Adaptive tools in virtual environments

Independent component analysis for multimedia

Thomas Kolenda

LYNGBY 2002

IMM-PHD-2002-94

IMM

Adaptive tools in virtual environments

Independent component analysis for multimedia

Thomas Kolenda

LYNGBY 2002

IMM-PHD-2002-94

IMM

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: Adaptive tools in
virtual environments
Independent component analysis for multimedia

Author: Thomas Kolenda

TECHNICAL UNIVERSITY OF DENMARK
Informatics and Mathematical Modelling
Richard Petersens Plads, Building 321,
DK-2800 Kongens Lyngby, Denmark
IMM-PHD-2002-94

Lyngby 2002-01-31

Printed by IMM, Technical University of Denmark

ISSN 0909–3192

Abstract

The thesis investigates the role of independent component analysis in the setting of virtual environments, with the purpose of finding properties that reflect human context. A general framework for performing unsupervised classification with ICA is presented in extension to the latent semantic indexing model. Evidence is found that the separation by independence presents a hierarchical structure that relates to context in a human sense. Furthermore, introducing multiple media modalities, a combined structure was found to reflect context description at multiple levels. Different ICA algorithms were compared to investigate computational differences and separation results. The ICA properties were finally implemented in a chat room analysis tool and briefly investigated for visualization of search engines results.

Abstract (Danish)

Afhandlingen undersøger independent component analysis (ICA) i virtuelle verdener, med det formål at finde egenskaber der reflekterer menneskelige forståelse. På baggrund af ICA bliver en generel metode præsenteret til, at udføre ”unsupervised” klassifikation, der er en udvidelse af ”latent semantic indexing” modellen. Det blev fundet at uafhængighed reflektere en menneskelig naturlig måde at separere på, og at metoden viser en hierarkisk opdeling. Ved endvidere at introducere flere medietyper, blev en samlet struktur fundet, der beskrev indholdet på flere niveauer. Forskellige ICA algoritmer blev undersøgt mht. beregningskompleksitet og separationsresultater. ICA egenskaberne blev til sidst implementeret i et chat rums analyse værktøj, og kort undersøgt i henholdt til visualisering at Internet søgemaskineresultater.

Preface

This thesis was prepared at the Department of Mathematical Modelling, Technical University of Denmark, in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering. The work was funded by the Danish Research Councils through the Intermedia plan for multimedia research. The project commenced in October 1998 and was completed in January 2002, with a short interruption due to service for her Majesty Queen Margrete II of Denmark. Throughout the period, the project was supervised by Lars Kai Hansen, Jan Larsen and Niels Jørgen Christensen from IMM, DTU.

The thesis reflects the work done during the Ph.D. project, that relates to software agents in artificial intelligence, independent component analysis, and multimedia modeling. The thesis is roughly divided up into chapters accordingly, for the reader to decide where to put emphasis according to background. As a whole the thesis aims to build a bridge across the communities, applying independent component analysis to multimedia in a virtual environment setting, for the purpose of finding computational reasonable properties that reflect human context.

The thesis is printed by IMM, Technical University of Denmark, and available as softcopy at http://www.imm.dtu.dk/documents/ftp/publphd.html .

Publication notes

Parts of the work presented in this thesis have previously been published and presented at conferences. Following are the most important paper contributions in the context of this thesis:

T. Kolenda, L.K. Hansen and S. Sigurdsson
Independent Components in Text
in M. Girolami (ed.) Advances in Independent Component Analysis, Springer-Verlag, chapter 13 229-250, 2000.

L.K. Hansen, J. Larsen and T. Kolenda
On Independent Component Analysis for Multimedia Signals
in L. Guan et al. (eds.) Multimedia Image and Video Processing, chapter 7, 175-200, 2000.

T. Kolenda, L.K. Hansen and J. Larsen
Signal Detection using ICA: Application to Chat Room Topic Spotting
in proc. ICA'2001, 2001.

Nomenclature

An attempt has been made to use standard symbols and operators consistently throughout the thesis. Symbols and operators are introduced along the way as they appear and the reader should have no trouble understanding the meaning.

In general matrices are presented in uppercase bold letters, vectors are presented in lowercase bold letters, and scalars in plain lowercase. Number of elements in a matrix or vector is mostly written as N_x where the suffix indicates the matrix reference. Probability density functions are written in a short form e.g. $p(x)$ meaning $p_x(x)$ if nothing else is specified.

Acknowledgements

I truly want to thank my supervisors Lars Kai Hansen and Jan Larsen for our working relationship, and understanding towards my familiarity difficulties occurring during the Ph.D. work. I would also like to acknowledge my colleagues at the Department of Signal Processing at IMM, DTU, and especially my office mate Sigurdur Sigurdsson and department secretary Ulla Nørhave for having to put up with me.

In the developments of visualizations of search engine results I would also like to thank Ankiro for the use of their database, and especially Steen Bøhm Andersen.

Finally, I sincerely want to thank my loving wife Sanne and children Mikkel and Ellen for the love and support they have given me.

Technical University of Denmark, January 2002

Thomas Kolenda

”I was sitting writing at my text book, but the work did not progress; my thoughts were elsewhere. I turned my chair to the fire, and dozed. Again the atoms were gamboling before my eyes. This time the smaller groups kept modestly in the background. My mental eye, rendered more acute by repeated visions of this kind, could now distinguish larger structures of manifold conformations; long rows, sometimes more closely fitted together; all twisting and turning in snake-like motion. But look! What was that? One of the snakes had seized hold of its own tail, and the form whirled mockingly before my eyes. As if by a flash of lightning I woke;... I spent the rest of the night working out the consequences of the hypothesis. Let us learn to dream, gentlemen, and then perhaps we shall learn the truth.”

- August Kekule von Stradonitz year 1858

Contents

CHAPTER 1

Introduction

Advancements in the last century in computer processing power, network and storage capabilities have given rise to massive shared virtual environments. Accordingly the amount of data has in the recent years grown beyond the capability of traditional data handling methods. To for example, overlook, handle and find data, tools are needed as context navigators and interpreters. Software agents have been envisioned for handling these tasks, and thus to roam the virtual worlds for the purpose of serving man. In contrast to our physical world, no one set of underlying laws define the virtual, but it is a combination of many different ontologies and media modalities. It is therefore no simple task when defining a software agent. In general it has to have properties of being autonomous, very adaptive, and most importantly to fulfill the purpose of its creators, i.e. to acknowledge what defines human context. Statistical methods has until recently not been applicable in a practical sense when handling these massive amounts of data. Through the development in computer power and the growing user demand for more powerful and complex methods, statistical methods have become tractable. A primary problem to be looked at is how statistics can define rules that reveal context in a human sense[82].

Recent research has suggested that imposing *independence* is a good criteria in unsupervised separation. In for example, sound and image separation, obtain-ing independence captures important information of the generating sources, see e.g. [19, 34]. Furthermore, in regards to the human brain, the independence paradigm is observed in the primary visual cortex of the receptive fields that resembles edge like filters. Hence, the same result is found when constraining the criteria of independence between small patches of natural images[86, 6].

The notion of independence is not easy to explain. By definition, we think of independence as the natural criteria to reduce redundancy in a system. Hence in order to obtain the best statistical independent separation, the separation must be done so as to minimize the redundancy. Also, in physics terms one must minimize the marginal entropies of the observations[12], where the entropy is a measure of how uncertain a grouping is.

In separating multimedia, the context can be explained on different levels of description regarding the general notion of human context. The image in figure 1.1[1], can be described for example by color as being *overall cold with hot spots* or by accompanying text as *science fiction and surrealism*. Furthermore combining the two gives mutual added information, as what must be assumed to be closer to what humans acknowledge when a media is observed.

In this thesis we investigate the properties of independency with regards to multiple types of media and employ the *independent component analysis* algorithm. In that regard, different algorithms are used and compared. Extracting features from different media modalities we extend the *vector space model* [91] and *latent semantic indexing* [25] framework to ICA classification. ICA proves to describe the grouping structure better and finds context in a human sense. Combining multimedia furthermore show to improve the unsupervised classification, and find a hierarchical context taxonomy towards the used media modalities.

Finally the ICA algorithm is implemented in a chat room and a search engine visualization setting, to demonstrate its online capabilities for use with software agents.

[1] The image was retrieved on the Internet WWW, from http://www.ikolenda.demon.co.uk/. However coincidence with author name, there are no direct family connection.

Figure 1.1 Art work by Ian Kolenda, titled *Rebirth Of The New Adam*.

Reading guide

The contents of this thesis are roughly divided into four parts between chapter 2 to 5. Readers that are familiar with the topic of chapter 2 or 3 can skip over them as they are largely reviews. The main research contributions in this thesis are found in chapters 4 and 5, together with papers described in appendixB.

Chapter 2 We start by outlining the general framework of the virtual worlds and software agents using the Internet as a reference point. The ever evolving structure and ontologies of the virtual worlds makes properties of software agents hard to define, and so tools reflecting human context become our prime objective, thus the focus on independent component analysis.

Chapter 3 The ICA model and algorithm is presented in the form of a maximum likelihood, mean field and a dynamic ICA model by Molgedey and Schuster. We further look at BIC for model selection, and PCA as general preprocessing tools in multimedia applications.

Chapter 4 The chapter is divided in two parts. At first ICA is used directly on the raw sound and images. We compare ICA algorithms and discuss positive constraints. The second part describes the ICA classification framework and application with text and images, individually and in combination.

Chapter 5 Exploiting the ICA properties, separation is done both in the context of chat rooms, and for classifying and visualizing search engine results.

CHAPTER 2

Modeling virtual environments

Virtual environments are computer generated cyperspaces or virtual spaces, only existing trough symbolic representation of data in different media. The purpose of the virtual environments (VE) are to share, store or process data in the sense of a meaningful human context.

In this chapter we look at what the virtual environments consist of and who inhabits them. We will in general reference the Internet, given that it is the largest and most rapidly expanding shared virtual environment today. Since it truly started in 1987 with 28,000 hosts [99], it has grown to over 160 million today [21]. Each host contains numerous services and web pages holding many independent contexts.

2.1 Structure and Ontologies of the Internet

The Internet is a network of computers that either provide and/or access information. The network communication protocol is *Transmission Control Pro-*

tocol/Internet Protocol (TCP/IP) and with this the Internet offers a number of services, e.g.

- Telnet, for accessing and exploiting other computers.
- File Transfer Protocol (FTP), for the up or downloading of files.
- Internet Relay Chat (IRC), lets users communicate through text online, giving simultaneous multiuser environments.
- Electronic mail (e-mail), gives access to send/recive mail messages and join discussion groups.
- World Wide Web (WWW or The Web), the fastest growing service that largely communicates using hypertext pages.

In many aspects the Internet is evolving in the same way as an ecosystem in the physical world about nature[18]. It has started out in its basic form by just being able to send simple one character messages, and grew into hypertext pages. As we see and envision it today, there are endless possibilities in its use. The structure of the Internet is changing all the time. No one person or company has the influence to change it into something that would just stay static for a reasonable short time frame. When new needs arise, new structure and ontologies are developed. If they prove to be of added value then they stay as a part of the virtual environment of the Internet. As such, we can see how some things have been further improved and other simply disappeared. Two good examples of this are *hypertext modeling language* HTML and *virtual reality modeling language* VRML on WWW. A basic homepage is structured in HTML and since the need for home pages has grown exponentially, the need for more advanced features has lead to many improved versions. In contrast to this VRML has not been an success. VRML was meant to be the equivalent 3-dimensional version of HTML. Many interesting features were added in a second version, in the form of movable and interactive objects. This gave the possibility for multiuser environments, where users could project themselves into the virtual environment in the form of for example, an avatar that could roam the virtual environment. However the 3D homepages never became a success and VRML is only used to a limited extent today. The reasons for this are many, but basically it was not a structure that most people could use in practice, other alternatives were better and so the strongest win. The Internet is therefore made up of many different and changing structures, which also makes the life of software agents difficult, as we shall see in the next section.

The structures of data and communication that the Internet services use are described by their *ontologies*. The word ontology is used in many different scientific communities and the meaning of the word is therefore somewhat ambiguous[33]. The philosophically meaning of the word *ontologos* is a neologism meaning ”to reason about being” and so the dictionary of *Merriam-Webster*[47] defines ontology as,

> **Ontology**: *”particular theory about the nature of being or the kinds of existents”.*

In our discussion ontology refers to a particular definition of a structure, e.g. a web page is implemented by the ontology of HTML and in short, we describe HTML as being the ontology of the web page.

The ontologies of the services provided by the Internet are basically designed either for human or computer to interpret. A computer displaying a WWW HTML page does not know the ”meaning” of its context for e.g. assisting search engines or roaming software agents. As the Internet has grown and new user-needs have evolved this has become an issue. Handling of large amounts of data for more complex tasks is needed, and so ontologies that hold information for both human and computer are needed.

One answer to this seems to lie in e.g.*extended markup language* (XML) which defines an ontology where the human semantics can be labeled with machine understandable tags. The tags are not predefined, so XML basically provide the foundation for higher level ontologies to specify these, depending on the more specific purpose. The XML ontology is proposed by The World Wide Web Consortium (W3C) [23] which has played a major role in developing for example, HTML. Examples of extensions to XML are,

- The Semantic Web[8], a new upcoming general extension to the world wide web, where information is given well defined meaning in the sense of both humans and computers.
- Resource Description Framework (RDF), providing a lightweight ontology system to support the exchange of knowledge on the Web. Appli-

cations include grouping and managing of news, software, photos and events, with relation to the user.

- Scalable Vector Graphics (SVG), a language for describing 2-D graphics. It holds basic forms, filtering effects, and scripting for dynamic and interactive objects.

At last we should also mention another likewise development in *moving picture experts group* (MPEG) research, that we will refer to in chapter 4. MPEG has been used for many years in digital video and audio compression, and lately more extensive on the Internet. As a part of a new upcoming version 7, it is intended to hold the video in a structure much like that of XML's ontology.

2.2 Software agents

The history of software agents started roughly around 1980, and became a real buzzword in both the popular computing press and the artificial intelligence community around 1994 [79]. The birth of the Internet had an exploding effect on this fairly new area, and the word agent is today used and misused in respect to many types of applications. A new community has evolved with its own journals, books and conferences. Some of the ongoing conferences are Agent-Oriented Information Systems (AOIS), Autonomous Agents (AA) and International Joint Conference on Artificial Intelligence (IJCAI). In all, more than two hundred conferences and workshops have been held the last couple of years. Among the largest journals focusing on software agents, are Artificial Intelligence from Elsevier Science, Autonomous Agents and Multi-Agent Systems from Kluwer Academic Publishers and Knowledge and Information Systems (KAIS) from Springer-Verlag.

Software agents, or just agents as we will refer to them in short, are used in numerous areas and it can therefore be hard to define a direct meaning of the word[103]. In the following we will give a brief overview of where we encounter agents today and what general properties they consists of. For current developments we point to [28, 32, 90].

2.2.1 Agent properties

Defining what an agent is in the software agent community is just as difficult as it is for the artificial intelligence community to define intelligence. In the literature we find three main communities that have an active interest in the field: computer science, artificial intelligence, and artificial life. Each group or community has its own perspective of an agent and its defining properties.

Computer science hold people that emerge from software application groups. They represent the largest group, with a foundation in computer programming and engineering. Most working software agents that are on the market today have been developed from computer science. Computer science looks at agents as being anything from a relatively simple program, to a fully grown autonomous application. They define the properties of an agent by the task it fulfills, e.g. an *e-mail agent* thus handles incoming e-mail.

The artificial intelligence (AI) community is more concerned with the aspects of science rather then commercial interests. They define an agent to be able to solve complex tasks, and explain its properties by its ”mental” behavior in doing so, e.g. by an agents knowledge, belief, intention and obligation.

The last of the three groups is the artificial life (ALife). ALife use bottom-up studies commonly associated with living organisms. Agent properties are associated with self-replication, evolution, adaptation, self-organization, parasitism, competition, cooperation and social network formation.

The different perspective of the three overall groups adds noise to the word agent. Over the last decade, some consensus as to the general properties of an agent have developed. A collection of the most general properties from the agent community literature is listed below,

Autonomous Operate without the direct control of humans or other agents. They must have at least partial control over their own actions and internal state.

Responsive Observe the surroundings and act accordingly on changes.

Proactive Be able to see opportunities as they arise or by themselves initiate one.

Social Contact other agents or humans when necessary.

Cooperate Mediate on communication with other agents.

Learn Adapt to the surroundings and internal states.

Mobility Exist in different surroundings.

Interfacing Man-machine interfacing.

The properties arise from the individual aims that the people in the agent community have taken. Specific agent development is therefore usually based on only a few of the properties. A deeper discussion of the topic can be found in [78, 51, 103, 98].

2.2.2 Multi-agent system

Multi-agent systems interconnect separately developed agents. In doing so larger tasks can be solved, e.g. by parallelization or better resource distribution. This has however not been explored very much. In [79] this point has been criticized for the lack of interest by the agent community. Given the short lifespan of the community and the Internet, we suspect that this kind of development has not been ready to evolve. This might although soon change in the years to come, with the introduction of XML based ontologies creating a common based environment for agents.

2.2.3 Agents today

Various agents are being used everyday. In the following we list a variety of agents that more or less live up to the properties of a software agent. It can easily be argued that some of the listed agents are simple programs, and should not belong there. However, given the loose definition of an agent, the list reflects more or less the point in development that we are at today. We expect that these agents will be the building blocks of more complex agents yet to come.

Virus Traditionally the computer virus has not been viewed upon as an agent, but given its autonomous nature and mobility its can be regarded as such.

Generally virus agents have some kind of destructive nature or enable a remote user to get access to a secure computer system. They usually spread very fast in order to survive, and some even have the ability to mutate over time in order not to get detected. Lately viruses with good intention have also been constructed. One examples of this is the Cheese Worm that makes its way around the web, checking computers for vulnerabilities and closing them if any are found[67]. Another virus called Noped checks computers for child pornography, and informs specified government agencies if any are found[26].

Help agent When looking for information, help agents can guide the way. The best known help agent is probably Microsofts Clippy in their MS-Office products. It helps the user via a common language text interface to look up answers in the online help[102]. Another help agent application is the help desk for e-mails. Support hotlines usually have to answer to the same questions again and again. This can be done by an agent or the agent can forward it to a human supporter if the answer is not known. An overview is given in[69]. Help agents are also more and more common in chat rooms, where they are called *bots*. They can help novice chat users and make sure that the chat stays active. The chat bot Poptoesen at the Jubii[17] chat room is one example of this.

Personal interest As a background agent these agents search for general information of interest for the user. Usually the agents are connected to a specific database as with the home buying agent from Nybolig[81] that alerts the user by e-mail when a purchase of interest is available. When surfing on the WWW the Surf Safari helps by predicting what pages are of interest. This is done by background searching the WWW and comparing results with the user's fields of interest[49].

Entertainment Computer game agents have in principle always been used in more or less sophisticated ways. A game like The Sims simulates an family of autonomous figures that the user can interact with. When left alone the family members (agents) go about their own business[75]. The need for human-like agents is also needed in the movie industry. Lately, more and more computer animated movies are produced, and movies like Final Fantasy: The Spirits Within was able to produce very realistic movements of the animated characters[85].

e-Commerce The introduction of the Internet has resulted in a new way of business called *e-Commerce*. In e-Commerce people can for example, buy directly on the WWW from their homes. In doing so, the behavior of the buyer can be traced by software agents that can exploit this to assist. One example of this is Amazone.com, one of the biggest Internet outlets which generally deals with books, music and films[45]. Likewise, people at the Pioneer Investment homepage can be guided by an agent on how to invest and in what[48].

2.2.4 Defining an agent

As stated in the previous sections we are not able to get a clear picture of what defining properties an agent should have. It is however clear that it ultimately should be a tool for humans. The agent or the multi-agent system, must therefore be able to handle both the human context paradigm and the often vast amount of high dimensional data in the virtual environments.

When speculating about an agent's most advanced form, it should be able to adopt to any given task. This calls for adaptiveness, like we see in living organisms. Also, agents should be able to form communities in order to solve larger tasks and parallel tasks. This is in principle what the artificial intelligence and artificial life community are trying to solve from their own perspective[29, 95]. As the software agent community evolves together with the virtual environments, we suspect that agents will evolve into virtual living and intelligent entities. In order to serve humans they need to learn reasoning and meaning in a human context. We will not speculate on the possibility that software agents can evolve further than humans[1], or if they will become conscience at some point, since this is long past the topic of this thesis.

Thus, as presented in the introduction, we aim to investigate the use of the independency criteria for the use in VE tools, given that it should reflect separation properties natural in a human sense. For this we use independent component analysis. We therefore turn our attention to the ICA algorithm in the next chapter, regarding its properties and framework.

[1] It is not clear in what regards the human paradigm is optimal.

CHAPTER 3

Independent component analysis

Achieving *blind source separation* (BSS) with *independent component analysis* (ICA) is a fairly new and fast growing field. In BSS the word blind refers to the fact that we do not know how the the signals were mixed or how they were generated. As such, the separation is in principal impossible. Allowing some relatively indirect and general constrains, we however still hold the term BSS valid, and separate under these conditions.

A classic problem in BSS is the *cocktail party problem*, as shown in figure 3.1. The objective is to sample a mixture of spoken voices, with a given number of microphones - *the observations*, and then separate each voice into a separate speaker channel - *the sources*. The BSS is unsupervised and thought of as a black box method. In this we encounter many problems, e.g. time delay between microphones, echo, amplitude difference, voice order in speaker and underdetermined mixture signal.

At seminar work in 1986, Herault and Jutten pursued the idea that the separation could be done by reducing redundancy between signals, in a artificial neural network like architecture. This approach initially lead to what is known as independent component analysis today. The fundamental research involved

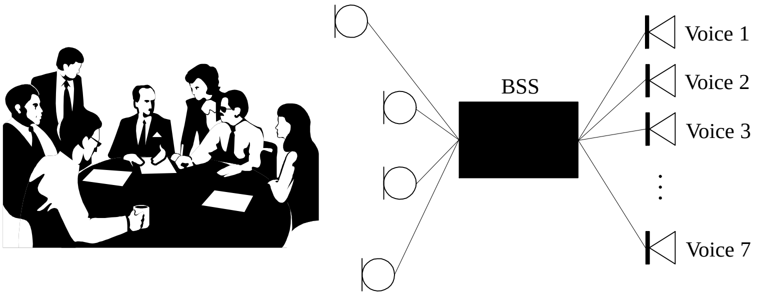


Figure 3.1 The figure illustrates the cocktail party problem. Seven people are talking and the BSS task is to separate each of the speakers voices, without knowledge of the mixing or generation of the voices.

only a handful of researchers up until 1995. It was not until then, when Bell and Sejnowski [7] published a relatively simple approach to the problem named *infomax*, that many became aware of the potential of ICA. Since then a whole community has evolved around ICA, centralized around some large research groups[1] and its own ongoing conference, *International Conference on independent component analysis and blind signal separation*[80]. ICA is used today in many different applications, e.g. medical signal analysis, sound separation, image processing, dimension reduction, coding and text analysis.

In ICA the general idea is to separate the signals, assuming that the original underlying source signals are mutually independently distributed. Due to the field's relatively young age, the distinction between BSS and ICA is not fully clear. When regarding ICA, the basic framework for most researchers has been to assume that the mixing is instantaneous and linear, as in infomax. ICA is often described as an extension to PCA, that uncorrelates the signals for higher order moments and produces a non-orthogonal basis. More complex models assume for example, noisy mixtures[72, 34], nontrivial source distributions[52, 97], convolutive mixtures[4, 63], time dependency, underdetermind sources[68, 41], mixture and classification of independent component[64, 57]. A general introduction and overview can be found in [62].

In the following we will look at the properties of ICA, and present the ICA

[1] e.g. at Computational Neuroscience Lab lead by Terry Sejnowskis[22], Laboratory of Computer and Information Science at Helsinki University of Technology lead by professor E. Oja[38] and TSI Department Signal-Images lead by J. Cardoso[94]

algorithms used in this text. Finally we address the topics of preprocessing with PCA, and model selection using the Bayesian information criterion.

3.1 Model

The general model for ICA is that the sources are generated through a linear basis transformation, where additive noise can be present. In this text we consider the model to be,

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \boldsymbol{\Gamma}, \qquad \mathbf{X}_{m,n} = \sum_{k=1}^{N_k} \mathbf{A}_{m,k}\mathbf{S}_{k,n} + \boldsymbol{\Gamma}_{m,n}, \tag{3.1}$$

where $\mathbf{X}$ is the matrix holding the N_m mixed or observed signals in each row with N samples, $\mathbf{A}$ is the $N_m \times N_k$ basis transformation or mixing matrix, and $\mathbf{S}$ is the matrix holding the N_k independent source signals in rows of N samples. The noise is added by the $N_m \times N$ matrix $\boldsymbol{\Gamma}$ that is generally defined to be Gaussian or fully neglected.

3.2 Properties of ICA

Independent sources

The fundamental principle in ICA is that the sources are independent of each other. By this we mean that they are statistically independent, thus the joint probability of a given multivariate sample $\mathbf{s} = [s_1, s_2, ..., s_{N_k}]^\top$ is therefore equal to the product of its marginal distributions as,

$$p_{\mathbf{s}}(\mathbf{s}) = \prod_{k=1}^{N_k} p_s(s_k). \tag{3.2}$$

In terms of optimization, the ICA algorithm can therefore directly or indirectly be defined as minimizing the *Kullback-Leibler* (KL) divergence between the estimated joint distribution and the product of the marginal,

$$KL(\hat{p}_{\mathbf{s}}(\mathbf{s})||\Pi\hat{p}_s(s_k)) = \int_{-\infty}^{\infty} \hat{p}_{\mathbf{s}}(\mathbf{s}) \log \frac{\hat{p}_{\mathbf{s}}(\mathbf{s})}{\prod_{k=1}^{N_k} \hat{p}_s(s_k)}\, d\mathbf{s}. \tag{3.3}$$

The KL divergence measures the distance between two probability distributions, and becomes zero when the distributions are equal. However it should be noted that the KL divergence is not symmetrical.

The KL divergence can only rarely be solved analytically. In infomax[7] the mutual entropy of the estimated sources is maximized. That essentially is equivalent to minimizing eq. (3.3) when employing a non-linear function[2]. In [19] and [2] the KL divergence was estimated using *Gram-Charlier* or *Edgeworth* expansion of moments.

Since ICA is an unsupervised algorithm, the estimated sources will converge to a false optimum if the true sources are not independent.

