation sample. Not surprisingly, the value of this field is best predicted by Day Profile. For the rest of the blocks Personal Profile imputate with the smallest error. The situation is similar for the KNN model. However, it is possible to see (also from the figures 1, 2 and 3), that the error rate does not decrease much with increased size of the training set.

Tables 1 and 2 present error correlation matrices for Gaussian and K-Nearest Neighbor model, respectively, for three methods: Personal, Day and Complete Diary Profile. The E_{ij} entry of error correlation matrix [3] is defined as $E_{ij} = \text{Prob}\{\text{error in method } i \wedge \text{error in method } j\}$. The error rates are shown for two extreme training set sizes, namely 5 and 100 samples.

	PP	DP	CDP
PP	0.2481	0.0518	0.1701
DP	0.0518	0.1566	0.1100
CDP	0.1701	0.1100	0.2634

	PP	DP	CDP
PP	0.1695	0.0351	0.2096
DP	0.0351	0.1484	0.1705
CDP	0.2096	0.1705	0.2668

Table 1: Error correlation table for GM. Left and right tables present data for small training set (5 samples) and large training set (100 samples), respectively. Used abbreviations: PP - Personal Profile, DP - Day Profile, CDP - Complete Diary Profile.

	PP	DP	CDP
PP	0.1754	0.0255	0.4093
DP	0.0255	0.0870	0.1004
CDP	0.4093	0.1004	0.2024

	PP	DP	CDP
PP	0.2217	0.0405	0.2294
DP	0.0405	0.1359	0.1240
CDP	0.2294	0.1240	0.2486

Table 2: Error correlation table for KNN model. Left and right tables present data for small training set (5 samples) and large training set (100 samples), respectively. Used abbreviations: PP - Personal Profile, DP - Day Profile, CDP - Complete Diary Profile.

Ideally uncorrelated methods would return errors only on the main diagonal. In such case the Voting procedure would produce optimal results.

Figure 5 shows the error rate for 100 validation samples as a function of training set size. All the methods share the same set of validation samples. It is interesting to see that for some of the validation samples, the error does not depend on which method or model is used or the size of the training set. This phenomenon is even stronger when using KNN model (for example, sample no. 25). In other cases, increased size of the training set will reduce the error rate (sample no. 18 for the Gaussian model). It can also be seen that for other validation samples, the error rate varies from method to method and between the models. In this cases, Voting may return the lowest error rate.

CONCLUSIONS

It is generally expected that the models perform better for large training sets. However, the error rate is strongly sample related, i.e., it can increase significantly with the one “unlucky” sample.

Applying different methods depending on the block number can be relevant for this data set. In this case using Day Profile in the prediction of the value of the block no. 2 and Personal Profile for the rest of the blocks may give considerable improvement in the error rate. However, such mixing of the methods is highly data dependent and has to be tuned manually.

In conclusion, for the present data set, the Gaussian model is superior to the non-parametric K-nearest neighbor model although the Gaussian model assumptions is violated for binary data vectors. The Day Profile method gave best results indicating a strong daily variation. If the errors made by different methods had been uncorrelated, the results returned by the Voting would give the best imputation performance of missing data. For small training sets

Voting resulted in improved performance, while severe correlation among the errors of the methods disfavors Voting for large training sets. In addition. the use of overlapping training sets additionally improved correlation among the methods.

REFERENCES

[1] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in J. D. Cowan, G. Tesauro and J. Alspector (eds.), **Advances in Neural Information Processing Systems**, Morgan Kaufmann Publishers, Inc., 1994, vol. 6, pp. 120–127.

[2] Z. Ghahramani and M. I. Jordan, "Mixture models for Learning from incomplete data," in T. P. R. Greiner and S. Hanson (eds.), **Computational Learning Theory and Natural Learning Systems**, Cambridge, MA: The MIT Press, 1997, vol. IV: Making Learning Systems Practical, pp. 67–85.

[3] L. K. Hansen, C. Lissberg and P. Salamon, "Ensemble Methods for Handwritten Digit Recognition," in S. Kung, F. Fallside, J. Sørensen and C. Kamm (eds.), **Neural Networks for Signal Processing II**, IEEE, 1992, pp. 333–342.

[4] R. Little and D. Rubin, **Statistical Analysis with Missing Data**, John Wiley and Sons, New York, 1987.

[5] D. Rubin, **Multiple Imputation for Nonresponse in Surveys**, J. Wiley and Sons, New York, 1987.

[6] V. Tresp, S. Ahmad and R. Neuneier, "Training Neural Networks with Deficient Data," in J. D. Cowan, G. Tesauro and J. Alspector (eds.), **Advances in Neural Information Processing Systems**, Morgan Kaufmann Publishers, Inc., 1994, vol. 6, pp. 128–135.

[7] V. Tresp, R. Neuneier and S. Ahmad, "Efficient Methods for Dealing with Missing Data in Supervised Learning," in G. Tesauro, D. Touretzky and T. Leen (eds.), **Advances in Neural Information Processing Systems**, The MIT Press, 1995, vol. 7, pp. 689–696.

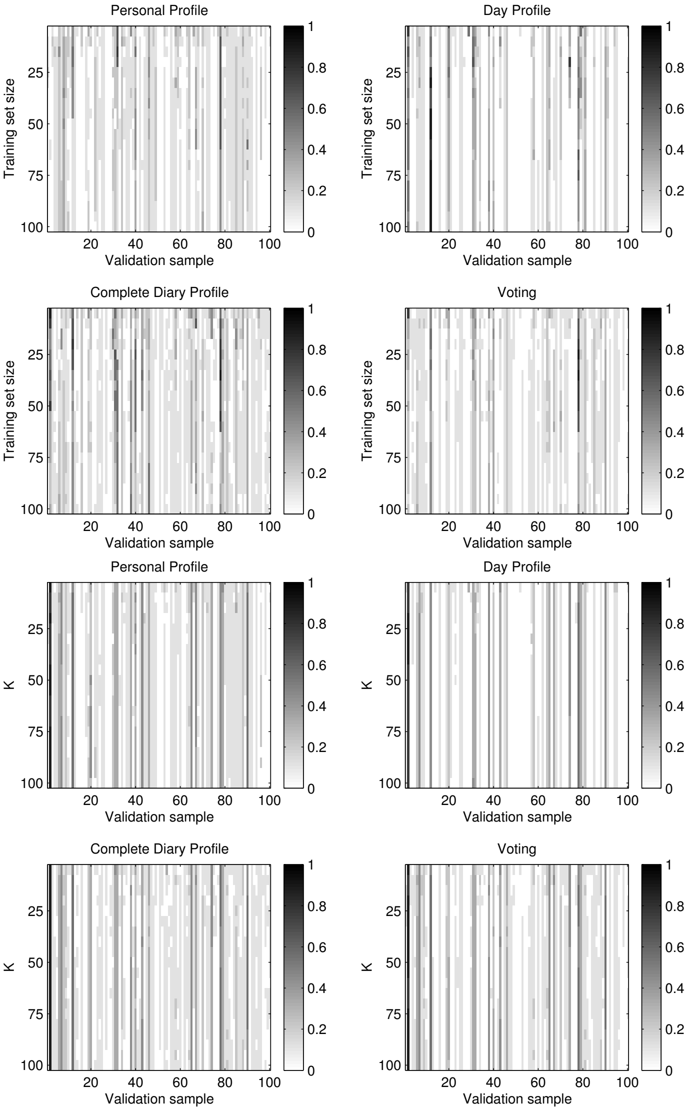


Figure 5: Error rate for different profiles in the GM (the upper four plots) and in the KNN (bottom four plots). Figures show dependency of training set size for specific validation samples. The methods share the same set of validation samples.

APPENDIX C

Contribution: KES 2001

This appendix contain the article *Hierarchical Clustering for Data mining.* , in International Journal of Knowledge-Based Intelligent Engineering Systems, volume: 6(1), pages: 56-62. Author list: J. Larsen, A. Szymkowiak-Have, L.K. Hansen. Own contribution approximated to 30%.

Hierarchical Clustering for Datamining

Anna Szymkowiak, Jan Larsen, Lars Kai Hansen
Informatics and Mathematical Modeling Richard Petersens Plads, Build. 321,
Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark,
Web: http://eivind.imm.dtu.dk, Email: asz,jl,lkhansen@imm.dtu.dk

Abstract. This paper presents hierarchical probabilistic clustering methods for unsupervised and supervised learning in datamining applications. The probabilistic clustering is based on the previously suggested Generalizable Gaussian Mixture model. A soft version of the Generalizable Gaussian Mixture model is also discussed. The proposed hierarchical scheme is agglomerative and based on a $\mathcal{L}_2$ distance metric. Unsupervised and supervised schemes are successfully tested on artificially data and for segmention of e-mails.

1 Introduction

Hierarchical methods for unsupervised and supervised datamining give multilevel description of data. It is relevant for many applications related to information extraction, retrieval navigation and organization, see e.g., [1, 2]. Many different approaches to hierarchical analysis from divisive to agglomerative clustering have been suggested and recent developments include [3, 4, 5, 6, 7]. We focus on agglomerative probabilistic clustering from Gaussian density mixtures. The probabilistic scheme enables automatic detection of the final hierarchy level. In order to provide a meaningful description of the clusters we suggest two interpretation techniques: 1) listing of prototypical data examples from the cluster, and 2) listing of typical features associated with the cluster. The Generalizable Gaussian Mixture model (GGM) and the Soft Generalizable Gaussian mixture model (SGGM) are addressed for supervised and unsupervised learning. Learning from combined sets of labeled and unlabeled data [8, 9] is relevant in many practical applications due to the fact that labeled examples are hard and/or expensive to obtain, e.g., in document categorization. This paper, however, does not discuss such aspects. The GGM and SGGM models estimate parameters of the Gaussian clusters with a modified EM procedure from two disjoint sets of observations that ensures high generalization ability. The optimum number of clusters in the mixture is determined automatically by minimizing the generalization error [10].

This paper focuses on applications to textmining [8, 10, 11, 12, 13, 14, 15, 16] with the objective of categorizing text according to topic, spotting new topics or providing short, easy and understandable interpretation of larger text blocks; in a broader sense to create intelligent search engines and to provide understanding of documents or content of webpages like Yahoo's ontologies.

2 The Generalizable Gaussian Mixture Model

The first step in our approach for probabilistic clustering is a flexible and universal Gaussian mixture density model, the generalizable Gaussian mixture model (GGM) [10, 17, 18], which models the density for d-dimensional feature vectors by:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P(k)p(\boldsymbol{x}|k), \;\; p(\boldsymbol{x}|k) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right) \quad (1)$$

where $p(\boldsymbol{x}|k)$ are the component Gaussians mixed with the non-negative proportions $P(k)$, $\sum_{k=1}^{K} P(k)$. Each component k is described by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$. Parameters are estimated with an iterative modified EM algorithm [10] where means are estimated on one data set, covariances on an independent set, and $P(k)$ on the combined set. This prevents notorious overfitting problems with the standard approach [19]. The optimum number of clusters/components is chosen by minimizing an approximation of the generalization error; the AIC criterion, which is the negative log-likelihood plus two times the number of parameters.

For unsupervised learning parameters are estimated from a training set of feature vectors $\mathcal{D} = \{\boldsymbol{x}_n; n = 1, 2 \ldots, N\}$, where N is the number of samples. In supervised learning for classification from a data set of features and class labels $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{1, 2, \ldots, C\}$ we adapt one Gaussian mixture, $p(\boldsymbol{x}|y)$, for each class separately and classify by Bayes optimal rule by maximizing $p(y|\boldsymbol{x}) = p(\boldsymbol{x}|y)P(y)/\sum_{y=1}^{C} p(\boldsymbol{x}|y)P(y)$ (under 1/0 loss). This approach is also referred to as mixture discriminant analysis [20].

The GGM can be implemented using either hard or soft assignments of data to components in each EM iteration step. In the hard GMM approach each data example is assigned to a cluster by selecting highest $p(k|\boldsymbol{x}_n) = p(\boldsymbol{x}_n|k)P(k)/p(\boldsymbol{x}_n)$. Means and covariances are estimated by classical empirical estimates from data assigned to each component. In the soft version (SGGM) e.g., the means are estimated as weighted means $\boldsymbol{\mu}_k = \sum_n p(k|\boldsymbol{x}_n) \cdot \boldsymbol{x}_n / \sum_n p(k|\boldsymbol{x}_n)$.

Experiments with the hard/soft versions gave the following conclusions. Per iteration the algorithms are almost identical, however, SGGM requires typically more iteration to converge, which is defined by no changes in assignment of examples to clusters. Learning curve[1] experiments indicate that hard GGM has slightly better generalization performance for small N while similar behavior for large N - in particular if clusters are well separated.

3 Hierarchical Clustering

In the suggested agglomerative clustering scheme we start by K clusters at level $j = 1$ as given by the optimized GGM model of $p(\boldsymbol{x})$, which in the case of supervised learning is $p(\boldsymbol{x}) = \sum_{y=1}^{C} \sum_{k=1}^{K_y} p(\boldsymbol{x}|k, y)P(k)P(y)$, where K_y is the optimal number of components for class y. At each higher level in the hierarchy two clusters are merged based on a similarity measure between pairs of clusters. The procedure is repeated until we reach one cluster at the top level. That is, at level $j = 1$ there are K clusters and 1 cluster at the final level, $j = 2K - 1$. Let $p_j(\boldsymbol{x}|k)$ be the density for the k'th cluster at level j and $P_j(k)$ as its mixing proportion, i.e., the density model at level j is $p(\boldsymbol{x}) = \sum_{k=1}^{K-j+1} P_j(k)p_j(\boldsymbol{x}|k)$. If clusters k and m at level j are merged into ℓ at level $j + 1$ then

$$p_{j+1}(\boldsymbol{x}|\ell) = \frac{p_j(\boldsymbol{x}|k) \cdot P_j(k) + p_j(\boldsymbol{x}|m) \cdot P_j(m)}{P_j(k) + P_j(m)}, \;\; P_{j+1}(\ell) = P_j(k) + P_j(m) \quad (2)$$

The natural distance measure between the cluster densities is the Kullback-Leibler (KL) divergence [19], since it reflects dissimilarity between the densities in the probabilistic space. The drawback is that KL only obtains an analytical expression for the first level in the

[1] Generalization error as as function of number of examples.

hierarchy while distances for the subsequently levels have to be approximated [17, 18]. Another approach is to base distance measure on the $\mathcal{L}_2$ norm for the densities [21], i.e., $D(k,m) = \int (p_j(\boldsymbol{x}|k) - p_j(\boldsymbol{x}|m))^2 \, dx$ where k and m index two different clusters. Due to Minkowksi's inequality $D(k,m)$ is a distance measure. Let $\mathcal{I} = \{1, 2, \cdots, K\}$ be the set of cluster indices and define disjoint subsets $\mathcal{I}_\alpha \cap \mathcal{I}_\beta = \emptyset$, $\mathcal{I}_\alpha \subset \mathcal{I}$ and $\mathcal{I}_\beta \subset \mathcal{I}$, where $\mathcal{I}_\alpha$, $\mathcal{I}_\beta$ contain the indices of clusters which constitute clusters k and m at level j, respectively. The density of cluster k is given by: $p_j(\boldsymbol{x}|k) = \sum_{i \in \mathcal{I}_\alpha} \alpha_i p(\boldsymbol{x}|i)$, $\alpha_i = P(i) / \sum_{i \in \mathcal{I}_\alpha} P(i)$ if $i \in \mathcal{I}_\alpha$, and zero otherwise. $p_j(\boldsymbol{x}|m) = \sum_{i \in \mathcal{I}_\beta} \beta_i p(\boldsymbol{x}|i)$, where β_i obtains a similar definition. According to [21] the Gaussian integral $\int p(\boldsymbol{x}|i) p(\boldsymbol{x}|\ell) \, dx = G(\boldsymbol{\mu}_i - \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_\ell)$, where $G(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2} \cdot |\boldsymbol{\Sigma}|^{1/2} \cdot \exp(-\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}/2)$. Define the vectors $\boldsymbol{\alpha} = \{\alpha_i\}$, $\boldsymbol{\beta} = \{\beta_i\}$ of dimension K and the $K \times K$ symmetric matrix $\boldsymbol{G} = \{G_{i\ell}\}$ with $G_{i\ell} = G(\boldsymbol{\mu}_i - \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_\ell)$, then the distance can be then written as $D(k,m) = (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top \boldsymbol{G} (\boldsymbol{\alpha} - \boldsymbol{\beta})$. Figure 1 illustrates the hierarchical clustering for Gaussian distributed toy data.

A unique feature of probabilistic clustering is the ability to provide optimal cluster and level assignment for new data examples which have not been used for training. $\boldsymbol{x}$ is assigned to cluster k at level j if $p_j(k|\boldsymbol{x}) > \rho$ where the threshold ρ typically is set to 0.9. The procedure ensures that the example is assigned to a wrong cluster with probability 0.1.

Interpretation of clusters is done by generating likely examples from the cluster, see further [17]. For the first level in the hierarchy where distributions are Gaussian this is done by drawing examples from a super-eliptical region around the mean value, i.e., $(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) < \textit{const}$. For clusters at higher levels in the hierarchy samples are drawn from each Gaussian cluster with proportions specified by $P(k)$.

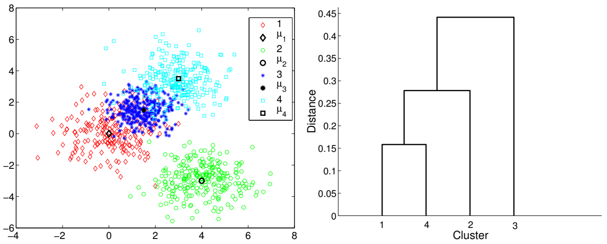


Figure 1: Hierarchical clustering example. Left panel is a scatter plot of the data. Clusters 1,2 and 4 have wide distributions while 3 has a narrow one. Since the distance is based on the shape of the distribution and not only its mean location, clusters 1 and 4 are much closer than any of these to cluster 3. Right panel presents the dendrogram.

