Classification of Breast Lesions in Digital Mammograms

Master Thesis

547

M.R.M. Samulski

11th June 2006

University Medical Center Nijmegen
Department of Radiology
Supervisor: Dr. Ir. Nico Karssemeijer

Radboud University Nijmegen
Information and Knowledge Systems
Supervisors: Dr. Peter Lucas
Dr. Perry Groot

Table of contents

Abstract

Breast cancer is the most common life-threatening type of cancer affecting women in The Netherlands. About 10% of the Dutch women have to face breast cancer in their lifetime. The success of the treatment of breast cancer largely depends on the stage of a tumor at the time of detection. If the size of the invasive cancer is smaller than 20 mm and no metastases are found, chances of successful treatment are high. Therefore, early detection of breast cancer is essential. Although mammography screening is currently the most effective tool for early detection of breast cancer, up to one-fifth of women with invasive breast cancer have a mammogram that is interpreted as normal, i.e., a false-negative mammogram result. An important cause are interpretation errors, i.e., when a radiologist sees the cancer, but classify it as benign. In addition, the number of false-positive mammogram results is quite high, more than half of women who undergo a biopsy actually have breast cancer.

To overcome such limitations, Computer-Aided Diagnosis (CAD) systems for automatic classification of breast lesions as either benign or malignant are being developed. CAD systems help radiologists with the interpretation of lesions, such that they refer less women for further examination when they actually have benign lesions.

The dataset we used consists of mammographic features extracted by automated image processing algorithms from digitized mammograms of the Dutch screening programme. In this thesis we constructed several types of classifiers, i.e., Bayesian networks and support vector machines, for the task of computer-aided diagnosis of breast lesions. We evaluated the results with receiver operating characteristic (ROC) analysis to compare their classification performance. The overall conclusion is that support vector machines are still the method of choice if the aim is to maximize classification performance. Although Bayesian networks are not primarily designed for classification problems, they did not perform drastically lower. If new datasets are being constructed and more background knowledge becomes available, the advantages of Bayesian networks, i.e., incorporating domain knowledge and modeling dependencies, could play an important role in the future.

List of acronyms

BN	Bayesian network
CAD	computer aided detection
CC	cranio caudal
CPD	conditional probability distribution
DAG	directed acyclic graph
DCIS	ductal carcinoma in situ
EM	expectation-maximization
EWD	equal width discretization
EFD	equal frequency discretization
GMM	Gaussian mixture model
IC	inductive causation
LCIS	lobular carcinoma in situ
LDA	linear discriminant analysis
MCMC	Markov Chain Monte Carlo
MLO	medio-lateral oblique
MRI	magnetic resonance imaging
MWST	maximum weighted spanning tree
NDD	non-disjoint discretization
NN	neural network
PCA	principal component analysis
PDAG	partially directed acyclic graph
PKID	proportional k-interval discretization
ROC	receiver operating characteristic
ROI	region of interest
SVM	support vector machines
TDLU	terminal ductal lobular unit

Acknowledgements

There are a number of people who have, in one way or another, made it possible for me to write this thesis.

First and foremost, I would especially like to thank my supervisors, Dr. Peter Lucas, Dr. Perry Groot, and Dr. Ir. Nico Karssemeijer. This thesis may never have been completed without their support and participation in every step of the process.

I have to thank Peter Lucas for introducing me into the field of Bayesian networks and for the initiation of this thesis project. He has provided me with advice on which direction my project should take and his extensive knowledge about Bayesian networks has been essential to this thesis.

Also, this thesis undoubtedly benefited from Perry Groot's thorough comments and advice. During my thesis writing, he provided feedback on various draft versions of the thesis to make it as accurate as possible and more readable.

It was an honor to have the opportunity to work with Nico Karssemeijer, one of the finest people in the field of radiology. Besides creating the facilities that were needed to conduct this research, he never refused giving his time and advice when needed.

I am really indebted to Drs. Sheila Timp who helped me in the first four months of the project while finishing her PhD thesis. Her patience with my numerous questions and inexperience with scientific research has been invaluable. Especially her critical questions and careful listening helped me to understand and further investigate certain observations.

Also, I would like to thank Drs. Marcel van Gerven for steering me in the right direction in choosing the appropriate software for constructing Bayesian networks and pointing out interesting literature.

And not to forget, a lot of friends and acquaintances have helped to take my mind off work from time to time, I hope that we will be able to stay in touch. I would especially like to acknowledge the contribution of Christian Gilissen, not just for his insightful comments and endless support, but also for him being a reliable friend during all the years we have known each other.