Higher order moments

Independence can also be expressed directly in terms of moments. If the signals are uncorrelated for all moments, including the higher order moments, then they are considered independent[4]. A signal $\mathbf{s}_a = [s_{a_1}, s_{a_2}, ..., s_{a_N}]$ and $\mathbf{s}_b = [s_{b_1}, s_{b_2}, ..., s_{b_N}]$ are independent if,

$$E_c[\mathbf{s}_a^p \cdot \mathbf{s}_b^q] = E_c[\mathbf{s}_a^p] \cdot E_c[\mathbf{s}_b^q], \quad \forall p, q > 0, \tag{3.4}$$

where $E_c[\mathbf{s}^m] = E[(\mathbf{s} - \mu)^m]$ with μ being the mean over N samples. It has although been shown in [12] that it is sufficient to achieve independence by estimating no more than fourth order moments. In the often assumed case, where the source signals have zero mean and the source distributions are symmetric around zero, only the second and fourth order moments are left to find. The second order moments can be found using e.g. principal component analysis, thus ICA amounts to finding the fourth order, that is equivalent to a rotation basis[44].

Gaussian distributed signals only hold unique moments up to the second order, where higher order moments can be described by these. As such, ICA algorithms cannot separate Gaussian signals from each other, given that information on the higher order moments is missing[19].

[2] The non-linear function is called a squashing function and amounts to being the c.d.f. of the sources.

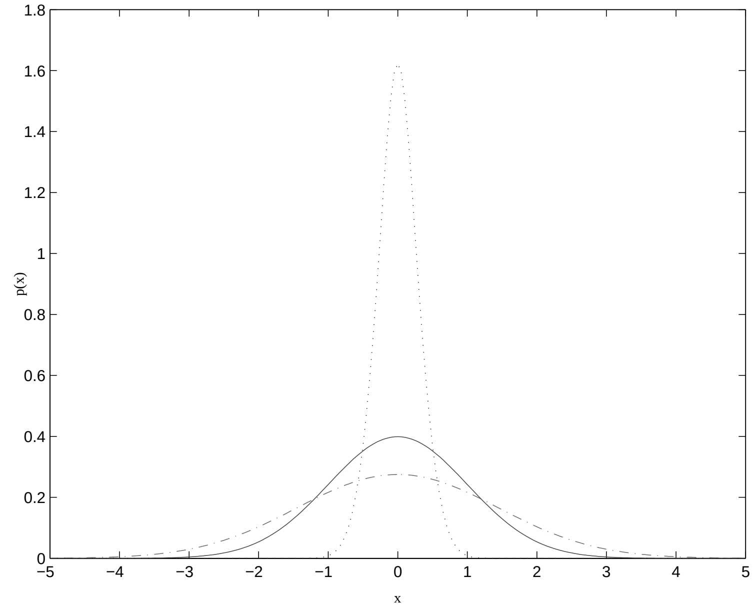


Figure 3.2 Distributions from signals with different kurtosis: $[\cdot -]$ A sub Gaussian signal have negative kurtosis, and can generally be described as being more uniformly distributed compared to Gaussian signals. $[\cdot\cdot]$ A super Gaussian signal is characterized as a heavy tailed and typically a sparse signal that is mostly distributed around zero. Speech signals are typically super Gaussian. $[-]$ A Gaussian signal. The kurtosis are respectively -1.3, 12.4 and 0.0.

The fourth order moment can be expressed as the signal's kurtosis γ, and describes the ”top-steep-ness” of a signal $\mathbf{s} = [s_1, s_2, ..., s_{N_N}]$,

$$\gamma = \frac{E[(\mathbf{s} - \mu)^4]}{(\sigma^2)^2} - 3, \tag{3.5}$$

where μ and σ^2 are respectively the mean and variance over the signal samples. The kurtosis becomes zero in the case of a Gaussian signal, positive in the case of a super Gaussian signal and negative in the case of a sub Gaussian signal, as shown in figure 3.2. Thus, $N_k - 1$ signals need to have a kurtosis different from zero for the separation to be possible [65].

Source probability distribution

Recovering the source signals involves more or less directly the source signals probability distributions. Many different approaches have been used, ranging from fully parameterized to static functions. Surprisingly, the latter has proved to be remarkably robust, even with large deviations from the true distribution. In infomax the sources cumulative density functions are needed, as in the equivalent *maximum likelihood* (ML) case that we discuss later. It was suggested in [7] that a simple monotonically growing function is sufficient to estimate the c.d.f., and that it is merely used to bound the parameters. This does though, limit the source signals to be either sub- or super-Gaussian, if the algorithm does not take this into account, e.g. as in the *extended infomax*[65]. In the case of zero mean probability distributions, the error made by not matching the source distributions (if not too gross) results in merely a scaling of the estimated signals[11].

Expecting e.g. more skew or fully positive source distributions, as implemented in [52, 97], can be a vital criteria in order to e.g. avoid *anti-correlated* components, as we shall look at later regarding images and text analysis. The basic properties of the underlying source distributions need therefore to be respected, although it might not make the optimization of the ICA algorithm unstable.

Mixing matrix

The mixing matrix in eq. (3.1) can be thought of as being a non-orthogonal transformation basis. The columns in $\mathbf{A}$ are linearly independent and must have full rank. The matrix can at best be recovered from the true mixing matrix up to a scaling and permutation of the matrix rows. We think of the mixing matrix as,

$$\mathbf{A} = \widetilde{\mathbf{A}}\boldsymbol{\Delta}\boldsymbol{\Pi}, \tag{3.6}$$

where $\boldsymbol{\Delta}$ is a $N_k \times N_k$ diagonal matrix containing the scaling (including possible sign), and $\boldsymbol{\Pi}$ is a $N_k \times N_k$ permutation matrix that interchanges rows, having one unique element in each row set to one and the rest zero. The $\widetilde{\mathbf{A}}$ matrix is the original $N_m \times N_k$ mixing matrix that we cannot recover without further information.

The number of sources, hence columns in $\mathbf{A}$, are generally not known and must

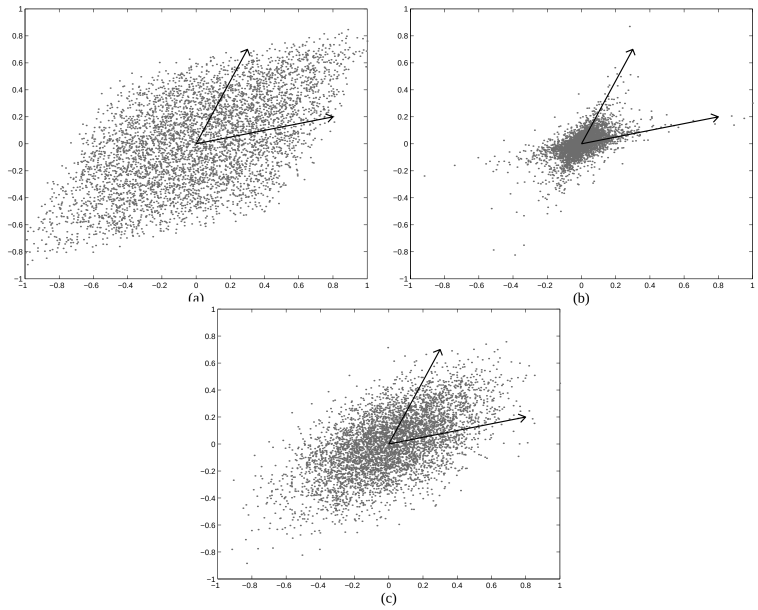


Figure 3.3 The scatter plots form signals with different kurtosis in pairs, that have been mixed linear. The arrows show the basis of the mixing matrix. (a) Sub Gaussian signals gives a scatter plot that becomes more rectangular on the edges. (b) Super Gaussian signals give a scatter plot that is mostly distributed around zero and that has distinct tails. (c) Gaussian signals show a oval scatter plot. In the sub and super Gaussian plots the edges and tails align with the mixing matrix basis and so give evidence for separation, unlike in the pure Gaussian signals. The distributions from which the signals were drawn from are shown in figure 3.2.

be estimated. In the case where the number of sources and number of observed signals are the same, the problem simplifies and the un-mixing matrix can be found as the inverse of $\mathbf{A}$. In the case where more observations are present than sources, the mixing matrix does not have full rank, and is said to be *under-complete*. The last case is where the number of observations are less than the number of sources. The information for estimating the sources is therefore *underdetermined*, and called the *overcomplete* case in regards to the mixing matrix. Development in this field has not matured yet, but ICA algorithms have been able to handle this under reasonable conditions, e.g. [68].

Independence measure

Ensuring that an ICA algorithm has converged to independent source signals is normally hard to determine if the sources are not known. Straightforward approaches give an estimate of the KL divergence [2], but drawing the histograms of the source signals, which represent the joint and marginal probability distributions, might also give evidence to the success of the separation [55].

We found that drawing the scatter plots for the signals in pairs was the most reliable tool. In figure 3.3, scatter plots from signals drawn from the distributions presented in figure 3.2 are shown. In the case of sub- and super-Gaussian signals we can recognize structure for the separation, as opposed to the Gaussian scatter plot, where we cannot find the basis of the mixing proportions in the plot. For Example in text analysis the signals are generally super Gaussian distributed, and we therefore expect the signals to be scattered along the source dimensions, i.e. axis of the scatter plot.

ICA approaches

There are largely two main approaches to ICA at the current date. One can be traced back to a probabilistic framework, where we formulate our problem to solve the maximum likelihood of the observed signals [72]. Here we also have the infomax algorithm as the simplest case, but also being very robust. The second approach is based on joint diagonalization[53], e.g. as in the Molgedey and Schuster algorithm.

Other ICA-like algorithms exist, e.g. *complexity pursuit* [43], but we do not regard them as true ICA algorithms unless the objective of independent sources is present.

3.3 Probabilistic ICA

In probabilistic ICA we think of eq. (3.1) as being a generative model. The source signals are latent variables and the mixed signals are the observations. Both are described by their probability distributions. The noise is regarded as Gaussian distributed by $\mathbf{\Gamma} \sim \mathcal{N}(0, \mathbf{\Sigma})$. The objective is hereby to find an estimate of $\mathbf{S}$, $\mathbf{A}$ and $\mathbf{\Sigma}$ for a given model $\mathcal{M}$, where we know the number of observation N_m and sources N_k, and we are given the mixed signals $\mathbf{X}$, sampled independently in time.

Using Bayes theorem the relationship between the probability distributions of $\mathbf{X}$ and $\mathbf{S}$ can be inferred,

$$
\begin{aligned}
p(\mathbf{X}, \mathbf{S}|\mathbf{A}, \mathbf{\Sigma}) &= p(\mathbf{S}|\mathbf{X}, \mathbf{A}, \mathbf{\Sigma})p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}) \qquad (3.7)\\
&\text{and}\\
p(\mathbf{X}, \mathbf{S}|\mathbf{A}, \mathbf{\Sigma}) &= p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma})p(\mathbf{S}). \qquad (3.8)
\end{aligned}
$$

Eq. (3.7) is trivial, given that the mixed signals are generated from the mixing matrix and noise. In eq. (3.8) we have that $p(\mathbf{S}|\mathbf{A}, \mathbf{\Sigma}) = p(\mathbf{S})$, since the true sources are not dependent on the mixing matrix or the noise. Furthermore we now can impose the constraint of independence from eq. (3.2) that $p(\mathbf{S}) = \prod_{k=1}^{N_k} p(\mathbf{S}_k)$, where $\mathbf{S}_k$ is a row in the source matrix.

In the following we will look at two approaches to solve the probabilistic ICA, either by directly maximizing the likelihood or by a mean field approach.

3.3.1 Maximum likelihood

In the maximum likelihood approach we marginalize over the latent variables. This involves solving an integral that might not always be trivial and therefore not attractive. In the following we formulate this approach based mainly on the work of [72, 34, 11], and look closer at the special case with a square mixing matrix and where no noise is present to derive the equivalent infomax solution.

The likelihood of the mixed signals is defined as the product over each multivariate sample distribution given the mixing matrix and noise covariance matrix, $p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{A}, \mathbf{\Sigma})$. Assuming that the source signals are the latent variables, we can write the likelihood as the marginal distribution and using eq. (3.8) we get,

$$
p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}) = \int p(\mathbf{X}, \mathbf{S}|\mathbf{A}, \mathbf{\Sigma}) d\mathbf{S} = \int p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) \prod_{k=1}^{N_k} p(\mathbf{S}_k)\, d\mathbf{S}, \quad (3.9)
$$

where we imposed the independence criteria on the source prior, with $p(\mathbf{S}_k)$ as the probability distribution of the k'th source component. By $p(\mathbf{X}|\mathbf{S},\mathbf{A},\mathbf{\Sigma}) = p(\mathbf{AS}+\mathbf{\Gamma}|\mathbf{S},\mathbf{A},\mathbf{\Sigma})$, we have that $\mathbf{A}$ and $\mathbf{S}$ become constants by the conditioning, and using the property of linear transformation between probability functions[3] we have,

$$p(\mathbf{X}|\mathbf{A},\mathbf{\Sigma}) = \int p(\mathbf{X}-\mathbf{AS}|\mathbf{\Sigma}) \prod_{k=1}^{N_k} p(\mathbf{S}_k)\; d\mathbf{S}, \tag{3.10}$$

where the probability $p(\cdot|\mathbf{\Sigma})$ is now the Gaussian noise function,

$$p(\mathbf{\Gamma}) = (\det 2\pi\mathbf{\Sigma})^{-\frac{N}{2}}\; e^{-\frac{1}{2}\mathrm{Tr}\,\mathbf{\Gamma}^\top\mathbf{\Sigma}^{-1}\,\mathbf{\Gamma}}. \tag{3.11}$$

No noise case

In the special case when assuming that the mixing matrix is an invertible square matrix and that no noise is present, we get the infomax solution as shown by [72, 11].

If we assume that the covariance matrix $\mathbf{\Sigma}$ of the noise distribution has elements that are infinitesimal small, the noise distribution becomes a delta function. We also assume that the number of sources are equal to the number of mixed signals, $m = k$. The mixing matrix is therefore square, and if it has full rank, we can find the unmixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ as follows. The likelihood can be written as,

$$p(\mathbf{X}|\mathbf{A}) = \int \prod_{m=1}^{N_m} \delta_m(\mathbf{X}-\mathbf{AS}) \prod_{k=1}^{N_k} p(\mathbf{S}_k)\; d\mathbf{S}, \tag{3.12}$$

where the product over the delta function comes from the fact that it is the noise function, thus independent between samples and channels. This integral can be solved[4], and writing it as the log likelihood we get,

$$\log p(\mathbf{X}|\mathbf{A}) = N \log\det \mathbf{A}^{-1} + \sum_{k=1}^{N_k} \log p(\mathbf{S}_k) \tag{3.13}$$

[3] For $x = ay + b$ the relation between the probability functions of x and y is $p_x(x) = \frac{1}{|a|} p_y(\frac{x-b}{a})$ where a and b are constants.

[4] For scalars we have $\int \delta(x-as)p(s)\; ds = \frac{1}{|a|} p(x/a)$ [72].

Substituting and differentiating with respect to $\mathbf{W}$ we can obtain the gradient for updating the unmixing matrix in an iterative optimization method,

$$\frac{\partial}{\partial \mathbf{W}} \log p(\mathbf{X}|\mathbf{A}) = \frac{\partial}{\partial \mathbf{W}} N \log \det \mathbf{W} + \sum_{k=1}^{N_k} \frac{\partial \log p(\mathbf{S}_k)}{\partial \mathbf{S}_k} \frac{\partial \mathbf{S}_k^\top}{\partial \mathbf{W}} \tag{3.14}$$

where $\mathbf{\Phi}(\mathbf{S}_k) = \frac{\partial}{\partial \mathbf{S}_k} \log p(\mathbf{S}_k)$, that we replace with a static sigmoid function. Solving the derivative amounts to,

$$\frac{\partial}{\partial \mathbf{W}} \log p(\mathbf{X}|\mathbf{A}) = N(\mathbf{W}^\top)^{-1} + \mathbf{\Phi}(\mathbf{S})\mathbf{X}^\top \tag{3.15}$$

Choosing the function of $\mathbf{\Phi}$ is not gravely important as pointed out in the above section, and setting $\mathbf{\Phi} = -\tanh$ matches directly that of the infomax solution[7] to separate super-Gaussian signals. This implies a source distribution $P(s) = 1/\pi \exp(-\log \cosh s)$. The source signals can hereafter be found as $\mathbf{S} = \mathbf{W}\mathbf{X}$.

In extension to the gradient in eq. (3.15), a remarkable improvement has been done in terms of optimization by Amari[2], where the gradient is corrected in each iteration to follow the *natural gradient* instead. The natural gradient takes into account how the parameter space is conditioned locally. When optimizing, the update with the natural gradient is found in [2] to be $[\frac{\partial}{\partial \mathbf{W}} \log p(\mathbf{X}|\mathbf{A})]\mathbf{W}^\top\mathbf{W}$, which also takes care of the matrix inversion in eq. (3.15).

3.3.2 Mean field

To avoid the often intractable integral in eq. (3.9) we can use *mean field* (MF). In the mean field approximation (MF) we find the mean of the sources and their covariance matrix, and use these to describe the sources, mixing matrix and the noise covariance matrix, thus they describe the sufficient statistics of the model.

Mixing matrix and noise covariance matrix

The derivative of the log likelihood can be formulated in the mean field sense without the integral. As shown from appendix A.1 we can write,

$$\frac{\partial}{\partial \mathbf{A}} \log p(\mathbf{X}|\mathbf{A},\boldsymbol{\Sigma}) = \langle \frac{\partial}{\partial \mathbf{A}} \log p(\mathbf{X}|\mathbf{S},\mathbf{A},\boldsymbol{\Sigma}) \rangle_{p(\mathbf{S}|\mathbf{X}\mathbf{A}\boldsymbol{\Sigma})} \tag{3.16}$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \log p(\mathbf{X}|\mathbf{A},\boldsymbol{\Sigma}) = \langle \frac{\partial}{\partial \boldsymbol{\Sigma}} \log p(\mathbf{X}|\mathbf{S},\mathbf{A},\boldsymbol{\Sigma}) \rangle_{p(\mathbf{S}|\mathbf{X}\mathbf{A}\boldsymbol{\Sigma})} \tag{3.17}$$

where $\langle \cdot \rangle$ is the posterior average over the sources, and will be implied in the following when used. The log likelihood of the mixed signals conditioned on the mixing matrix, the noise covariance matrix and the sources, was found in the above section as the Gaussian distribution, thus from eq. (3.11) and (3.10) we get,

$$p(\mathbf{X}|\mathbf{S},\mathbf{A},\boldsymbol{\Sigma}) = (\det 2\pi\boldsymbol{\Sigma})^{-\frac{N}{2}}\ e^{-\frac{1}{2}\mathrm{Tr}\,(\mathbf{X}-\mathbf{A}\mathbf{S})^\top \boldsymbol{\Sigma}^{-1}\,(\mathbf{X}-\mathbf{A}\mathbf{S})}. \tag{3.18}$$

Evaluating the ML on the right hand side of eq. (3.16) and (3.17) w.r.t. either the mixing matrix or the noise covariance matrix, and then setting them equal to zero, amounts to a mean field solution,

$$\mathbf{A} = \mathbf{X}\langle\mathbf{S}\rangle\langle\mathbf{S}\mathbf{S}^\top\rangle^{-1} \tag{3.19}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\langle(\mathbf{X}-\mathbf{A}\mathbf{S})(\mathbf{X}-\mathbf{A}\mathbf{S})^\top\rangle. \tag{3.20}$$

In the case of i.i.d. noise, the noise covariance matrix simplify to a diagonal matrix with elements $\sigma^2 = \frac{1}{N_m}\mathrm{Tr}\boldsymbol{\Sigma}$.

In [97] the mixing matrix is found through the *maximum a posterior* (MAP) solution, having $p(\mathbf{A}|\mathbf{X},\boldsymbol{\Sigma}) \propto p(\mathbf{X}|\mathbf{A},\boldsymbol{\Sigma})p(\mathbf{A})$. Conditions on the mixing matrix can hereby nicely be imposed through $p(\mathbf{A})$, as e.g. positive mixing coefficients.

Source signals

In the mean field solution we found that the mixing matrix and the noise covariance matrix could be described by $\langle\mathbf{S}\rangle$ and $\langle\mathbf{S}\mathbf{S}^\top\rangle$, hence being the sufficient statistics. Different approaches can be taken to find these, and following [34] we will assume that,

$$\langle\mathbf{S}\rangle = \widehat{\mathbf{S}}\ \ ,\ \ \langle\mathbf{S}\mathbf{S}^\top\rangle = \widehat{\mathbf{S}}\widehat{\mathbf{S}}^\top + \beta\mathbf{1}, \tag{3.21}$$

where $\widehat{\mathbf{S}}$ is the solution of the MAP estimate of the sources, and $\mathbf{1}$ is the $N_k \times N_k$ identity matrix. Solving for the mixing matrix in eq. (3.20) the noise covariance term vanishes when setting the derivative of the log likelihood to zero. In [34] it is therefore argued that inserting β helps to ensure stability, if the source covariance matrix is badly conditioned. Estimating the value of β can be done in the low noise limit, based on a Gaussian approximation to the likelihood [34]. Other approaches in determining the sufficient statistics, e.g. *variational*, *linear response* and *adaptive TAP*, has in general proved to give better estimates [97], but outside of the scope of this writing.

In the MAP estimate we maximize w.r.t. the sources on the full conditioned source distribution. Equating eq. (3.7) and (3.8) we get,

$$p(\mathbf{S}|\mathbf{X},\mathbf{A},\mathbf{\Sigma}) \propto p(\mathbf{X}|\mathbf{S},\mathbf{A},\mathbf{\Sigma})p(\mathbf{S}) = p(\mathbf{X}-\mathbf{AS}|\mathbf{\Sigma})\prod_{k=1}^{N_k} p(\mathbf{S}_k). \tag{3.22}$$

Inserting eq. (3.18) and introducing the log on both sides leads to the same form as we saw in the ML case of eq. (3.13).

$$\log p(\mathbf{S}|\mathbf{X},\mathbf{A},\mathbf{\Sigma}) \propto -\frac{1}{2}\mathrm{Tr}\,(\mathbf{X}-\mathbf{AS})^\top\mathbf{\Sigma}^{-1}(\mathbf{X}-\mathbf{AS}) + \sum_{k=1}^{N_k}\log p(\mathbf{S}_k), \tag{3.23}$$

where we have omitted the log determinant term, given that it is not dependent on the sources. Differentiating w.r.t. the sources, we identify $\mathbf{\Phi}(\mathbf{S}_k) = \frac{\partial}{\partial \mathbf{S}_k}\log p(\mathbf{S}_k)$. Setting $\mathbf{\Phi} = -\tanh$ as before we get,

$$\frac{\partial}{\partial \mathbf{S}}\log p(\mathbf{S}|\mathbf{X},\mathbf{A},\mathbf{\Sigma}) = \mathbf{\Sigma}^{-1}(\mathbf{A}^\top\mathbf{X} - \mathbf{A}^\top\mathbf{A}\widehat{\mathbf{S}}) - \tanh(\widehat{\mathbf{S}}). \tag{3.24}$$

This can be used directly an in iterative gradient optimization method, or as proposed by [34], solve for the optimum when setting it to zero, and getting a faster and more stable convergence.

Solving the full ICA problem then amounts to alternately updating of both the mixing matrix and noise covariance matrix, and estimating the sources.

3.4 Molgedey and Schuster

The Molgedey and Schuster (MS) ICA algorithm is based on time delayed decorrelation of the mixed signals, thus the signals need to be correlated in time. The sources called dynamic components, are assumed to be Gaussian distributed with unique autocorrelation functions, and so higher order moments are not necessary for separation. The algorithm is based on the joint diagonalization approach, and simply amounts to solving an eigenvalue problem of a quotient matrix. The quotient matrix holds among other the mixed signals to a given delay τ, that is the only parameter to be specified.

In the joint diagonalization for ICA problems, the idea is to solve a series of matrices to be diagonal under the constraint of independence, e.g. cumulant matrices in Jade by Cardoso[13]. Given a set of $\mathbf{M}_1, ..., \mathbf{M}_L$ rectangular real matrices, we want to find a non-orthogonal matrix $\mathbf{A}$ that holds,

$$\mathbf{M}_l = \mathbf{A}\mathbf{\Lambda}_l\mathbf{A}^{-1}, \tag{3.25}$$

where $l = 1, ..., L$ and each $\mathbf{\Lambda}_l$ is a diagonal matrix corresponding to a given $\mathbf{M}_l$ [53].

In the following we will derive the MS separation for a square mixing matrix. We will look at finding the delay τ, and finally write out its likelihood in order to handle model selection.

3.4.1 Source separation

Let $\mathbf{X}$ be the matrix holding the mixed signals that are correlated in time. We write a τ time shifted matrix of the mixed signals as $\mathbf{X}_\tau$, that can either be cyclic or truncated, depending on its border conditions. We now want to solve the simultaneous eigenvalue problem of $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}_\tau\mathbf{X}^\top$ by defining a quotient matrix,

$$\mathbf{Q} \equiv \mathbf{X}_\tau\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}. \tag{3.26}$$

Having no noise and inserting eq. (3.1) with a square mixing matrix, we can write

$$\mathbf{Q} = \mathbf{A}\mathbf{S}_\tau\mathbf{S}^\top\mathbf{A}^\top(\mathbf{A}\mathbf{S}\mathbf{S}^\top\mathbf{A}^\top)^{-1}. \tag{3.27}$$

In the limit when the number of samples goes to infinity, we have that the cross-correlation is equal to a diagonal matrix, given the sources are independent and time correlated ergodic for,

$$\lim_{N\to\infty}\frac{1}{N}\mathbf{S}\mathbf{S}^\top = \mathbf{C}_0 \ \ , \ \ \lim_{N\to\infty}\frac{1}{N}\mathbf{S}_\tau\mathbf{S}^\top = \mathbf{C}_\tau. \tag{3.28}$$

The crosscorrelation matrix of the sources and the time shifted crosscorrelation matrix are written as $\mathbf{C}_0$ and $\mathbf{C}_\tau$ respectively. Inserted into eq. (3.27) we get,

$$\mathbf{Q} = \mathbf{A}\mathbf{C}_\tau\mathbf{C}_0^{-1}\mathbf{A}^{-1}, \tag{3.29}$$

where we identify the multiplication of $\mathbf{C}_\tau\mathbf{C}_0^{-1}$ as a diagonal matrix. Solving this eigenvalue problem, we get the mixing matrix directly,

$$\mathbf{Q} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^{-1}, \tag{3.30}$$

where $\mathbf{\Lambda} = \mathbf{C}_\tau\mathbf{C}_0^{-1}$. The sources can then be found by $\mathbf{S} = \mathbf{A}^{-1}\mathbf{X}$.