4 Experiments

The hierarchical clustering is illustrated for segmentation of e-mails. Define term-vector as a complete set of the unique words occurring in all the emails. An email histogram is the vector containing frequency of occurrence of each word from the term-vector and defines the content of the email. The term-document matrix is then the collection of histograms for all emails in the database. After suitable preprocessing[2] the term-document matrix contains 1405 (702 for training and 703 for testing) e-mail documents, and the term-vector 7798 words. The emails where annotated into the categories: *conference*, *job* and *spam*. It is possible to model

[2] Words which are too likely or too unlikely are removed. Further only word stems are kept.

directly from this matrix [8, 15], however we deploy Latent Semantic Indexing (LSI) [22] which operates from a latent space of feature vectors. These are found by projecting term-vectors into a subspace spanned by the left eigenvectors associated with largest singular value of a singular value decomposition of the term-document matrix. We are currently investigating methods for automatic determination of the subspace dimension based on generalization concepts. We found that a 5 dimensional subspace provides good performance using SGGM.

A typical result of running supervised learning is depicted in Figure 2. Using supervised learning provides a better resemblance with the correct categories at the level in the hierarchy as compared with unsupervised learning. However, since labeled examples often are lacking or few the hierarchy provides a good multilevel description of the data with associated interpretations. Finding typical features as described on page 3 and back-projecting into original term-space provides keywords for each cluster as given in Table 1.

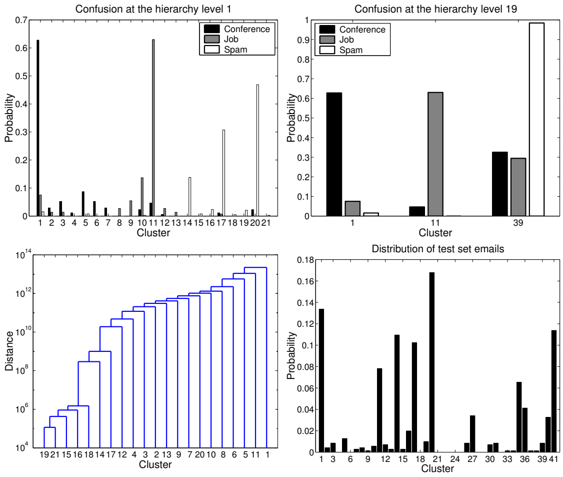


Figure 2: Supervised hierarchical clustering. Upper rows show the confusion of clusters with the annotated email labels on the training set at the first level and the level where 3 clusters remains, corresponding to the three categories *conference*, *job* and *spam*. At level 1 clusters 1,11,17,20 have big resemblance with the categories. In particular *spam* are distributed among 3 clusters. At level 19 there is a high resemblance with the categories and the average probability of erroneous category on the test set is 0.71. The lower left panel shows the dendrogram associated with the clustering. The lower right panel shows the histogram of cluster assignments for test data, cf. page 3. Clearly some samples obtain a reliable description at the first level (1–21) in the hierarchy, whereas others are reliable at a higher level (22–41).

5 Conclusions

This paper presented a probabilistic agglomerative hierarchical clustering algorithm based on the generalizable Gaussian mixture model and a $\mathcal{L}_2$ metric in probabilty density space. This leads to a simple algorithm which can be used both for supervised and unsupervised learning. In addition, the probabilistic scheme allows for automatic cluster and hierarchy level assignment for unseen data and further a natural technique for interpretation of the clusters

Table 1: Keywords for supervised learning

1	research,university,conference	8	neural,model	15	click,remove,hottest,action
2	university,neural,research	9	university,interest,computetion	16	free,adult,remove,call
3	research,creativity,model	10	research,position,application	17	website,adult,creativity,click
4	website,information	11	science,position,fax	18	website,click,remove
5	information,program,computation	12	position,fax,website	19	free,call,remove,creativity
6	research,science,computer,call	13	research,position,application	20	mac
7	website,creativity	14	free,adult,call,website	21	adult,government

1	research,university,conference	11	science,position,fax	39	free,website,cal l,creativity

via prototype examples and features. The algorithm was successfully applied to segmentation of emails.

References

[1] J. Carbonell, Y. Yang and W. Cohen, Special Isssue of Machine Learning on Information Retriceal Introduction, *Machine Learning* **39**, (2000) 99–101.

[2] D. Freitag, Machine Learning for Information Extraction in Informal Domains, *Machine Learning* **39**, (2000) 169–202.

[3] C.M. Bishop and M.E. Tipping, A Hierarchical Latent Variable Model for Data Visualisation, *IEEE T-PAMI* **3**, 20 (1998) 281–293.

[4] C. Fraley, Algorithms for Model-Based Hierarchical Clustering, *SIAM J. Sci. Comput.* **20**, 1 (1998) 279–281.

[5] M. Meila and D. Heckerman, An Experimental Comparison of Several Clustering and Initialisation Methods. In: Proc. 14th Conf. on Uncert. in Art. Intel., Morgan Kaufmann, 1998, pp. 386–395.

[6] C. Williams, A MCMC Approach to Hierarchical Mixture Modelling. In: Advances in NIPS 12, 2000, pp. 680–686.

[7] N. Vasconcelos and A. Lippmann, Learning Mixture Hierarchies. In: Advances in NIPS 11, 1999, pp. 606–612.

[8] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, **39** 2–3 (2000) 103–134.

[9] D.J. Miller and H.S. Uyar, A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data. In: Advances in NIPS 9, 1997, pp. 571–577.

[10] L.K. Hansen, S. Sigurdsson, T. Kolenda, F.Å. Nielsen, U. Kjems and J. Larsen, Modeling Text with Generalizable Gaussian Mixtures. In: Proc. of IEEE ICASSP'2000, vol. 6, 2000, pp. 3494–3497.

[11] C.L. Jr. Isbell and P. Viola, Restructuring Sparse High Dimensional Data for Effective Retrieval. In: Advances in NIPS 11, MIT Press, 1999, pp. 480–486.

[12] T. Kolenda, L.K. Hansen and S. Sigurdsson Indepedent Components in Text. In: Adv. in Indep. Comp. Anal., Springer-Verlag, pp. 241–262, 2001.

[13] T. Honkela, S. Kaski, K. Lagus and T. Kohonen, Websom — self-organizing maps of document collections. In: Proc. of Work. on Self-Organizing Maps, Espoo, Finland, 1997.

[14] E.M. Voorhees, Implementing Agglomerative Hierarchic Clustering Ulgorithms for Use in Document Retrieval, *Inf. Proc. & Man.* **22** 6 (1986) 465–476.

[15] A. Vinokourov and M. Girolami, A Probabilistic Framework for the Hierarchic Organization and Classification of Document Collections, submitted for *Journal of Intelligent Information Systems*, 2001.

[16] A.S. Weigend, E.D. Wiener and J.O. Pedersen Exploiting Hierarchy in Text Categorization, *Information Retrieval*, **1** (1999) 193–216.

[17] J. Larsen, L.K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda, Webmining: Learning from the World Wide Web, *Computational Statistics and Data Analysis* (2001).

[18] J. Larsen, L.K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda, Webmining: Learning from the World Wide Web. In: Proc. of Nonlinear Methods and Data Mining, Italy, 2000, pp. 106–125.

[19] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[20] T. Hastie and R. Tibshirani Discriminant Analysis by Gaussian Mixtures, *Jour. Royal Stat. Society - Series B*, **58** 1 (1996) 155–176.

[21] D. Xu, J.C. Principe, J. Fihser, H.-C. Wu, A Novel Measure for Independent Component Analysis (ICA). In: Proc. IEEE ICASSP98, vol. 2, 1998, pp. 1161–1164.

[22] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, Indexing by Latent Semantic Analysis, *Journ. Amer. Soc. for Inf. Science.*, **41** (1990) 391–407.

APPENDIX D

Contribution: International KES Journal 2002

This appendix contain the article *Probabilistic Hierarchical Clustering with Labeled and Unlabeled Data* in Proceedings of KES-2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, pages: 261–265. Author list: A. Szymkowiak, J. Larsen, L.K. Hansen. Own contribution approximated to 80%.

Probabilistic Hierarchical Clustering with Labeled and Unlabeled Data

J. Larsen, A. Szymkowiak, L.K. Hansen

Informatics and Mathematical Modeling, Technical University of Denmark
Richard Petersens Plads, Build. 321, DK-2800 Kongens Lyngby, Denmark
Web: http://eivind.imm.dtu.dk, Emails: jl,asz,lkh@imm.dtu.dk

Abstract. *This paper presents hierarchical probabilistic clustering methods for unsupervised and supervised learning in datamining applications, where supervised learning is performed using both labeled and unlabeled examples. The probabilistic clustering is based on the previously suggested Generalizable Gaussian Mixture model and is extended using a modified Expectation Maximization procedure for learning with both unlabeled and labeled examples. The proposed hierarchical scheme is agglomerative and based on probabilistic similarity measures. Here, we compare a $\mathcal{L}_2$ dissimilarity measure, error confusion similarity, and accumulated posterior cluster probability measure. The unsupervised and supervised schemes are successfully tested on artificially data and for e-mails segmentation.*

1 Introduction

Hierarchical methods for unsupervised and supervised datamining provide multilevel description of data, which is relevant for many applications related to information extraction, retrieval navigation and organization of information, see e.g., [4, 7]. Many different approaches to hierarchical analysis from divisive to agglomerative clustering schemes have been suggested, and recent developments include [3, 6, 16, 20, 24]. In this paper we focus on agglomerative probabilistic clustering from Gaussian density mixtures based on earlier work [14, 15, 19] but extended by suggesting and comparing various similarity measures in connection with cluster merging. An advantage of using the probabilistic clustering scheme is automatic detection of the final hierarchy level for new data not used for training. In order to provide a meaningful description of the clusters we suggest two interpretation techniques: listing of prototypical data examples from the cluster, and listing of typical features associated with the cluster.

The generalizable Gaussian mixture model (GGM) [8] and the soft generalizable Gaussian mixture model (SGGM) [19] are basic model for supervised and unsupervised learning. We extend this framework to supervised learning from combined sets of labeled and unlabeled data [9, 17, 18] and present a modified version of the approach in [17] called the unsupervised/supervised generalizable Gaussian mixture model (USGGM). Supervised learning from combined sets is relevant in many practical applications due to the fact that labeled examples are hard and/or expensive to obtain, for instance in document categorization or medical applications. The models estimate parameters of the Gaussian clusters with a modified EM procedure from two disjoint data sets to prevent notorious infinite overfit problems and ensuring good generalization ability. The optimum number of clusters in the mixture is determined automatically by minimizing an estimate of the generalization error [8].

This paper focuses on applications to textmining [8, 11, 12, 13, 18, 22, 21, 23] with the objective of categorizing text according to topic, spotting new topics or providing short, easy and understandable interpretation of larger text blocks – in a broader sense to create intelligent search engines and to provide understanding of documents or content of webpages like Yahoo's ontologies.

In Section 2, various GGM models for supervised and unsupervised learning are discussed, in particular we introduce the USGGM algorithm. The hierarchical clustering scheme is discussed in section 3 and introduces three similarity measures for cluster merging. Finally, Section 4 provide numerical experiments for segmentation of e-mails.

2 The Generalizable Gaussian Mixture Model

The first step in our approach for probabilistic clustering is a flexible and universal extension of Gaussian mixture density model, the generalizable Gaussian mixture model [8, 14, 15, 19] with the aim of supervised learning from unlabeled and labeled data. Define $\boldsymbol{x}$ as the d-dimensional input feature vector and the associated output, $y \in \{1, 2, \cdots, C\}$, of class labels, assuming C mutually exclusive classes. The joint input/output density is modeled as the Gaussian

This research is supported by the Danish Research Councils through Distributed Multimedia Technologies and Applications program within Center for Multimedia and the Signal and Image Processing for Telemedicine (SITE) program.

mixture in [17][1]

$$p(y, \boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} P(y|k)p(\boldsymbol{x}|k)P(k) \qquad (1)$$

$$p(\boldsymbol{x}|k) = \qquad (2)$$
$$\frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

where K is the number of components, $p(\boldsymbol{x}|k)$ are the component Gaussians mixed with the non-negative priors $P(k)$, $\sum_{k=1}^{K} P(k) = 1$ and the class-cluster posteriors $P(y|k)$, $\sum_{y=1}^{C} P(y|k) = 1$. The k'th Gaussian component is described by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$. $\boldsymbol{\theta}$ is the vector of all model parameters, i.e., $\boldsymbol{\theta} \equiv \{P(y|k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, P(k) : \forall k, y\}$. Since the Gaussian mixture is an universal approximator, the model Eq. (1) is rather flexible. One restriction, however, is that the joint input/output for each components is assumed to factorize, i.e., $p(y, \boldsymbol{x}|k) = P(y|k)p(\boldsymbol{x}|k)$.

The input density associated with Eq. (1) is given by

$$p(\boldsymbol{x}|\boldsymbol{\theta}_u) = \sum_{y=1}^{C} p(y, \boldsymbol{x}) = \sum_{k=1}^{K} p(\boldsymbol{x}|k)P(k), \qquad (3)$$

where $\boldsymbol{\theta}_u \equiv \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, P(k) : \forall k, y\}$. Assuming a 0/1 loss function the optimal Bayes classification rule is $\widehat{y} = \max_y P(y|\boldsymbol{x})$ where[2]

$$P(y|\boldsymbol{x}) = \frac{p(y, \boldsymbol{x})}{p(\boldsymbol{x})} = \sum_{k=1}^{K} P(y|k)P(k|\boldsymbol{x}) \qquad (4)$$

with $P(k|\boldsymbol{x}) = p(\boldsymbol{x}|k)P(k)/p(\boldsymbol{x})$.

Define the data set of unlabeled examples $\mathcal{D}_u = \{\boldsymbol{x}_n; n = 1, 2, \cdots, N_u\}$ and a set of labeled examples $\mathcal{D}_l = \{\boldsymbol{x}_n, y_n; n = 1, 2, \cdots, N_l\}$. The objective is to estimate $\boldsymbol{\theta}$ from the combined set $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ with $N = N_l + N_u$ examples ensuring high generalizability. If no labeled data are available we can merely perform unsupervised learning of $\boldsymbol{\theta}_u$, however, if a number of labeled data are available, estimation from both data sets is possible as $p(y|\boldsymbol{x})$ and $p(\boldsymbol{x})$ share the model parameters $\boldsymbol{\theta}_u$ [9]. The negative log-likelihood for the data sets, which are assumed to consist of independent examples, is given by

$$L = -\log p(\mathcal{D}|\boldsymbol{\theta}) \qquad (5)$$
$$- \sum_{n \in \mathcal{D}_l} \log \sum_{k=1}^{K} P(y_n|k)p(\boldsymbol{x}_n|k)P(k)$$
$$-\lambda \sum_{n \in \mathcal{D}_u} \log \sum_{k=1}^{K} p(\boldsymbol{x}_n|k)P(k)$$

where $0 \leq \lambda \leq 1$ is a discount factor. If the model is unbiased (realizable), the estimation $\boldsymbol{\theta}_u$ from either labeled or unlabeled data will result in identical optimal setting and thus $\lambda = 1$ is optimal. On the other hand, in the typical case of a biased mode, it is advantageous to discount the influence of unlabeled data [9, 18].

```
Initialization
1. Choose values for $K$ and $0 \leq \lambda \leq 1$.
2. Let $\boldsymbol{i}$ be $K$ different randomly selected indices from $\{1, 2, \cdots, N\}$, and set $\boldsymbol{\mu}_k = \boldsymbol{x}_{\boldsymbol{i}_k}$.
3. Let $\boldsymbol{\Sigma}_0 = N^{-1} \sum_{n \in \mathcal{D}} (\boldsymbol{x}_n - \boldsymbol{\mu}_0)(\boldsymbol{x}_n - \boldsymbol{\mu}_0)^\top$, where $\boldsymbol{\mu}_0 = N^{-1} \sum_{n \in \mathcal{D}} \boldsymbol{x}_n$, and set $\forall k : \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_0$.
4. Set $\forall k : P(k) = 1/K$.
5. Compute class prior probabilities: $P(y) = N_l^{-1} \sum_{n \in \mathcal{D}_l} \delta(y_n - y)$, where $\delta(z) = 1$ if $z = 0$, and zero otherwise. Set $\forall k : P(y|k) = P(y)$.
6. Select a split ratio $0 < \gamma < 1$. Split the unlabeled data set into disjoint sets as $\mathcal{D}_u = \mathcal{D}_{u,1} \cup \mathcal{D}_{u,2}$, with $|\mathcal{D}_{u,1}| = [\gamma N_u]$ and $|\mathcal{D}_{u,2}| = N_u - |\mathcal{D}_{u,1}|$. Do similar splitting for the labeled data set $\mathcal{D}_l = \mathcal{D}_{l,1} \cup \mathcal{D}_{l,2}$.
Repeat until convergence
1. Compute posterior component probabilities: $p(k|\boldsymbol{x}_n) = p(\boldsymbol{x}_n|k)P(k) / \sum_k p(\boldsymbol{x}_n|k)P(k)$, for all $n \in \mathcal{D}_u$, and for all $n \in \mathcal{D}_l$,
       $p(k|y_n, \boldsymbol{x}_n) = \frac{P(y_n|k)p(\boldsymbol{x}_n|k)P(k)}{\sum_k P(y_n|k)p(\boldsymbol{x}_n|k)P(k)}$.
2. For all $k$ update means
       $\boldsymbol{\mu}_k = \frac{\sum_{n \in \mathcal{D}_{l,1}} \boldsymbol{x}_n P(k|y_n, \boldsymbol{x}_n) + \lambda \sum_{n \in \mathcal{D}_{u,1}} \boldsymbol{x}_n P(k|\boldsymbol{x}_n)}{\sum_{n \in \mathcal{D}_{l,1}} P(k|y_n, \boldsymbol{x}_n) + \lambda \sum_{n \in \mathcal{D}_{u,1}} P(k|\boldsymbol{x}_n)}$
3. For all $k$ update covariance matrices
       $\boldsymbol{\Sigma}_k = \frac{\sum_{n \in \mathcal{D}_{l,2}} \boldsymbol{S}_{kn} P(k|y_n, \boldsymbol{x}_n) + \lambda \sum_{n \in \mathcal{D}_{u,2}} \boldsymbol{S}_{kn} P(k|\boldsymbol{x}_n)}{\sum_{n \in \mathcal{D}_{l,2}} P(k|y_n, \boldsymbol{x}_n) + \lambda \sum_{n \in \mathcal{D}_{u,2}} P(k|\boldsymbol{x}_n)}$
   where $\boldsymbol{S}_{kn} = (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$. Perform a regularization of $\boldsymbol{\Sigma}_k$ (see text).
4. For all $k$ update cluster priors
       $P(k) = \frac{\sum_{n \in \mathcal{D}_l} P(k|y_n, \boldsymbol{x}_n) + \lambda \sum_{n \in \mathcal{D}_u} P(k|\boldsymbol{x}_n)}{N_l + \lambda N_u}$
5. For all $k$ update class cluster posteriors
       $P(y|k) = \frac{\sum_{n \in \mathcal{D}_l} \delta(y - y_n) P(k|y_n, \boldsymbol{x}_n)}{\sum_{n \in \mathcal{D}_l} P(k|y_n, \boldsymbol{x}_n)}$
```

Figure 1: The USGGM algorithm.