Some practical problems arise from the fact that we are dealing with a limited number of samples N. We know that $\mathbf{C}_\tau$ needs to be a diagonal matrix, and this is only true if the matrix $\mathbf{X}_\tau\mathbf{X}^\top$ is symmetric, given it must hold that $\mathbf{X}_\tau\mathbf{X}^\top = \mathbf{A}\mathbf{C}_\tau\mathbf{A}^\top$ for real values of $\mathbf{A}$. We must therefore ensure that $\mathbf{X}_\tau\mathbf{X}^\top$ is symmetric, thus the quotient matrix can be written as,

$$\mathbf{Q}_s = \frac{1}{2}(\mathbf{X}_\tau\mathbf{X}^\top + \mathbf{X}\mathbf{X}_\tau^\top)(\mathbf{X}\mathbf{X}^\top)^{-1}. \tag{3.31}$$

3.4.2 Determination of τ

Experiments have shown that choosing the value of τ has a crucial influence on the separation. We might use a model selection approach with an exhaustive search of the best delay τ, as we describe in the next section. This proves although too computational costly in order to preserve the otherwise fast property of the MS algorithm. We therefore look closer at the problems around determining τ.

First we recognize the problem if τ is not chosen such that the quotient matrix becomes non trivial. In the case of over sampled mixed signals and e.g. setting the value of $\tau = 1$ as is often seen, will result in an quotient matrix close to the unit matrix. Likewise if the time shifted mixed signals are uncorrelated by e.g. choosing a value of τ that is too large, then the quotient matrix degenerates. Choosing τ with these considerations in mind is a reasonable task given a specific data set, and so we address the second problem that seem to have a great impact. For the eigenvalue problem in eq. (3.30) to have a unique solution, the eigenvalues in $\mathbf{\Lambda}$ must be unique. In figure 3.4 (top left) the eigenvalues as a function of τ are plotted, thus it becomes clear that there is a connection

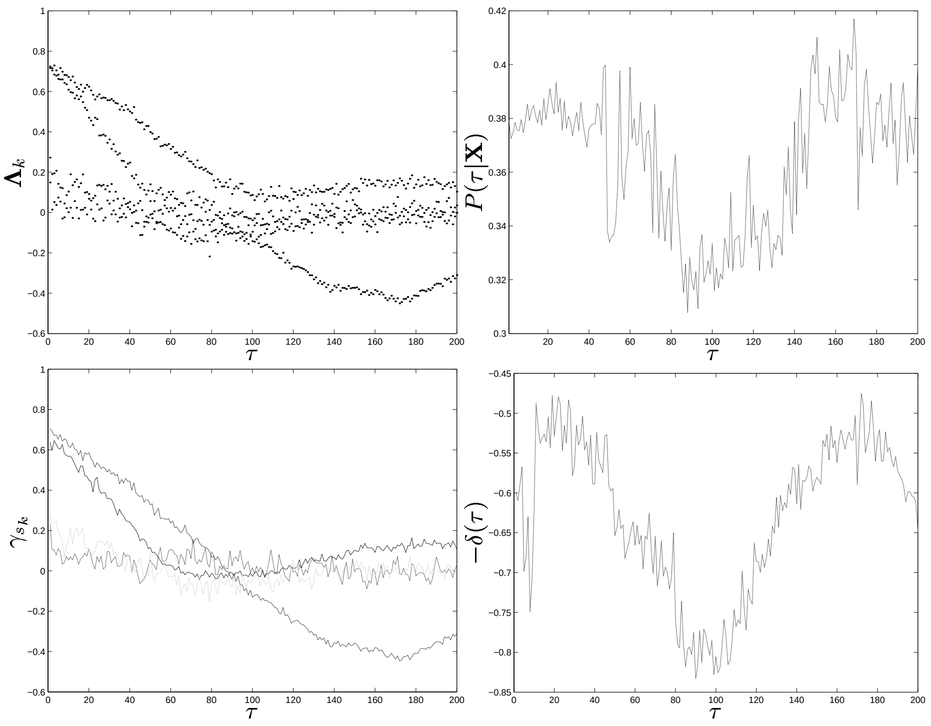


Figure 3.4 For the eigenvalue problem to have a unique solution, the eigenvalues themselves (top left) need to be unique . The autocorrelations of the sources (bottom left) resemble the eigenvalues closely. A Bayesian scheme (top right) for estimating the optimal lag value τ is compared with a computationally much simpler approach (bottom right), where the τ is chosen to be equal to the lag of which provides the most widely distributed autocorrelation function values of the sources (bottom left). The best τ for the Bayesian approach was $\tau = 169$, and for the δ-function $\tau = 172$ in this chat room example.

between the two. The data used in the figure is from a chat room experiment, and is described in chapter 5 when separated for 4 sources. In eq. (3.30) we have that $\mathbf{\Lambda} = \mathbf{C}_\tau \mathbf{C}_0^{-1}$, meaning that the eigenvalues can be described by the sources autocorrelation for a given τ. In figure3.4 (bottom left) the autocorrelation functions of the sources $\gamma_{s,t} = \sum_{n=1}^{N-t} s_n s_{n+t}$ are plotted for $t = 1 = \tau$. A close resemblance is observed between the eigenvalues and the autocorrelation functions, thus the MS separation seem to succeed reasonably on the basis of just one time shifted joint diagonalization. It was suggested in [104] that using multiple time lags of τ might improve the separation. In preliminary tests we did however not find evidence of this, both when selecting a wide range e.g. $\tau \in [1..N/2]$, or when hand picking multiple selected values of τ.

Comparing the autocorrelations with the Bayes optimal model selection from eq. (3.38) using BIC that we describe later, we observed a clear reduction in probability when the autocorrelation of the sources were overlapping, as seen in figure 3.4 (right top). Investigating this further, we formulated an objective function δ for identification of τ, enforcing sources with autocorrelation values which are as widely distributed as possible. For a specific value of τ we have

$$\delta(\tau) = \sum_{i=1}^{K-1} \left| \rho_{s_{i+1}}(\tau) - \rho_{s_i}(\tau) - \frac{1}{K-1} \right| , \tag{3.32}$$

where $\rho_{s_{i+1}}(\tau) > \rho_{s_i}(\tau)$ are the sorted normalized autocorrelations $\rho_{s_i}(m) = \gamma_{s_i}(m)/\gamma_{s_i}(0)$. When comparing the selection according to $\delta(\tau)$ and the Bayes optimal model selection procedure it clearly showed similar behavior, as seen in figure 3.4 (right bottom).

The procedure for determination of τ thus consists of first estimating the sources and associated normalized autocorrelation functions for a initial value, e.g. $\tau = 1$. Second, select the τ with the smallest $\delta(\tau)$, and reestimate the ICA. In principle this procedure is iterated until the value of τ stabilizes, which in experiments always was obtained in less than 5 iterations.

3.4.3 Likelihood

The likelihood of the mixed signals can be written in the same framework as in the ML setting.

$$p(\mathbf{X}|\mathbf{A}) = \int \prod_{m=1}^{N_m} \delta_m(\mathbf{X} - \mathbf{AS}) \prod_{k=1}^{N_k} p(\mathbf{S}_k) \; d\mathbf{S}. \tag{3.33}$$

The sources are believed to be generated from filtered white Gaussian signals, thus the source probability distribution is,

$$p(\mathbf{S}_k) = (\det 2\pi \mathbf{\Sigma}_k)^{-\frac{1}{2}} \; e^{-\frac{1}{2}\mathrm{Tr}\; \mathbf{S}^\top \mathbf{\Sigma}_k^{-1} \; \mathbf{S}}, \tag{3.34}$$

where the k source covariance matrix is estimated to be

$$\mathbf{\Sigma}_k = \mathrm{Toeplitz}(\gamma_{s_k,0}, ..., \gamma_{s_k,N-1}), \tag{3.35}$$

having the autocorrelation function $\gamma_{s,t} = \sum_{n=1}^{N-t} s_n s_{n+t}$. Solving the integral we can write the likelihood as,

$$p(\mathbf{X}|\mathbf{A}) = (\det \mathbf{A}^{-1})^N \prod_{k=1}^{N_k} (\det 2\pi \mathbf{\Sigma}_k)^{-\frac{N}{2}} \; e^{-\frac{1}{2}\mathrm{Tr}\; \mathbf{S}^\top \mathbf{\Sigma}_k^{-1} \; \mathbf{S}}. \tag{3.36}$$

3.5 PCA preprocessing

In order to comply with a number of issues that typically arise in multimedia applications, we use *principal component analysis* (PCA) as means of preprocessing, when dealing with zero mean signals. A general description of PCA can be found in [10].

Dealing with samples fewer than observations, $N < N_m$, we face a so-called *ill-posed learning problem*. This can be ”cured” without loss of generality by the PCA projection onto a N dimensional subspace[59]. Thus we use the N dimensional PCA as input to the ICA algorithm.

We often face under-complete mixing, hence where the number of sources are less than the number of observations, $N_k < N_m$. Having a square ICA mixing matrix as in the case of the no noise ML or MS, we can use PCA for reducing the dimension prior to the ICA. Choosing the N_k principal components (PC) with the highest variance preserves the most information[10]. In the MS algorithm this can be implemented directly into the solution as shown in [35]. By experience, when comparing results from ICA algorithms using PCA as dimension reduction, to algorithms that can handle the under-complete mixing by themselves, we found no significant difference in results. Likewise, PCA also tend to handle badly conditioned separations better, hence being more stable.

Finally PCA can be used as pre-whitening to the ICA. In PCA we recover the second order moments and so naturally aid the ICA decomposition in getting faster convergence when using the PCA solution as input to the ICA algorithm.

3.6 Model selection

To determine the optimal model in regards to e.g. ICA algorithm and hyper parameters, we use a model selection method based on *Bayes information criterion* (BIC). Bayes optimal decision rule under the $1/0$ loss function leads to the optimal model [89],

$$\mathcal{M}_{opt} = \arg\max_{\mathcal{M}} p(\mathcal{M}|\mathbf{X}). \tag{3.37}$$

Using Bayes rule, we find the probability of a specific model given the observed data to be,

$$p(\mathcal{M}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathcal{M})p(\mathcal{M})}{p(\mathbf{X})}, \tag{3.38}$$

The denominator functions as the normalizer $p(\mathbf{X}) = \sum_{N_{\mathcal{M}}} p(\mathbf{X}|\mathcal{M})p(\mathcal{M})$. The probability p($\mathcal{M}$) is the prior of the models and often assumed to be uniform distributed. We are therefore left to find the likelihood for the observed signals given the specific models $p(\mathbf{X}|\mathcal{M})$.

In a true Bayes framework we need to integrate over all parameters in our model to obtain the best generalizing solution[10]. For a particular choice of model, e.g. a particular number of sources, we have

$$p(\mathbf{X}|\mathcal{M}) = \int p(\mathbf{X},\theta|\mathcal{M})\ d\theta = \int p(\mathbf{X}|\theta,\mathcal{M})p(\theta|\mathcal{M})\ d\theta, \tag{3.39}$$

where θ hold the model parameters. Inferring $e^{\log}$, we have

$$p(\mathbf{X}|\mathcal{M}) = \int e^{\log p(\mathbf{X}|\theta,\mathcal{M})+\log p(\theta|\mathcal{M})}\, d\theta = \int e^{-f(\theta)}\, d\theta. \tag{3.40}$$

We now consider a second order Taylor expansion of $f(\theta)$ around θ^*, that holds the parameters where the likelihood function has its maximum. Thus the gradient is zero, and we write,

$$f(\theta) \approx f(\theta^*) + \frac{1}{2}(\theta-\theta^*)^\top \mathbf{H}(\theta-\theta^*), \tag{3.41}$$

where $\mathbf{H} = \frac{\partial f^2}{\partial\theta\partial\theta^\top}$ is the Hessian matrix. For large number of samples N, θ is close to θ^* and so the likelihood is close to maximum. In this limited Taylor expansion, parameters θ that deviate away from the optimum θ^* makes the likelihood drop rapidly, and will therefore not have much influence in the integral of eq. (3.40). Thus we approximate the likelihood as,

$$p(\mathbf{X}|\mathcal{M}) \approx e^{-f(\theta^*)} \int e^{-\frac{1}{2}(\theta-\theta^*)^\top \mathbf{H}(\theta-\theta^*)}\, d\theta. \tag{3.42}$$

The exponential in front of the integral is the likelihood with the optimal parameters, and the integral can be seen to have a Gaussian form. From [10] we can now write the likelihood as,

$$p(\mathbf{X}|\mathcal{M}) \approx p(\mathbf{X}|\theta^*,\mathcal{M})p(\theta^*|\mathcal{M})\ (2\pi)^{\frac{D}{2}}\ |\mathbf{H}|^{-\frac{1}{2}}, \tag{3.43}$$

where D is the number of free parameters. In the BIC estimate we only consider the terms that contribute to the largest errors regarding number of samples N. Neither the prior $p(\theta^*|\mathcal{M})$, nor the term $(2\pi)^{\frac{D}{2}}$ are functions of N, and are therefore neglected. The $D \times D$ Hessian holds a product over samples, that we can factor out as $|\mathbf{H}| = N^D\ |\widetilde{\mathbf{H}}|$. We hereby also neglect the remaining determinant $|\widetilde{\mathbf{H}}|$, thus not having to estimate the often computational tedious Hessian matrix. The likelihood can finally be written as,

$$p(\mathbf{X}|\mathcal{M}) \approx p(\mathbf{X}|\theta^*,\mathcal{M})\ N^{-\frac{D}{2}}. \tag{3.44}$$

As is general with model selection, the number of samples needs to be large. The Taylor expansion will otherwise favor a wrong optimum, and the neglected terms from eq. (3.42) has high influence on the likelihood. For a more detailed discussion on BIC see e.g. [87].

We hereby identify the likelihood for a particular ICA model $\mathcal{M}$ with Gaussian noise, as

$$p(\mathbf{X}|\theta,\mathcal{M}) = p(\mathbf{X}|\mathbf{A},\mathbf{\Sigma}). \tag{3.45}$$

Using PCA preprocessing

In the case where we use PCA as preprocessing, the likelihood in eq. (3.45) will factorize into to parts of what we respectively regard as signal and noise. We use the PCA model introduced in [36, 76], where the signal space spanned by the first N_k principal components has full covariance structure. The space $\mathcal{U}$ spanned by the remaining $N_m - N_k$ principal components are assumed to be Gaussian noise and isotropic. The covariance matrix is thus diagonal, with elements of $\sigma^2_{\mathcal{U}} = (N_m - N_k)^{-1} \sum_{i=N_k+1}^{N} \Delta^2_{ii}$, where Δ is a matrix holding the eigenvalues corresponding to the principal components. Given that the noise and signal space are independent we can expand the likelihood from eq. (3.45) to,

$$p(\mathbf{X}|\theta, \mathcal{M}) = p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}) p(\mathcal{U}|\sigma^2_U) \tag{3.46}$$

where $\mathbf{X} = \mathbf{A}_{\mathrm{PCA}}\mathbf{Z}$, thus $\mathbf{Z}$ holds the N_k principal components and is the input for the ICA algorithm. The distribution $p(\mathcal{U}|\sigma^2_{\mathcal{U}})$ is set to be Gaussian, thus

$$p(\mathcal{U}|\sigma^2_U) = (2\pi\sigma^2_{\mathcal{U}})^{-N(N_m - N_k)/2}\, e^{-N(N_m-N_k)/2} \tag{3.47}$$

CHAPTER 4

Multimedia separation

Virtual environments consists of many different media modalities. In the following chapter we look at how ICA separation can be applied on sound, text and images.

In the first part of this chapter we investigate the independent source separation on raw sound and images. Regarding sound, we compare the performance of the PCA and ICA algorithms to get an idea of their separation properties. In image separation we look at the case of independence between pixels or images. The constraint of positive components and mixing is thereafter introduced, which seem to correspond well with the human paradigm.

ICA classification is introduced in the second part of the chapter, when extending the *latent semantic indexing*[25] with text onto ICA, from features instead of raw data. The ICA classification is further used on images using the same framework, and finally in combination of both texts and images from HTML Internet pages. In the ICA classification we investigate among other the grouping structure of the independency criteria regarding number of classes.

The chapter is partly a summary of the two published articles [57] and [37], both shortly described in appendix B.

ICA algorithms

In the following we employ different ICA algorithms, as to investigate their different properties. We will refer to them in short as follows,

ML ICA The maximum likelihood ICA algorithm with no noise and square mixing matrix or equivalent the Infomax algorithm, described in section 3.3.1. The natural gradient and line search is used to aid the optimization.

MS ICA The Molgedey and Schuster ICA algorithm is non iterative, but requires square mixing matrix and time correlated samples, thus is a dynamic ICA algorithm, described in section 3.4. In optimization the symmetric quotient matrix and automatic detection of τ is used.

MF ICA The mean field ICA algorithm with non square mixing matrix and noise model. The MAP solution to the source posterior is used here in estimating the sufficient statistics, described in section3.3.2 and [34]. We thank Lars K. Hansen for the use of his code.

positive MF ICA The positive mean field ICA imposes positive constraints on the mixing coefficients and sources using the exponential distribution, described in section 3.3.2 and [97]. The sufficient statistics are found using adaptive TAP. We thank Pedro H. Sorensen and Ole Winther for the use of their code.

Matlab code for the ICA algorithms can be found at http://isp.imm.dtu.dk/toolbox/ on the Internet.

4.1 Source separation

Separating the raw data finds the underlying and generating components of the data. Regarding sound, this could be the sound sources as of the peoples voices in the *cocktail party* problem[7]. Dealing with images, this could be images of face features in *face recognition problems*[5], or brain activation images in *functional magnetic resonance imaging* studies[84].

4.1.1 Sound

Sound in a virtual environments are found in two forms as either music or sound effects. However obvious the need for separations tasks are in the physical world, we do not have many likewise parallels in the VE at the present time, because focus has mainly been on images and text as in e.g. the Internet.

Separation of raw sound is although also interesting because of historical reasons in the development of ICA methods. In the following we separate some artificial mixed natural signals to recover the source signals using the linear ICA model. In general these assumptions would not hold in the physical world due to echo, noise, delay, and different kind of nonlinear effects. In such cases more elaborate source separation is needed, as described e.g. in [3, 4, 24]. We also demonstrate differences between PCA, Molgedey and Schuster ICA, and maximum likelihood ICA algorithms in their separation.

The present example deals with speech from 3 persons which are assumed statistically independent, and scaled to have variance one. The sampling frequency of the signals is $11025\,\mathrm{Hz}$ and each consist of 50000 samples. A linear instantaneous mixing with fixed known 3×3 mixing matrix is deployed and enables a quantitative evaluation of the ICA separation. The source and mixing signals are shown in figure 4.1. In order to evaluate the results of the separation, we consider a *system matrix* defined as

$$\mathbf{C} = \mathbf{A}\mathbf{W}, \tag{4.1}$$

where $\mathbf{A}$ is the true mixing matrix that we used to generate the mixed signals, and $\mathbf{W}$ is the estimated unmixing matrix from the separation. If the separation is successfully, the system matrix equals the identity matrix, w.r.t. scaling of the estimated sources to have variance one and permutation of the unmixing matrix

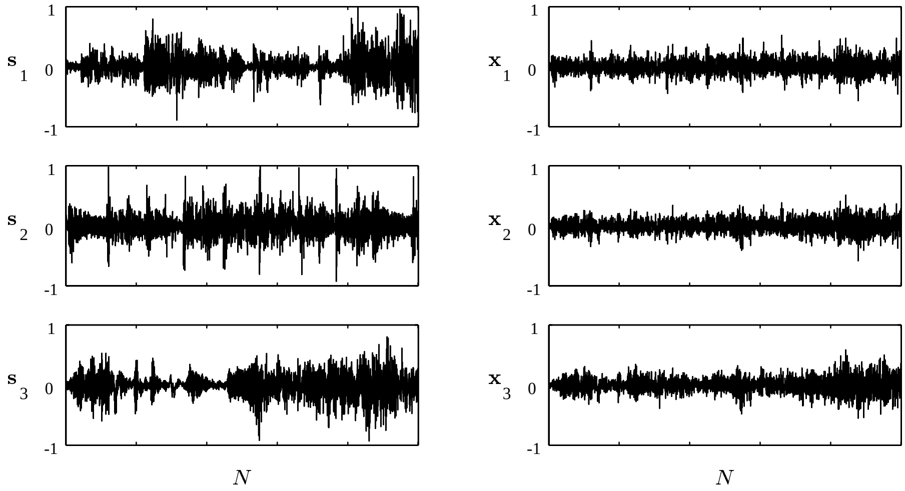


Figure 4.1 (Left) The orginal source sound signals $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]^\top$ consist of 50000 samples (N) and are assumed to be statistically independent. (Right) The mixture signals $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]^\top$ are linear instantaneous combinations of the source signals.

rows, to compensate for the unknowns described in eq. (3.6). We also plot the residual error signal $\mathbf{e}_i = \mathbf{s}_i - \widehat{\mathbf{s}}_i$ for signals $i = 1..3$, where $\mathbf{s}$ holds the original sources and $\widehat{\mathbf{s}}$ holds the estimated sources.

4.1.1.1 PCA

Principal component analysis is often used because it is simple and relatively fast. Moreover it offers the possibility of reducing the number of sources by ranking sources according to variance. See section 3.5.

The result of the PCA separation is shown in figure 4.2 and the corresponding system matrix in table 4.1. Obviously the result is poor when comparing estimated sources to the original sources in figure 4.1. This is also confirmed by inspecting the system matrix in table 4.1.

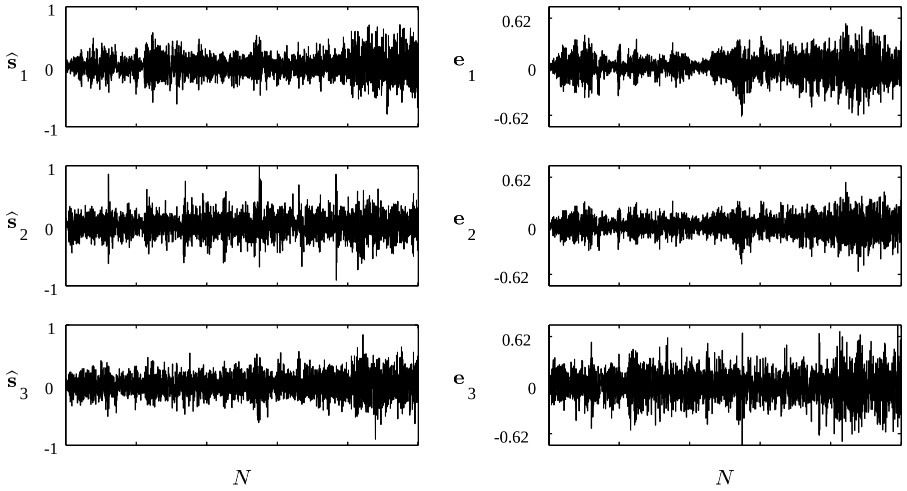


Figure 4.2 Separated sound source signals using PCA. The right panels shows error residual signals.

$$\mathbf{C}_{\mathrm{PCA}} = \begin{bmatrix} 0.56 & 0.98 & 0.62 \\ 0.28 & 0.72 & 0.23 \\ 0.18 & 0.50 & 0.06 \end{bmatrix}$$

Table 4.1 System matrix for the PCA separation of sound signals.

4.1.1.2 MS ICA

The main advantage of the MS ICA algorithm is that it is non-iterative, and consequently very fast. In figure 4.3 the estimated sound signals from the separation are shown. Comparison with original source signals in figure4.1 indicates very good separation. The system matrix in table 4.2 and an additional listening test also confirms this result.

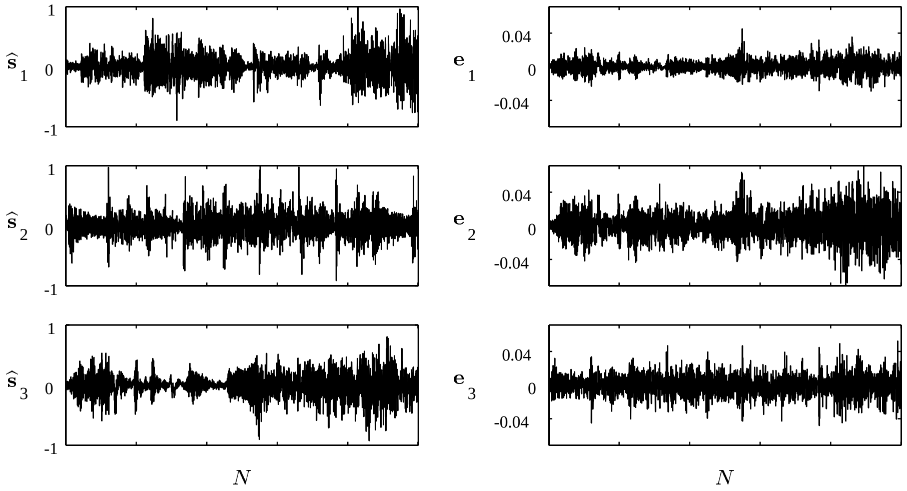


Figure 4.3 Separated sound source signals using Molgedey-Schuster ICA. The right panels shows error residual signals.

$$\mathbf{C}_{\mathrm{MS\ ICA}} = \begin{bmatrix} 1.00 & 0.02 & 0.03 \\ 0.02 & 1.00 & -0.01 \\ -0.03 & -0.03 & -1.00 \end{bmatrix}$$

Table 4.2 System matrix for the Molgedey-Schuster ICA separation of sound signals.

4.1.1.3 ML ICA

The ML ICA or Infomax algorithm is very commonly used. Because of its iterative nature it is much more time consuming than the Molgedey-Schuster algorithm or PCA. In figure 4.3 and table 4.3 the results of the separation are shown. Clearly, the system matrix is closer to the identity matrix than that of Molgedey-Schuster at the expense of increased computational burden.

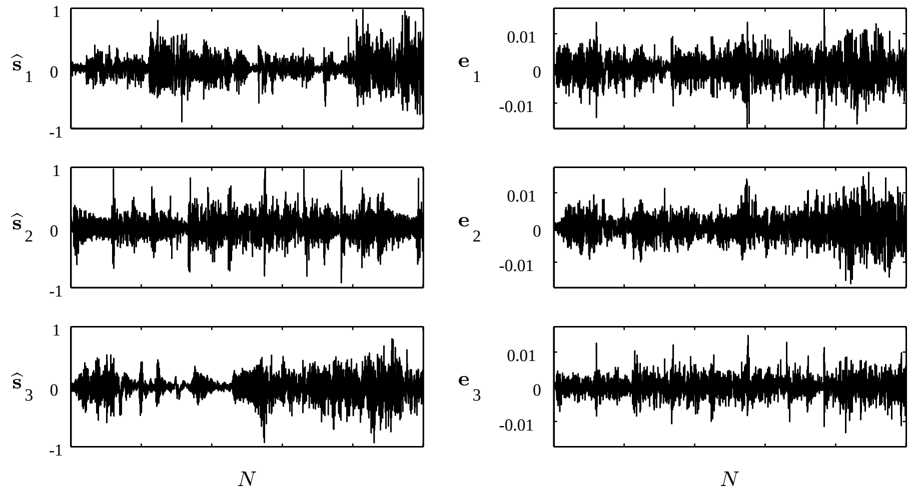


Figure 4.4 Separated sound source signals using ML ICA. The right panels shows error residual signals.

$$\mathbf{C}_{\text{ML ICA}} = \begin{bmatrix} 1.00 & -0.01 & 0.01 \\ 0.00 & 1.00 & -0.01 \\ 0.01 & 0.01 & 1.00 \end{bmatrix}$$

Table 4.3 System matrix for the ML ICA separation of sound signals.