[1] In [17] referred to as the generalized mixture model.

[2] The dependence on $\boldsymbol{\theta}$ is omitted.

2.1 The USGGM Algorithm

The model parameters are estimated with an iterative modified EM algorithm [8], where means and covariance matrices are estimated from independent data sets, and $P(y|k)$, $P(k)$ from the combined set. This approach prevents overfitting problems with the standard approach [2]. It is designated the generalizable Gaussian mixture model with labeled and unlabeled data (USGGM) and may be viewed as an extension of the EM-I algorithm suggested in [17]. The GGM can be implemented using either hard or soft assignments of data to components in each EM iteration step. In the hard GGM approach each data example is assigned to a cluster by selecting highest $P(k|\boldsymbol{x})$. Means and covariances are estimated by classical empirical estimates from data assigned to each component. In the soft version (SGGM) [19] means and covariances are estimated as weighted quantities, e.g., $\boldsymbol{\mu}_k = \sum_n p(k|\boldsymbol{x}_n)\boldsymbol{x}_n / \sum_n p(k|\boldsymbol{x}_n)$. GGM provides a biased estimate, which gives better results for small data sets [19], however, in general the soft version is preferred. The USGGM algorithm is summarized in Fig. 1 and is based on the soft approach. The main iteration loop is aborted[3] when no change in example cluster assignment is noticed. Labeled examples are assigned to clusters $k_n = \arg\max_k P(k|y_n, \boldsymbol{x}_n)$, $n \in \mathcal{D}_l$, and unlabeled to $k_n = \arg\max_k P(k|\boldsymbol{x}_n)$, $n \in \mathcal{D}_u$. In contrast to EM algorithms there is no guarantee that each iteration leads to improved likelihood, however, practical experience indicates that the updating scheme is sufficiently robust. Potential poor conditioned covariance matrices for clusters where few examples are assigned is avoided by regularizing towards the overall input covariance matrix $\boldsymbol{\Sigma}_0$ (defined in Fig. 1) as $\boldsymbol{\Sigma}_k \leftarrow \boldsymbol{\Sigma}_k + \alpha\boldsymbol{\Sigma}_0$. α is selected as the smallest positive number, which ensures that the resulting condition number is smaller than $1/(d \cdot \epsilon)$, where ϵ is the floating point machine precision.

Essential algorithm parameters are the number of components K and the weighting factor λ. In principle, these parameters should be chosen as to maximize generalization performance. One method is to pick K and λ so that the cross-validation estimate of the classification error is minimized. A less computational cumbersome method is to select K based on the AIC estimate of the generalization error [1, 8, 19], which is the negative log-likelihood plus the number of parameters in the model, $K(d(d+1)/2+C)-1$. The only remaining algorithm parameter to determine is the split ratio γ, which in principle also should be selected to achieve high generalization performance. Practical simulations show that $\gamma = 0.5$ is a proper choice in most cases.

2.2 Unsupervised GGM Model

If only input data are available one has to perform unsupervised learning. In this case the object of modeling is the input density Eq. 3, which can be trained using the SGGM algorithm[4] [19].

2.3 Supervised GGM Model

Clearly USGGM can be used in the case of no unlabeled examples. Another choice is to use separate GGM models for the class conditional input densities, i.e., $p(\boldsymbol{x}|y) = \sum_{k=1}^{K_y} p(\boldsymbol{x}|y,k)P(k|y)$ with $p(\boldsymbol{x}|y,k)$ defined by Eq. (2) and where K_y is the number of components. Using Bayes optimal rule and assuming a 1/0 loss function, classification is done by maximizing $p(y|\boldsymbol{x}) = p(\boldsymbol{x}|y)P(y)/\sum_{y=1}^{C} p(\boldsymbol{x}|y)P(y)$. The approach is also referred to as mixture discriminant analysis [10] and seems more flexible than the model in Eq. (1). However, it does not use discriminative training, i.e., minimizing the classification error or negative log-likelihood $L = -\sum_n \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are model parameters. Modeling instead $p(\boldsymbol{x}|y)$ will provide reasonable estimates of $p(y|\boldsymbol{x})$ in the entire input space, whereas discriminative learning will use the data to obtain relatively better estimates of $p(y|\boldsymbol{x})$ close to the decision boundaries. The model in Eq. (1) describes the joint input-class probability $p(y,\boldsymbol{x}) = p(y|\boldsymbol{x})p(\boldsymbol{x})$ and may be interpreted as a partial discriminative estimation procedure.

3 Hierarchical Clustering

In the case of unsupervised learning, i.e., learning $p(\boldsymbol{x})$, hierarchical clustering concerns identifying a hierarchical structure of clusters in the feature space $\boldsymbol{x}$. In the suggested agglomerative clustering scheme we start by K clusters at level $j = 1$ as given by the optimized GGM model of $p(\boldsymbol{x})$. At each higher level in the hierarchy two clusters are merged based on a similarity measure between pairs of clusters. The procedure is repeated until we reach one cluster at the top level. That is, at level $j = 1$ there are K clusters, and one cluster at the final level, $j = K$.

For supervised learning one can either identify a hierarchical structure common for all classes, i.e., working from the associated input density $p(\boldsymbol{x})$, or identifying individual hierarchies for each class by working from the class conditional input densities $p(\boldsymbol{x}|y)$. For the model in Eq. (1) $p(\boldsymbol{x})$ is given by Eq. (3) and

$$p(\boldsymbol{x}|y) = \frac{p(y,\boldsymbol{x})}{P(y)} = \sum_{k=1}^{K} p(\boldsymbol{x}|k)P(k|y) \qquad (6)$$

[3] Convergence criteria based on changes in the negative log-likelihood can also be formulated.

[4] The SGGM is similar to USGGM in Fig. 1 and is essentially obtained by setting $\lambda = 1$, neglecting steps 5 of the initialization and main iteration loop, and further neglecting sums over labeled data.

where $P(k|y) = P(y|k)P(k)/\sum_k P(y|k)P(k)$. Let $p_j(\boldsymbol{x}|y,k)$ be the density for the k'th cluster at level j, and $P_j(k|y)$ the mixing proportion, which in the general case both may depend on y. Further, the (class conditional) density model at level j is $p(\boldsymbol{x}|y) = \sum_{k=1}^{K-j+1} P_j(k|y)p_j(\boldsymbol{x}|y,k)$. If clusters ℓ and m at level j are merged into i at level $j+1$ then

$$p_{j+1}(\boldsymbol{x}|y,i) = \tag{7}$$
$$\frac{p_j(\boldsymbol{x}|y,\ell)P_j(\ell|y) + p_j(\boldsymbol{x}|y,m)P_j(m|y)}{P_j(\ell|y) + P_j(m|y)},$$
$$P_{j+1}(i|y) = P_j(\ell|y) + P_j(m|y) \tag{8}$$

3.1 Level Assignment

A unique feature of probabilistic clustering is the ability to provide optimal cluster and level assignment for new data examples, which have not been used for training. $\boldsymbol{x}$ is assigned to cluster k at level j if

$$P_j(k|y,\boldsymbol{x}) = \frac{p_j(\boldsymbol{x}|y,k)P(k|y)}{p(\boldsymbol{x}|y)} > \rho \tag{9}$$

where the threshold ρ typically is set to 0.9. The procedure ensures that the example is assigned to a wrong cluster with probability 0.1.

3.2 Cluster Interpretation

Interpretation of clusters is done by generating likely examples from the cluster [14, 19] and displaying prototype examples and/or typical features. For the first level in the hierarchy in which distributions are Gaussian, prototype examples are identified as those who has highest density values. For clusters at higher levels in the hierarchy, prototype samples are drawn from each Gaussian cluster with proportions specified by $P(k)$ or $P(k|y)$. Typical features are in the first level found by drawing ancillary examples from a super-eliptical region around the mean value, i.e., $(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) < \textit{const.}$, and then listing associated typical features, e.g., keywords as demonstrated in Sec. 4. At higher levels we proceed as described above.

3.3 Similarity measures

Many different similarity measures may be applied in the framework of hierarchical clustering. The natural distance measure between the cluster densities is the Kullback-Leibler (KL) divergence [2], since it reflects dissimilarity between the densities in the probabilistic space. The drawback is that KL only obtains an analytic expression for the first level in the hierarchy, while distances for the subsequent levels have to be approximated [14, 15]. Consequently, we consider three different measures, which express similarity in probability space for models of $p(\boldsymbol{x})$ or $p(\boldsymbol{x}|y)$ (cf. Sec. 3) and can be computed exactly at all levels in the hierarchy[5]. Fig. 2 illustrates the hierarchical clustering for Gaussian distributed toy data.

3.3.1 $\mathcal{L}_2$ Dissimilarity Measure

The $\mathcal{L}_2$ distance for the densities [25] is defined

$$D(\ell,m) = \int \left(p_j(\boldsymbol{x}|\ell) - p_j(\boldsymbol{x}|m)\right)^2 dx \tag{10}$$

where ℓ and m index two different clusters. Due to Minkowksi's inequality, $D(\ell,m)$ is a distance measure, which also will be referred to as dissimilarity. Let $\mathcal{I} = \{1,2,\cdots,K\}$ be the set of cluster indices and define disjoint subsets $\mathcal{I}_\alpha \cap \mathcal{I}_\beta = \emptyset$, $\mathcal{I}_\alpha \subset \mathcal{I}$ and $\mathcal{I}_\beta \subset \mathcal{I}$, where $\mathcal{I}_\alpha$, $\mathcal{I}_\beta$ contain the indices of clusters, which constitute clusters ℓ and m at level j, respectively. The density of cluster ℓ is given by: $p_j(\boldsymbol{x}|\ell) = \sum_{i\in\mathcal{I}_\alpha} \alpha_i p(\boldsymbol{x}|i)$, $\alpha_i = P(i)/\sum_{i\in\mathcal{I}_\alpha} P(i)$ if $i \in \mathcal{I}_\alpha$, and zero otherwise. $p_j(\boldsymbol{x}|m) = \sum_{i\in\mathcal{I}_\beta} \beta_i p(\boldsymbol{x}|i)$, where β_i obtains a similar definition. According to [25], the Gaussian integral is given by $\int p(\boldsymbol{x}|a)p(\boldsymbol{x}|b)\,dx = \mathcal{N}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_a + \boldsymbol{\Sigma}_b)$, where $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-d/2} \cdot |\boldsymbol{\Sigma}|^{1/2} \cdot \exp(-\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/2)$. Define the vectors $\boldsymbol{\alpha} = \{\alpha_i\}$, $\boldsymbol{\beta} = \{\beta_i\}$ of dimension K and the $K \times K$ symmetric matrix $\boldsymbol{G} = \{G_{ab}\}$ with $G_{ab} = \mathcal{N}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_a + \boldsymbol{\Sigma}_b)$, then the distance can be then written as $D(\ell,m) = (\boldsymbol{\alpha}-\boldsymbol{\beta})^\top \boldsymbol{G}(\boldsymbol{\alpha}-\boldsymbol{\beta})$. It turns out (see Fig. 2) that it is important to include the prior of the component in the dissimilarity measure. The modified $\mathcal{L}_2$ is then given by $\widetilde{D}(\ell,m) = \int \left(p_j(\boldsymbol{x}|\ell)P_j(\ell) - p_j(\boldsymbol{x}|m)P_j(m)\right)^2 dx$, which easily can be computed using a modified matrix $\widetilde{G}_{ab} = P(a)P(b)G_{ab}$.

3.4 Cluster Confusion Similarity Measure

Another natural principle is based on merging clusters, which have the highest confusion. Thus, when merging two clusters, the similarity is the probability of misassignment (PMA) when drawing examples from the two clusters seperately. Let $\boldsymbol{x}$ be an example from cluster $\mathcal{C}_k$ denoted by $\boldsymbol{x} \in \mathcal{C}_k$ and let $m = \arg\max_j P(j|\boldsymbol{x})$ be the model estimate of the cluster, then the PMA for all $\ell \neq m$ is given by:

$$E(\ell,m) = P(\ell \neq m) = \tag{11}$$
$$\int_{\mathcal{R}_m} p(\boldsymbol{x}|\ell)P(\ell)d\boldsymbol{x} + \int_{\mathcal{R}_\ell} p(\boldsymbol{x}|m)P(m)d\boldsymbol{x}$$

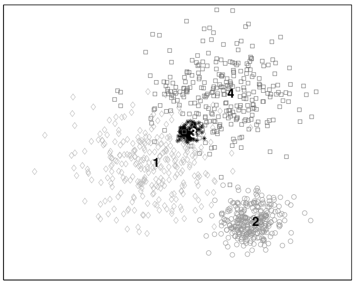
where $\mathcal{R}_m = \{\boldsymbol{x} : m = \arg\max_j P(j|\boldsymbol{x})\}$ and likewise for $\mathcal{R}_\ell$. In general, $E(\ell,m)$ can not be computed analytically, but can be approximated arbitrarily accurately by using an ancillary set of data samples drawn from the estimated model. That is, randomly select a cluster i with probability $P(i)$, draw a sample from $p(\boldsymbol{x}|i)$ and compute the estimated cluster $j = \arg\max_k P(k|\boldsymbol{x})$. Then estimate $P(\ell \neq m)$

[5] In the following sections we omit the possible dependence on y for notational convenience.

as the fraction of samples where $(i = \ell \wedge j = m)$ or $(j = \ell \wedge i = m)$.

3.5 Sample Dependent Similarity Measure

Instead of constructing a fixed hierarchy for visualization and interpretation of new data a sample dependent hierarchy can be obtained by merging a number of clusters relevant for a new data sample $\boldsymbol{x}$. The idea is based on level assignment described in Sec. 3.1. Let $P(k|\boldsymbol{x})$, $k = 1, 2, \cdots, K$, be the computed posteriors ranked in descending order and compute the accumulated posterior $A(\ell) = \sum_{k=1}^{\ell} P(k|\boldsymbol{x})$. The sample dependent cluster is then formed by merging the fundamental components $k = 1, 2, \cdots, m$ where $m = \min_\ell A(\ell) > \rho$, with e.g., $\rho = 0.9$.



Level	$\mathcal{L}_2$	modified $\mathcal{L}_2$	Error confus.
2	5={1,4} 2 3	5={1,4} 2 3	5={1,4} 2 3
3	6={1,2,4} 3	6={1,3,4} 2	6={1,3,4} 2

Figure 2: Hierarchical 2D clustering example with 4 Gaussian clusters. 1 and 4 have wide distributions, 2 more narrow, and 3 extremely peaked. The priors are $P(k) = 0.3$ for $k = 1, 2, 3$ and $P(3) = 0.1$. The table shows the construction of higher-level clusters, e.g., the $\mathcal{L}_2$ distance measure groups clusters 1 and 4 at level 2, which is due to the fact that distance is based on the shape of the distribution and not only its mean. This also applies to the other dissimilarity measures. At level 3, however, the $\mathcal{L}_2$ method absorbs cluster 4 into 5 to form cluster 6. The other methods absorbs cluster 3 at this stage. The reason is that the prior of cluster 3 is rather low, which is neglected in the $\mathcal{L}_2$ method.

4 Experiments

The hierarchical clustering is illustrated for segmentation of e-mails. Define the term-vector as a complete set of the unique words occurring in all the emails. An email histogram is the vector containing frequency of occurrence of each word from the term-vector and defines the content of the email. The term-document matrix is then the collection of histograms for all emails in the database. Suitable preprocessing of the data is required for good performance. This concerns: 1) removing words, which are too likely (stop words) or too unlikely[6]; 2) keeping only word stems; and 3) normalizing all histogram vectors to unit length[7]. After preprocessing the term-document matrix contains 1280 (640 for training and 640 for testing) e-mail documents, and the term-vector consists of 1652 words. The emails where annotated into the categories: *conference*, *job* and *spam*. It is possible to model directly from the term-document matrix, see e.g., [18, 22], however, we deploy the commonly used framework Latent Semantic Indexing (LSI) [5], which operates using a latent space of feature vectors. These are found by projecting term-vectors into a subspace spanned by the left eigenvectors associated with largest singular values of a singular value decomposition of the term-document matrix. We are currently investigating methods for automatic determination of the subspace dimension based on generalization concepts, however, in this work, the number of subspace components is obtained from an initial study of classification error on a cross-validation set. We found that a 5 dimensional subspace provides good performance. Fig. 3 presents a 3D scatter plot of the first 3 feature dimensions, viz. the largest principal components. Data seem to be well separated, however, parts of *job* and *conference* e-mails are mixed. Fig. 4 shows the performance of the USGGM algorithm, and in Fig. 5 the hierarchical representations are illustrated.

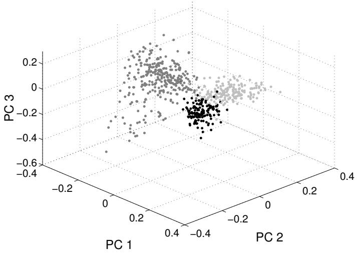


Figure 3: 3D scatter plot of the data. Three largest out of five principal components are displayed. Light grey color - *conference*, black - *job*, dark grey - *spam*. Data is well separated, however, there exists small confusion between job and conference e-mails.

[6] A threshold value for unlikely word up to approx. 100 occurrences has little influence on classification error. In the simulation the threshold was set to 40 occurrences.

[7] Another approach is to normalize the vectors to represent estimated probabilities, i.e., let vector sum to one. However, extensive experiments indicate that this approach give a feature space, which is not very appropriate for Gaussian mixture models.