4.1.1.4 Summary

In table 4.4 we list the norm of the system matrix deviation from the identity matrix as well as computation time.

PCA was out-performed by both ICA algorithms due to very restricted separation capabilities. Both ICA algorithms performed very well. The major difference is computation time, thus MS ICA was more than 200 times faster than the ML ICA. The advantage of the ML ICA algorithm is that the system matrix can be significantly closer to unity provided sufficient computation time. By listening to the separated signals it was nearly impossible to tell the difference

	$\lvert\mathbf{C} - \mathbf{I}\rvert$	**Comp. time (sec.)**
PCA	1.21	0.25
MS ICA	0.05	0.25
ML ICA 22 iterations	0.05	56.10
ML ICA 56 iterations	0.01	152.18

Table 4.4 Norm of the system matrix deviation from the identity matrix and computation time in seconds. MS ICA is the Molgedey-Schuster ICA, ML ICA is the no noise maximum likelihood ICA for 22 and 56 iterations.

between the ICA results.

4.1.2 Image

Images in virtual environments are found in two forms as either natural images or graphics, where the latter can be regarded as a close to noiseless special case of natural images. From applications with natural images we know that ICA finds interesting structures, hence source images, see e.g. [5, 84, 34, 40, 42, 66]. In the following we will look at how ICA can be applied to image signals using either the *pixel-independence* or the *image-independence* assumption, and later by also imposing a positive constraint on the mixing and sources.

To illustrate basic properties we first constructed a simple series of faces that we regard as the mixed signals, described in figure 4.5.

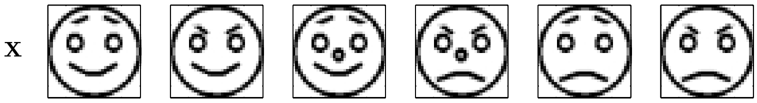


Figure 4.5 The mixed or observed image signals $\mathbf{X}$ consist of 6 images with faces. Each 91×100 size image has been arranged into an vector of $N = 9100$ pixels, thus $\mathbf{X}$ is a 6×9100 matrix.

From the images in figure 4.5 we would like to extract the distinct features of the faces as being the source images. In our simple example no noise is present and so pixels align perfectly if the images were stacked on top of each other, when features are identical. This is done intentionally to illustrate the separated properties more clearly.

4.1.2.1 PCA

First we look at the PCA solution, given it is a commonly used method in image analysis. Figure 4.6 shows the result with the separated PC images in the top row and its corresponding mixing proportions on the bottom row being the PCA basis. From this we read that e.g. the first image in the $\mathbf{X}$ matrix is generated from the first PC image component, subtracted by the second and finally added some of the last PC component. Image features are not easily recognizable in the PCA solution, except maybe the last image, as being the *nose*.

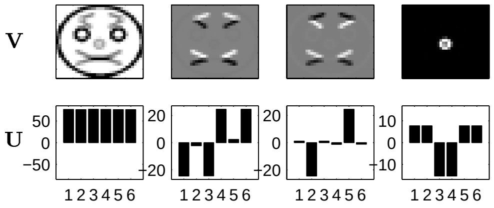


Figure 4.6 Projecting the faces along the PCA axis using the SVD method, four directions were enough to represent the six mixed image signals. The upper row U shows the PC image components, where as the lower row V shows the linear projection between the observed faces and the PC components, thus the mixing proportions. Only 4 components are found since mouth and eyebrows are anti-correlated when being up or down. Hence, they can be represented in fewer components when placing some as positive and others as negative.

4.1.2.2 ICA

Using the MS ICA algorithm we recognize the underlying mixing as being under-complete from the PCA. This is given from the fact that the *mouth* and *eyebrows* are either up or down, thus can be represented in the same component since we use a symmetric model that holds both positive and negative mixing. PCA is used as preprocessing for dimension reduction by using the PCA solution from figure 4.6. The ICA separation can hereafter be achieved by independence between pixels when using the mixing matrix directly as $\mathbf{X}$, or by independence between images using $\mathbf{X}^\top$ as input for the ICA algorithm.

In figure 4.7 we used the *pixel-independence assumption*, i.e., $\mathbf{X}$ is the signal matrix. The estimated IC image components are shown in the top row and associated mixing matrix in the bottom row. Unlike PCA in figure 4.6, MS ICA does not mix eyebrows and mouths together, i.e. the separation is more meaningful in regard to face features.

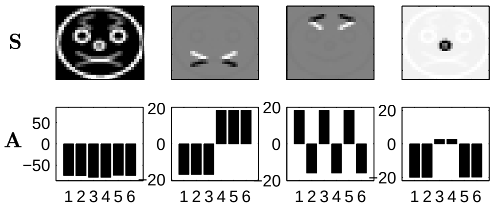


Figure 4.7 Separating with the Molgedey Schuster ICA algorithm and imposing independence between pixels. The upper row shows the IC image components and the lower row their linear projection between observations and IC components in the form of the mixing matrix.

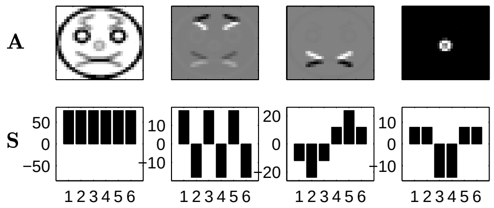


Figure 4.8 Separating with the Molgedey Schuster ICA algorithm and imposing independence between images. The upper row shows the IC image components and the lower row their linear projection between observations and IC components.

Transposing the mixed signal matrix we impose the *image-independence assumption* as shown in figure 4.8. The separated image features are not nearly as meaningful in this case compared to the pixel independent ICA solution. Although this is not always the case as discussed in [84], and should be determined from case to case depending on the true source properties.

4.1.2.3 Positive ICA

In the previous solutions we have accepted negative components and mixing, thus the *mouth* and *eyebrows* could be present in the same component when both are up or down. If we think of the underlying problem as the face consisting of features that are added to the image, it might be more natural in a human understandable sense. The mixing components can therefore not be negative, thus the sources are likewise not negative. Imposing positive constraint on both the mixing matrix and sources we used the positive MF ICA. The ICA separation was done with the image independence assumption and the result shown in figure 4.9.

All the face features was separated nicely into 6 components, with the exception of an underlying *face* repeating in each component the solution is very clear. Components that are *anti-correlated* as the mouth or eyebrow component are hereby avoided.

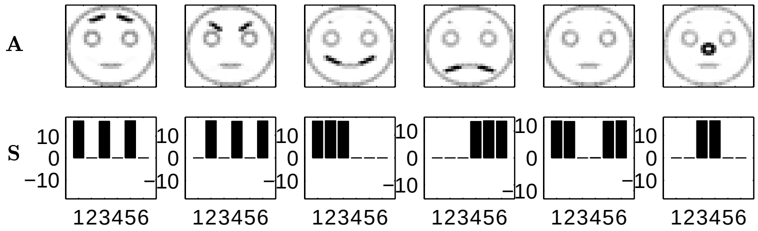


Figure 4.9 Separating with the probabilistic mean field ICA algorithm and imposing independence between images and positive mixing matrix and sources. The upper row shows the IC image components and the lower row their linear projection between observations and IC components.

4.1.2.4 Face data

Stepping back from this nicely constructed example, we now turn to real data with images of faces as found in face recognition problems, see [5] for a detailed discussion on the subject. We use the Yale Face Database[1] that consists of 15 subjects posing in 11 different ways as described in figure 4.10. Again we look for source images that describe interesting facial parts as in the case of

[1] We thank Sebastian Seung at MIT for giving access to the data.

the artificial face data, and compare results using different algorithms with 10 source components.

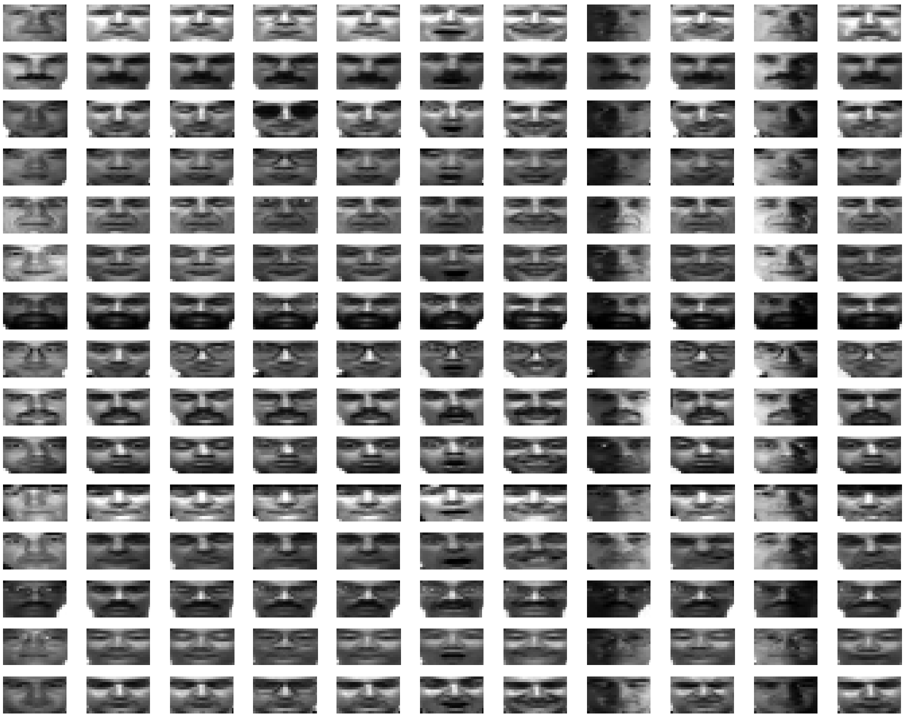


Figure 4.10 The face dataset consist of 15 subjects posing in 11 different ways, thus giving categories of: *center light, no glasses, sleepy, glasses, normal, surprised, happy, right light, wink, left light* and *sad*. Each image is 50×60 pixels, thus giving a matrix $\mathbf{X}$ size 165×3000. Eyes and mouth was center aligned by manual translocation to give the best possible overlap between faces.

In figure 4.11 and 4.12 separation is done respectively by PCA and ML ICA. In both cases the source distributions are assumed symmetric and with negative mixing allowed. Facial image parts are recognized in both cases, but it is not all clear what each component represent in a unique way. Anti-correlated components are also recognized, e.g. in figure 4.12 of both image 3 and 4, where the one side of a face (see figure 4.10 where one image category has intense light from one side) is found together with the opposite part from under the eyes, and vise versa.

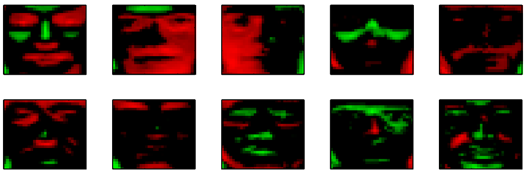


Figure 4.11 PCA separation. Pixels with a threshold at ±20% of the mean intensity was removed, thus to enhance the separation result more clearly. Green intensity represent positive and red represent negative values.

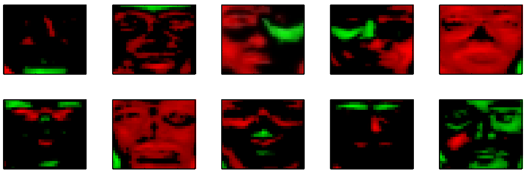


Figure 4.12 ML ICA was done imposing independence between pixels. Pixels with a threshold at ±20% of the mean intensity was removed, thus to enhance the separation result more clearly. Green intensity represent positive and red represent negative values.

In figure 4.13 the positive MF ICA separation was done. The result show clear evidence of finding more clear and interesting facial parts as opposed to the PCA or ML ICA.

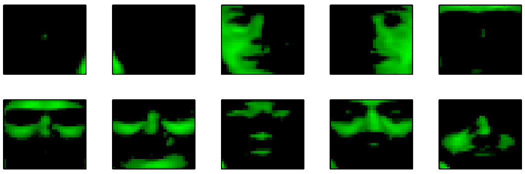


Figure 4.13 Positive ICA separation was done imposing independence between pixels. Pixels with a threshold at below 40% intensity was removed, thus to enhance the separation result more clearly.

Work done by [61] show that the positive or non-negative constraint is very strong, and decomposing without the independence criteria holds similar results. In figure 4.14 we separated using the *non-negative matrix factorization* (NMF) [60] that decomposes the model holding only the criteria of positive mixing and source matrix. Comparing the positive ICA and NMF result they are very close to being identical.

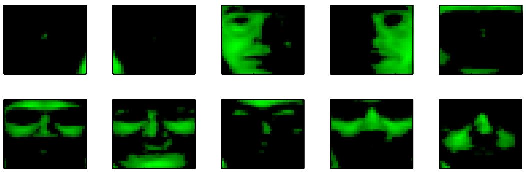


Figure 4.14 NMF separation. Pixels with a threshold at below 40% intensity was removed, thus to enhance the separation result more clearly. Images was likewise ordered manually.

4.1.2.5 Summary

The separation result of images from the artificial and real face data seem highly governed by constraining to positive separation. The positive ICA and NMF algorithm both produced easy reconcilable face components as opposed to both PCA and the ML ICA. The possibility of having simultaneous positive and negative components does not seem to correspond well with the underlying human paradigm, thus producing anti-correlated components. In general we must assume that this holds in many related image separation cases, and underlines the importance of taking this aspect into account when choosing the ICA model in regard to images.

4.2 ICA classification

Opposed to the direct source separation presented in the previous section, we now extract features from each media modality to form feature histograms for each data sample. This approach has been used widely in text analysis in connection with information retrieval through the *vector space model*[91]. State of the art data mining tools are based on statistical pattern recognition, working from the relatively basic features such as e.g. term frequency histograms. Since feature lists most often are high-dimensional and we typically have access to rather limited labeled databases, representation becomes an important issue. The problem of high dimensions has been approached with principal component analysis, that in text mining is called *latent semantic indexing*[25]. Lately this has also been done regarding image retrieval and in the combined image and text retrieval, using the LSI framework, see e.g. [83, 15, 100].

In this section we extend the LSI with ICA, thus to find a separation that align the sources grouping structure, and to exploit that for classification. For the purpose of text retrieval has ICA separation likewise been done in [50].

Performing unsupervised ICA classification we seek grouping structures that explain meaningful context in a human sense. We thus investigate the role of independency, and the independent context taxonomy that lives in different levels, depending on the number of components. When comparing the classification to human made labels we look for the *description level*, as the number of components that describe the classes best.

As for historical reasons, in respect to work published in [57] and the general research on LSI, we describe the ICA classification framework using text. Later we extract features from images for classification, and finally in combination of both texts and images from HTML Internet pages.

4.2.1 Text

In text separation the data is presented in the form of terms[2]. We collect the terms into frequency histograms as purposed in the vector space model (VSM)[91]. The VSM presents a natural good distance measure between text

[2] A term is one word or a small set of words that present a meaning.

samples, that we in general call documents. Using latent semantic indexing (LSI)[25] trough PCA we approach the problem of high dimensions. To this we apply the MF ICA algorithm, which is able to identify a low-dimensional basis set in the face of high-dimensional noisy data. The major benefit of using ICA is that the representation is better aligned with the content group structure than LSI.

We apply the ICA separation to two public domain datasets: a subset of the MED medical abstracts database and the CRAN set of aerodynamics abstracts.

4.2.1.1 Vector space representations

The vector space model is used for feature extraction and involves three steps: Indexing, term weighting and a similarity measure [91]. Features are based on single word statistics, hence, first we create a term set of all words occurring in the database. This term set is screened for words that do not help to differentiate documents in terms of relevance. This is called *indexing*. In this *stop list* we find very frequent words like and, is and the. We also eliminate infrequent words that occur only in a few documents. The use of term frequency within a document to discriminate content bearing words from the function words has been used for a long time [71]. Elimination of high frequency terms is necessary as they can be very dominant in the term frequency histogram as shown in figure 4.15.

When the term set for the document collection has been determined, each document can now be described with a vector. For document j the document vector is $\mathbf{d}_j = [w_{1j}\ w_{2j}\ \cdots\ w_{N_m j}]^\top$, where N_m is the number of terms in the term list, e.g. the union of content bearing words for all documents in the collection. We will form the term-document matrix for convenience, given by

$$\mathbf{X} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & & \vdots \\ w_{N_m 1} & w_{N_m 2} & \cdots & w_{N_m N} \end{bmatrix} \tag{4.2}$$

where N is the number of documents in the database.

Determining the normalization of the weights is called *term weighting*. There have been suggested a number of different term weighting strategies [92]. The

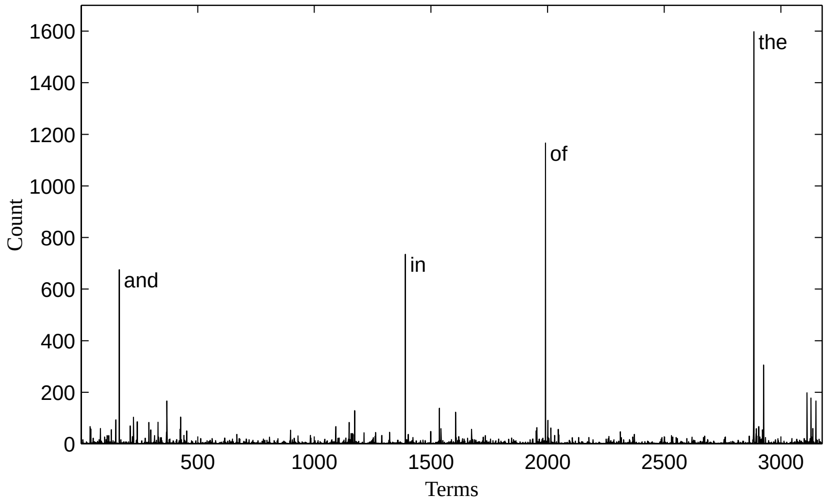


Figure 4.15 Text mining is based on simple term frequency histograms. We show a histogram prior to screening for high frequency words. Note that common words like and, is and the totally dominate the histogram typically without much bearing on the subsequent classification.

weights can be determined from single documents independent of the other documents, or by using database wide statistical measures. The simplest term weighting scheme is to use the raw term frequency value as weights for the terms. If we assume that document length is not important for the classification, this vector can be normalized to unit length,

$$w_{ij} = \frac{f_{ij}}{\sum_{i=1}^{N_m} f_{ij}}, \tag{4.3}$$

where f_{ij} is the frequency of term i in document j. This however is not always a good weighting scheme when e.g. dealing with Internet HTML pages. These documents are often of very different sizes, thus terms in short documents will get much higher weight than terms in long documents.

The document *similarity measure* is usually based on the inner product of the document weight vectors, but other metrics can be argued for.

Figure 4.16 shows a normalized term-document matrix with function words removed. The data used for visualization are the first 5 groups in the MED

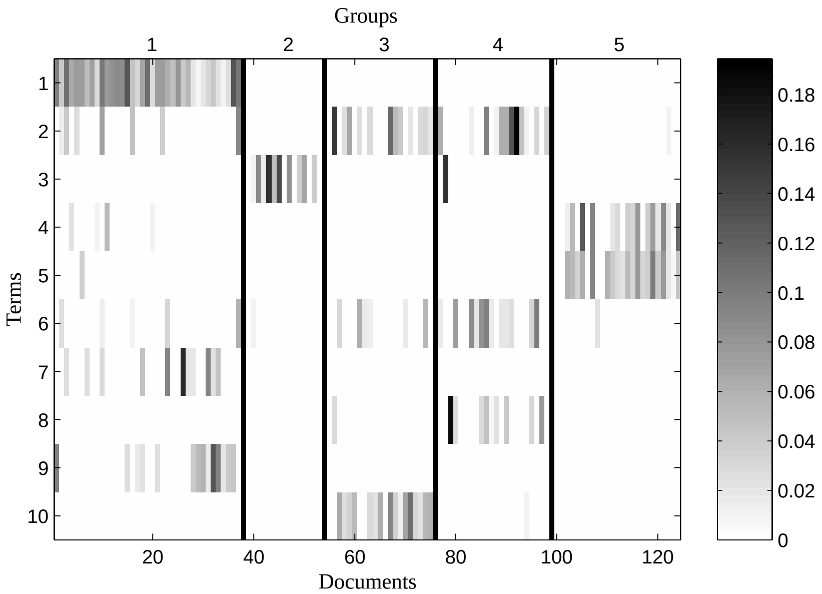


Figure 4.16 The figure shows 10 terms with the largest variance in the first 5 groups in the a document dataset. The columns are sorted by the group numbers from (1) to (5). Some of the terms are clearly "keywords".

data, which will be described later in this chapter. Only 10 terms with the highest occurrence variance are shown.

4.2.1.2 *Latent Semantic Indexing*

All document classification methods that use single word statistics have well known language related ambiguities: polysemy and synonomy [25]. *Polysemy* refers to the problem of words have more than one meaning. An example of this is the word *jaguar* which depending on context represents a sports car or a cat. *Synonomy* is used to describe the fact that different words have the similar meanings. An example of this are the words car and automobile.

Latent semantic indexing [25] is the PCA of the vector space model. The main objective is to uncover hidden linear relations between histograms, by rotating the vector space basis. If the major content differences form uncorrelated (orthogonal) combinations, LSI will find these. The technique used for this transformation is the well known singular value decomposition (SVD). With the use of SVD the term-document matrix $\mathbf{X}$ is decomposed into singular values and singular vectors, given by

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{L} \cdot \mathbf{D}^{\top}, \tag{4.4}$$

where $\mathbf{T}$ is $N_m \times r$, $\mathbf{L}$ is $r \times r$, $\mathbf{D}$ is $N \times r$, and r is the rank of $\mathbf{X}$. $\mathbf{L}$ is a diagonal matrix of singular values and $\mathbf{T}$ and $\mathbf{D}$ hold the singular vectors for the terms and documents respectively. The terms and documents have been transformed to the same space with dimension r. The columns of $\mathbf{T}$ and $\mathbf{D}$ are orthogonal, i.e., uncorrelated. If $\mathbf{X}$ is full rank the dimension of the new space is N.

If the database is indeed composed from a few independent contents each characterized by a class histogram, we would expect relatively few relevant singular values, the remaining singular values being small and their directions representing noise. By omitting these directions in the term vector space we can improve the signal to noise ratio and effectively reduce the dimensionality of the representation. If the singular values are ordered by decreasing value, the reduced model using the $N_p < r$ largest singular values, will have $\mathbf{T}$ as $N_m \times N_p$, $\mathbf{L}$ as $N_p \times N_p$, and $\mathbf{D}$ as $N \times N_p$.

The selection of the number of dimensions or N_p is not trivial. The value of N_p should be large enough to hold the latent semantic structure of the database, but at the same time we want it as small as possible to obtain the optimal signal to noise ratio.

In figure 4.17 the ten largest principal components of the $\mathbf{D} \cdot \mathbf{L}$ matrix is shown using the MED data. In the upper row of figure 4.18 we show scatter plots of projections on PC2 vs. PC1 and PC2 vs. PC3, and note that the documents (color coded according to class) fall in non-orthogonal rays emanating from origo. This strongly suggests the use of a non-orthogonal algorithms as is ICA. Decomposing the same data along the ICA basis is shown in the middle row of figure 4.18.

4.2.1.3 Learning ICA text representations on the LSI space

As we typically operate with $1000+$ words in the terms list and much fewer documents, we face a so-called *ill-posed learning problem*. Using the PCA as preprocessing, the problem can be “cured” without loss of generality[59], by choosing the N largest principal components as input to the ICA algorithm. We

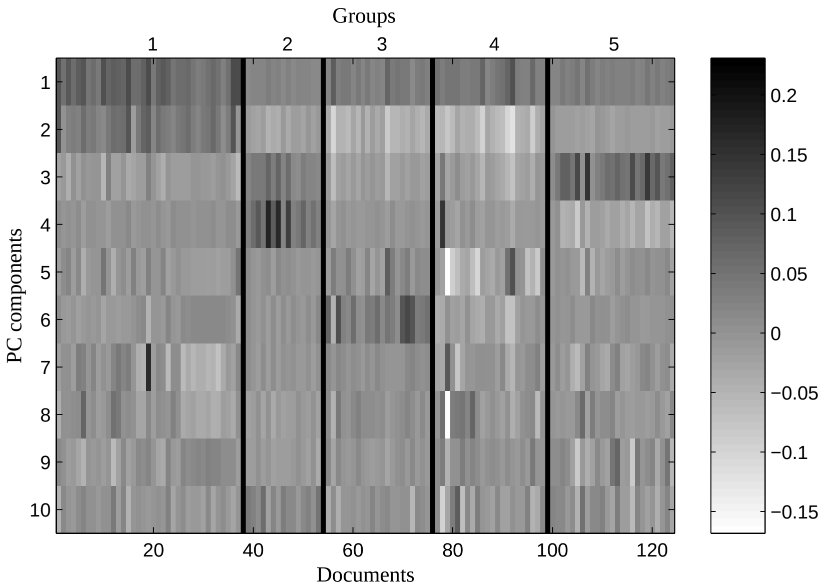


Figure 4.17 The figure shows the ten first principal components of the $\mathbf{D} \cdot \mathbf{L}$ matrix for the first 5 groups in the MED dataset. The columns are sorted by the groups (1) to (5). The first components are clearly assigned to specific groups in the dataset.

note that it often may be possible to further limit the dimensionality of the PCA subspace, hence further reducing the histogram dimensionality of the remaining problem, as described in section 3.5 for the under-complete case.

The LSI model is merely performing a PCA on top of the vector space model and thus learning the ICA text representation can be viewed as a post-processing step for the LSI model. Inserting the ICA decomposition into eq. (4.4) we decompose the term-document matrix into,

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{S} \tag{4.5}$$

where $\mathbf{T}$ holds the term eigenvectors of $N_m \times N$, $\mathbf{A}$ is the $N \times N_k$ IC document projections on the PC basis and $\mathbf{S}$ is $N_k \times N$ thus holds the N_k separated sources.