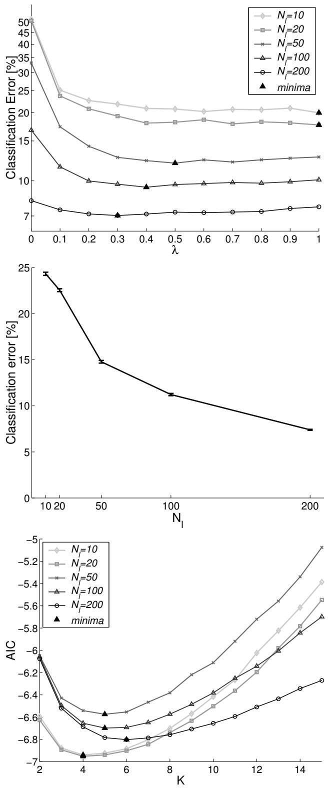


Figure 4: Average performance of the USGGM algorithm over 1000 repeated runs using $N_u = 200$ unlabeled examples and a variable number of labeled examples N_l. The algorithm parameter is set to $\gamma = 0.5$. Upper panel shows the performance as a function of the discount factor λ for unlabeled examples ($\lambda = 0$ corresponds to no unlabeled data). As expected, if few unlabeled examples are available, $N_l = 10, 20$, the optimal λ is close to one, and all available unlabeled data are fully used. As N_l increases λ decreases towards 0.3 for $N_l = 200$, indicating the reduced utility of unlabeled examples. The classification error is reduced approx. 26% using unlabeled data for $N_l = 10$, gradually decreasing to 1% for $N_l = 200$. The classification error for optimal λ as a function of N_l is shown in the middle panel. The lower panel shows number of components selected by the AIC criterion for optimal λ as described in Sec. 2.1. As N_l increase, also it is advantageous to increase the number of components.

y	k	$P(k\|y)$	Keywords
1	1	.7354	information, conference, call, workshop, university
	3	.0167	remove, address, call, free, business
	4	.2297	call, conference, workshop, information, submission, paper, web
	6	.0181	research, position, university, interest, computation, science
2	2	.6078	research, university, position, interest, science, computation, application, information
	6	.3922	research, position, university, interest
3	3	.6301	remove, call, address, free, day, business
	5	.3698	free, remove, call

Table 1: Keywords for the USGGM model. $y = 1$ is *conference*, $y = 2$ is *jobs* and $y = 3$ is *spam*.

Typical features as described in Sec. 3.2 and backprojecting into original term-space provides keywords for each cluster as given in Tab. 1. In Fig. 5 we choose to illustrate the hierarchies of individual class dependent densities $p(\boldsymbol{x}|y)$ using the modified $\mathcal{L}_2$ dissimilarity only. The cluster confusion measure is computational expensive if little overlap exist as many ancillary data are required. The modified $\mathcal{L}_2$ is computational inexpensive and basically treat dissimilarity as the cluster confusion, while the standard $\mathcal{L}_2$ do not incorporate priors. The *conference* class is dominated by cluster 1. This has keywords listed in Tab. 1, which are in accordance with the meaning of conference. The lower left panel shows the cluster level assignment distribution of test set emails, which are classified as conference emails cf. Sec. 3.1. Some obtain significant interpretation at level 1 (clusters 1-6), while others at a high level (cluster 9). Similar comments can be made for the *jobs* and *spam* classes.

For comparison, we further trained an unsupervised SGGM model and the results for a typical run are presented in Fig. 6. The top row illustrate the hierarchy formed by using the sample dependent, the modified $\mathcal{L}_2$ dissimilarity, and the cluster confusion similarity measures. For the sample dependent measure the numbers on top of the bars indicate the most frequent combinations of first level clusters. Clearly there is a significant resemblance among the sample dependent and the cluster confusion similarity hierarchies, e.g., higher level clusters formed by $\{1, 3\}$ and $\{2, 10\}$. However, inspection of the bottom row panels, which show the cluster confusion with the class labels, indicate that the cluster combinations of the sample dependent method is better aligned with the class labels. The modified $\mathcal{L}_2$ provides the best alignment of clusters with class labels at level 8 and is in

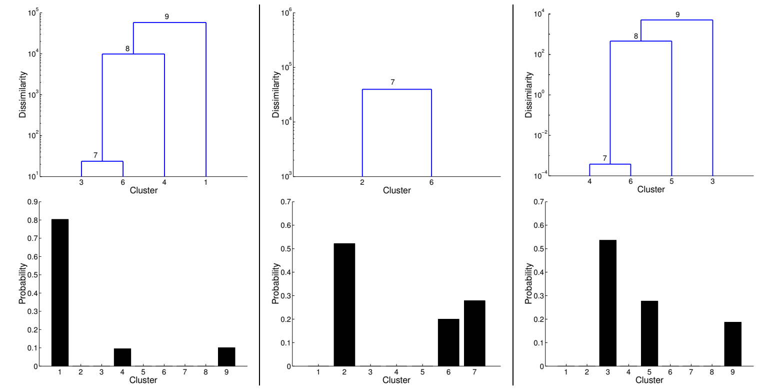
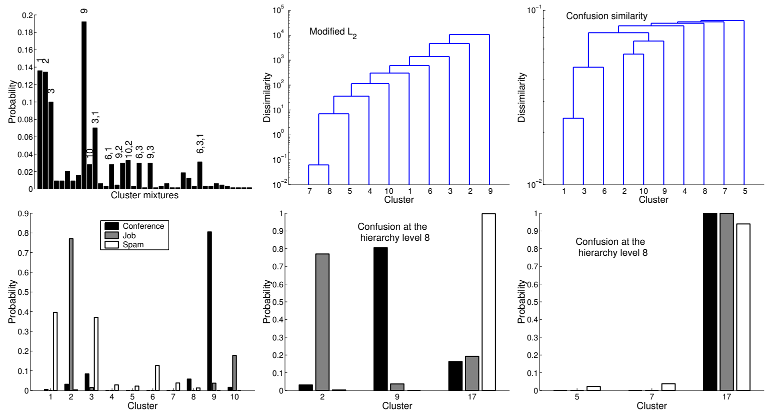


Figure 5: Hierarchical clustering using the USGGM model. Left column is class $y = 1$ *conference*, middle column $y = 2$ *jobs*, and right column is for $y = 3$ *spam*. Upper rows show the dendrogram using the modified $\mathcal{L}_2$ dissimilarity for each class, and the lower row the histogram of cluster level assignments for test data, cf. Sec 4.



1	free,remove	2	research,university,position	3	free,information,address,day
4	call,remove,address	5	free,service,site,sit,internet	6	call,remove,address,business
7	call,remove	8	free,address,state,quantity	9	information,conference,workshop
10	research,university,interest,position				

Figure 6: Unsupervised SGGM modeling of $p(\boldsymbol{x})$. Upper rows show the hierarchical structure. Left panel illustrates the sample dependent similarity measure Sec. 3.5, the middle panel the modified $\mathcal{L}_2$ dissimilarity measure, Sec. 3.3.1, and the right panel the cluster confusion measure Sec. 3.4. measure. Lower rows show the confusion of clusters with the annotated email labels at the first level in the hierarchy (left panel) and at level 8, where 3 clusters remain for the modified $\mathcal{L}_2$ dissimilarity (middle) and the cluster confusion measure (right panel). E.g., the black bars are the fraction of conference labeled test set emails ending up in a particular cluster. In addition, keywords for each cluster of the first level are also provided.

that respect superior to the other methods for the current data set. The keywords for clusters 2,10 and 9 provide perfect description of the *jobs* and *conference* emails, respectively. Keywords for the other clusters indicate that these mainly belong to the broad *spam* category.

5 Conclusions

This paper presented probabilistic agglomerative hierarchical clustering schemes based on the introduced unsupervised/supervised generalizable Gaussian mixture model (USGGM), which is an extension of [17]. The ability to learn from both labeled and unlabeled examples is important for many real world applications, e.g., text/webmining and medical decision support. The USGGM was successfully tested on a textmining example concerning segmentation of emails.

Using a probabilistic scheme allows for automatic cluster and hierarchy level assignment for unseen data, and provides further a natural technique for an interpretation of the clusters via prototype examples and features. In addition, three different similarities measures for cluster merging were presented and compared.

References

[1] H. Akaike: "Fitting Autoregressive Models for Prediction," *Ann. of the Inst. of Stat. Math.*, vol. 21, 1969, pp. 243–247.

[2] C.M. Bishop: *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[3] C.M. Bishop and M.E. Tipping: "A Hierarchical Latent Variable Model for Data Visualisation," *IEEE T-PAMI* vol. 3, no. 20, 1998, pp. 281–293.

[4] J. Carbonell, Y. Yang and W. Cohen: "Special Isssue of Machine Learning on Information Retriceal Introduction," *Machine Learning* vol. 39, 2000, pp. 99–101.

[5] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman: "Indexing by Latent Semantic Analysis," *Journ. Amer. Soc. for Inf. Science.*, vol. 41, 1990, pp. 391–407.

[6] C. Fraley: "Algorithms for Model-Based Hierarchical Clustering," *SIAM J. Sci. Comput.* vol. 20, no. 1, 1998, pp. 279–281.

[7] D. Freitag: "Machine Learning for Information Extraction in Informal Domains," *Machine Learning* vol. 39, 2000, pp. 169–202.

[8] L.K. Hansen, S. Sigurdsson, T. Kolenda, F.Å. Nielsen, U. Kjems and J. Larsen: "Modeling Text with Generalizable Gaussian Mixtures," In *Proc. of IEEE ICASSP'2000*, vol. 6, 2000, pp. 3494–3497.

[9] L.K. Hansen: "Supervised Learning with Labeled and Unlabeled Data," submitted for pulication 2001.

[10] T. Hastie and R. Tibshirani: "Discriminant Analysis by Gaussian Mixtures," *Jour. Royal Stat. Society - Series B*, vol. 58, no. 1, 1996, pp. 155–176.

[11] T. Honkela, S. Kaski, K. Lagus and T. Kohonen: "Websom — Self-organizing Maps of Document Collections, in *Proc. of Work. on Self-Organizing Maps*, Espoo, Finland, 1997.

[12] C.L. Jr. Isbell and P. Viola: "Restructuring Sparse High Dimensional Data for Effective Retrieval," in *Advances in NIPS 11*, MIT Press, 1999, pp. 480–486.

[13] T. Kolenda, L.K. Hansen and S. Sigurdsson: "Indepedent Components in Text," in *Adv. in Indep. Comp. Anal.*, Springer-Verlag, pp. 241–262, 2001.

[14] J. Larsen, L.K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda: "Webmining: Learning from the World Wide Web," *Computational Statistics and Data Analysis*, 2001.

[15] J. Larsen, L.K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda: "Webmining: Learning from the World Wide Web," in *Proc. of Nonlinear Methods and Data Mining*, Rome, Italy, 2000, pp. 106–125.

[16] M. Meila and D. Heckerman: "An Experimental Comparison of Several Clustering and Initialisation Methods," in *Proc. 14th Conf. on Uncert. in Art. Intel.*, Morgan Kaufmann, 1998, pp. 386–395.

[17] D.J. Miller and H.S. Uyar: "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," in *Advances in NIPS 9*, 1997, pp. 571–577.

[18] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell: "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, vol. 39, 2000, pp. 103–134.

[19] A. Szymkowiak, J. Larsen and L.K. Hansen: "Hierarchical Clustering for Datamining," in *Proc. 5th Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies KES'2001*, Osaka and Nara, Japan, 6–8 Sept., 2001.

[20] N. Vasconcelos and A. Lippmann: "Learning Mixture Hierarchies," in *Advances in NIPS 11*, 1999, pp. 606–612.

[21] E.M. Voorhees: "Implementing Agglomerative Hierarchic Clustering Ulgorithms for Use in Document Retrieval," *Inf. Proc. & Man.*, vol. 22, no. 6, 1986, pp. 465–476.

[22] A. Vinokourov and M. Girolami: "A Probabilistic Framework for the Hierarchic Organization and Classification of Document Collections," submitted for *Journal of Intelligent Information Systems*, 2001.

[23] A.S. Weigend, E.D. Wiener and J.O. Pedersen: "Exploiting Hierarchy in Text Categorization," *Information Retrieval*, vol. 1, 1999, pp. 193–216.

[24] C. Williams: "A MCMC Approach to Hierarchical Mixture Modelling," in *Advances in NIPS 12*, 2000, pp. 680–686.

[25] D. Xu, J.C. Principe, J. Fihser and H.-C. Wu: "A Novel Measure for Independent Component Analysis (ICA)," in *Proc. IEEE ICASSP98*, vol. 2, 1998, pp. 1161–1164.

APPENDIX E

Contribution: NNSP 2002

This appendix contain the article *Clustering of Sun Exposure Measurements.* in Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII, editor: D. Miller and T. Adali and J. Larsen and M. Van Hulle and S. Douglas, pages: 727–735. Author list: A. Szymkowiak-Have, J. Larsen, L.K. Hansen, P.A. Philipsen, E. Thieden and H.C. Wulf. Own contribution approximated to 50%.

CLUSTERING OF SUN EXPOSURE MEASUREMENTS

A. Szymkowiak-Have[1], J. Larsen[1], L.K. Hansen[1],
P.A. Philipsen[2], E. Thieden[2], H.C. Wulf[2]

[1]Informatics and Mathematical Modelling, Building 321
Technical University of Denmark, DK-2800 Lyngby, Denmark
Phone: +45 4525 3900,3923,3889 Fax: +45 4587 2599
E-mail: asz,jl,lkh@imm.dtu.dk Web: eivind.imm.dtu.dk

[2]Department of Dermatology, Bispebjerg Hospital
University of Copenhagen, Bispebjerg Bakke 23
DK-2400 Copenhagen, Denmark

Abstract. In a medically motivated sun-exposure study, questionnaires concerning sun-habits were collected from a number of subjects together with UV radiation measurements. This paper focuses on identifying clusters in the heterogeneous set of data for the purpose of understanding possible relations between sun-habits exposure and eventually assessing the risk of skin cancer. A general probabilistic framework originally developed for text and web mining is demonstrated to be useful for clustering of behavioral data. The framework combined Latent Semantic indexing like approach with probabilistic clustering based on the generalizable Gaussian mixture model.

INTRODUCTION

In the studied sun-exposure experiment, questionnaires concerning sun-habits were collected from 187 subjects. In addition, daily UV radiation were measured at a 10 minute sampling rate using a specially designed "sun-watch". The ultimate objective is to relate the heterogeneous data of sun-habits, UV dose and other data (e.g., medical records) with the purpose of assessing the risk of skin cancer for individual subjects. This paper focuses on the subtask of identifying relevant structure in the combined data set of sun habit diaries and daily UV dose measurements. We aim at identifying relevant structure using hierarchical probabilistic clustering. Although the method presented in [7] can be invoked for hierarchical clustering, we resort to simple probabilistic clustering in this work. The diary records can be viewed as a vector of categorical data, whereas the daily UV dose is a continuous measurement which is measured for different persons during 138 days. The long-term theoretical aim is to identify a hierarchical probabilistic clustering model which efficiently handles combinations of categorical and continuous data. However, the idea of the present paper is to study the capabilities of our flexible multimedia text and images data [4, 5, 6, 7, 9] mining framework for analysis and understanding of behavioral data.

SUN EXPOSURE STUDY

A specially designed device, measuring received sun radiation (*PID*), was given to the group of subjects. In addition, subjects were requested to fill out a diary concerning their sun behaviors during each day of the study (for more details, see [10]). Eight selected questions are presented here:

Variable	**Values**
1. Holiday	yes/no
2. Abroad	yes/no
3. Sun Bathing	yes/yes-solarium/no
4. Naked Shoulders	yes/no
5. On the Beach/Water	yes/no
6. Sun Factor Number	no/26 values in range 1-60
7. Sunburned	no/red/hurts/blisters
8. Size of Sunburned Area	no/little/medium/large

Thus, two types of data were collected: continuous measurements of the sun UV radiation (*PID*) and categorical diary records. Each diary record is represented by an 8 dimensional vector and describes a specific behavior of the particular person during the particular day. The total number of possible patterns for the presented set of questions equals 20736, however, only a small fraction of 423 patterns actually exist in the investigated data set.

PREPROCESSING

Latent Semantic Indexing (LSI) [2] was developed for text and multimedia mining, see [4, 5]. In this study we pursue a similar idea, which enables to combining different types of data into a common framework. Figure 1 presents the general framework of preprocessing, clustering and data post-processing. In the first step, data is windowed creating vectors that contain data from consecutive days. The optimal size of the window is an issue to be addressed. For example, taking the full set of records belonging to a given person will produce a set of points in the space that will not form any particular clusters, since each of them will contain most of the observed patterns. On the other hand, taking one diary record at the time will significantly increase the computational complexity. In the experiments a window of size 7

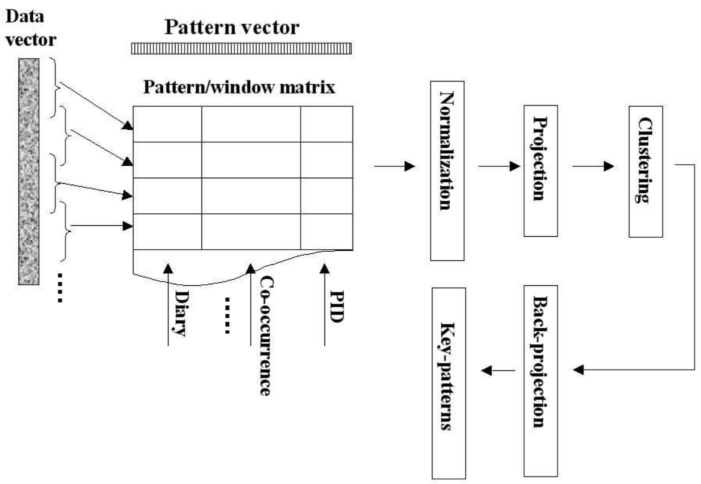


Figure 1: Framework for data clustering: 1) the data is windowed into several histogram vectors and together with the co-occurrence matrix and the *PID* histogram forms a pattern/window matrix. 2) data is then normalized and projected onto the orthogonal singular value decomposition space. 3) the Gaussian mixture algorithm is used to cluster the data. 4) In order to interpret the results, cluster centers are back-projected to the original space where key-patterns are identified.

is used. This was decided after several experiments, taking into account stationarity of the clustering and complexity level. In the final paper we plan to invoke the concept of generalization for optimal window selection, see further [3].

Originally, the pattern/window matrix is formed from the histogram vectors achieved by counting occurrences of every found pattern in the window. However, the histograms does not convey time ordering information. It is possible to include time information by considering the co-occurrence matrix of joint occurrences of neighbor patterns in the window. There are 20736^2 possible co-occurences but only 1509 were present in the actual data set. The continuous sun radiation measurements were quantized in order to fit the presented framework. Both diary histograms, the co-occurrence matrix and sun radiation are screened against rare patterns by removing patterns which have occurrence below a certain threshold.

The next step involves normalization of the pattern/window matrix. Two types of normalization are performed. First, each window vector is scaled to unity length, and then, pattern vectors are scaled to zero mean and unit variance over training samples. The three component matrices (diary-window histograms, co-occurrence and *PID* histograms) are then separately projected onto the few principal component directions found by singular value decomposition (SVD). Finally, the generalizable Gaussian mixture model is used for clustering in the subspace.

UNSUPERVISED GAUSSIAN MIXTURE MODEL

The Gaussian mixture model was previously addressed in [4, 6, 8]. The K component mixture of Gaussian densities of the d-dimensional feature vector $\boldsymbol{x}$ is defined as:

$$p(\boldsymbol{x}) = \sum_k p(\boldsymbol{x}|k) \cdot p(k) \tag{1}$$

where $p(\boldsymbol{x}|k) \equiv \mathcal{N}(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ are Gaussian densities and $p(k)$ are nonnegative mixture proportions such that $\sum_k p(k) = 1$. The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are estimated from the training data set $\mathcal{D} = \{\boldsymbol{x}_n, n = 1 \ldots N\}$ by minimizing negative log-likelihood cost function of the form: $\mathcal{L} = -\sum_n log(p(\boldsymbol{x}_n|k))$ through expectation-maximization method. In order to ensure generalizability, parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are estimated from the disjoint sets of observations and the optimal number of mixture components is found by the AIC-criterion [1, 3]. The complete algorithm for generalizable Gaussian mixture model (GGM) can be found in [4, 7].

OUTLIER DETECTION

The Gaussian mixture model models the data density In order to spot an outlier, which indicate non-stationarity in data, the cumulative probability [6, 7] is computed $Q(t) = \text{Prob}(x \in \mathcal{R}), \mathcal{R} = \{\boldsymbol{x} : p(\boldsymbol{x} < t)\}$ for all thresholds t. Thus, the outliers occupy lower part of the cumulative curve.

PROTOTYPES

In order to find key-patterns corresponding to each of the clusters, centers $\boldsymbol{\mu}$ need to be back-projected to the original space of normalized histograms[1]. Furthermore, the used framework makes it possible to describe the behavior of every new person in the experiment by using both cluster assignment and associated key-patterns. The confidence of assigning the person into the given cluster k can be expressed by the posterior probability:

$$p(k|Per) = \frac{1}{N} \sum_i p(k|Per, \boldsymbol{x}_i) \cdot p(\boldsymbol{x}_i), \tag{2}$$

where $\boldsymbol{x}_i$ is a feature vector of the size d and $i = 1, 2, \ldots, N$. The number of feature vectors N is different for every person and depends on the number of returned diary records and the window size.

[1] Another way would be to project the most probable feature vectors from each of the clusters found e.g. by Monte Carlo sampling.

RESULTS

The set of 19171 diary records and corresponding *PID* values were selected for the clustering experiments. Data are complete i.e., there is no missing records or *PID* values. The missing record problem for the current data set was partly addressed in [10]. The sun behaviors of 187 subjects during summer period were collected. Of this 10 persons were hold out for testing. Sun exposure measurements were quantized into 4 values. The slicing window of size 7 was applied forming 2580 training and 158 test feature vectors. Each feature vector consist of the diary histogram, the co-occurrence matrix and the *PID* histogram. The diary histogram is reduced from 423 to 97 patterns by removing rare patterns. In a similar way, the co-occurrence matrix is reduced from 1509 to the 80 most often occurring pairs of patterns. Each of these matrices are projected separately on the orthogonal directions found by SVD. For both diary and co-occurrence the 9 largest eigenvalues is used and 3 for *PID* data[2].

The investigation was performed of the importance of the co-occurrence matrix and the *PID* histograms for the clustering. The results of the experiments are collected in the tables 1, 2, 3 and 4.

In the experiments hard assignment GGM model [4] is used, i.e., the parameters of the clusters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ were estimated from the set of samples assigned to each of the clusters. In order to achieve a more detailed cluster structure one could use soft GGM [9, 7].

In the tables 1, 2, 3 and 4 the results of back-propagation are shown. The key-patterns, associated probabilities and description of the clusters are provided. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The third gives the probabilities for the key-patterns and the fourth column presents a general description of the cluster based on the key-patterns.

In table 1 the results of clustering of the diary histograms are shown. The presented patterns are equivalent to the set of questions given in section: *Sun Exposure Study*. For example: pattern 10111 describes the following set of answers: 1. holiday - "yes", 2. abroad - "no", 3. sun bathing - "yes", 4. naked shoulders - "yes", 5. on the beach - "yes", remaining questions 6,7 and 8 - "no", or pattern 0: all the questions where answered "no" or pattern 1: 1. holiday - "yes" and the rest of the questions from 2 to 8 - "no". This rule for describing patterns hold as well in the case of table 2, 3 and 4.

Table 2 presents key-patterns for clustering diary histograms combined with *PID* histograms. Eight clusters were found. Diary key-patterns are explained in table 1. Patterns corresponding to the *PID* histograms are marked with the subscript "*PID*". Four different values of *PID* from 0 to 3 are observed: 0 corresponds to the very low sun radiation and 3 describes very high one. This rule for describing *PID*-patterns hold as well in the case of table 4.

[2] The decision was made based on the shape of the eigenvalue curve but a more elaborate selection can be invoked using the concept of generalization [4].

#.	Key-Pattern	Probability.	Description
1.	10001,11,10111	0.33,0.32,0.19	holiday, on the beach, sun bathing
2.	0	0.98	working - no sun
3.	1	0.9	on holiday - no sun
4.	0,0001,1	0.4,0.27,0.18	working naked shoulders - no sun
5.	1,1101	0.67,0.17	holiday, naked shoulders
6.	1011,1001,10011	0.47,0.17,0.16	holiday , sun bathing
7.	11	0.5	holiday abroad - no sun
8.	10111,0001,1001	0.45,0.17,0.13	holiday, sun bathing, naked shoulders
9.	0000001	0.05	no sun, sunburned - red
10.	0	0.99	working - no sun

Table 1: Key-patterns for clustering diary histograms. In the first column the cluster number is shown. Second column contains the most probable patterns for the cluster. The presented pattern numbers are equivalent to the set of questions given in section: *Sun Exposure Study.* For example: pattern 10111 gives the following set of answers: holiday - yes, abroad - no, sun bathing - yes, naked shoulders - yes, on the beach - yes, remaining questions 6,7 and 8 - no, or pattern 0 means that all the questions where answered "no". Third column gives the probabilities for the key-patterns, and fourth column presents a general description of cluster.

In table 3 the key-patterns for clustering diary histograms combined with co-occurrence matrix are presented. The diary key-patterns are explained in table 1. The co-occurring patterns are shown with the dash between them e.g., "0-1" means that a pattern working is followed by pattern holiday, pattern "1-10011" means that holiday without sun was followed by holiday spent on the beach. This rule for describing co-occurrence patterns hold as well in the case of table 4.

Table 4 shows the key-patterns for clustering diary histograms combined with co-occurrence matrix and *PID* histograms. The diary key-patterns are explained in table 1. The co-occurred patterns are explained in table 3 and the *PID* patterns in table 2. Both the *PID* values and the co-occurrence pairs are likely to appear as key-patterns. This could suggest that joining time information and the sun exposure measurements are important for the clustering. Moreover, the description of the clusters is more explicit.

In figure 2 the probability of observing certain groups of behaviors in the clusters together with registered sun exposure values are presented. Clustering was done using full pattern/window matrix for which keywords are displayed in table 4. Five behaviors are specified: *working - no sun exposure*, *holiday - no sun exposure*, *sun exposure* describes mild sun behaviors often on the beach or naked shoulders without sun-screen and without sunburns, *using sun-block* and diary records with reported *sunburns.* In the bottom figure the observed sun exposure measurements are presented. For example cluster number 6 groups behaviors marked as *working - no sun* and corresponding *PID* values are low. Opposite, cluster no. 5 contains records with reported sunburns, sun exposure and using sun-block and consequently *PID* values are high.

For the same clustering setting the cluster probabilities were calculated Eq. (2) for 10 test subjects. Together with key-patterns presented in table 4

#	Key-Pattern	Probability.	Description
1.	1001,1000, 1_{PID},10011	0.31,0.26, 0.16,0.11	holiday, naked shoulders, small *PID*
2.	11,0001,0, 2_{PID},0_{PID}	0.29,0.2,0.17, 0.16,0.15	holiday abroad, working
3.	1,11	0.39,0.12,	holiday
4.	1011,2_{PID}, 3_{PID},0001	0.0.31,0.25, 0.14,0.13	naked shoulders, high sun radiation
5.	1,2_{PID},3_{PID},10001	0.2,0.17,0.16,0.14,0.12,0.1	holidays, high *PID*
6.	1_{PID},0_{PID}	0.14,0.13	low *PID*
7.	3_{PID},1001, 2_{PID},10011	0.22,0.22, 0.16,0.15	holiday, naked, shoulders, high *PID*
8.	0,0_{PID}	0.6,0.4	no sun

Table 2: Key-patterns for clustering diary histograms combined with *PID* histograms. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The diary key-patterns are explained in table 1. Patterns corresponding to the *PID* histograms are marked with the subscript "*PID*". Four different values of *PID* are observed: 0 corresponds to very low sun radiation and 3 describes very high one. Third column gives the probabilities for the key-patterns and fourth column presents general description of cluster based on the key-patterns.

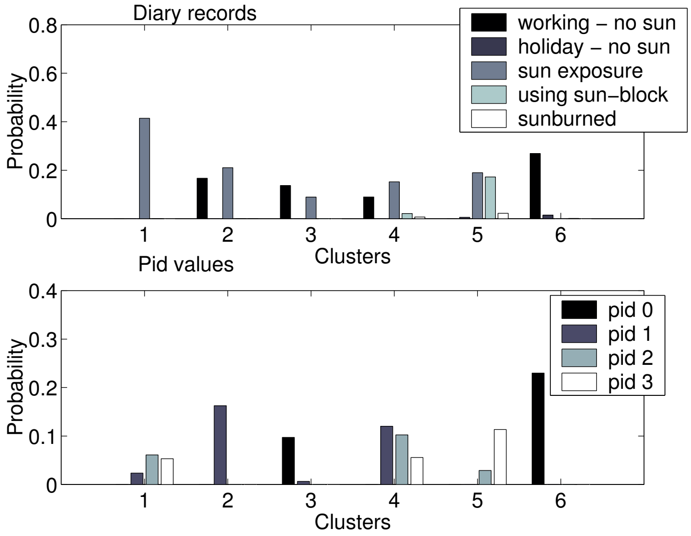


Figure 2: The probability of observing certain groups of behaviors in the clusters together with registered sun exposure values. Key-patterns for the clusters are presented in the table 4. For each cluster grouped behaviors from diary records are presented on the upper plot and corresponding *PID* is shown on the lower figure.

it gives a good description of the behavior of the particular persons during the whole period of the experiment. For all test persons there is a large

#	Pattern	Probability.	Description
1.	1001,1101-1101	0.27,0.13	holiday,naked sholders
2.	1001,1101-1101,1	0.26,0.21,0.1	holiday,naked sholders
3.	0001,1001-0, 10111,0-1001	0.17,0.12, 0.11,0.1	working, naked shoulders
4.	11,11-11	0.14,0.11	holiday, abroad
5.	0001,1001-0, 1001,0-1001	0.27,0.14, 0.13,0.1	holiday or working, naked shoulders
6.	1001,0,0-1,1-0,1,1-1	0.29,0.19,0.16,0.14,0.1,0.09	work - holiday, no sun
7.	10011	0.19	holiday, on the beach
8.	10001,1-10011	0.21,0.12	holiday, on the beach
9.	0-0,0,0-1,1-0	0.36,0.35,0.12,0.12	working - no sun
10.	1001,1-1101,1101-1, 1101-1101,1011	0.26,0.16,0.12, 0.12,0.11	holiday, naked shoulders

Table 3: Key-patterns for clustering diary histograms combined with co-occurrence matrix. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The diary key-patterns are explained in table 1. The co-occurring patterns are shown with the dash between them e.g., "0-1" means that a pattern working is followed by pattern holiday. Third column gives the probabilities for the key-patterns and fourth column presents general description of cluster based on the key-patterns.

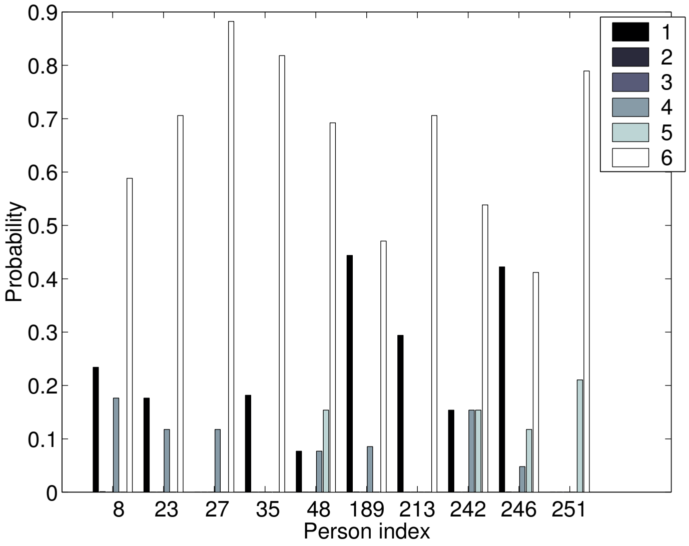


Figure 3: Cluster probabilities calculated for the 10 test persons Eq. (2). Person index is shown on the x-axes and different grey level colors corresponds to six clusters. Key-patterns are given in table 4.

probability of the cluster no. 6 that describes working and no sun exposure. However, some of the periods are described by other behaviors. For example for person no. 251 there is high probability component for cluster no. 5

#	Pattern	Probability	Description
1.	1001,0001,1101-1101	0.15,0.1,0.09	naked shoulders
2.	0,1_{PID},0-0,0001	0.17,0.16,0.14,0.13	working, low sun radiation
3.	1001-0,0-1001, 0,1-0,0-1,0_{PID}	0.17,0.14 ,0.14,0.12,0.11,0.1	no sun radiation, holiday-work
4.	1_{PID},2_{PID}	0.12,0.1	medium sun exposure
5.	3_{PID},11,11-11	0.11,0.11,0.09	holiday, high sun radiation
6.	0-0,0,0_{PID}	0.29,0.27,0.23	working, no sun

Table 4: Key-patterns for clustering the diary histograms combined with the co-occurrence matrix and the *PID* histograms. In the first column the cluster number is displayed. Second column contains the most probable patterns for the cluster. The diary key-patterns are explained in table 1. The co-occurred patterns are explained in table 3 and the *PID* patterns in table 2. Third column gives the probabilities for the key-patterns and fourth column presents general description of cluster based on the key-patterns.

describing holidays with high sun radiation. Persons no. 213 and 35 can be well described by clusters 6 (working, no sun) and 1 (naked shoulders) while person no. 23 by clusters 6, 1 and 4 (medium sun exposure).

CONCLUSION

This paper discusses using an Latent Semantic Indexing like method for processing and clustering categorical data. Moreover, it provides the possibility for combining multiple date types into a common vector space framework. We applied the method to analysis a combination of categorical diary data and real valued sun radiation measurements. Using the analogy to textmining we proposed methods for interpretation of the identified clusters. This scheme allows for evaluating the significance of various feature representations. For the specific data set we addressed the role of different representations. Preliminary results indicate that the sequence information and UV dose measurements contribute to stabilizing the clustering model and its interpretation.

REFERENCES

[1] H. Akaike, "Fitting Autoregresive Models for Predition," **Ann. of the Ins. of Stat. Math.**, vol. 21, pp. 243–247, 1969.

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis," **J. Amer. Soc. for Inf. Science**, vol. 41, pp. 391–407, 1990.

[3] L. Hansen and J. Larsen, "Unsupervised Learning and Generalization," in **Proceedings of the 1996 IEEE International Conference on Neural Networks**, Washington DC, USA, 1996, pp. 25–30.

[4] L. Hansen, S. Sigurdsson, T. Kolenda, F. Nielsen, U. Kjems and J. Larsen, "Modeling text with generalizable gaussian mixtures," in **Proceedings of IEEE ICASSP'2000**, 2000, vol. VI, pp. 3494–3497.

[5] T. Kolenda, L. K. Hansen, J. Larsen and O. Winther, "Independent component analysis for understanding multimedia content," 2002, **submitted for NNSP2002**.

[6] J. Larsen, L. Hansen, A. Szymkowiak-Have, T. Christiansen and T. Kolenda, "Webmining: Learning from the World Wide Web," **Computational statistics and data analysis**, vol. 38, pp. 517–532, 2002.

[7] J. Larsen, A. Szymkowiak and L. Hansen, "Probabilistic Hierarchical Clustering with Labeled and Unlabeled Data," **International Journal of Knowledge-Based Intelligent Engineering Systems**, vol. 6, no. 1, pp. 56–62, 2002.

[8] B. Ripley, **Pattern Recognition and Neural Networks**, Cambridge University Press, 1996.

[9] A. Szymkowiak, J. Larsen and L. Hansen, "Hierarchical Clustering for datamining," in N. Babs, L. Jain and R. Howlett (eds.), **Proc. 5th Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies**, 2001, pp. 261–265.

[10] A. Szymkowiak, P. Philipsen, J. Larsen, L. Hansen, E. Thieden and H. Wulf, "Impuating missing values in diary records of sun-exposure study," in D. Miller, T. Adali, J. Larsen, M. V. Hulle and S. Douglas (eds.), **Proceedings of IEEE Workshop on Neural Networks for Signal Processing XI**, Falmouth, Massachusetts, 2001, pp. 489–498.

APPENDIX F

Contribution: special issue CSDA 2002

This appendix contain the article *Webmining: Learning from the World Wide Web.* in special issue of Computational Statistics and Data Analysis, volume: 38, pages: 517–532. Author list: J. Larsen, L.K. Hansen, A. Szymkowiak-Have, T. Christiansen and T. Kolenda. Own contribution approximated to 10%.