As shown in figure 4.18 the IC projections are not symmetrical around zero,

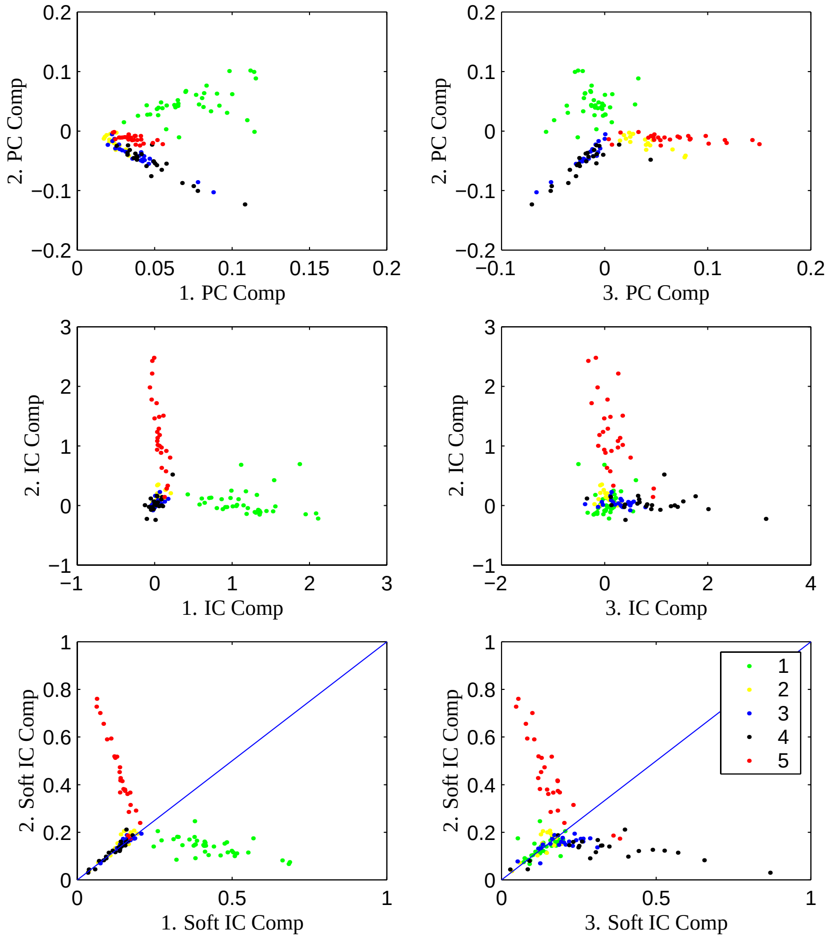


Figure 4.18 Analysis of the MED set of medical abstracts, labeled in five classes here coded in colors. The two upper panels show scatter plot of documents in the latent semantic or principal component basis. In the middle panels we show the document location as seen through the ICA representation. Note that while the group structure is clearly visible in the PCA plots, only in the ICA plots is the group structure aligned with independent components. In the lower panels the result from passing the IC components through softmax for classification. The diagonal is a simple classification decision boundary.

as our ICA model imposes. We overcome this problem by changing the corresponding sign in $\boldsymbol{\Delta}$ of eq. 3.6 for components with negative mean. In regards

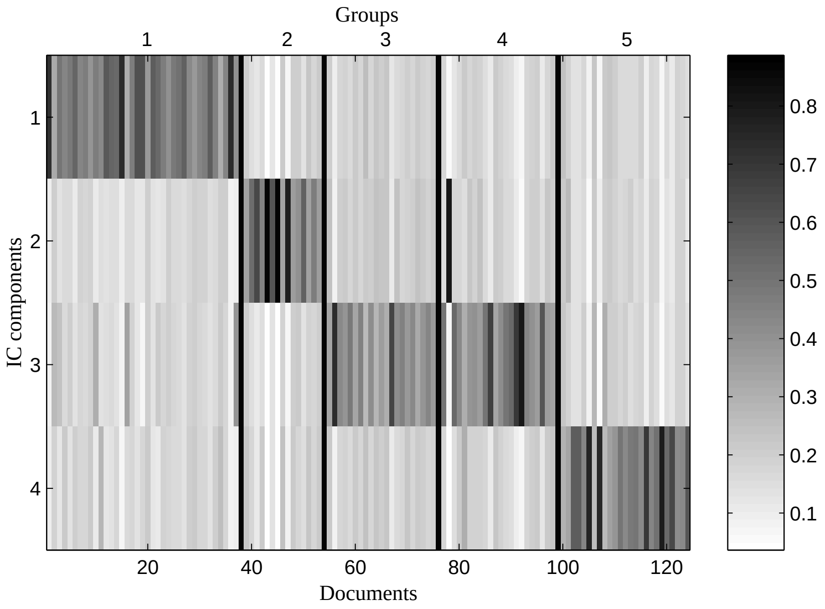


Figure 4.19 The figure shows the IC components with 4 channels using first 5 groups in the MED dataset. The columns are sorted by the group numbers from (1) to (5). The channel value is clearly related to the class number.

to scaling, each IC component is assume to have equal variance of one, thus to classify relative to magnitude.

An example of the estimated source matrix is shown in figure 4.19 using the MED data. The **S** matrix is normalized with softmax so the outputs can be interpreted as the probability of a document belonging to each class. In this case the unsupervised ICA is able to determine a group structure which very closely coincide with the "human" labels 1,2,5 but lumping groups 3 & 4 in one group. Interestingly running ICA with five or more component does not resolve groups 3 and 4 but rather finds a independent mixture within class groups.

4.2.1.4 *Classification based on independent components*

To quantify the ability of the ICA to group documents we convert the separated signal to "class probabilities" using the standard *softmax* [10] normalization on the recovered source signals,

$$\phi_{kn} = \frac{\exp(\mathbf{S}_{kn})}{\sum_{1=k}^{N_k} \exp(\mathbf{S}_{kn})} \quad , \tag{4.6}$$

The estimated ICA class label for a given document or sample n, is identical to the component number k with the highest probability ϕ_{kn}. This is the same as assigning a given document to the IC component with which it forms the smallest angle, i.e. distance.

Since the ICA classification is unsupervised we need to match classes when comparing results with manual labels.

4.2.1.5 Keywords from context vectors

Finding characteristic *keywords* to describe the context in a given independent component can be obtained by back projection of the documents to the original vector histogram space. This amounts to projection onto the identity matrix through the PCA and ICA bases. From eq. 4.5 we find that $\mathbf{T} \cdot \mathbf{A}$ is the basis change where columns represent the weight of the terms in each output, see figure 4.20. Depending on how many and their weight, we choose the keywords above a specified threshold after normalizing, as in table4.5.

4.2.1.6 Text examples

We will illustrate the use of ICA in text mining on two public domain datasets both available on the WWW [96]. The MED dataset has been known to produce good results in most search and classification models and therefore serves as a good starting point. The second dataset CRAN is a more complex set of documents with overlapping class labels and less obvious group structure.

In general when constructing the histogram term-document matrix, words that occurred in more than one document and was not present in a given list of stop words, was chosen as a term word. The length of each document histogram (the columns) was normalized to one to remove the effect of the document length.

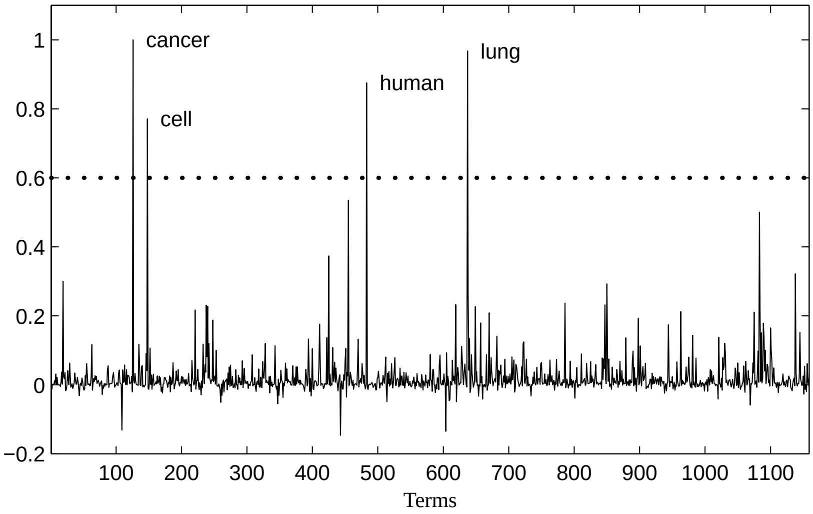


Figure 4.20 In analysis of the MED dataset keywords can be found by back projection for a given component. Keywords above a specified threshold are chosen as words that best describe the given components context.

When converting ICA recovered sources to classifications using eq.4.6 we also matched the unsupervised ICA classes to the manual labels.

MED *dataset results*

The MED dataset is a commonly studied collection of medical abstracts. In total it consists of 1033 abstracts, of which 30 labels has been applied to 696 of the documents. For simplicity we here consider 124 abstracts associated with the first five groups in the MED dataset. An 1159 terms were hereby used to build the term-document matrix.

Brief outline of the MED abstract groups:

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	36	1	0	0	2	lens crystallin
$\mathbf{IC_2}$	1	15	22	23	24	cell lung tissue alveolar normal cancer human

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	36	0	0	0	0	lens crystallin
$\mathbf{IC_2}$	0	16	0	1	24	fatty acid blood glucose oxygen free maternal plasma level tension newborn
$\mathbf{IC_3}$	1	0	22	22	2	cell lung tissue alveolar normal

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	36	0	0	0	0	lens crystallin
$\mathbf{IC_2}$	0	16	0	1	0	oxygen tension blood cerebral pressure arterial
$\mathbf{IC_3}$	1	0	22	21	2	cell lung tissue alveolar normal
$\mathbf{IC_4}$	0	0	0	1	24	fatty acid glucose blood free maternal plasma newborn fat level

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	35	0	0	0	0	lens crystallin
$\mathbf{IC_2}$	0	16	0	1	0	oxygen tension blood cerebral pressure arterial
$\mathbf{IC_3}$	2	0	15	10	0	cells alveolar normal
$\mathbf{IC_4}$	0	0	7	12	2	cancer lung human cell growth tissue found virus acid previous
$\mathbf{IC_5}$	0	0	0	0	24	fatty acid glucose blood free maternal plasma newborn level fat

Table 4.5 Confusion matrix and keywords from classification of MED with 2 to 5 output IC components. The confusion matrix compares the classification of the ICA algorithm to the labeled documents. Each IC component likewise produced a set of keywords, that are ordered by the size of the projection starting with the largest.

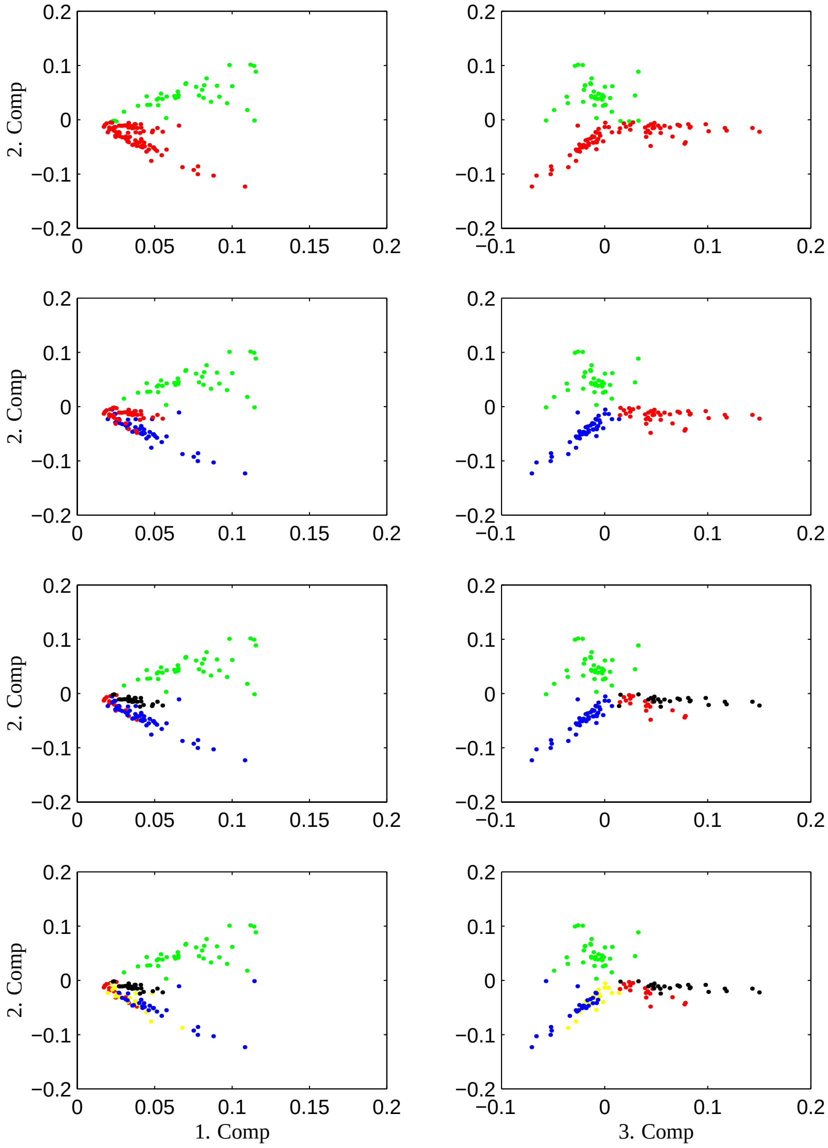


Figure 4.21 The MED dataset of medical abstracts. The dataset consists of 124 documents in five topics. The "source signals" recovered in the ICA have been converted by a simple softmax classifier, and we have coded these classes by different colors. From top to bottom we show scatterplots in the principal component representation PC2 vs. PC1 and PC2 vs. PC3, with colors indicating the classification proposed by the ICA with 2,3,4,5 independent components respectively.

$\mathbf{C}_1$	The crystalline lens in vertebrates, including humans.
$\mathbf{C}_2$	The relationship of blood and cerebrospinal fluid oxygen concentrations or partial pressures. A method of interest is polarography.
$\mathbf{C}_3$	Electron microscopy of lung or bronchi.
$\mathbf{C}_4$	Tissue culture of lung or bronchial neoplasms.
$\mathbf{C}_5$	The crossing of fatty acids through the placental barrier. Normal fatty acid levels in placenta and fetus.

In figure 4.18 we show scatterplots in the largest principal components and the most variant independent components. While the distribution of documents forms rather well-defined group structure in the PCA scatterplots, clearly the ICA scatterplots are much better axis aligned. We conclude that the non-orthogonal basis found by ICA better "explains" the group structure. To further illustrate this finding we have converted the ICA solution to a pattern recognition device by a simple heuristic. Normalizing the IC output values through softmax, showed evidence that comparing the magnitude of the recovered source signals produced a method for unsupervised classification.

In table 4.5 we show that this device is quite successful in recognizing the group structure although the ICA training procedure is completely unsupervised. For an ICA with three independent components two are recognized perfectly, and three classes are lumped together. The four component ICA recognizes three of the five classes almost perfectly and confuses the two classes 3 & 4. Inspecting the groups we found that the two classes indeed are on very similar topics (they both concern medical documents on diseases of the human lungs), and investigating classifications for five or more ICA component did not resolve the ambiguity between them. The ability of the ICA-classifier to identify the context structure is further illustrated in figure 4.21 where we show the PC scatterplots color coded according to ICA classifications.

Finally, we inspect the components produced by ICA by backprojection using the PCA basis. Thresholding the ICA histograms we find the salient terms for the given component. These terms are keywords for the given topic as shown in table 4.5.

CRAN *dataset results*

The CRAN dataset is a collection of aerodynamic abstracts. In total it consists of 1398 abstracts with 225 different labels and some were labeled as belonging to more than one group. Furthermore , inspecting the abstracts we found a greater content class overlap, hence we expect discrimination to be much harder. Because of the high number of classes some clusters were very small and we selected five content classes with a total number of 138 documents. In those groups the overlap were especially present in class 1 with 3 and class 2 with 5. A total of 1115 terms were used in the term-document matrix.

Brief description of the CRAN abstracts groups:

$\mathbf{C}_1$	What are the structural and aeroelastic problems associated with flight of high speed aircraft.
$\mathbf{C}_2$	How can the effect of the boundary-layer on wing pressure be calculated, and what is its magnitude.
$\mathbf{C}_3$	What similarity laws must be obeyed when constructing aeroelastic models of heated high speed aircraft.
$\mathbf{C}_4$	How can the aerodynamic performance of channel flow ground effect machines be calculated.
$\mathbf{C}_5$	Summarizing theoretical and experimental work on the behaviour of a typical aircraft structure in a noise environment is it possible to develop a design procedure.

To recognize possible group structure we show scatterplots for the first tree PC and IC components in figure 4.22. Classes that overlap are marked both with a dot and a circle having colors representing their classes. Comparing the CRAN PC scatterplots with the MED in figure 4.18 it is clear that the CRAN data is much more heterogeneous in contents. From figure 4.22 it is clear that ICA has identified some group structure while not as convincingly so as in the MED data. This is also borne out in figures 4.24 and 4.25 imaging the document projection relations.

The classification confusion matrix is found in table4.6. If we focus on the three source solution we find that ICA isolates class 1 and IC1, while the expected overlap between class 2 and 5 is seen in IC2. Class 4 is placed in a component overlapping with class 2 and 3 in IC3.

In table 4.7 we have illustrated the classification consistency among ICA's with different number of components. First we adapted a five component system and we recorded the ICA class labels for each document. We next adapted ICA's with two, three, and four components and created class labels. The confusion matrices show that although the ICA “unsupervised” labels are only in partial agreement with the manual labels they are indeed consistent and they show a taxonomy with hierarchical structure similar to the MED data.

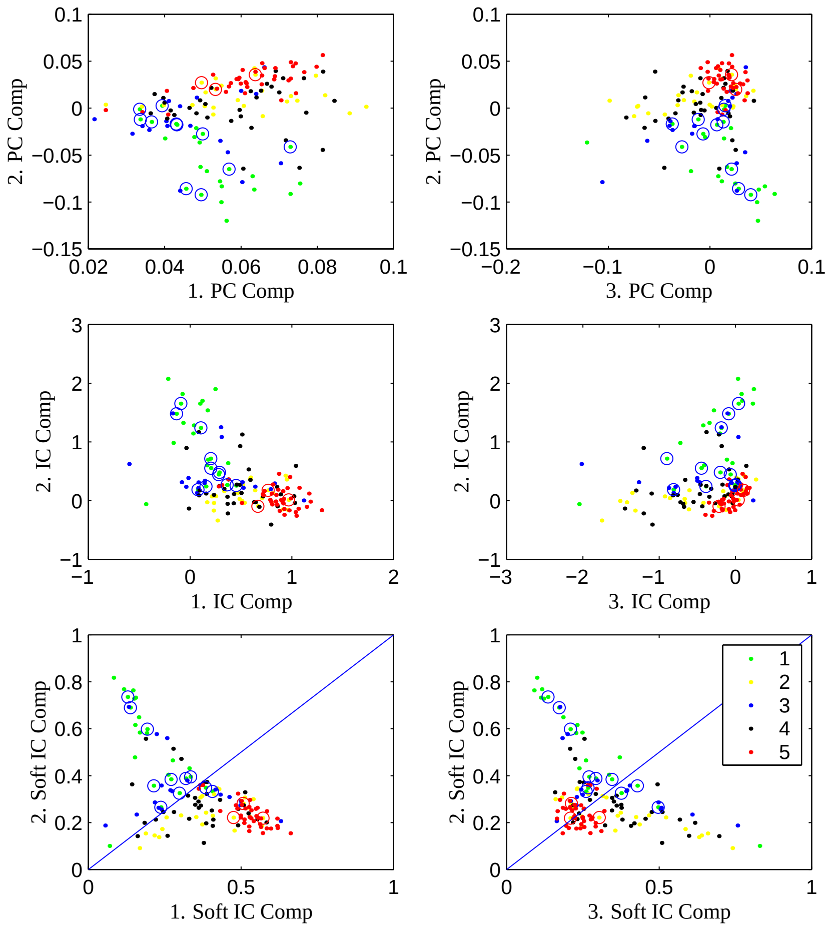


Figure 4.22 Analysis of the CRAN set labeled in five classes here coded in colors. The two upper panels show scatterplot of documents in the Latent Semantic or Principal Component basis. In the middle panels we show the document location as seen through the ICA representation. Note that while the group structure is clearly visible in the PCA plots, only in the ICA plots is the group structure aligned with independent components. In the lower panels the result from putting the IC components through softmax for classification. The diagonal line shows the decision boundary.

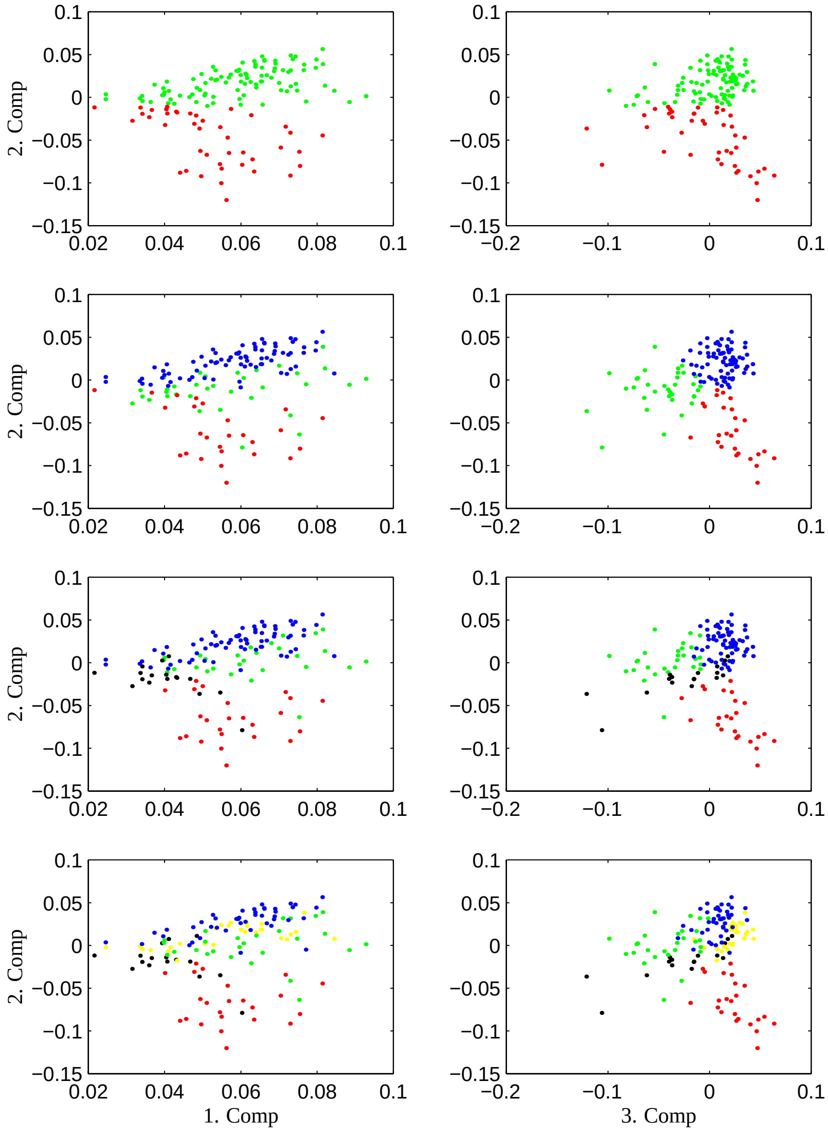


Figure 4.23 The CRAN dataset of aerodynamic abstracts. The dataset consists of 138 documents in five topics. The “source signals” recovered in the ICA has been converted to a simple classifier, and we have coded these classes by different colors. From top to bottom we show scatterplots in the principal component representation 1 vs. 2 and 3 vs. 2., with colors signifying the classification proposed by the ICA with 2,3,4,5 independent components respectively.

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	23	0	12	7	0	flutter panel
$\mathbf{IC_2}$	2	25	6	26	37	flow body pressure mach theory

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	19	0	4	3	0	flutter panel
$\mathbf{IC_2}$	2	16	6	16	37	flow pressure body mach number shock hypersonic
$\mathbf{IC_3}$	4	9	8	14	0	wing body

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	17	0	3	3	0	flutter panel
$\mathbf{IC_2}$	2	13	6	11	34	flow pressure mach number hypersonic shock heat layer body boundary transfer
$\mathbf{IC_3}$	5	0	9	0	1	wing thermal temperature stress aerodynamic supersonic
$\mathbf{IC_4}$	1	12	0	19	3	wing body theory lift flow

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	17	0	3	3	0	flutter panel
$\mathbf{IC_2}$	1	11	0	14	1	wing body lift theory
$\mathbf{IC_3}$	5	0	11	0	0	thermal wing temperature stress heat
$\mathbf{IC_4}$	0	7	1	12	24	flow body
$\mathbf{IC_5}$	2	7	3	4	12	mach pressure number heat

Table 4.6 Confusion matrix and keywords from classification of CRAN with 2 to 5 output IC components. The confusion matrix compares the classification of the ICA algorithm to the labeled documents. Each IC component likewise produced a set of keywords, that are ordered by the size of the projection starting with the largest.

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	23	5	13	0	1	flutter panel
$\mathbf{IC_2}$	0	22	3	44	27	flow body pressure mach theory number

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	23	0	2	0	1	flutter panel
$\mathbf{IC_2}$	0	21	11	3	0	wing body
$\mathbf{IC_3}$	0	6	3	41	27	flow pressure body mach number shock hypersonic

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	23	0	0	0	0	flutter panel
$\mathbf{IC_2}$	0	26	0	7	1	wing body theory lift flow
$\mathbf{IC_3}$	0	0	13	0	2	wing thermal temperature stresses
$\mathbf{IC_4}$	0	1	3	37	25	flow pressure mach number hypersonic shock heat layer body boundary transfer

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**keywords**
$\mathbf{IC_1}$	23	0	0	0	0	flutter panel
$\mathbf{IC_2}$	0	27	0	0	0	wing body lift theory
$\mathbf{IC_3}$	0	0	16	0	0	thermal wing temperature stress heat
$\mathbf{IC_4}$	0	0	0	44	0	flow body
$\mathbf{IC_5}$	0	0	0	0	28	mach pressure number heat

Table 4.7 Confusion matrix from classification of CRAN with 2 to 5 output IC components. The confusion matrix compares the classification of the ICA algorithm to the five ICA estimated classes.

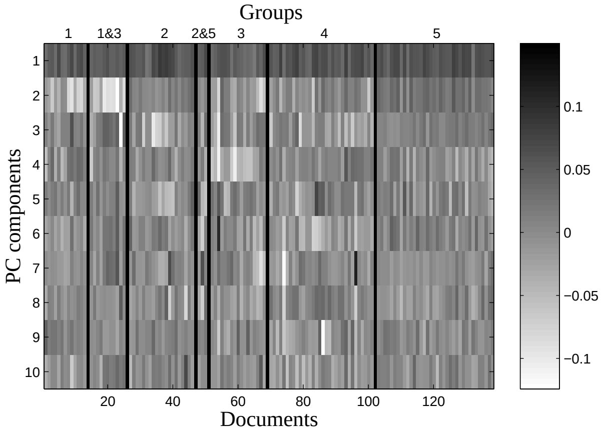


Figure 4.24 The figure shows the 10 first principal components of the $\mathbf{D} \cdot \mathbf{L}$ matrix for the groups (1) to (5) and (1&3) and (2&5) in the CRAN dataset. The columns are sorted by groups. Relations between principal components and groups can be observed, e.g., the second principal component seems to represent group (1) and (1&3).