Webmining: Learning from the World Wide Web

Jan Larsen, Lars Kai Hansen, Anna Szymkowiak,
Torben Christiansen and Thomas Kolenda*
Department of Mathematical Modeling,
Richard Petersens Plads, Building 321
Technical University of Denmark
DK-2800 Kongens Lyngby, Denmark
Phone: +45 4525 3923,3889,
Email: jl,lkhansen@imm.dtu.dk, Web: eivind.imm.dtu.dk

Abstract: Automated analysis of the world wide web is a new challenging area relevant in many applications, e.g., retrieval, navigation and organization of information, automated information assistants, and e-commerce. This paper discusses the use of unsupervised and supervised learning methods for user behavior modeling and content-based segmentation and classification of web pages. The modeling is based on independent component analysis and hierarchical probabilistic clustering techniques.

Keywords: Webmining, unsupervised learning, hierarchical probabilistic clustering

1. Introduction

Webmining is an increasingly important and very active research field which adapts advanced machine learning techniques for understanding the complex information flow of the world wide web, see e.g., (Nigam 00, Weigend 99). Web data are fundamentally multimedia streams of text, sound, images, and various database information. While optimal information retrieval, navigation or organization requires mining of all media modalities, this paper focuses on textmining and user behavior modeling.

Textmining (Hansen 00, Landuaer 98) is used to categorize text according to topic, to spot new topics, and in a broader sense to create more intelligent searches, e.g., by WWW search engines. Textmining proceeds by pattern

*This research is supported by the Danish Research Councils through the THOR Center for Neuroinformatics, Distributed Multimedia Technologies and Applications program within Center for Multimedia, and the Signal and Image Processing for Telemedicine (SITE) program. We would like to thank Jens Hørlück for assistance and support in setting up the user behavior modeling project.

recognition based on text features, typically document summary statistics. While numerous high-level language models for extraction of text features exists, simple summary statistics are still preferred because they are compact representation and can be adapted automatically and continuously, without costly manual intervention of language expertise.

Modeling the user's behavior when navigating a web site is very relevant in e-commerce applications (Cooley 99, Mobasher 99, Pei 00, Shahabi 97, Spiliopoulou 99, Yan 96). User modeling can be divided in three levels of functionality: the first level concerns automatic segmentation of users who display similar behavior. Second level concerns automatic classification of users using expert annotations of identified user segments. The third, and most elaborate level, involves interactive web pages continuously adapted to the user's behavior. This paper addresses merely automatic segmentation.

Section 2. describes a probabilistic hierarchical clustering framework based on the generalizable Gaussian mixture (GGM) model, which is reviewed. In section 3. we discuss the use of the GGM for supervised learning. Section 4. presents webmining applications using the methods of Sections 2.–3. covering: classification of webpages, hierarchical segmentation of emails, and user behavior segmentation.

2. Hierarchical Probabilistic Clustering

2.1. Generalizable Gaussian Mixture Model

The Gaussian mixture model is a very flexible pattern recognition device, see, e.g., (Ripley 96) for a review. The K component Gaussian mixture density of a feature vector $\boldsymbol{x}$ of dimension d, is defined as

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} P(k)p(\boldsymbol{x}|k,\boldsymbol{\theta}_k), \quad p(\boldsymbol{x}|k,\boldsymbol{\theta}_k) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \tag{1}$$

where the component Gaussians are mixed with proportions $\sum_k P(k) = 1$, $P(k) \geq 0$, and $\boldsymbol{\theta}_k \equiv \{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k\}$ is a parameter vector. The parameters are estimated from a set of examples $\mathcal{D} = \{\boldsymbol{x}_n | n = 1, \cdots, N\}$. Traditionally mixture densities are estimated using maximum likelihood (ML), e.g., through various expectation-maximization (EM) methods (Ripley 96). The (negative log-) likelihood cost function is defined by $S_N(\boldsymbol{\theta}) = \sum_{n=1}^{N} -\log p(\boldsymbol{x}_n|\boldsymbol{\theta})$ and $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} S_N(\boldsymbol{\theta})$ are the estimated parameters. The objective of modeling is to ensure that the generalization error, defined as the expected cost on independent data, $G(\widehat{\boldsymbol{\theta}}) = -\int \log p(\boldsymbol{x}|\widehat{\boldsymbol{\theta}})p^\circ(\boldsymbol{x})\,d\boldsymbol{x}$ is minimal. Here $p^\circ(\boldsymbol{x})$ denotes the "true" density.

The Gaussian mixture model is extremely flexible and simply minimizing the above cost function will lead to an "infinite overfit". This solution is optimal for the training set, but unfortunately has a generalization error roughly equal to that of the single component Gaussian model, as the singular components have zero measure w.r.t. test data[1]. This instability has lead to much confusion in the literature and needs to be addressed carefully. Basically, there is no way to distinguish generalizable from non-generalizable solutions if we only consider the likelihood function. The only way to ensure generalizability is to invoke the concept of generalization in the estimation procedure. The most common remedy is to bias the distributions so that they have a common shared covariance matrix, see e.g., (Hastie 96). In fact, classical EM algorithms only work under this assumption. A more principled method is to invoke regularization in terms of priors in a Bayesian framework (Rasmussen 00).

Here we adopt the Generalizable Gaussian Mixture model presented in (Hansen 00) which combines three approaches to ensure generalizability. First, we compute centers and covariances on different resamples of the data set. Secondly, we make an exception rule for sparsely populated components in which the covariance matrix defaults to the scaled full-sample covariance matrix. Thirdly, we estimate the number of mixture components by the AIC-criterion (Akaike 69, Hansen 96). The algorithm allows for individual component covariance matrices which enables a flexible local metric in contrast to methods assuming common covariance matrix, hence a global metric.

The Generalizable Gaussian Mixture algorithm is a modified EM procedure (Dempster 77) and is provided in Figure 1 for a fixed number of mixture components, K.

2.2. Hierarchical Clustering

There are numerous contributions within hierarchical clustering (see e.g., (Ripley 96)). Here the focus is to construct a relatively simple agglomerative hierarchical clustering using a probabilistic model which is based on the work in (Szymkowiak 01). For recent approaches to full hierarchical probabilistic clustering techniques the reader is referred to (Vasconcelos 99, Williams 00).

Define $p_j(\boldsymbol{x}|k)$ as the conditional probability[2] density of $\boldsymbol{x}$ for cluster $\mathcal{C}_k^j$ $k = 1, 2, \cdots, K - j + 1$ in layer $j = 1, 2, \cdots, K$ of a hierarchy. Further define $P_j(k)$ as the priors of the clusters (mixing proportions). At the most detailed level $j = 1$, the density is modeled by the GGM described above, i.e., $p_1(\boldsymbol{x}|k)$ are Gaussian densities. At each consecutive level two clusters with minimum distance are merged until we reach one cluster at level $j = K$. As distance measure we suggest to use the symmetric Kullback-Leibler divergence[3] between

[1] The cost function has a trivial (infinite) minimum attained by setting $\boldsymbol{\mu}_k = \boldsymbol{x}_k$ for $k = 1, \cdots, K-1$, and $\boldsymbol{\Sigma}_k = \mathbf{0}$. The remaining K'th Gaussian is adapted to the remaining $N - K + 1$ data points, with $\boldsymbol{\mu}_K = (N - K + 1)^{-1} \sum_{n=K}^{N} \boldsymbol{x}_n$, and $\boldsymbol{\Sigma}_K = (N - K + 1)^{-1} \sum_{n=K}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}_K)(\boldsymbol{x}_n - \boldsymbol{\mu}_K)^\top$.

[2] For notation convenience, we omitted the condition on the model parameters in what follows.

[3] See e.g., (Ripley 96) for the classical Kullback-Leibler definition.

Figure 1: *Generalizable Gaussian Mixture Algorithm.*

```
Initialization for $K$ components
    1. Compute the mean vector $\boldsymbol{\mu}_0 = N^{-1}\sum_n \boldsymbol{x}_n$.
    2. Compute the covariance matrix of the data set:
       $\boldsymbol{\Sigma}_0 = N^{-1}\sum_n(\boldsymbol{x}_n - \boldsymbol{\mu}_0)(\boldsymbol{x}_n - \boldsymbol{\mu}_0)^\top$.
    3. Initialize $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.
    4. Initialize $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_0$.
    5. Initialize $P(k) = 1/K$.
Repeat until convergence
    1. Compute $p(k|\boldsymbol{x}_n) = p(\boldsymbol{x}_n|k)p(k)/\sum_\ell p(\boldsymbol{x}_n|\ell)p(\ell)$ and assign $\boldsymbol{x}_n$
       to the most likely component.
    2. Split the data set in two parts $\mathcal{D}_{\boldsymbol{\mu}}$, $\mathcal{D}_{\boldsymbol{\Sigma}}$. Often 50/50 splitting is
       used.
    3. For each $k$ estimate $\boldsymbol{\mu}_k$ on the points in $\mathcal{D}_{\boldsymbol{\mu}}$ assigned to component $k$.
    4. For each $k$ estimate $\boldsymbol{\Sigma}_k$ on the points in $\mathcal{D}_{\boldsymbol{\Sigma}}$ assigned to component $k$. If the number of data points assigned to the $k$'th component,
       $N_k$, is less than $d+1$, then $\boldsymbol{\Sigma}_k \leftarrow (N_k\boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_0)/(N_k+1)$.
    5. Estimate $P(k)$ as the frequency of assignments to component $k$.
```

the mixture components, as defined by

$$D(k_1,k_2) = \frac{1}{2}\int p(\boldsymbol{x}|k_1)\log\frac{p(\boldsymbol{x}|k_1)}{p(\boldsymbol{x}|k_2)}\,d\boldsymbol{x} + \frac{1}{2}\int p(\boldsymbol{x}|k_2)\log\frac{p(\boldsymbol{x}|k_2)}{p(\boldsymbol{x}|k_1)}\,d\boldsymbol{x} \tag{2}$$

For layer $j=1$ in which the cluster densities are Gaussian the distance can be expressed as (Szymkowiak 01):

$$\begin{aligned} D_1(k_1,k_2) = {} & -\frac{d}{2} + \frac{1}{4}\left(\mathrm{Tr}[\boldsymbol{\Sigma}_{k_1}^{-1}\boldsymbol{\Sigma}_{k_2}] + \mathrm{Tr}[\boldsymbol{\Sigma}_{k_2}^{-1}\boldsymbol{\Sigma}_{k_1}]\right) + \\ & \frac{1}{4}(\boldsymbol{\mu}_{k_1} - \boldsymbol{\mu}_{k_2})^\top(\boldsymbol{\Sigma}_{k_1}^{-1} + \boldsymbol{\Sigma}_{k_2}^{-1})(\boldsymbol{\mu}_{k_1} - \boldsymbol{\mu}_{k_2}) \end{aligned} \tag{3}$$

When proceeding from level j to $j+1$ suppose that clusters $\mathcal{C}_{k_1}^j$ and $\mathcal{C}_{k_2}^j$ are merged. Then the merged density of cluster $\mathcal{C}_k^{j+1}$ at level $j+1$ is a mixture given by:

$$p_{j+1}(\boldsymbol{x}|k) = \frac{P_j(k_1)p_j(\boldsymbol{x}|k_1) + P_j(k_2)p_j(\boldsymbol{x}|k_2)}{P_j(k_1) + P_j(k_2)}, \quad P_{j+1}(k) = P_j(k_1) + P_j(k_2) \tag{4}$$

The remaining densities are unchanged.

At level 1 the expression for the distance in Eq. (4) is exact, while exact calculation at other levels cannot be cast into a simple analytical form. Consequently, we suggest to use a simple combination rule in which the distances to a merged cluster is original distances weighted by the mixing proportions, as in Eq. (4), i.e., $D_{j+1}(k,\ell) = (P_j(k_1)D_j(k_1,\ell) + P_j(k_2)D_j(k_2,\ell))/(P_j(k_1) + P_j(k_2))$, where clusters $\mathcal{C}_{k_1}^j, \mathcal{C}_{k_2}^j$ have been merged into $\mathcal{C}_k^{j+1}$ at level j, and ℓ indexes a cluster at level $j+1$.

Using a Bayes optimal decision strategy (assuming simple 0/1 loss function, see e.g., (Ripley 96)), a specific training example $\boldsymbol{x}_n$ is assigned to cluster k if

$$k = \arg\max_{\ell} P_j(\ell|\boldsymbol{x}_n) = \arg\max_{\ell} \frac{p_j(\boldsymbol{x}|\ell)P_j(\ell)}{\sum_{i=1}^{K-j+1} p_j(\boldsymbol{x}|i)P_j(i)} \tag{5}$$

If clusters $\mathcal{C}_{k_1}^j, \mathcal{C}_{k_2}^j$ have been merged into $\mathcal{C}_k^{j+1}$ at level j, then $P_{j+1}(k|\boldsymbol{x}_n) = P_j(k_1|\boldsymbol{x}_n) + P_j(k_2|\boldsymbol{x}_n)$. Thus, all posterior cluster probabilities are easily computed from the level 1 posteriors $P_1(k|\boldsymbol{x}_n)$.

Once the hierarchy is constructed we want to determine cluster/level membership of new examples. For this purpose we chose the following criterion: If $P_j(k|\boldsymbol{x}) = \arg\max_\ell P_j(\ell|\boldsymbol{x}) > \rho$ then $x \in \mathcal{C}_k^j$, where $\min_k P_1(k) < \rho \leq 1$ is a prescribed threshold, e.g., $\rho = 0.9$. This corresponds to accepting that $\boldsymbol{x}$ is assigned to a wrong cluster in with probability 0.1.

2.3. Interpretation of Clusters

Interpretation of clusters in the hierarchy is important for webmining applications. Suppose that each original example in our database is a set of elements drawn from finite number of possible elements (often large). Each example could for instance be a html-document consisting of a number of elements, i.e., words from a large vocabulary. The set of elements of each example is encoded into the feature vector $\boldsymbol{x}$. Basically two methods exist for a cluster interpretation: The first consist in listing a number of representative examples from the available training data set which are member of the cluster to be interpreted. The second method consists in listing typical elements associated with the cluster.

2.3.1. Prototype Examples

Representative examples of a specific cluster can be defined as the ones which are most probable. Since $p(\boldsymbol{x}|k)$ is a probability density the values are not directly comparable. Instead we compute the probability[4] $Q(t) = \mathrm{Prob}(\boldsymbol{x} \in \mathcal{R})$, $\mathcal{R} = \{\boldsymbol{x} : p(\boldsymbol{x}|k) < t\}$, for all thresholds t. We aim at identifying the t-value corresponding to the most probable example for the major part of the probability mass. This value is found as $t_{\max} = \arg\max_t Q(t) \leq Q_{\max}$, where that $Q_{\max}$ is a high threshold, e.g., 0.9. Practically, $Q(t)$ is computed from the training data assigned to cluster k, say $\mathcal{D}_k = \{\boldsymbol{x}_n \in \mathcal{C}_k\}$, as follows: rank $t_n = p(\boldsymbol{x}_n|k)$, $\boldsymbol{x}_n \in \mathcal{D}_k$ in ascending order, $t_1 \leq t_2 \leq \cdots \leq t_{N_k}$, where $p(\boldsymbol{x}_n|k)$ are model density values, and $N_k = |\mathcal{D}_k|$ is the number of example in $\mathcal{D}_k$. Finally, let $Q(t_n) = n/N_k$. Prototype examples are then a number of high ranked examples having t_n near $t_{\max}$.

[4] This idea relates to the concept of highest probability density regions (Box 92, Ch. 2.8).

2.3.2. Prototype Elements

In order to list representative elements associated with a cluster we start by finding most probable feature vectors from each cluster, basically using the method described in the previous section. An large surrogate data set can be generated by drawing Monte Carlo random samples from the estimated Gaussian mixture. From these data typical feature vectors are those having t-values for which $Q(t)$ is sufficiently high. Finally, the generated feature vectors are back-projected into original element space.

2.3.3. Novelty Detection

When the estimated density model is applied to new data there is a risk that these can not meaningfully be described by the model; in other words, we need to address the novelty problem. In line with recent work (Baker 99, Bishop 94, Nairac 97, Basseville 93), we suggest a novelty detector based on total input density $p(\boldsymbol{x})$. The method described in Section 2.3.1. can be used to form a $Q(t)$-function for $p(\boldsymbol{x})$. We then set a low threshold $Q_{\min}$ and find the corresponding $t_{\min}$ as $t_{\min} = \arg\min_t Q(t) \geq Q_{\min}$. Finally, novel events are detected as those having density values less than $t_{\min}$.

3. Generalizable Gaussian Mixture Classifier

If the feature vectors $\boldsymbol{x}$ are annotated by providing class labels, we are able to perform supervised learning using the GGM model. Consider a data set $\mathcal{D} = \{(\boldsymbol{x}_n, c_n) \,|\, n = 1, 2, \cdots, N\}$ where $c_n \in \{1, 2, \cdots, C\}$ is the class associated with example n. The joint density of feature vectors $\boldsymbol{x}$ and class labels c is $p(\boldsymbol{x}, c) = p(\boldsymbol{x}|c)P(c)$, where $p(\boldsymbol{x}|c)$ is the class conditioned density and $P(c)$ is the marginal class probabilities. The classifier is designed by adapting GGM's to each class separately. Hence, the class conditional density can be written as $p(\boldsymbol{x}|c) = \sum_{k=1}^{K_c} p(\boldsymbol{x}|k, c)P(k|c)$, where $P(k|c)$ and K_c are the mixture component probabilities and number components used for class c, respectively.

Labels are assigned to a new data point in accordance with the optimal Bayes classification (under the 0/1 loss) rule by selecting the maximum posterior probability, $P(c|\boldsymbol{x}) = p(\boldsymbol{x}|c)P(c)/\sum_{c=1}^{C} p(\boldsymbol{x}|c)P(c)$.

3.1. Unsupervised-then-Supervised Gaussian Mixture Model

In (Nigam 00) the interplay between supervised and unsupervised learning was discussed. To estimate the role of the labels for the GGM model first perform an GGM input density estimate $p(\boldsymbol{x}) = \sum_{k=1}^{K} P(k)p(\boldsymbol{x}|k)$. Next estimate $P(c|k)$ for each component k from the joint feature/label training data set as N_{ck}/N_k, where N_{ck} is the number of data samples of component k assigned class label c, and N_k is the number of data samples of component k. Finally, estimate the conditional class probability by

$$P(c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c)P(c)}{p(\boldsymbol{x})} = \frac{\sum_{k=1}^{K} p(\boldsymbol{x}|k,c)P(c|k)P(k)}{p(\boldsymbol{x})} = \frac{\sum_{k=1}^{K} p(\boldsymbol{x}|k)P(c|k)P(k)}{p(\boldsymbol{x})}. \quad (6)$$

The classification of examples using Eq. (6) can be compared to that of the supervised GGM classifier, illustrating the role of labels during training.