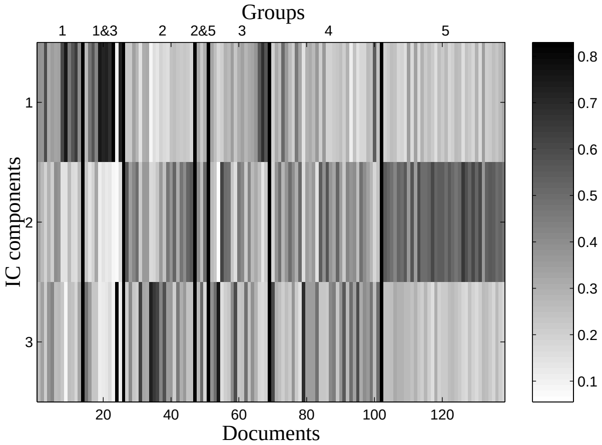


Figure 4.25 The figure shows the IC components after softmax using 4 channels in the CRAN dataset. The columns are sorted by group. Groups (1), (1&3) and (5) clearly visible in the channels, but the other groups overlap.

4.2.2 Image

The features we extract from images must be good distance measures to the VSM, and so we keep the framework presented in the previous section on ICA classification on text. We hereby seek to build a *feature-image* matrix as opposed to the *term-document* matrix. Work by [83, 15, 100] has investigated the latent semantic indexing model using images in information retrieval for the purpose of search, and found that it improves the result. As in the text separation we therefore use the ICA algorithm as an extension to LSI and hereby do unsupervised classification on the images.

The features we use are the lowest level image features purposed by the MPEG-7 standard, thus texture and color frequency histograms. MPEG-7 is an ISO/IEC standard developed by Moving Picture Experts Group. The standard is formally named Multimedia Content Description Interface, and quote ”aims to create a standard for describing the multimedia content data that will support some degree of interpretation of the information’s meaning, which can be passed onto, or accessed by, a device or a computer code” [74], that likewise reflect our goal. As such, the features used are color and texture, that we implement respectively with HSV encoding and Gabor filters. To enhance sensitivity for the overall shape as e.g. background, we divide each image into 4×4 image parts, as shown in figure 4.26. Thus the color and texture features are extracted from each partial image and collected into one overall feature vector for each image, hence a texture-image and color-image matrix. We found it crucial for the classification to add this shape information that represents a somewhat higher level description of the image context.

4.2.2.1 Texture features

By definition a texture is a spatially extended pattern build by repeating similar units called texels. Texture segmentation involves subdividing an image into differently textured regions. We use a texture segmentation scheme that is based on a filter-bank model, where the filters are derived from *Gabor elementary functions*. Any band-limited signal of finite duration can be represented by a finite superposition of Gabor elementary functions[73]. The goal is to transform texture differences into detectable filter output that describe the features. Each filter is therefore constructed to reflect a specific texture frequency and direction, that all together in the filter-bank describe the image textures.

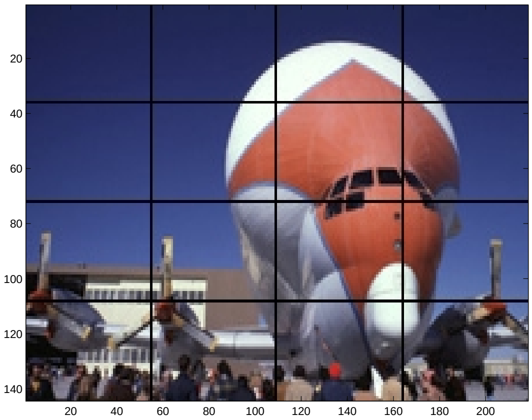


Figure 4.26 A typical image in the image collection that we are going to separate. Each image is divided in 4×4 sub-images, thus to capture the structure of the image.

A Gabor function can be described as a Gaussian function modulated by complex sinusoids[27]. In 2-dimensions we write a symmetric Gabor filter as,

$$h(n,m) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{(n^2+m^2)}{2\sigma^2}}\, e^{j2\pi(Un+Vm)}, \tag{4.7}$$

where the center frequency $f = \sqrt{U^2+V^2}$ is to capture the repetition of the texels in the direction of $\angle = tan^{-1}(V/U)$, and the width of the function is described by σ. The output of a filtered image i is then given as the convolution,

$$o(n,m) = i(n,m) * h(n,m). \tag{4.8}$$

The image texture feature for a given Gabor filter in the filter-bank is the energy of the filtered output using quadrature[54],

$$w_i = \frac{1}{N_{nm}} \sum_{n,m} [\mathrm{RE}\ o(n,m)]^2 + [\mathrm{IM}\ o(n,m)]^2, \tag{4.9}$$

where N_{nm} is the number of pixels in the partial image. The filter-bank we use in the following is largely defined in [100] and experimentally shown to be feasible. The Gabor filters are shown in figure 4.27 (left), and shows (right) the Gabor filtered output on the image from figure 4.26.

Gabor elementary functions are often used for modeling the simple cells function in the primary visual cortex (V1) of the receptive fields in the brain. Their function are that of edge detectors [86], and in relation to this has ICA been known likewise to produce edge detector filter images from natural images[6, 39]. This give some evidence that the Gabor filters are a reasonable choice.

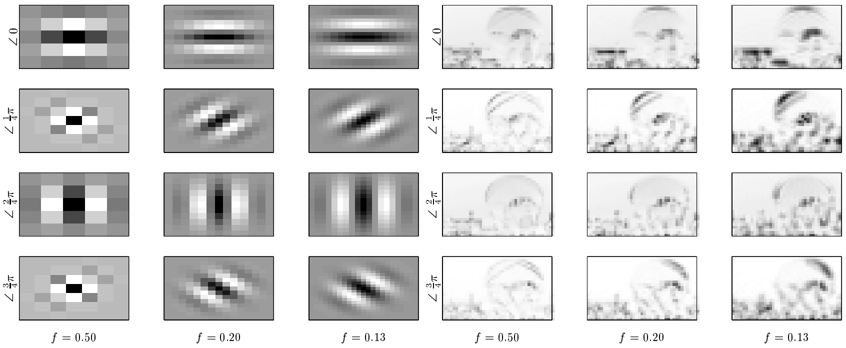


Figure 4.27 Gabor filters are used to extract the texture features. (Left) Filters representing four directions $\angle = [0, \frac{1}{4}\pi, \frac{2}{4}\pi, \frac{3}{4}\pi]$ and three frequencies $f = [0.50, 0.20, 0.13]$ are used in combination to a total of 12 filters. The width of the filters are determined by $\sigma = 2$. (Right) When running the Gabor filters on e.g. the image from figure 4.26, we enhance directions and resolution according to the given filter used.

One feature is found from each filter and normalized to sum to one. In total each image is represented by $4 \times 4 \times 12 = 192$ texture features.

4.2.2.2 Color features

In agreement with [100] we use the *hue, saturation and value* (HSV) color representation for the color features. The HSV color space corresponds better to how people experience color than e.g. the *red, green and blue* (RGB) color space. In figure 4.28[3] the HSV color space is sketched. The ”true” color as we think of e.g. red and yellow, is represented by the *hue* value on a color weel, and is often described as the dominant wavelength. How much the hue dominates is given by the *saturation* value. As the saturation goes from fully saturated to un-saturated the color disappears until just gray tones are present. The last color component is the *value* that describes the lightness–darkness of the color space.

[3] The image is borrowed from the Matlab Image Processing Toolbox documentation.

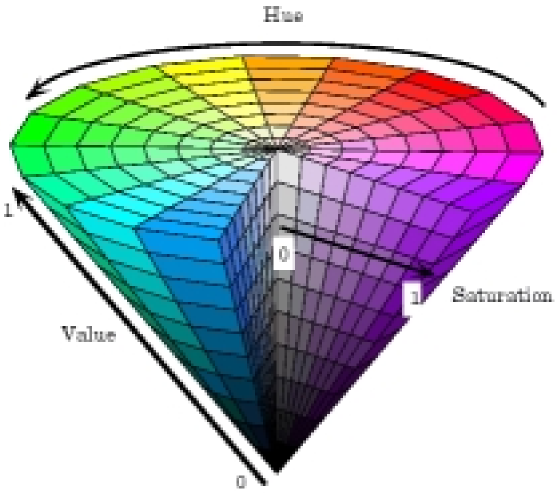
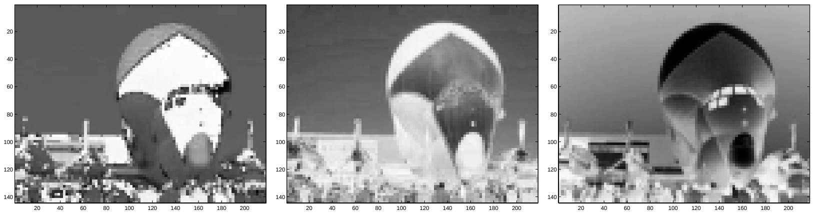


Figure 4.28 The HSV color space.

The color features we use are hereby given as the frequency histograms of the three color channels of 16 bins each. In figure 4.29 the image from figure 4.26 is decomposed into each color channel to demonstrate there significance.



Hue Saturation Value

Figure 4.29 Frequency histograms for each color channel (Hue, Saturation, Value) are used as color features. From the image in figure 4.26 the intensity map from each channel is shown respectively.

Each partial image produces hereby 3×16 features that are normalized to sum to one. In total each image is represented by $4 \times 4 \times 3 \times 16 = 768$ color features.

4.2.2.3 HTML database

The image collection we investigate consist of images retrieved from the Internet WWW. Three categories $\mathbf{C}_1 ... \mathbf{C}_3$ on www.yahoo.com were chosen as starting point for the retrieval:

$\mathbf{C}_1$	**Aviation**	Business and Economy → Transportation → Aviation → Pictures
$\mathbf{C}_2$	**Travel**	Recreation → Travel → Photos
$\mathbf{C}_3$	**Sports**	Recreation → Sports → Pictures

Simultaneously to downloading the images we retrieved neighboring text. The text that were in an above or below paragraph or in the same table row were accepted. Patches of WWW HTML were hereby collected. To ensure consistency, the images were to be of no less than 72×72 in height and width, stored in the jpg image format, and some text had also to be attached. In all a total of 292 images / texts were stored in the database, and features for color and text were extracted. LSI was performed on each of the feature modalities as described in the previous section about text separation. The ICA algorithm used for classification was ML ICA with LSI as preprocessing.

Our attention in the following will be to look at how well the different media modalities describe the data through the ICA classification, and for comparison between modalities. To quantify on the classification success for a given media, we measure how well each component describe the labeled categories $C_1, ..., C_3$. Except for the largest value in each row of the confusion matrix (in the two component classification it is the two largest values) we count the other values as miss-classification errors. The classification with the lowest error describe the categories best and thus the number of components equals the ICA *description level* of the categories in the given data and media.

We now seek to use the ICA classification method on each of the image features, and later in the next section on the joint features, including the text.

Image feature results

In table 4.8 the result from the ICA classification using texture and color features is presented by 2 to 4 components. The best ICA description level for color is 2 and for texture it is 3. The general evolving of the confusion matrices

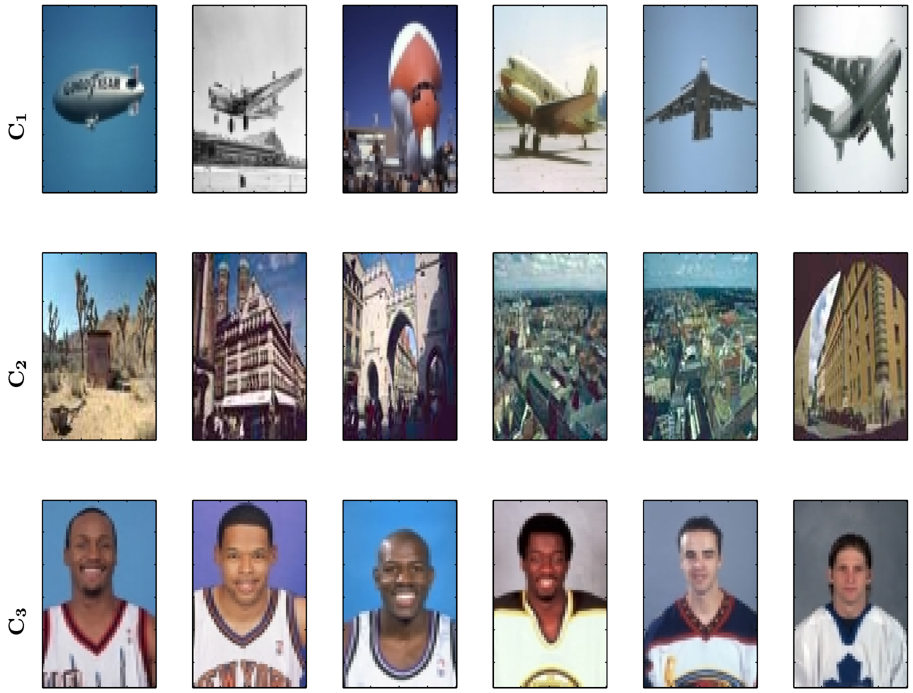


Figure 4.30 Images collected from the categories C_1:Aviation, C_2:Travel and C_3:Sports on www.yahoo.com.

when employing more and more IC components does only weakly suggests that there is some underlying hierarchical structure as we saw it in the previous text classifications. We do however find recognizable classes in general throughout the classification e.g. as for the 4 IC component classification with color, where the $\mathbf{C}_3$ mainly is divided in *basketball* and *icehockey* images.

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	97	61	7
$\mathbf{IC_2}$	3	31	93

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	92	7	4
$\mathbf{IC_2}$	6	83	2
$\mathbf{IC_3}$	2	2	94

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	79	5	5
$\mathbf{IC_2}$	6	76	2
$\mathbf{IC_3}$	1	1	72
$\mathbf{IC_4}$	14	10	23

Texture features

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	95	90	5
$\mathbf{IC_2}$	5	2	95

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	85	70	3
$\mathbf{IC_3}$	2	2	73
$\mathbf{IC_2}$	13	20	24

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	75	65	3
$\mathbf{IC_2}$	12	24	6
$\mathbf{IC_3}$	8	3	50
$\mathbf{IC_4}$	5	0	41

Color features

Table 4.8 Confusion matrix from classification with 2 to 4 output IC components using either texture or color features. The confusion matrix compares the classification of the ICA algorithm to the categories $\mathbf{C}_1$:Aviation, $\mathbf{C}_2$:Travel and $\mathbf{C}_3$:Sports. Texture class errors $[41, \mathbf{23}, 44]$, and color class errors $[\mathbf{12}, 114, 102]$.

4.2.3 Combined media

In ICA classification using features in the form of an term-document matrix or as feature-image matrix we used the same framework. We therefore seek to combine the features by simply stacking the feature matrices on top of one another. This has been investigated in [83, 15, 100] for information retrieval using LSI. The combination shows improvements and so we seek to investigate this further for the ICA classification.

The database we use is a collection of HTML patches, consisting of images and surrounding text, that were specified in the last section4.2.2.3 together with the model framework. A total of 1139 terms is used for the term-document matrix, and so the full feature matrix is of size 2099×292 where all separate feature modalities have vector length one for normalization purposes.

Text features results

At first we classify using only the term-document matrix for 2 to 5 components. The result is shown in table 4.9 and recognizable structures are found, this matches the findings from the previous section about text separation. The best description level is found to be with 5 components. A hierarchical structure is clear in this classification for 3 and more IC components. This is not all surprising given the result previous on text classification, but also since text can be regarded as describing the category labels, that also are man made.

Combined image and text features

Combining the three different modalities of texture, color and text can be done using only two modalities or all in one combined feature matrix. We present here the result of the all three combined, that also show the general trend from the other combinations. Table 4.10 states the confusion matrix using 2 to 5 IC components in the classification.

The classification with 2 to 4 components show a more clear grouping structure compared to using just one media. The classification where each modality seem to be the most dominant improved collectable. In the 3 IC component single modality classification the texture classification was best, but both text and

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	70	55	55
$\mathbf{IC_2}$	30	37	45

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	100	92	0
$\mathbf{IC_2}$	0	0	55
$\mathbf{IC_3}$	0	0	45

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	45	1	0
$\mathbf{IC_2}$	55	91	0
$\mathbf{IC_3}$	0	0	55
$\mathbf{IC_4}$	0	0	45

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	80	1	0
$\mathbf{IC_2}$	19	0	0
$\mathbf{IC_3}$	1	91	0
$\mathbf{IC_4}$	0	0	55
$\mathbf{IC_5}$	0	0	45

Table 4.9 Confusion matrix from classification with 2 to 4 output IC components using text. The confusion matrix compares the classification of the ICA algorithm to the categories $\mathbf{C}_1$:Aviation, $\mathbf{C}_2$:Travel and $\mathbf{C}_3$:Sports. Text class errors are $[122, 92, 56, \mathbf{2}]$

color modality prefers to divide the sports category in two, and they seem to be the stronger parts. Thus even though the ICA classification seemingly does something sensible when looking at the confusion matrix, the error is high since it does not describe the categories well. For the 4 component classification no single modality classification work good, but the combined did. The 2 component classification had a lower error than the 4 component, but also has lesser components than classes. The 4 component result is therefore the more interesting in regard to classification of the categories, and seem to be a consensus of the 3 media.

The ICA classification can be explained by backprojection of the independent components. In table 4.11 we show the keywords belonging to the 4 component

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	100	92	0
$\mathbf{IC_2}$	0	0	100

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	100	92	0
$\mathbf{IC_2}$	0	0	54
$\mathbf{IC_3}$	0	0	46

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	92	2	0
$\mathbf{IC_2}$	5	90	0
$\mathbf{IC_3}$	0	0	55
$\mathbf{IC_4}$	0	0	45

	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
$\mathbf{IC_1}$	66	4	0
$\mathbf{IC_2}$	0	76	0
$\mathbf{IC_3}$	0	0	52
$\mathbf{IC_4}$	0	0	45
$\mathbf{IC_5}$	34	12	3

Table 4.10 Confusion matrix from classification with 2 to 5 output IC components using text and image features. The confusion matrix compares the classification of the ICA algorithm to the categories $\mathbf{C}_1$:Aviation, $\mathbf{C}_2$:Travel and $\mathbf{C}_3$:Sports. Combined media class errors are $[0, 92, 7, 19]$

classification. The keywords somewhat underline the class categories and e.g. verify that component 4 hold a ice hockey class in the sports category by the word nhl.

4.2.4 Summary

ICA seem to identify the grouping structure in the feature data better than in the LSI model. This we must assume is partially because ICA is not restricted to an orthogonal basis as is LSI.

IC components	keywords
$\mathbf{IC_1}$	afb air wing overcast aluminum space boeing photographer lockett airshow airplane stratofortresses
$\mathbf{IC_2}$	view building dome park garden place fall
$\mathbf{IC_3}$	weight height position lbs born college
$\mathbf{IC_4}$	draft weight position lbs born height selected round nhl

Table 4.11 Keywords from the 4 component ICA classification using all three feature modalities.

Exploiting this property we can use ICA for unsupervised classification. Regarding text, the number of components seem to project a hierarchical structure that correspond to human labeling, thus a human context taxonomy. Evidence of this was present, but not all clear when using image features. The image features used are low level - color and texture, thus describing context more in general, as does text. As such, the description level of the data for each of the media where different: color 2, texture 3 and text 5. We will therefore expect this kind of ordering regarding most multimedia data sets. Another reason that we do not see the grouping structure so clearly in image features is that classifying the data in more components than are natural present, does not comply well with the independent ”ray” like classification that ICA exploits.

In combination of all three modalities - text, color and texture, the overall grouping structure in the classification was strengthened. This presents evidence that all modalities adds valuable information.

In regards to ICA algorithm, we used models with symmetric source probability function, that in principal is not the natural choice giving the feature data is strictly positive. From experience we do however find that ”flipping” the components by changing component sign in general works fine, as opposed to the results from separation of raw images. In the online chat room application presented in the next chapter 5, we did however experience anti-correlated components from time to time. This present an interesting social point regarding chat room behaviour, thus when a given semantic (vocabulary) is used, another is definitely not.

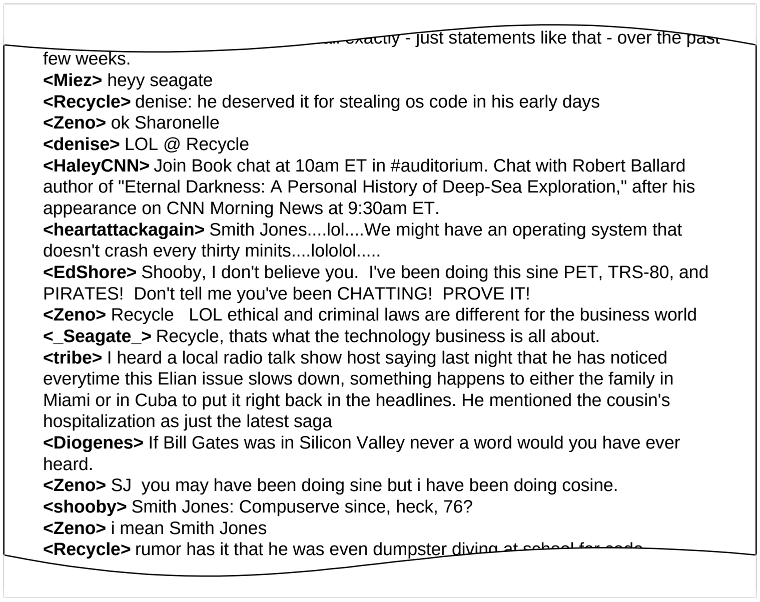
CHAPTER 5

Applications of ICA in virtual environments

5.1 ICA in chat rooms

Internet chat rooms are getting more and more popular in various relations. They define in principle their own contexts and often with a mixture of topics at the same time. This is especially true for the *cafe* like chat rooms, where no or little interference is present from a supervisor or moderator. Figure 5.1 shows a small sample of such chat room. In spite of this anarchy, valuable information can be obtained from monitoring these activities. Information about e.g. peoples general thoughts on the daily news and trends. Another purpose, is that of presenting the chat users with the resent discussed topics in a chat room before entering, or giving notice when a topic is being discussed for he/she to participate.

Related research areas are found in *topic detection and tracking*[1] where generally news streams are analysed for the purpose of collecting overall reports. In the chat room text streams we do however have topics mixed together without clear beginnings and endings, and so separating with ICA seem the obvious choice. Related work can be found in [9] that extends this framework in *pro-*



exactly - just statements like that - over the past few weeks.
<Miez> heyy seagate
<Recycle> denise: he deserved it for stealing os code in his early days
<Zeno> ok Sharonelle
<denise> LOL @ Recycle
<HaleyCNN> Join Book chat at 10am ET in #auditorium. Chat with Robert Ballard author of "Eternal Darkness: A Personal History of Deep-Sea Exploration," after his appearance on CNN Morning News at 9:30am ET.
<heartattackagain> Smith Jones....lol....We might have an operating system that doesn't crash every thirty minits....lololol.....
<EdShore> Shooby, I don't believe you. I've been doing this sine PET, TRS-80, and PIRATES! Don't tell me you've been CHATTING! PROVE IT!
<Zeno> Recycle   LOL ethical and criminal laws are different for the business world
<_Seagate_> Recycle, thats what the technology business is all about.
<tribe> I heard a local radio talk show host saying last night that he has noticed everytime this Elian issue slows down, something happens to either the family in Miami or in Cuba to put it right back in the headlines. He mentioned the cousin's hospitalization as just the latest saga
<Diogenes> If Bill Gates was in Silicon Valley never a word would you have ever heard.
<Zeno> SJ you may have been doing sine but i have been doing cosine.
<shooby> Smith Jones: Compuserve since, heck, 76?
<Zeno> i mean Smith Jones
<Recycle> rumor has it that he was even dumpster diving at school for code

Figure 5.1 The chat consists of a mixture of contributors discussing multiple concurrent topics. The figure shows a small sample of the a CNN.com chat room, April 5, 2000.

jection pursuit.

In the following we use the ICA text classification previously presented. The Molgedey and Schuster ICA algorithm is especially attractive given the dynamic nature of chat data, and the minor model complexity for online purposes. At first we look at a retrospective analysis of a whole day to illustrate the principals, and secondly present the online WebChat Internet page.

5.1.1 Chat data

The dataset used to demonstrate the chat room analysis is generated from the daily chat at CNN.com in channel #CNN. In this particular chat room daily news topics are discussed by non-experts. A CNN moderator supervises the chat to prevent non-acceptable contributions and for occasional comments. All

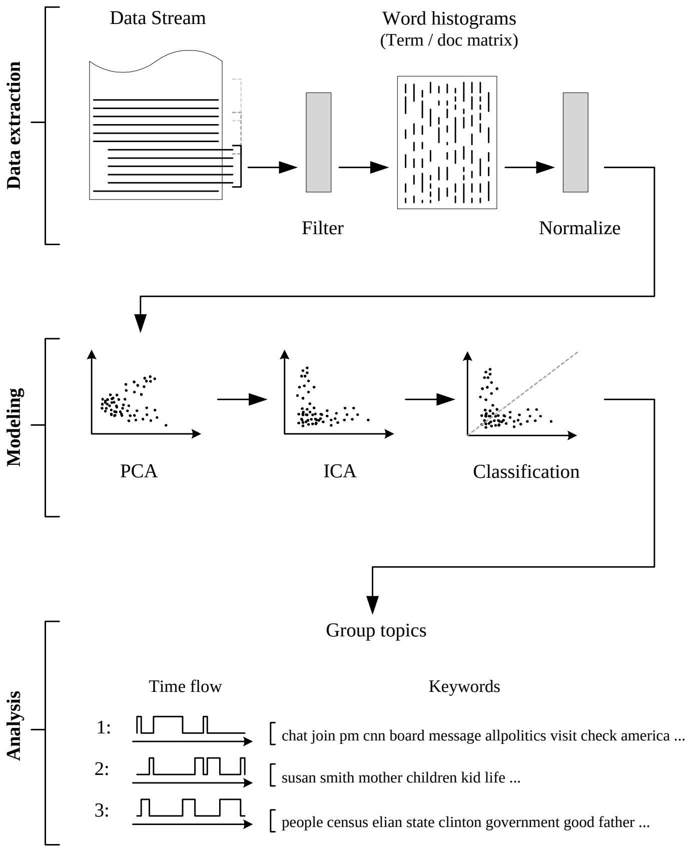

Data extraction
Data Stream
Word histograms
(Term / doc matrix)
Filter
Normalize
Modeling
PCA
ICA
Classification
Analysis
Group topics
Time flow
Keywords
1:
chat join pm cnn board message allpolitics visit check america ...
2:
susan smith mother children kid life ...
3:
people census elian state clinton government good father ...

Figure 5.2 The text analysis process is roughly divided into three phases: Feature extraction and construction of the term histograms and the analysis where the group structure is visualized and the dynamic components presented.

chat was logged in a period of 8.5 hours on April 5, 2000, generating a dataset of 4900 lines with 128 different users chatting. The data set was cleaned by removal of non-user generated text, all users names, stop words and non-alphabetic characters. The remaining text was merged into one string and a window of size 300 characters was used to segment the contiguous text in pseudo-documents. The window was moved forward (without breaking words apart), leaving an overlap of 50% between each window. Term histograms were generated from each pseudo-document forming a 2495×1114 term-document matrix. The general framework for ICA text classification was used as introduced in section 4.2.1, and the whole process is summarized in figure 5.2.

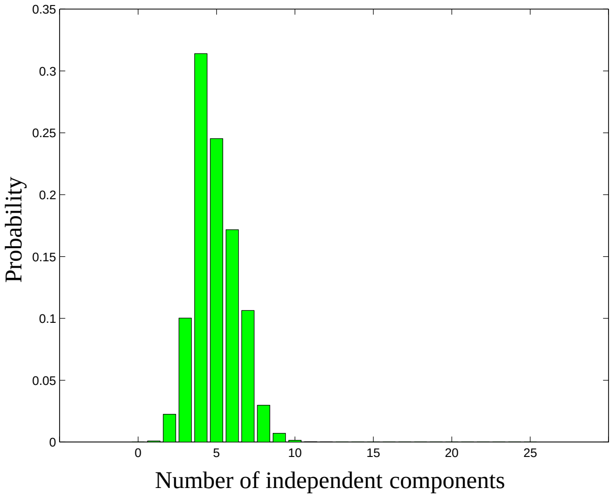


Figure 5.3 The BIC approximation is used to calculate the most probable number of components, that show to be 4 components. The log probabilities are normalized by number of samples, thus providing a conservative estimate. The zero components model corresponds to a white noise null hypothesis.

5.1.2 Retrospective analysis

In retrospective analysis of a whole day we use the full term-document matrix as described above. We hereby use ICA classification to find the most independent significant underlying topics, and use the Bayesian information criterion to decide on the number of components, hence number of topics.

In figure 5.3 the posterior probability using BIC is plotted as a function of IC components. The most probable model is 4 components. The resulting IC components show both short and long time scale rhythms as seen in figure5.4 (top), with their corresponding classification (bottom). A rejection group R was used for samples where the largest probability was not above 0.2 of the others. The keywords for the content of the topics spotted by the 4 IC components are: The first topic is dominated by the CNN moderator and immediate responses to these contributions, the second is a discussion on gun control, the third is concerned with the Susan Smith killings and her mother who appeared live on

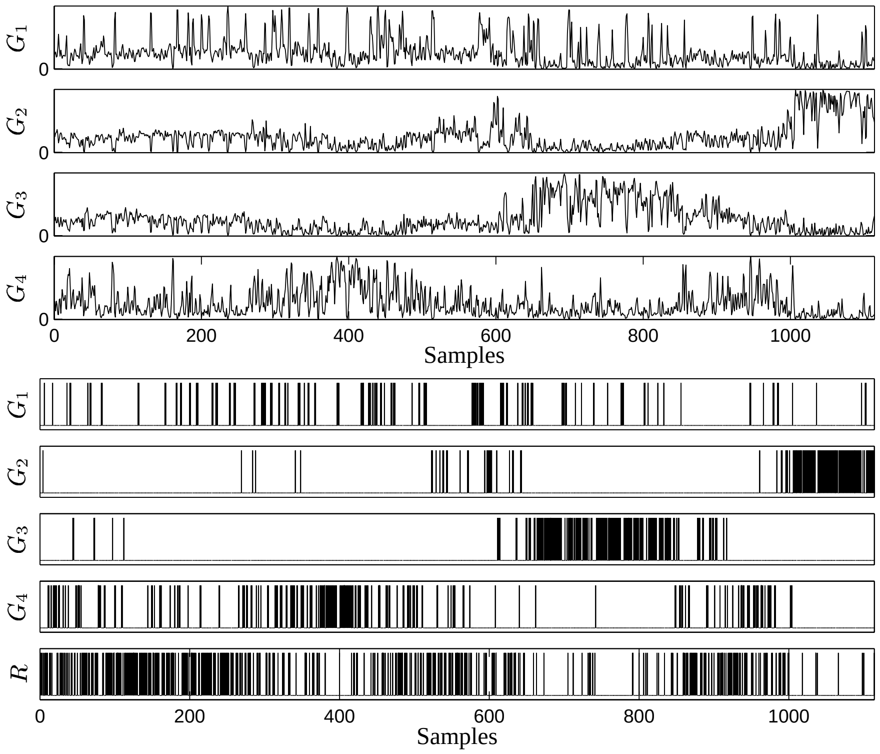


Figure 5.4 The figure shows the ICA components (top) and the result of the classification (bottom) of topic groups, as function of samples, that equal the linear time during the 8.5 hours of chat.

CNN, and the fourth is an intense discussion of the Cuban boy Elian's case. Hence, topics of high current public interest at the present time. The CNN moderators participated in the discussions, but also made announcements frequently doing the day e.g about Susan Smiths mother appearing on CNN. Back projecting the full chat line data (no overlapping) on the found ICA basis, we count the number of lines each user participated with in the 4 topics, as shown in figure 5.5. The first discussion is mainly dominated by one user (#20) that turns out to be a CNN moderator, and this concurs nicely with the keywords. In the other discussions more users are active and some participate in more that one topic (as e.g. #2 and #117). Somewhat strangely one non-moderator user (#40) is also indicated to be very active in the first topic. Although this user did not do CNN announcements, other underlying topics are in the chat, that are

	keywords
Group 1	cnn join pm burden proof
Group 2	gun show
Group 3	susan smith mother children burden kill kid life proof
Group 4	people census elian state clinton government thing year good father time

Table 5.1 Projecting the independent components found in figure 5.4 back to the term (histogram) space produces a histogram for each IC component. We select the terms with the highest backprojects to produce a few keywords for each component. The first topic is dominated by the CNN moderator and immeadiate responses to these contributions, the second is a discussion on gun control, the third is concerned with the Susan Smith killings and her mother who appeared live on CNN, and the fourth is an intense discussion of the Cuban boy Elian's case.

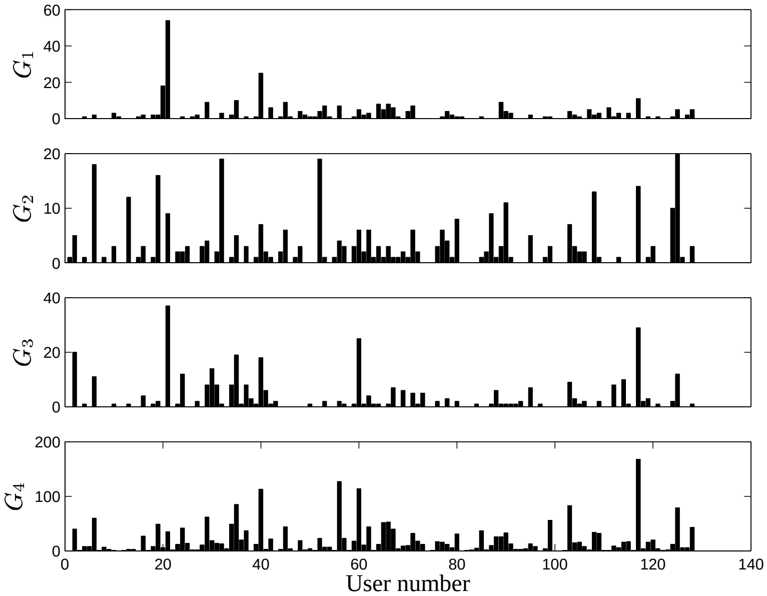
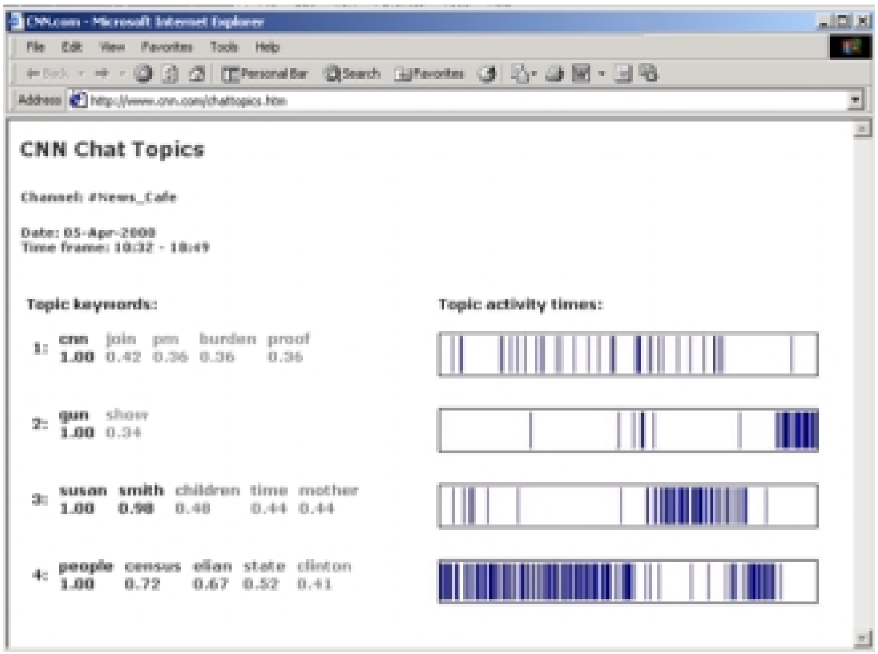


Figure 5.5 For each topic class the number of chat lines are presented for each user, represented by a number $1..128$.

closer to this topic than the other three.


CNN.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Personal Bar Search Favorites
Address http://www.cnn.com/chattopics.htm

CNN Chat Topics

Channel: #News_Cafe

Date: 05-Apr-2000
Time frame: 10:32 - 18:49

Topic keywords:

1: cnn join pm burden proof
1.00 0.42 0.36 0.36 0.36

2: gun show
1.00 0.34

3: susan smith children time mother
1.00 0.98 0.40 0.44 0.44

4: people census elian state clinton
1.00 0.72 0.67 0.52 0.41

Topic activity times:

Figure 5.6 The online WebChat programs monitors an ongoing chat room at CNN in channel #NEWSCAFE. The current Internet address for WebChat is http://base.imm.dtu.dk/webchat/index.htm.

5.1.3 WebChat

To present the user with the previous discussed topics before entering a char room, we implemented an online version of the ICA chat analysis - WebChat. The online version analysis the last 1000 chat lines of the CNN chat room in channel #NEWSCAFE as seen in figure 5.6. The dynamic classification is presented with keywords and number of topics are chosen using the BIC approximation. In connection to each keyword is a weight underneath (and the color gray-scaling of the keyword), that describes the specific keywords significance in the topics. Clicking on the component classification graph links the user to the collected chat lines belonging to the given topic.

Monitoring WebChat trough several hours, topics appeared and disappeared continuously, as did the number of components vary depending on the discussions.

5.2 ICA in web search

Internet users ranked *search* in a survey [88] as their most important activity on the Internet web, and awarded it 9.1 out of 10 points. Different approaches have been proposed to improve the search itself by commercial and research interests, but seem to be lagging the user paradigm[82]. In general the field of visually presenting the search results is trailing behind, and thus commercial search engines usually order the results in a simple ranked list. Many different approaches has although been proposed by the research community as e.g. [14, 77, 31], of which some we most likely will see more of in the future.

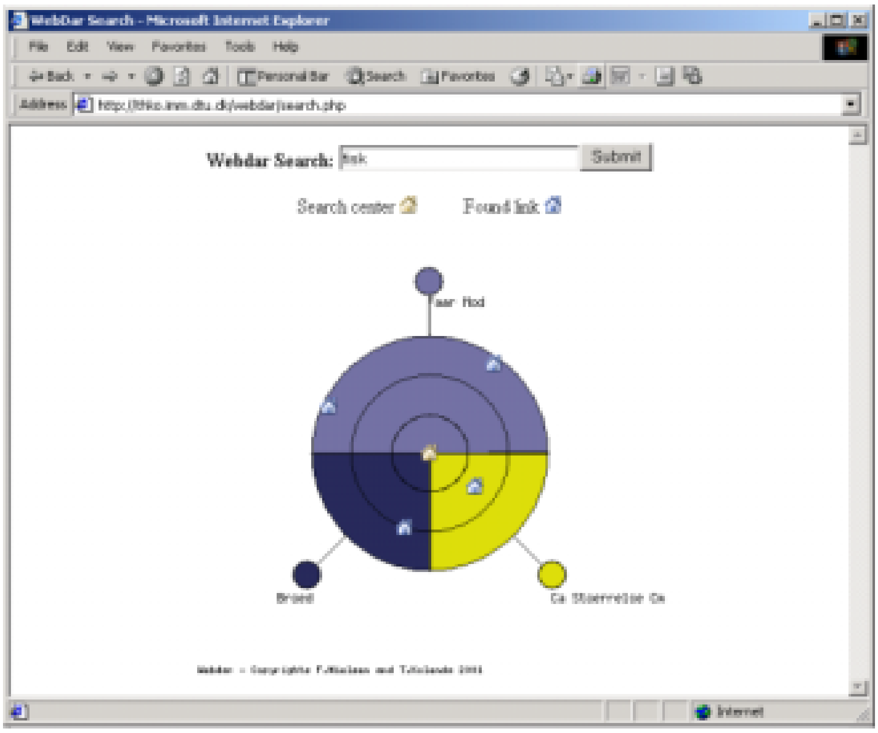
In general close to halve of all Internet users searching only write one query for a given search; and does hereafter not refine the search, but look trough the often huge amount of search results[16]. In addition to this, is the average query string only made up of $1\frac{1}{2}$ words[1]. Some commercial search engines do although aid the user in searching by relating the search result to categories as Google[30] or NorthernLight[70], or by purposing new search query, based on what other users have done as AltaVista[20] or Ask Jeevs[46].

Using ICA we take a look at how its properties could be set to use in search. Our strategy is to use its unsupervised classification to group the search results, and further to use the IC component class keywords to purpose a new and more refined query. The search is therefore hierarchical, and an interaction between the user and the search. Finally we want to exploit the ICA basis in visualization of the search results, to show the relative distance between the found links. From experience with 3D navigation in VRML we omit this idea because navigation is to difficult, and since it normally is viewed on a 2D screen. This leaves us with 2 dimensions as in e.g. topographic maps, and using the idea from [77] of a radar like projection.

5.2.1 WebDar

To test the above stated ideas we implemented a Internet application WebDar, that works as an extension to the Jubii search engine with ML ICA. Unfortunately did implementation difficulties prevent that the full version of the WebDar came to life before finishing this writing, thus we present a minor working

[1] By experience of the commercial web company Ankiro.



Webdar Search: link

WebDar Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Personal Bar Search Favorites
Address http://thko.imm.dtu.dk/webdar/search.php
Submit
Search center
Found link
Brød
Internet

Figure 5.7 The WebDar takes advantage of the ICA properties in dividing the search results in classes and purposing new word for the query. The current Internet address for the demo of WebDar is http://thko.imm.dtu.dk/webdar/search.htm.

demo that outlines the general ideas, see figure 5.7. The data we use consist of 750 Danish homepages from the Jubii.dk[2] database.

The search query is the center (yellow house) of the radar or pie image, and the distance to the found search results (blue houses) on the radar are equal to the Jubii search ranking. The number of pie slices are determined by the ICA and the size is relative to the number of search results in a given class. Thus, the size gives the user an idea of how narrow the class is. Each IC class is represented as a pie, with a pin as the given IC basis. The IC basis vectors are sorted according to how close they are to each other. A given search result is placed at an angle relative to the two IC basis's it lie between, hereby showing

[2] We thank the companies Jubii A/S and Ankiro for letting us use their database and support.

the distance to them. The maximum number of IC classes are not limited, but should in principal not be more that 3 for the angle distance measure to hold. The maximum number of search results shown is user defined, but default 10 as most search engines.

Using WebDar the user submits a query to the search engine. The result is classified by the ICA algorithm and keywords calculated. After presenting the user with the WebDar radar image the links with the highest ranking is presented according to the above description, and the keywords belonging to each class are written with its given pins. Clicking on the blue houses links to the found results, or clicking on the pins submits a new search including the keywords belonging to the pin.

In figure 5.8 a search is done (in Danish but translated in this text) with the word fish. The result showed 4 links and 3 classes:

Class keywords	**link description**
Ca Size Cm Farm	A fish farm homepage
Bread	Menu from a restaurant
Eat	Person writing about his world experiences (left) Advice about healthy food homepage (right)

Pressing the pin with the keyword Bread submits a new query to the search engine with Fish Bread, and the following result is found:

Class keywords	**link description**
Cheese	Menu from a restaurant
Fruit	Advice about healthy food homepage

It should be noted that the ”advice about healthy food homepage” was present in two different classes, although close in angle. This is fully acceptable. Studies show that people are not generally interested in the words on a homepage but its contents and is known as the *paraphrase problem*[101]. Thus choosing one or the other keyword that are close in angle should not be a hard decision.

In further development we plan to improve the graphical presentation of the found homepages, to reveal the homepage descriptions belonging to a class,

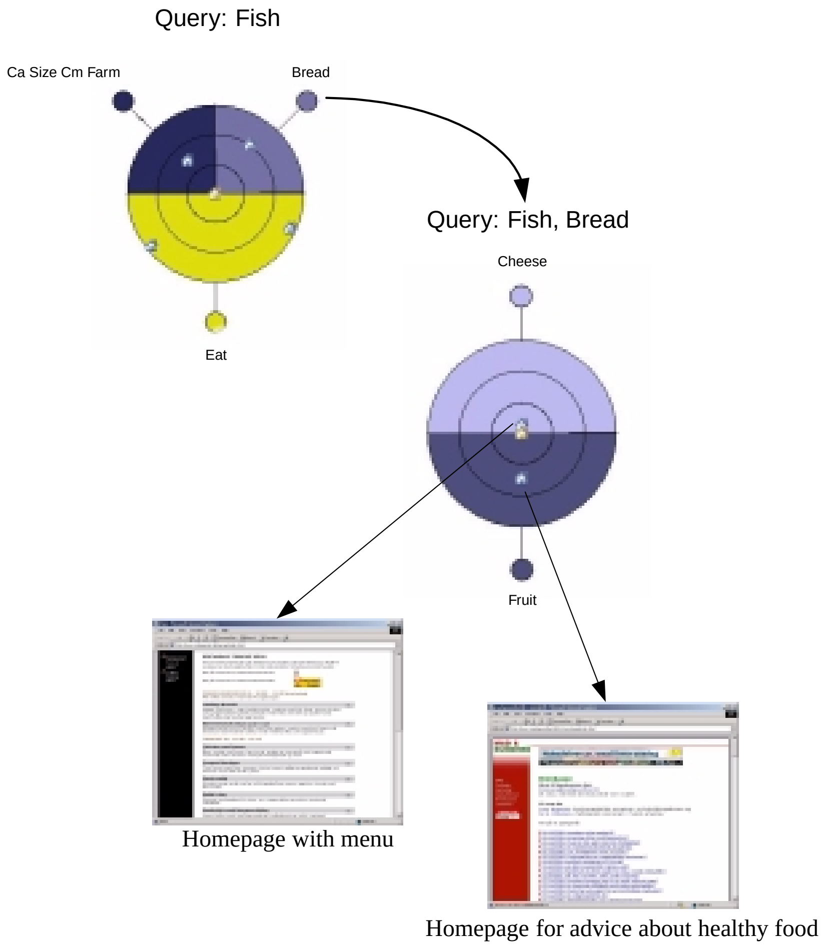


Figure 5.8 A search is started by submitting the word Fish (upper left), and futher extended by pressing the ”Bread” class pin to submit the words: Fish Bread. Clicking on the blue houses opens a given search result.

when the mouse is over its given pie. Also to let the user choose between keywords by clicking on them directly. Finally, going online with the full Jubii.dk database of 1.2 mill. homepages, is crucial, to see if the ICA properties truly can add value in searching the web.

CHAPTER 6

Conclusion

The focus of the Ph.D. thesis has been to find new tools for software agents in virtual environments as for example, the Internet. A primary problem is to look at how statistics can define rules that reveal context in a human sense. We recognized independency as a natural criteria for separation in a early stage of the project, thus focusing on this using independent component analysis (ICA).

In general we used a linear ICA model with possible gaussian noise. A short introduction to the properties of ICA is presented and the framework for probabilistic ICA algorithms, using a maximum likelihood (ML) and a mean field (MF) formulation. Further we present the dynamic Molgedey and Schuster (MS) ICA algorithm based on joint diagonalization. In the latter case we acknowledge the use of a single time delay to be enough, but from experiments also recognize how sensitive the algorithm is to the choice of the delay parameter. We therefore formulate a method for finding the delay parameter τ and verified the results by both exhaustive search using Bayes information criterion (BIC) and from experiments.

Using the criteria of independence on different media, we investigated the properties of the ICA separation. The two ICA frameworks with ML and MS, respectively with or without use dynamic input, were tested for computational speed and accuracy on artificially mixed sound signals, against the traditional principal component analysis (PCA) algorithm. Of the two ICA algorithm, MS was the fastest and ML the most accurate. The quality of the separation when listening was good in both cases, as opposed to the PCA solution.

Separation of images was looked into, both by achieving independency between pixels and images. For that we employed the MS and positive MF algorithms using artificial data. The positive MF assumes positive source distributions and mixing. Nothing conclusive could be said about which method is best, and instead it should be decided giving a concrete problem. Regarding positive and non-positive constraint ICA, the positive clearly showed a better result in regards to better interpretability of the separated images. Comparing the positive ICA with the non-negative matrix factorization (NMF) algorithm, both gave close to the same result, thus we conclude that the positive constraint holds much stronger than the independency.

A general framework was presented for ICA classification on features in extension to the latent semantic indexing model (LSI). This was demonstrated on text and images, using term, texture and color frequencies. Evidence was found that the separation by independence presents a hierarchical structure that relates to context in a human sense. Towards manual labeling of a given data set, best description level was determined as to how few ICA components produced minimum classification error. In the setting of multiple media modalities a combined hierarchical structure was found to reflect the context described at multiple levels, thus to reflect the collectable impression of the context.

Employing the properties of ICA, online Internet applications were implemented in the setting of chat room analysis and visualization of search engine results. The analysis of chat rooms seem a natural choice of application, giving that chat room text streams are a mixture of simultaneous unsupervised discussions. The MS ICA algorithm was utilized due to its minor computational burden, and therefore ideal for online purposes. Model selection was done using BIC as to find the number of classes, hence topics. Finally the ongoing separation for a fixed number of chat lines was presented on an Internet web page, showing topic keywords and activation times. In the application of visualizing search engine results, the result of a given search can be grouped, thus presenting the user with a better overview. Using a pie radar-like visualization, the information on both search engine ranking and intermediate distance between search results can be shown, relative to the classification. The keywords from the ICA classification is likewise presented to the user, to suggest new words for further search. The application was implemented as extension to a public Danish search engine, but because of implementation problems it was not tested on the full database in due time.

APPENDIX A

Detailed equations

This appendix hold more detailed calculations of equations mentioned in the main text. In general [93] were used throughout the thesis for matrix calculation.

A.1 Mean Field Likelihood equality

When differentiating with respect to the parameters e.g. **A**, we can write the connection between the log likelihood of the mixed signals given the parameters, and the log likelihood of the mixed signals given the parameters and the source signals.

$$
\begin{aligned}
&\frac{\partial}{\partial \mathbf{A}} \log p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}) \\
&= \frac{\partial}{\partial \mathbf{A}} \log \int p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) p(\mathbf{S}) \; d(\mathbf{S}) \\
&= \left(\int p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) p(\mathbf{S}) \; d(\mathbf{S}) \right)^{-1} \cdot \frac{\partial}{\partial \mathbf{A}} \int p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) p(\mathbf{S}) \; d(\mathbf{S}) \\
&= p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma})^{-1} \int p(\mathbf{S}) \frac{\partial}{\partial \mathbf{A}} p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) \; d(\mathbf{S}) \\
&= \int \frac{p(\mathbf{S}|\mathbf{X}\mathbf{A}\mathbf{\Sigma})}{p(\mathbf{X}|\mathbf{S}\mathbf{A}\mathbf{\Sigma})} \frac{\partial}{\partial \mathbf{A}} p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) \; d(\mathbf{S}) \\
&= \int p(\mathbf{S}|\mathbf{X}\mathbf{A}\mathbf{\Sigma}) \frac{\partial}{\partial \mathbf{A}} \log p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) \; d(\mathbf{S}) \\
&= \left\langle \frac{\partial}{\partial \mathbf{A}} \log p(\mathbf{X}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) \right\rangle_{p(\mathbf{S}|\mathbf{X}\mathbf{A}\mathbf{\Sigma})}
\end{aligned}
\tag{A.1}
$$

, where $\langle\cdot\rangle$ denotes the sources posterior average given the mixed signals, the mixing matrix and the noise covariance matrix. We used Bayes rule from the equality of (3.7) and (3.7), and the integral from (3.9). Also we used that $\frac{\partial}{\partial x} \log f(x) = \frac{1}{f(x)} \frac{\partial}{\partial x} f(x)$ twice. Once to remove the $\log$ and later to insert it again.

APPENDIX B

Papers

The most important papers published in relation to work done on this thesis are shortly described in this appendix.

B.1 Independent Components in Text

[57] T. Kolenda, L.K. Hansen and S. Sigurdsson, *Independent Components in Text* in M. Girolami (ed.) Advances in Independent Component Analysis, Springer-Verlag, chapter 13 229-250, 2000.

Description: We introduce a framework for ICA classification in text, and analyze the feasibility of ICA for dimensional reduction and representation of word histograms. The study is carried out using mean field that allows for estimating the noise level. We also discuss the generalizability of the estimated models and show that an empirical test error estimate may be used to optimize model dimensionality, in particular the optimal number of sources. When applied to word histograms ICA is shown to produce representations that are better aligned with the group structure in the text data than the LSA.

Contributions in this paper from this writings author is largely found in section 4.2.1 about text separation.

B.2 On Independent Component Analysis for Multimedia Signals

[37] L.K. Hansen, J. Larsen and T. Kolenda *On Independent Component Analysis for Multimedia Signals*. in L. Guan et al. (eds.) Multimedia Image and Video Processing, chapter 7, 175-200, 2000.

Abstract: We discuss the independent component problem within a context of multimedia applications. The literature offers several independent component analysis schemes which can be applied in this context, and each have its own trade-off between flexibility, complexity and computational effort. The specific applications investigated in this chapter comprise modeling of speech/sound, images, and text data.

Contributions in this paper from this writings author is largely found in section 4.1.1 about separation of sound signals, and in section4.1.2 about separation of images.