4. Experiments

4.1. Classification of Web Pages

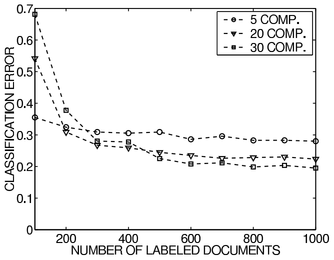
The focus is on understanding the textual content of a web page based on statistical features. Here we consider the single word statistics; frequency of word occurrence, hence disregarding order and association. Word frequencies have been used in the vector space model (Luhn 58, Salton 89) for decades. In practice words which high and low frequencies have little discriminative power. High frequency words are typically function words, e.g., is and the. Such words are removed by comparing the document with a list of stop words, i.e., a dictionary of common words. Also low frequency words are removed since they do not represent any common meaning among a number of web pages. In addition, we will consider to remove words with common stem, i.e., words like worked and working are represented by their stem work. Typically the number of words/terms after such parsing is still a very large compared to the number of documents available for learning. Since learning algorithms often fail to generalize in high dimensions there is a need for efficient and robust means for data reduction and feature extraction. Latent Semantic Indexing (LSI) (Deerwester 90) is a method to generate a reasonable low dimensional feature vector, and is further believed to handle polysemy and synonomy problems. Polysemy refers to the problem that words often have more than one meaning, whereas synonomy refers to the problem of different words with similar meaning.

LSI is based on the $T \times N$ term-document matrix, $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N]$, where $\boldsymbol{z}_n$ represent term frequency of document n, i.e., z_{in} is the probability of term i in document n.[5] The term frequencies are projected on a orthogonal set of eigen-histograms found by singular value decomposition (SVD). LSI can aid interpretation by visualizing group structure in the set of documents, typically by scatter plots of the term histograms on a reduced set of salient eigen-histograms. Another virtue of this representation is that it can be used as a dimensionality reduction scheme. First we remove the mean value $\tilde{\boldsymbol{z}}_n = \boldsymbol{z}_n - \widehat{\boldsymbol{\mu}}$, where $\widehat{\boldsymbol{\mu}} = N^{-1}\sum_{n=1}^{N} \boldsymbol{z}_n$. Then the SVD is given by $\tilde{\boldsymbol{Z}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top = \sum_{i=1}^{R} \boldsymbol{u}_i D_{i,i} \boldsymbol{v}_i^\top$, where the $T \times R$ matrix $\boldsymbol{U} = \{u_{ti}\} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_R]$, with R being the rank[6] of $\boldsymbol{Z}$, and the $N \times R$ matrix $\boldsymbol{V} = \{v_{ni}\} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_R]$

[5] The probabilities as normalized so that $\sum_i z_{in} = 1$.

[6] Since $T \gg N$, then for independent documents the rank is $R = N$.

Figure 2: *Learning curves for supervised learning of the generalizable Gaussian mixture classifier using WebKB data set.*



Confusion Matrix

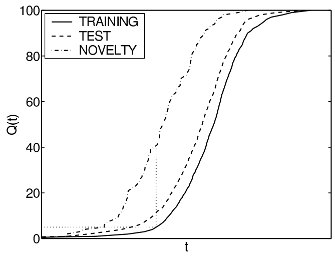
$d = 30$ and $N_{\text{train}} = 1000$

	True			
Estimated	Course	Fac.	Proj.	Stud.
Course	0.92	0.03	0.05	0.02
Fac.	0.04	0.64	0.10	0.13
Proj.	0.03	0.09	0.75	0.02
Stud.	0.01	0.24	0.10	0.83

represent the orthonormal basis vectors (i.e., eigenvectors of the symmetric matrices $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{X}^\top\boldsymbol{X}$, respectively). $\boldsymbol{\Lambda} = \{\lambda_{ii}\}$ is a $R \times R$ diagonal matrix of singular values ranked in decreasing order. Many singular values will be small and are regarded as artifacts or noise. Consequently, the subspace associated with these should be omitted while maintaining the latent semantic structure. The projection onto the d dimensional latent subspace is given by $\boldsymbol{X} = \widetilde{\boldsymbol{U}}^\top \bar{\boldsymbol{Z}}, \widetilde{\boldsymbol{U}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_d]$.

The CMU WebKB repository (CMU homepage) consist of 2240 web pages labeled according to the following categories: Course (24.7%), Faculty (21.6 %) Project (15.7%), Student (38.0%). A term list of 13071 words that occurred in two or more documents was defined without screening for stopwords. Latent semantic analysis is performed using feature dimensions of $d = 5, 20, 30$. In Figure 2 learning curves for the GGM classifier Section 3. were estimated by cross-validation. Data are randomly split 10 times into a test set of ($N_{\text{test}} = 1240$) and training sets of increasing sizes, $N_{\text{train}} = 100$–1000. Learning curves were estimated as the averaged test error as a function of d. A generalization cross-over, as function of the dimension, is noticed, i.e., the larger dimensional representations requires more samples to generalize. The proposed GGM classifier achieves classification rates and learning curves comparable to those found in (Nigam 00). The GGM model, however, achieves this performance based on the full 13071 dimensional term-frequency showing the strength of Latent Semantic Analysis representation. This allows for handling more complex webmining problems and also avoiding the selection of terms as in (Nigam 00). The interplay between supervised and unsupervised learning was further addressed in (Nigam 00). To estimate the role of the labels for the GGM model, we have carried out a similar learning curve experiment for the unsupervised-then-supervised Gaussian mixture model Section 3.1. It turns out that learning is much less efficient for the unsupervised-then-supervised procedure indicating significant class overlap.

Figure 3: *Novelty detection using web 173 pages from the* Department *group of the WebKB data set. The model has* $d = 30$ *dimensions and both the training and test sets contained 1120 documents. Threshold* t *for* $p(\boldsymbol{x})$ *is selected for* $Q = 5\%$.



4.1.1. Novelty Detection

Since the GGM classifier produces conditional probabilities we obtain in this way a clue to the "internal" confidence. The magnitude of the probabilities is determined by proximity of the decision boundary of the closest competing class. The overall test error rate give a clue to our confidence in the probabilities obtained from the system. However, when applied to new data the possibility exist, of course, that the new data can not in a meaningful way be assigned to any of the classes in the training data. In other words we need to address the novelty problem by identifying outliers in $p(\boldsymbol{x})$ as described in Section 2.3.3. Figure 3 shows $Q(t)$ based on training and a test set gathered from the documents above. We note that the test data are not rejected at reasonable Q-levels. The third curve is obtained from a third independent set of documents Department not related in an obvious way to the training and test sets. This data is declared novelty at levels below $Q_{\min} = 5\%$.

4.1.2. Web Navigation

A possible application is a navigation tool that can assist the user by combining the supervised and unsupervised classification schemes. At first the supervised part uses a list of labeled web pages, as typically can be found in a bookmark/favorite list ordered in folders for which the folder name serves as label for the underlying web pages (links). The GGM classifier classifies new pages into known bookmark labels. Documents not qualifying w.r.t. the current list of topics are detected as novel and using unsupervised GGM clustering of the pages and evaluating representative keywords for each mixture component, we are able to get an overall description of the document. Keywords are generated by back-projecting cluster centers into term-frequency space and then selecting most probable terms. Using e.g., Other/Misc pages of the WebKB data set 40% of the pages in this group are detected as novel, and these were subsequently clustered into 4 new groups. Keywords suggested the 4 groups could be interpreted as: Places, Spare time, Computer systems and Multimedia as indicated by Table 1.

Table 1: *Keywords associated with novel WebKB group* Other/Misc.

Multimedia	Computer sys.	Spare time	Places
eros	up	page	mississippi
random	readme	webteam	detroit
np	cache	visits	university
u	incoming	funny	military
player	msdos	uva	saint
ramifications	directory	today	macon
gif	windows	museum	williamsburg
format	mac	totals	rolla
slide	unix	robins	aeronautical
modulo	wie	total	louis

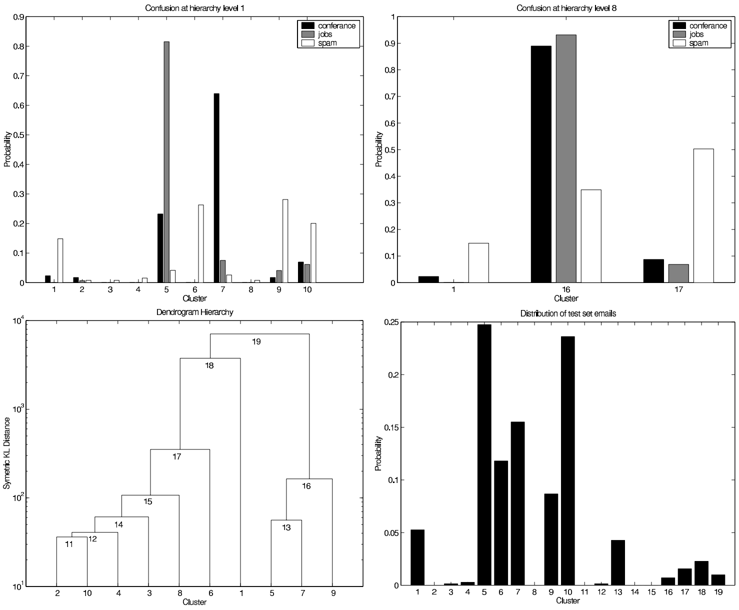
4.2. Email Segmentation using Hierarchical Probabilistic Clustering

Consider hierarchical segmentation of emails. A database of 1443 English emails categorized in three groups conference, jobs, and spam were collected. Only the text contained in subject and body was considered. As in Section 4.1. we performed LSI using a stop word list of 577 words, removed words which occurred less than 4 times, and finally we discarded emails which contained less than 2 words. Only one word for words with a common stem was maintained by discarding 13 different endings. After preprocessing we had 1405 emails divided into 702 for training and 703 for testing. Each email was represented by it's term-histogram of 7798 terms. Using a latent subspace of $d = 30$ components[7] resulted in GGM models with optimal number of clusters in level 1 in the range 6–10. We chose to illustrate a model consisting of $K = 10$ clusters. Performing hierarchical clustering on top of the GGM, as described in Section 2.2., results in a dendrogram hierarchy depicted in lower left panel of Figure 4. Numbers refer to cluster numbers, e.g., 12 is the merging of clusters 4 and 11. The confusion matrices computed from training examples for hierarchy levels 1 and 8 are shown in the upper panels of Figure 4. It is noted that at level 1 the conference category is mainly represented by cluster 7 and 5, jobs by cluster 5, and spam by clusters 9, 6, 10 and 1. At level 8, corresponding to three clusters, clusters 1 and 17 mainly represent spam whereas cluster 16 mainly represents both conference and jobs. Consequently, the unsupervised hierarchical clustering is not able to distinguish these categories. Also notice that cluster 5 and 7 which largely represent these categories are merged at an early level into cluster 13. For comparison, supervised learning was also implemented. As expected, it performs much better regarding cluster separation. Then confusion on the first level of hierarchical clustering

[7] A method for selecting the subspace dimension based on generalization error in described in (Szymkowiak 01).

was much smaller comparing to the unsupervised. However, since the goal of the algorithm is to extract hidden common sense in the text documents, the exact classification can be misleading. In the tested database the clustering algorithm seems to confuse big parts of the conference and jobs group. This happens both for the unsupervised and supervised learning algorithm. The KL divergence measure, Eq. (4), indicates a small distance in the probabilistic space between these two clusters, and the generated keywords (see Table 2) are closely related which explains the small distance. When filtering test set emails through the hierarchy we assign a specific email to the cluster at which the posterior probability is above 0.9, according to Section 2.2. The right lower panel of Figure 4 shows the fraction of test set emails ending up in different clusters. We notice that several email first obtain a meaningful interpretation at high level in the hierarchy (i.e., cluster number larger than 10).

Figure 4: *Dendrogram for hierarchical email clustering and distribution of test set emails among clusters.*



Keywords are generated by back-projecting most probable features from each cluster at any level in the hierarchy as outlined in Section 2.3. The back-projection intro term-frequency space is given by $\tilde{\boldsymbol{z}} = \widetilde{\boldsymbol{U}}\boldsymbol{x}$, where $\boldsymbol{x}$ is a probable feature vector and $\widetilde{\boldsymbol{U}}$ is the 1405 × 30 projection matrix. The keywords are then found as the most likely terms, i.e., highest values[8] of $\bar{\boldsymbol{z}}$.

Table 2: *Keywords for email cluster hierarchy in Figure 4.*

Cluster	Keywords
1	free address government
2	fax
3	subject future remove computer
4	adult free
5	fax interest computation computer web science position research university
6	good mac
7	fax year message call conference information computer address
8	call girl
9	good year receive product special make month day future mail friend quick line state send offer
10	government remove adult
11	action address check hottest click site creativity call website government web free remove mac adult
12	hey jessica site remove call government creativity web website mac adult fax free
13	conference send information application computation year interest science address call fax computer
14	website mac adult remove fax free computer
15	website revolution remove fax free call adult girl computer
16	year interest computation fax address science web computer position university research
17	mail government subject creativity call website free future fax food adult remove computer
18	food web free mac government adult
19	message fax computer science list call subject university money position information address dear

4.3. User behavior modeling

User behavior modeling is an important aspect of e-commerce systems. The current examples is based on our work reported in (Christiansen 01) which studied an e-commerce company selling articles via the web. Web log-data was recorded for half a year and resulted in 31700 sessions for which all user actions where mapped into 60 unique events. Events could be pressing a buy button, selecting a certain group of articles, or following a link to a another web page. Each session is thus a variable length sequence of events from the 60 element event-alphabet $\mathcal{B} = \{1, 2, \cdots, 60\}$, $B = |\mathcal{B}| = 60$. In general, it might be difficult to map the details of the web server log file into a unique event space unless the logging has been designed with this purpose in mind.

The log-on to the site could be done in two ways, either as member login with personal password, or as a guest assigned a pseudo user-id. Each session was numbered in succession, i.e., repeated log-on from the same user is mapped to different session numbers. Using the industry standard, sessions are interrupted and the user automatically logged-off after 30 minutes of no activity.

Too short sessions will not reflect a real interest in the web site (Yan 96). Hence, the minimum session length was set to four events, corresponding to the shortest way into the "shopping area" from the opening site. A total of 4339 sessions remained of which 1089 randomly was selected as test set, leaving

[8] Due to using a low-dimensional subspace of $d = 30$, $\bar{\boldsymbol{z}} + \widehat{\boldsymbol{u}}$ typically does take values in the range $[0; 1]$ nor is $\sum_i \bar{z}_i + \widehat{u}_i = 1$. In principle, we could feed the values trough a softmax-function (Ripley 96), which, however, will not change the ranking.

3250 sessions for training.

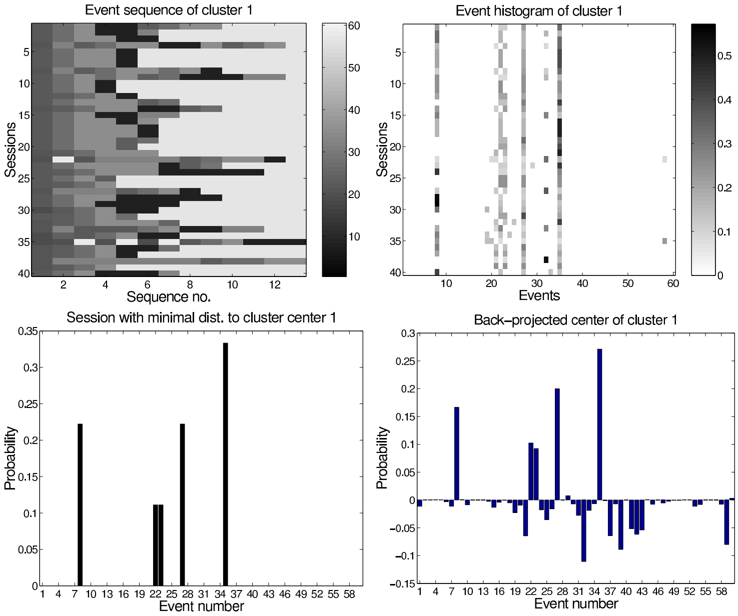
Let $s_{\ell n} \in \mathcal{B}$ represent session n consisting of L_n events, $\ell = [1; L_n]$. As in (Yan 96), we deploy histogram statistics representation of the sessions by computing the frequency of events: $z_{in} = L_n^{-1} \sum_{\ell=1}^{L_n} \delta(i - s_{\ell n})$, where $i \in \mathcal{B}$, $\delta(\cdot)$ is the Kronecker delta-function, and $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n]$ is denoted the histogram matrix. It is possible to use second order statistics, i.e., co-occurrence matrices. The $B \times B$ co-occurrence matrix for session n and displacement τ is defined as, $c_{ij}(n, \tau) = (L_n - 1)^{-1} \sum_{\ell=1}^{L_n - 1} \delta(i - s_{\ell,n}) \cdot \delta(j - s_{\ell+\tau,n})$, $\forall i, j \in \mathcal{B}$, and expresses the frequency of events i and j in distance τ of the sequence. Co-occurrence features have be used in e.g., (Faisal 99) and will be further addressed in (Christiansen 01). In this study we merely address the use of the histogram and also neglect to include the duration of a session as a feature. In order to obtain a compact feature space we apply singular value decomposition (see p. 7) of the zero mean $B \times N$ histogram matrix $\bar{\boldsymbol{Z}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}$ defined by $\bar{\boldsymbol{z}}_n = \boldsymbol{z}_n - \widehat{\boldsymbol{u}}$, where $\widehat{\boldsymbol{u}} = N^{-1} \sum_{n=1}^{N} \boldsymbol{z}_n$. Then we project onto the d-dimensional latent subspace spanned by the largest singular values as given by $\boldsymbol{X} = \widetilde{\boldsymbol{U}}^{\top} \bar{\boldsymbol{Z}}$, where $\widetilde{\boldsymbol{U}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_d]$.