B.3 Signal Detection using ICA: App. to Chat Room Topic Spotting

[56] T. Kolenda, L.K. Hansen and J. Larsen, *Signal Detection using ICA: Application to Chat Room Topic Spotting* in proc. ICA'2001, 2001.

Abstract: There is an increasing interest in the application of machine learning methods in text analysis. We apply independent component analysis to dynamic text gathered in a CNN chat room. Using dynamic decorrelation we find that there are stable dynamic components during eight hours contigous chat. The components have widely different dynamic structure as reflected in their temporal autocorrelation functions. Each component activates a distinct word

frequency histogram and these histograms are straightforward to relate to news topics on the given day.

Contributions in this paper from this writings author is largely found in section 3.4.2 about determining the value of tau, and in section 5.1 about ICA in chat applications.

Bibliography

[1] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. In *In proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[2] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 757–763. The MIT Press, 1996.

[3] H. Attias and C.E. Schreiner. Blind source separation and deconvolution by dynamic component analysis. *Neural Networks for Signal Processing VII, in proc. of the 1997 IEEE Workshop, 456-465 (1997)*, pages 456–465, 1997.

[4] H. Attias and C.E. Schreiner. Blind source separation and deconvolution: The dynamic component analysis algorithm. *Neural Computation*, 10:1373–1424, 1998.

[5] M.S. Bartlett, H.M. Lades, and T.J. Sejnowski. Idependent component representations for face recognition. *In proc. of the SPIE – The International Society for Optical Engineering*, 3299:528–539, 1998.

[6] A. Bell and T. Sejnowski. Edges are the independent components of natural scenes. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 831. The MIT Press, 1997.

[7] A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[8] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. Scientific American.com, Internet, 2001. http://www.scientificamerican.com/2001/ 0501issue/ 0501berners-lee.html.

[9] E. Bingham. Topic identification in dynamical text by extracting minimum complexity time components. *In proc. ICA'2001*, pages 546–551, 2001.

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

[11] J. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Process. Lett.*, 4(4):112–114, 1997.

[12] J. Cardoso. Statistical principles of source separation. *In proc. SYSID'97, 11th IFAC symposium on system identification*, pages 1837–1844, 1997.

[13] J. Cardoso and A. Souloumiac. Blind beamforming for non gaussian signals. *In proc. IEE - F*, 140:362–370, 1993.

[14] J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. *Computer Networks and ISDN Systems*, 29(11):1257–1267, 1997.

[15] M. Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. *In proc. IEEE Workshop on ContentBased Access of Image and Video Libraries – IEEE Computer Society*, pages 24–28, 1998.

[16] C.Bradford and I.W.Marshall. Analysing users www search behaviour. *In Colloquium on Lost in the Web: Navigation on the Internet, IEE*, pages 6/1–6/4, 1999.

[17] Jubii Chat. Internet, 2001. http://chat.jubii.dk/.

[18] K. Cla, Y. Tracie, E. Monk, and D. McRobb. Internet tomography, 1999. http://helix.nature.com.

[19] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[20] AltaVista Company. Internet, 2002. http://www.altavista.com/.

[21] Telcordia Technologies Inc. An SAIC Company. Evaluating the size of the internet. Internet, 2001. http://www.netsizer.com/.

[22] The Salk Institute Computational Neurobiology Lab. Cnl. Internet, 2001. http://www.cnl.salk.edu/.

[23] The World Wide Web Consortium. Internet, 2001. http://www.w3.org/.

[24] G. Deco and D. Obradovic. *An Information-Theoretic Approach to Neural Computing*. Springer-Verlag, 1996.

[25] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. for Inf. Science*, 41:391–407, 1990.

[26] M. Delio. Wired news. Internet, 2001. http://www.wired.com/ news/ technology/ 0,1282,44112,00.html.

[27] D. Dunn and E.William E. Higgins. Optimal gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, 4(7):947–964, 1995.

[28] T. Finin and Y. Labrou. Umbc agentweb. UMBC Laboratory for Advanced Information Technology, 2001. http://agents.umbc.edu/.

[29] Stan Franklin and Art Graesser. A taxonomy for autonomous agents. In *In proc. of the Third International Workshop on Agent Theories, Architectures, and Language*, 1996.

[30] Google. Internet, 2002. http://www.google.com/.

[31] R.S. Grewal, M. Jackson, P. Burden, and J. Wallis. A novel interface for representing search-engine results. *In Colloquium on Lost in the Web: Navigation on the Internet, IEE*, pages 7/1–7/10, 1999.

[32] INT Media Group. Botspot homepage. Internet, 2001. http://www.botspot.com/.

[33] N. Guarino. *Formal Ontology and Information Systems*. IOS Press, June 1998.

[34] L.K. Hansen. Blind separation of noicy image mixtures. *M. Girolami, editor, Advances in Independent Component Analysis, Springer-Verlag*, pages 159–179, 2000.

[35] L.K. Hansen and J. Larsen. Source separation in short image sequences using delayed correlation. *P. Dalsgaard and S.H. Jensen, editors, in proc. of the IEEE Nordic Signal Processing Symposium, Vigsø, Denmark 1998.*, pages 253–256, 1998.

[36] L.K. Hansen and J. Larsen. Unsupervised learning and generalization. *In proc. of IEEE International Conference on Neural Networks*, 1:25–30, 2000.

[37] L.K. Hansen, J. Larsen, and T. Kolenda. On independent component analysis for multimedia signals. *L. Guan, S.Y. Kung and J. Larsen , editors, Multimedia Image and Video Processing, CRC Press*, Chapter 7:175–200, 2000.

[38] Laboratory of Computer Helsinki University of Technology and Neural Network Research Center Information Science. Laboratory of computer and information science. Internet, 2001. http://www.cis.hut.fi/.

[39] P. Hoyer and A. Hyv. Independent component analysis applied to feature extraction from colour and stereo images. *Computation in Neural Systems*, 11(3):191–210, 2000.

[40] J. Hurri, A. Hyv, J. Karhunen, and E. Oja. Image feature extraction using independent component analysis. *In proc. NORSIG'96*, 1996.

[41] A. Hyv, R. Cristescu, and E. Oja. A fast algorithm for estimating overcomplete ica bases for image windows. *In proc. Int. Joint Conf on Neural Networks*, 1999.

[42] A. Hyv, r Oja, P. Hoyer, and J. Hurri. Image feature extraction by sparse coding and independent component analysis. *In proc. Int. Conf. on Pattern Recognition (ICPR'98)*, pages 1268–1273, 1998.

[43] A. Hyvärinen. Complexity pursuit: Separating interesting components from time-series. *Neural Computation*, 13(4):883–898, 2001.

[44] S. Icart and R. Gautier. Blind separation of convolutive mixtures using second and fourth order moments. In *ICASSP*, pages 3018–3021, 1996.

[45] Amazone.com Inc. Internet, 2001. http://www.amazone.com/.

[46] Ask Jeeves Inc. Internet, 2002. http://www.askjeeves.com/.

[47] Merriam-Webster Inc. Internet, 2001. http://www.m-w.com/.

[48] Pioneer Funds Distributor Inc. Internet, 2001. http://www.pioneerfunds.com/.

[49] Simpli.com Inc. Internet, 2001. http://www.simpli.com/.

[50] C.L. Isbell and P. Viola. Restructuring sparse high dimensional data for effective retrieval. *In proc. of NIPS'98*, 11:480–486, 1998.

[51] N. R. Jennings and M. J. Wooldridge. Applications of intelligent agents. In N. R. Jennings and M. J. Wooldridge, editors, *Agent Technology: Foundations, Applications, and Markets*, pages 3–28. Springer-Verlag: Heidelberg, Germany, 1998.

[52] A. Kabán and M. Girolami. Clustering of text documents by skewness maximisation. *In proc. of ICA'2000*, pages 435–440, 2000.

[53] P. Kidmose. *Blind Separation of Heavy Tail Signals*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2001.

[54] H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Workshop CAPAIDM*, Pasadena, CA, 1983.

[55] T. Kolenda. Independent component analysis - an introduction. Master's thesis, Technical University of Denmark, IMM, 1998.

[56] T. Kolenda, L.K. Hansen, and J. Larsen. Signal detection using ica: Application to chat room topic spotting. *In proc. ICA'2001*, 2001.

[57] T. Kolenda, L.K. Hansen, and S. Sigurdsson. Independent components in text. *M. Girolami, editor, Advances in Independent Component Analysis, Springer-Verlag*, pages 229–250, 2000.

[58] T. Kolenda, L.K. Hansen, O. Winther, and S. Sigurdsson. Dtu:toolbox. Internet, 2002. http://mole.imm.dtu.dk/toolbox/.

[59] B. Lautrup, L.K. Hansen, I. Law, N. Mørch, C. Svarer, and S.C. Strother. Massive weight sharing: A cure for extremely ill-posed problems. *In H.J. Hermanet et al., editors, Supercomputing in Brain Research: From Tomography to Neural Networks. World Scientific Pub. Corp.*, pages 137–148, 1995.

[60] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. *In proc. of NIPS'2000*, 13, 2000.

[61] D.D. Lee and H.S. Seungg. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[62] T. Lee. *Independent Component Analysis - Theory and Applications*. Kluwer Academic Publisher, 1998.

[63] T. Lee, A. Bell, and R. Lambert. Blind separation of delayed and convolved sources. *In proc. NIPS'97*, pages 758–764, 1997.

[64] T. Lee, M. Lewicki, and T. Sejnowski. Unsupervised classification with nongaussian mixture models using ica. *In proc. NIPS'99*, 1999.

[65] Te-Won Lee, Mark Girolami, and Terrence J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Computation*, 11(2):417–441, 1999.

[66] T.W. Lee, M.S. Lewicki, and T.J. Sejnowski. Ica mixture models for image processing. *In proc. of the 6th Joint Symposium on Neural Computation*, pages 79–86, 1999.

[67] R. Lemos. News: The cheese worm: A welcome helper? ZD Net News, Internet, 2001. http://www.zdnet.com/ zdnn/ stories/ news/ 0,4586,5083014,00.html.

[68] Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[69] M. Liptrot. The helpdesk market. Thor Project, DSP IMM Technical University of Denmark, 2000. http://eivind.imm.dtu.dk/ thor/ projects/multimedia/ textmining/ internal/ notes/.

[70] Northern Light Technology LLC. Internet, 2002. http://www.northernlight.com/.

[71] H.P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165,317, 1958.

[72] D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. in preparation, 1996.

[73] Bruce MacLennan. Gabor representations of spatiotemporal visual images. Technical Report CS-91-144, Computer Science Dep., University of Tennessee, 1994.

[74] J. M. Martínez. Overview of the mpeg-7 standard (version 5.0). Technical report, International Organisation for Standardisation ,ISO/IEC JTC1/SC29/WG11, Codeing of moving pictures and audio, 2001.

[75] Maxis. The sims. Electronic Arts Inc., 2001. http://thesims.ea.com/.

[76] T.P. Minka. Automatic choice of dimensionality for pca. *In proc. of NIPS'2000*, 13, 2000.

[77] Finn Årup Nielsen. *Neuroinformatics in Functional Neuroimaging*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2001.

[78] H. S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(2):205–244, 1995.

[79] Hyacinth S. Nwana and Divine T. Ndumu. A perspective on software agents research. *The Knowledge Engineering Review*, 14(2):1–18, 1999.

[80] International Conference on independent component analysis and blind signal separation. Internet, 2001. http://www.ica2001.org/.

[81] Nybolig Online. Internet, 2001. http://www.nybolig.dk/ nkm/ PrivatSearchForm.jsp ?kortID=NEkort-dk.

[82] J. Pagonis and M. Sinclair. Evolving personal agent environments to reduce internet information overload: Initial considerations. *In Colloquium on Lost in the Web: Navigation on the Internet, IEE*, pages 2/1–2/10, 1999.

[83] Z. Pecenovi'c. Image retrieval using latent semantic indexing. Master's thesis, AudioVisual Communications Lab, Ecole Polytechnique F'ed'erale de Lausanne, Switzerland, 1997.

[84] K.S. Petersen, L.K. Hansen, T. Kolenda, E. Rostrup, and S. Strother. On the independent components in functional neuroimages. *In proc. of ICA'2000*, 2000.

[85] Columbia Pictures. Final fantasy: The spirits within. Internet, 2001. http://www.finalfantasy.com/.

[86] M. Pötzsch and M. Rinne. Gabor wavelet transformation. Internet, 1996. http://www.neuroinformatik.ruhr-uni-bochum.de/ ini/ VDM/ research/ computerVision/ imageProcessing/ wavelets/ gabor/ contents.html.

[87] A.E. Raftery. Bayesian model selection in social research. Technical Report 94-12, University of Washington Demography Center, 1994.

[88] Jupiter Research. by Infoseek Press Release, 1999.

[89] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge: University Press, 1 edition, 1996.

[90] S. Russell and P. Norvig. Ai on the web. Computer Science Division Office, University of California, Berkeley, 2001. http://www.cs.berkeley.edu/ russell/ ai.html.

[91] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesle, 1989.

[92] G. Salton and C. Buckley. *Term Weighting Approaches in Automatic Text Retrieval*. Department of Computer Science, Cornell University, Technical Report TR87-881, 1987.

[93] S.M. Selby. *Standard Mathematical Tables*. CRC Press, 20 edition, 1972.

[94] TSI Department Signal-Images. Enst/tsi – image-signal department, 2001. http://sig.enst.fr/.

[95] M. Sipper. An introduction to artificial life. *Explorations in Artificial Life (special issue of AI Expert)*, pages 4–8, 1995.

[96] Smart. Department of computer science, cornell university. Public ftp, 1999. ftp.cs.cornell.edu/pub/smart/.

[97] P. H. Sorenson, O. Winther, and L.K. Hansen. Mean field approaches to independent component analysis. *Neural Computation, The MIT Press*, 14:889–918, 2002.

[98] L. Tesfatsion. *How Economists Can Get Alife*, volume XXVII. Addison-Wesley, 1997.

[99] Net Valley. History of the internet. Internet, 2001. http://www.netvalley.com/ archives/ mirrors/ davemarsh-timeline-1.htm.

[100] T. Westerveld. Image retrieval: Content versus context. *In proc. Content-Based Multimedia Information Access, RIAO 2000 – C.I.D.-C.A.S.I.S.*, pages 276–284, 2000.

[101] T. Westerveld, D. Hiemstra, and F. De Jong. Extracting bimodal representations for language-based image retrieval. *In proc. Eurographics Workshop, Multimedia '99*, pages 33–42, 2000.

[102] Working with the Office Assistant. Microsoft Corporation, 2001. http://msdn.microsoft.com/ library/ default.asp?url=/ library/ en-us/ modcore/ html/ deovrWorkingWithOfficeAssistant.asp.

[103] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[104] A. Ziehe and K. Muller. Tdsep – an efficient algorithm for blind separation using timestructure. *In proc. 8 th ICANN, Perspectives in neural computing*, pages 675–680, 1998.

Ph. D. theses from IMM

1. **Larsen, Rasmus.** (1994). *Estimation of visual motion in image sequences. xiv* + 143 pp.

2. **Rygard, Jens Moberg.** (1994). *Design and optimization of flexible manufacturing systems. xiii* + 232 pp.

3. **Lassen, Niels Christian Krieger.** (1994). *Automated determination of crystal orientations from electron backscattering patterns. xv* + 136 pp.

4. **Melgaard, Henrik.** (1994). *Identification of physical models. xvii* + 246 pp.

5. **Wang, Chunyan.** (1994). *Stochastic differential equations and a biological system. xxii* + 153 pp.

6. **Nielsen, Allan Aasbjerg.** (1994). *Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data. xxiv* + 213 pp.

7. **Ersbøll, Annette Kjær.** (1994). *On the spatial and temporal correlations in experimentation with agricultural applications. xviii* + 345 pp.

8. **Møller, Dorte.** (1994). *Methods for analysis and design of heterogeneous telecommunication networks.* Volume 1-2, *xxxviii* + 282 pp., 283-569 pp.

9. **Jensen, Jens Christian.** (1995). *Teoretiske og eksperimentelle dynamiske undersøgelser af jernbanekøretøjer. viii* + 174 pp.

10. **Kuhlmann, Lionel.** (1995). *On automatic visual inspection of reflective surfaces.* Volume 1, *xviii* + 220 pp., (Volume 2, *vi* + 54 pp., fortrolig).

11. **Lazarides, Nikolaos.** (1995). *Nonlinearity in superconductivity and Josephson Junctions. iv* + 154 pp.

12. **Rostgaard, Morten.** (1995). *Modelling, estimation and control of fast sampled dynamical systems. xiv* + 348 pp.

13. **Schultz, Nette.** (1995). *Segmentation and classification of biological objects. xiv* + 194 pp.

14. **Jørgensen, Michael Finn.** (1995). *Nonlinear Hamiltonian systems. xiv* + 120 pp.

15. **Balle, Susanne M.** (1995). *Distributed-memory matrix computations. iii* + 101 pp.

16. **Kohl, Niklas.** (1995). *Exact methods for time constrained routing and related scheduling problems. xviii* + 234 pp.

17. **Rogon, Thomas.** (1995). *Porous media: Analysis, reconstruction and percolation. xiv* + 165 pp.

18. **Andersen, Allan Theodor.** (1995). *Modelling of packet traffic with matrix analytic methods. xvi* + 242 pp.

19. **Hesthaven, Jan.** (1995). *Numerical studies of unsteady coherent structures and transport in two-dimensional flows.* Risø-R-835(EN) 203 pp.

20. **Slivsgaard, Eva Charlotte.** (l995). *On the interaction between wheels and rails in railway dynamics. viii* + 196 pp.

21. **Hartelius, Karsten.** (1996). *Analysis of irregularly distributed points. xvi* + 260 pp.

22. **Hansen, Anca Daniela.** (1996). *Predictive control and identification - Applications to steering dynamics. xviii* + 307 pp.

23. **Sadegh, Payman.** (1996). *Experiment design and optimization in complex systems. xiv* + 162 pp.

24. **Skands, Ulrik.** (1996). *Quantitative methods for the analysis of electron microscope images. xvi* + 198 pp.

25. **Bro-Nielsen, Morten.** (1996). *Medical image registration and surgery simulation. xxvii* + 274 pp.

26. **Bendtsen, Claus.** (1996). *Parallel numerical algorithms for the solution of systems of ordinary differential equations. viii* + 79 pp.

27. **Lauritsen, Morten Bach.** (1997). *Delta-domain predictive control and identification for control. xxii* + 292 pp.

28. **Bischoff, Svend.** (1997). *Modelling colliding-pulse mode-locked semiconductor lasers. xxii* + 217 pp.

29. **Arnbjerg-Nielsen, Karsten.** (1997). *Statistical analysis of urban hydrology with special emphasis on rainfall modelling.* Institut for Miljøteknik, DTU. *xiv* + 161 pp.

30. **Jacobsen, Judith L.** (1997). *Dynamic modelling of processes in rivers affected by precipitation runoff. xix* + 213 pp.

31. **Sommer, Helle Mølgaard.** (1997). *Variability in microbiological degradation experiments - Analysis and case study. xiv* + 211 pp.

32. **Ma, Xin.** (1997). *Adaptive extremum control and wind turbine control. xix* + 293 pp.

33. **Rasmussen, Kim Ørskov.** (1997). *Nonlinear and stochastic dynamics of coherent structures. x* + 215 pp.

34. **Hansen, Lars Henrik.** (1997). *Stochastic modelling of central heating systems. xxii* + 301 pp.

35. **Jørgensen, Claus.** (1997). *Driftsoptimering på kraftvarmesystemer.* 290 pp.

36. **Stauning, Ole.** (1997). *Automatic validation of numerical solutions. viii* + 116 pp.

37. **Pedersen, Morten With.** (1997). *Optimization of recurrent neural networks for time series modeling. x* + 322 pp.

38. **Thorsen, Rune.** (1997). *Restoration of hand function in tetraplegics using myoelectrically controlled functional electrical stimulation of the controlling muscle. x* + 154 pp. + Appendix.

39. **Rosholm, Anders.** (1997). *Statistical methods for segmentation and classification of images. xvi* + 183 pp.

40. **Petersen, Kim Tilgaard.** (1997). *Estimation of speech quality in telecommunication systems. x* + 259 pp.

41. **Jensen, Carsten Nordstrøm.** (1997). *Nonlinear systems with discrete and continuous elements.* 195 pp.

42. **Hansen, Peter S.K.** (1997). *Signal subspace methods for speech enhancement. x* + 226 pp.

43. **Nielsen, Ole Møller.** (1998). *Wavelets in scientific computing. xiv* + 232 pp.

44. **Kjems, Ulrik.** (1998). *Bayesian signal processing and interpretation of brain scans. iv* + 129 pp.

45. **Hansen, Michael Pilegaard.** (1998). *Metaheuristics for multiple objective combinatorial optimization. x* + 163 pp.

46. **Riis, Søren Kamaric.** (1998). *Hidden markov models and neural networks for speech recognition. x* + 223 pp.

47. **Mørch, Niels Jacob Sand.** (1998). *A multivariate approach to functional neuro modeling. xvi* + 147 pp.

48. **Frydendal, Ib.** (1998.) *Quality inspection of sugar beets using vision. iv* + 97 pp. + app.

49. **Lundin, Lars Kristian.** (1998). *Parallel computation of rotating flows. viii* + 106 pp.

50. **Borges, Pedro.** (1998). *Multicriteria planning and optimization. - Heuristic approaches. xiv* + 219 pp.

51. **Nielsen, Jakob Birkedal.** (1998). *New developments in the theory of wheel/rail contact mechanics. xviii* + 223 pp.

52. **Fog, Torben.** (1998). *Condition monitoring and fault diagnosis in marine diesel engines. xii* + 178 pp.

53. **Knudsen, Ole.** (1998). *Industrial vision. xii* + 129 pp.

54. **Andersen, Jens Strodl.** (1998). *Statistical analysis of biotests. - Applied to complex polluted samples. xx* + 207 pp.

55. **Philipsen, Peter Alshede.** (1998). *Reconstruction and restoration of PET images. vi* + 132 pp.

56. **Thygesen, Uffe Høgsbro.** (1998). *Robust performance and dissipation of stochastic control systems.* 185 pp.

57. **Hintz-Madsen, Mads.** (1998). *A probabilistic framework for classification of dermatoscopic images. xi* + 153 pp.

58. **Schramm-Nielsen, Karina.** (1998). *Environmental reference materials methods and case studies. xxvi* + 261 pp.

59. **Skyggebjerg, Ole.** (1999). *Acquisition and analysis of complex dynamic intra- and intercellular signaling events.* 83 pp.

60. **Jensen, Kåre Jean.** (1999). *Signal processing for distribution network monitoring. xv* + 199 pp.

61. **Folm-Hansen, Jørgen.** (1999). *On chromatic and geometrical calibration. xiv* + 238 pp.

62. **Larsen, Jesper.** (1999). *Parallelization of the vehicle routing problem with time windows.xx* + 266 pp.

63. **Clausen, Carl Balslev.** (1999). *Spatial solitons in quasi-phase matched struktures. vi* + (flere pag.)

64. **Kvist, Trine.** (1999). *Statistical modelling of fish stocks. xiv* + 173 pp.

65. **Andresen, Per Rønsholt.** (1999). *Surface-bounded growth modeling applied to human mandibles. xxii* + 125 pp.

66. **Sørensen, Per Settergren.** (1999). *Spatial distribution maps for benthic communities.*

67. **Andersen, Helle.** (1999). *Statistical models for standardized toxicity studies. viii* + (flere pag.)

68. **Andersen, Lars Nonboe.** (1999). *Signal processing in the dolphin sonar system. xii* + 214 pp.

69. **Bechmann, Henrik.** (1999). *Modelling of wastewater systems. xviii* + 161 pp.

70. **Nielsen, Henrik Aalborg.** (1999). *Parametric and non-parametric system modelling. xviii* + 209 pp.

71. **Gramkow, Claus.** (1999). *2D and 3D object measurement for control and quality assurance in the industry. xxvi* + 236 pp.

72. **Nielsen, Jan Nygaard.** (1999). *Stochastic modelling of dynamic systems. xvi* + 225 pp.

73. **Larsen, Allan.** (2000). *The dynamic vehicle routing problem. xvi* + 185 pp.

74. **Halkjær, Søren.** (2000). *Elastic wave propagation in anisotropic inhomogeneous materials. xiv* + 133 pp.

75. **Larsen, Theis Leth.** (2000). *Phosphorus diffusion in float zone silicon crystal growth. viii* + 119 pp.

76. **Dirscherl, Kai.** (2000). *Online correction of scanning probe microscopes with pixel accuracy.* 146 pp.

77. **Fisker, Rune.** (2000). *Making deformable template models operational. xx* + 217 pp.

78. **Hultberg, Tim Helge.** (2000).*Topics in computational linear optimization. xiv* + 194 pp.

79. **Andersen, Klaus Kaae.** (2001).*Stochastic modelling of energy systems. xiv* + 191 pp.

80. **Thyregod, Peter.** (2001). *Modelling and monitoring in injection molding. xvi* + 132 pp.

81. **Schjødt-Eriksen, Jens.** (2001). *Arresting of collapse in inhomogeneous and ultrafast Kerr media.*

82. **Bennetsen, Jens Christian.** (2000). *Numerical simulation of turbulent airflow in livestock buildings. xi* + 205 pp + Appendix.

83. **Højen-Sørensen, Pedro A.d.F.R.** (2001). *Approximating methods for intractable probabilistic models: - Applications in neuroscience. xi* + 104 pp + Appendix.

84. **Nielsen, Torben Skov.** (2001). *On-line prediction and control in nonlinear stochastic systems. xviii* + 242 pp.

85. **Öjelund, Henrik.** (2001). *Multivariate calibration of chemical sensors. xviii* + 182 pp.

86. **Adeler, Pernille Thorup.** (2001). *Hemodynamic simulation of the heart using a 2D model and MR data. xv* + 180 pp.

87. **Nielsen, Finn Årup.** (2001). *Neuroinformatics in functional neuroimaging.* 330 pp.

88. **Kidmose, Preben.** (2001). *Blind separation of heavy tail signals. viii* + 136 pp.

89. **Hilger, Klaus Baggesen.** (2001). *Exploratory analysis of multivariate data. xxiv* + 186 pp.

90. **Antonov, Anton.** (2001). *Object-oriented framework for large scale air pollution models.* 156 pp. + (flere pag).

91. **Poulsen, Mikael Zebbelin.** (2001). *Practical analysis of DAEs.* 130 pp.

92. **Keijzer, Maarten.** (2001). *Scientific discovery using genetic programming.*

93. **Sidaros, Karam.** (2002). *Slice profile effects in MR perfusion imaging using pulsed arterial spin labelling. xi* + 191 pp.

94. **Kolenda, Thomas.** (2002). *Adaptive tools in virtual environments – Independent component analysis for multimedia . xiii* + 112 pp.