Repeated training of the unsupervised GGM model using $d = 30$ features resulted in that the most generalizable model contained $K = 17$ components (clusters). Figure 5 shows the obtained analysis of cluster 1. The upper left panel shows the event sequences of the 40 sessions belonging to cluster 1, and are quite similar for the first few instances in the sequence. The upper right panel shows event histograms, and obviously most sessions use a rather limited number of events. In the lower panel the interpretation of cluster 1 is illustrated. The lower left panel shows the histogram of most the probable session, whereas the lower right panel shows the back-projection of the cluster center to histogram space. There is a significant resemblance indicating that the cluster can be interpreted by events (ordered in decreasing importance) as: $35, 27, 8, 22, 23$. From the actions associated with these events it seems that the cluster represents users attempting to register as a new members, while none of the users are able to get to the shopping web page. Other clusters can be interpreted using this technique. For instance, cluster 3 represents members who first login as guests, secondly choose a goods pick-up store, and then browse for while. However, almost 200 out of 708 in this cluster decide to quit after having watched the entry shopping web page. Cluster 15 represents a group of users which are not able to use the site correctly. They try use a search function before selecting preferred goods pick-up store, which turns out to be impossible. This way cluster 15 reveals a simple bug in the web site design.

5. Conclusion

This paper discussed the use of unsupervised and supervised methods for analysis and interpretation of world wide web data. A hierarchical probabilistic

Figure 5: *User behavior modeling. Analysis of cluster 1.*



clustering scheme based on the generalizable Gaussian mixture (GGM) model was described. In addition, methods for interpretation of the identified clusters were presented. The use of the GGM for supervised and unsupervised-then-supervised classification was also discussed. We successfully applied supervised GGM for classification of web pages. The unsupervised GGM was applied for hierarchical probabilistic clustering of emails and segmentation of user's behavior when shopping on a web site.

References

Akaike H. (1969) Fitting Autoregressive Models for Prediction, *Ann. of the Inst. of Stat. Math.*, vol. 21, pp. 243–247.

Baker L.D., Hofmann T., Maccallum A.K., Yang Y. (1999) A Hierarchical Probabilistic Model for Novelty Detection in Text, *CMU techincal report*, `http://www.cs.cmu.edu/People/mccallum/papers/tdt-nips99s.ps.gz`

Basseville M., Nikiforov I.V. (1993) *Detection of Abrupt Changes: Theory and Application*, Prentice-Hall.

Bishop C.M. (1994) Novelty Detection and Neural Network Validation, *IEE Proceedings - Vision Image and Signal Processing*, vol. 141, no. 4, pp. 217–222.

Box G.E.P., Tiao G.C. 1992 *Bayesian Inference in Statistical Analysis*, John Wiley & Sons.

Cooley R., Srivastava J., Mobasher, B. (1999) Data Preparation for Mining World Wide Web Browsing Patterns, *Journal of Know. and Inf. Sys.*, vol. 1, no. 1, pp. 5–32.

Christiansen T., Larsen J., Hansen L.K., Hørlück J. (2001) Understanding User Behavior on Web Sites. Forthcoming publication.

Carnegie Mellon University homepage, `http://www.cs.cmu.edu/~textlearning`

Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R. (1990) Indexing by Latent Semantic Analysis, *Journ. Amer. Soc. for Inf. Science.*, vol. 41, pp. 391–407.

Dempster A.P., Laird N.M., Rubin D.B. (1977) Maximum likelihood from incomplete data via the EM algorithmm, *Jour. R. Stat. Soc. B*, vol. 39, pp. 1–38.

Faisal A., Shahabi C., McLaughlin M., Betz F. (1999) A Generic Paradigm for Interpreting User-Web Space Interaction, in *Proc. of Web Inf. and Data Management (WIDM'99)*, Kansas City, USA, pp. 53–58.

Hansen L.K., Sigurdsson S., Kolenda T., Nielsen F.Å., Kjems U., Larsen J. (2000) Modeling Text with Generalizable Gaussian Mixtures, *Proeedings of IEEE ICASSP'2000*, Istanbul, Turkey, vol. VI, pp. 3494–3497.

Hansen L.K., Larsen J. (1996) Unsupervised Learning and Generalization, *Proc. of the IEEE Int. Conf. on Neural Networks 1996*, vol. 1, pp. 25–30.

Hastie T., Tibshirani R. (1996) Discriminant Analysis by Gaussian Mixtures, *Jour. Royal Stat. Society - Series B*, vol. 58, no. 1, pp. 155–176.

Landauer T.K., Laham D., Foltz P. (1998) Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report, *Advances in NIPS 10*, MIT Press, pp. 45–51.

Luhn H.P. (1958) The Automatic Creation of Literature Abstracts, *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165 and 317.

Mobasher B., Cooley R., Srivastava J. (1999) Creating Adaptive Web Sites Through Usage-Based Clustering of URLs, in *Proceedings KDEX'99.*

Nairac A., Corbett-Clark T., Ripley R., Townsend N., Tarassenko L. (1997) Choosing An Appropriate Model for Novelty Detection, *IEE 5th Int. Conf. on Art. NNs*, pp. 117–122.

Nigam K., McCallum A.K., Thrun S., Mitchell T. (2000) Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, vol. 39, no. 2–3, pp. 103–134.

Pei J., Han J., Mortazavi-Asl B., Zhu H. (2000) Mining Access Pattern Efficiently from Web Logs, *Proc. 2000 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00)*, Kyoto, Japan, pp. 396-407.

Rasmussen, C.E. (2000) The Infinite Gaussian Mixture Model, in S.A. Solla, T.K. Leen K.-R. Müller (eds.) *Advances in NIPS 12*, MIT Press, pp. 554–560.

Ripley B.D. (1996) *Pattern Recognition and Neural Networks*, Cambridge University Press.

Salton G. (1989) *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer* Addison-Wesley.

Shahabi C., Zarkesh A.M., Adibi J., Shah V. (1997) Knowledge Discovery from Users Web-page Navigation, *Proceedings Seventh International Workshop on Research Issues in Data Engineering*, (cat.no. 97TB100122), pp. 20-29.

Spiliopoulou M., Faulstich L.C., Winkler K. (1999) A Data Miner Analyzing the Navigational Behaviour of Web Users, *Proc. of the Workshop on Machine Learning in User Modelling of the ACAI'99 Int. Conf.*, Creta, Greece.

Szymkowiak A., Larsen J., Hansen L.K. (2001) Hierarchical Clustering for Datamining. Forthcoming publication.

Yan T.W., Jacobsen M., Garcia-Molina H., Dayal U. (1997) ¿From User Access Patterns to Dynamic Hypertext Linking, *Computer Networks and ISDN Systems*, vol. 28, no. 11.

Vasconcelos N., Lippmann A. (1999) Learning Mixture Hierarchies, in M. Kearns, S. Solla & D.A. Cohn (eds.) *Advances in NIPS 11*, pp. 606–612.

Weigend A.S., Wiener E.D., Pedersen J.O. (1999) Exploiting Hierarchy in Text Categorization, *Information Retrieval*, vol. 1, pp. 193–216.

Williams C. (2000) A MCMC Approach to Hierarchical Mixture Modelling, in T. Leen, S. Solla & K.R. Müller (eds.) *Advances in NIPS 12*, pp. 680–686.

List of Figures

List of Tables

Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. *2nd International Symposium on Information Theory*, pages 267–281, 1973.

[2] Hagai Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*, pages 21–30, 2000.

[3] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1988.

[4] Weizhu Bao and Shi Jin. The random projection method. In *Advances in Scientific Computing*, pages 1–11, 2001.

[5] Matthew J. Beal and Zoubin Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 7*. Oxford University Press, 2003.

[6] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 146–153, New Orleans, US, 2001. ACM Press, New York, US.

[7] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[8] Y. Bengio, P. Vincent, and J-F. Paiement. Learning eigenfuncions of similarity : Linking spectral clustering and kernel pca. Technical report, Département d'informatique et recherche opérationnelle, Université de Montréal, 2003.

[9] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Knowledge Discovery and Data Mining*, pages 245–250, 2001.

[10] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[11] Christopher M. Bishop and Michael E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.

[12] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59:291–294, 1988.

[13] Jaime Carbonell, Yiming Yang, and William Cohen. Special issue of machine learning on information retrieval introduction. *Machine Learning*, 39:99–101, 2000.

[14] Miguel Á. Carreira-Perpinán. A review of dimension reduction techniques. Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield, January 1997.

[15] R. B. Cattell. The scree test for the number of factors. J. Multiv. Behav. Res., 1966.

[16] A. Corduneanu and C.M. Bishop. Variational bayesian model selection for mixture distributions. In T. Richardson and T .Jaakkola (Eds.), editors, *Proceedings Eighth International Conference on Artificial Intelligence and Statistics*, pages 27–34, 2001.

[17] Vittorio Castelli Thomas M. Cover. On the exponential value of labeled samples. *Title Pattern Recognition Letters*, 16:105–111, 1995.

[18] J.M. Craddock and C.R. Flood. Eigenvectors fo representing the 500 mb. geopotential surface over the northen hemisphere. *Quarterly Journal of the Royal Meteorological Society*, 1969.

[19] Sanjoy Dasgupta. Experiments on random projection. In *In Proc. 16th Conf. Uncertainty in Artificial Intelligence*, 2000.

[20] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[22] K.I. Diamantaras and S.Y. Kung. Principal component neural networks: Theory and applications. *John Wiley & Sons Inc., New York*, 1996.

[23] George H. Dunteman. Principal components analysis. *Sage University Paper Series on Quantitative Applications in Social Sciences*, 69, 1989.

[24] H. T. Eastment and W. J. Krzanowski. Cross-validatory choice of the number of components from a principal components analysis. *Technometrics*, 24:73–77, 1982.

[25] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[26] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Knowledge Discovery and Data Mining*, pages 82–88, 1996.

[27] Chris Fraley. Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, 1999.

[28] Dayne Freitag. Machine learning for information extraction in informal domains. *Machine Learning*, 39:169–202, 2000.

[29] K. Fukunaga and D.R. Olsen. An algorithm for finding intrinsic dimensionality of data. In *IEEE Transactions on Computers*, volume 20, pages 165–171, 1976.

[30] Zoubin Ghahramani and Michael I. Jordan. Learning from incomplete data. Technical Report 108, MIT Center for Biological and Computational Learning, 1994.

[31] Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via an EM approach. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 120–127. Morgan Kaufmann Publishers, Inc., 1994.

[32] Mark Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[33] Mark Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14(3):669 – 688, 2002.

[34] Johannes Grabmeier and Andreas Rudolph. Techniques of cluster algorithms in data mining. *Data Mining and Knowledge Discovery*, 6(4):303–360, October 2002.

[35] H. Altay Güvenir, Gülsen Demiröz, and Nilsel Ilter. Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. *Aritificial Intelligence in Medicine*, 13(3):147–165, 1998.

[36] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. The MIT Press, Cambridge, Massachusetts 02142, August 2001.

[37] L.K. Hansen, J. Larsen, F. Nielsen, S. C. Strother, E. Rostrup, R. Savoy, N. Lange, J. Sidtis, C. Svarer, and O. B. Paulson. Generalizable patterns in neuroimaging: How many principal components? *NeuroImage*, pages 534–544, 1999.

[38] L.K. Hansen, C. Liisberg, and Peter Salamon. Ensemble methods for recognition of handwritten digits. In S.Y. Kung, F. Fallside, J. Aa. Sørensen, , and C.A. Kamm, editors, *Proceedings of Neural Networks For Signal Processing*, pages 540–549, 1992.

[39] L.K. Hansen, S. Sigurdsson, T. Kolenda, F. Nielsen, U. Kjems, and J. Larsen. Modeling text with generalizable gaussian mixtures. In *Proceedings of IEEE International Conference on Acustics Speach and Signal Processing*, volume VI, pages 3494–3497, 2000.

[40] Harry H Harman. *Modern Factor Analysis*. University of Chicago Press, 2 edition, 1967.

[41] John Hertz, Anders Krogh, Richard Palmer, and Roderick V. Jensen. Introduction to the theory of neural computation. *American Journal of Physics*, 62(7), 1994.

[42] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings Of The 22nd Annual Acm Conference On Research And Development In Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

[43] A.J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86:205–224, 1991.

[44] J. Edward Jackson. A user's guide to principal components. *Wiley Series in Probability and Mathematical Statistics*, 1991.

[45] I.T. Jolliffe. Principal component analysis. *Springer Series in Statistics*, 1986.

[46] N. Kambhatla and T.K. Lee. Dimension reduction by local principal component analysis. *Neural Computation*, 9:1493–1516, 1997.

[47] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: Algorithms and applications. *VLDB Journal: Very Large Data Bases*, 8(3–4):237–253, 2000.

[48] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[49] T. Kolenda. *Adaptive tools in virtual environments: Independent component analysis for multimedia*. PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, January 2002.

[50] T. Kolenda, L.K. Hansen, J. Larsen, and O. Winther. Independent component analysis for understanding multimedia content. In H. Bourlard, S. Bengio J. Larsen T. Adali, and S. Douglas, editors, *Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII*, pages 757–766, Piscataway, New Jersey, 2002. IEEE Press. Martigny, Valais, Switzerland, Sept. 4-6, 2002.

[51] Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.

[52] J. Larsen, L.K. Hansen, A. Szymkowiak-Have, T. Christiansen, and T. Kolenda. Webmining: Learning from the world wide web. *special issue of Computational Statistics and Data Analysis*, 38:517–532, 2002.

[53] J. Larsen, A. Szymkowiak-Have, and L.K. Hansen. Probabilistic hierarchical clustering with labeled and unlabeled data. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 6(1):56–62, 2002.

[54] Thorbjørn Larsen and Tobias Tobiasen. Automatisk fejldetektering i ukomplette data. Master's thesis, Department of Mathematical Modelling, Technical University of Denmark, 2001.

[55] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2000.

[56] Steven J. Leon. *Linear Algebra with Applications*. Prentice Hall, 6th edition, January 2002.

[57] Roderick J.A. Little and Donald B. Rubin. *Statistical Analysis With Missing Data*. Probability abd Mathematical Statistics. John Wiley & Sons, 1987.

[58] Marina Meila and David Heckerman. An experimental comparison of several clustering and initialization methods. In *In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 386–395, 1998.

[59] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In Todd Leen, Thomas Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 873–879. MIT Press, 2000.

[60] D.J. Miller and H.S. Uyar. A mixture of experts classifier with learning based on both labeled and unlabeled data. In *Advances in Neural Information Processing Systems*, volume 9, pages 571–578, 1997.

[61] Thomas P. Minka. Automatic choice of dimensionality for PCA. In *Advances in Neural Information Processing Systems*, pages 598–604, 2000.

[62] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *In Advances in Neural Information Processing Systems*, volume 14, pages 849–856, 2001.

[63] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[64] E. J. Nyström. Über die praktische auflösung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematica*, 4(15):1–52, 1928.

[65] E. Oja. Neural networks, principal components, and subspaces. *International Journal of Neural Systems*, 1(1):61–8, 1989.

[66] M. Partridge and R. Calvo. Fast dimensionality reduction and simple pca. *Intelligent Data Analysis*, 2:203–214, 1998.

[67] M. F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.

[68] Bhavani Raskutti, Herman Ferra, and Adam Kowalczyk. Combining clustering and co-training to enhance text classification using unlabelled data. In *Proceedinds of Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 620–625, 2002.

[69] S. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[70] Christian P. Robert. *The Bayesian Choice: a decision-theoretic motivation*. Springer-Verlag, 1994.

[71] D. Rubin. Bayesian inference for causal effects: The role of randomization. *Annals of Statistics*, 6:34–58, 1978.

[72] Donald B. Rubin. Inference and missing data. *Biometrica*, 63:581–592, 1976.

[73] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. J. Wiley and Sons, New York, July 1987.

[74] D. E. Rumelhart, G. E. Hinton, and J. L. McClelland. A general framework for parallel distributed processing. In D. E. Rumelhart, J. L. McClelland, et al., editors, *Parallel Distributed Processing: Volume 1: Foundations*, pages 45–76. MIT Press, Cambridge, 1987.

[75] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley, 1989.

[76] Lawrence Saul and Fernando Pereira. Aggregate and mixed-order Markov models for statistical language processing. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical*

Methods in Natural Language Processing, pages 81–89, Somerset, New Jersey, 1997. Association for Computational Linguistics.

[77] J.L. Schafer. *Analysis of Incomplete Multivariate Data*, volume 72 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1997.

[78] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[79] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[80] A. Szymkowiak, J. Larsen, and L.K. Hansen. Hierarchical clustering for datamining. In *Proceedings of KES-2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*, pages 261–265, 2001.

[81] A. Szymkowiak, P.A. Philipsen, J. Larsen, L.K. Hansen, E. Thieden, and H.C. Wulf. Impuating missing values in diary records of sun-exposure study. In D. Miller, T. Adali, J. Larsen, M. Van Hulle, and S. Douglas, editors, *Proceedings of IEEE Workshop on Neural Networks for Signal Processing XI*, pages 489–498, Falmouth, Massachusetts, 2001.

[82] A. Szymkowiak-Have, J. Larsen, L.K. Hansen, P.A. Philipsen, E. Thieden, and H.C. Wulf. Clustering of sun exposure measurements. In D. Miller, T. Adali, J. Larsen, M. Van Hulle, and S. Douglas, editors, *Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII*, pages 727–735, 2002.

[83] Anna Szymkowiak-Have, Mark Girolami, and Jan Larsen. A probabilistic approach to kernel principal component analysis. to be published in 2003.

[84] Michael Tipping and Christopher Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, September 1997.

[85] Volker Tresp, Subutai Ahmad, and Ralph Neuneier. Training neural networks with deficient data. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 128–135. Morgan Kaufmann Publishers, Inc., 1994.

[86] Volker Tresp, Ralph Neuneier, and Subutai Ahmad. Efficient methods for dealing with missing data in supervised learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 689–696. The MIT Press, 1995.

[87] N. Vasconcelos and A. Lippman. Learning mixture hierarchies. Technical report, MIT Media Laboratory, 1998.

[88] Chris William and Mathias Seeger. Using the nystrom method to speed up kernel machines. In Todd Leen, Thomas Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 682–688. MIT Press, 2000.

[89] C. Williams. A mcmc approach to hierarchical mixture modelling. In *Advances in Neural Information Processing Systems*, volume 12, pages 680–686, 2000.

[90] S. Wold. Cross-validatory estimation fo the numer of components in factor and principal components models. *Technometrics*, 20:397–405, 1978.

[91] D. Xu, J. Principe, and J. Fisher. A novel measure for independent component analysis (ICA). In *In: Proceedings of IEEE ICASSP98*, volume 2, pages 1161–1164, 1998.

[92] Lei Xu and Michael I. Jordan. On convergence properties of the EM algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.