Last, but certainly not least, I could not have completed this thesis without the help of my family. I want to thank my parents and my sister, for listening to my complaints and frustrations and for truly believing in me.

1 Introduction

Breast cancer is the most common life-threatening type of cancer affecting women in The Netherlands [Sta05].

About 10% of the approximately 8.25 million Dutch women have to face breast cancer. Every year there are around 11,000 newly diagnosed breast cancer patients. Men account for less than 1% of the diagnosed breast cancers. 25% of the newly diagnosed patients are detected by the breast cancer screening programme. The Dutch nationwide breast cancer screening program is offered to women aged 50-75 and about 76% of the women take part. The mammography screening takes place every 2 years. 100 of the 10,000 mammographically screened women are recalled for additional assessment. If further imaging confirms or reveals an abnormality, the woman may be referred for biopsy which happens 65 out of 100 times. Eventually 45 out of the 10,000 screened women have breast cancer [The03, WBMS03, OKH$^+$05]. This is schematically shown in Figure 1.1.

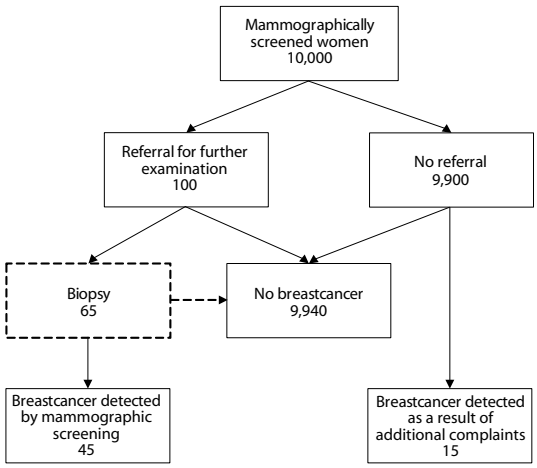


Figure 1.1: Dutch breast screening results per 10,000 women

Research [OKH$^+$05] shows that increasing the recall rate to 2% would increase the detection rate and result in about 260 extra tumors. To accomplish that result, an extra of 8500 women have to be examined.

Several studies have indicated that chances of successful treatment is high if the breast lesion can be detected at a size less than 2 cm, preferable even under 1 cm. Mammography screening, X-ray imaging of the breast, is currently the most effective tool for early detection of breast cancer.

1.1 Previous research

Machine learning techniques to diagnose breast cancer is a very active research area. Several Computer Aided Diagnosis (CAD) systems for automatic classification of breast lesions as either benign or malignant have been developed. Some of them are based on Bayesian networks learned on mammographic descriptions provided by radiologists [BRS00, KRSH97, KRW+95] or on features extracted by image processing [WZG+99, ZYHWG99, VRRL96]. Other classifying techniques that are used for the diagnosis of breast lesions are Support Vector Machines [Tim06, NAL+04, LDP04, MGD+04, BBB+00], Artificial Neural Networks [Tim06, MGD+04, AZC01, ZYHWG99, CDK99, DCBW95], Linear Classifiers [MGD+04, FWB+98] and Association Rule based classifiers [ZAC02]. Most of the computer aided diagnosis systems proved to be powerful tools that could assist radiologists in diagnosing a patient. In this thesis, we use two classification methods, namely Bayesian networks and support vector machines, and use techniques such as dimensionality reduction to improve the accuracy rate of the classifier. Recently, the combination of PCA and SVM has been used in medical imagery [LFK06, LCY06], where principal component analysis is applied to extracted image features and the results are used to train a SVM classifier, but not specifically for mammograms. To overcome the limitation of PCA that it can eliminate the dimension that is best for discriminating positive cases from negative cases, we also use a supervised dimension reduction technique FDA [DL88]. It is an extension of LDA [DHS01] such that we get more than only one optimal discriminating vector for using it as a dimensionality reduction technique rather than as a classifier. Also in previous research a combination of these two techniques is used [PSSM04, Joo03].

1.2 Purpose of the study

The aim of this project is to increase the quality and efficiency of computer aided diagnosis methods (CAD) used in breast cancer screening programs by means of Bayesian networks or classifiers and Support Vector Machines.

In order to achieve this goal we have set the following objectives:

Develop a novel classification technique using Bayesian networks or Bayesian classifiers

such that:

- The temporal pattern in the sequence of mammograms is captured by temporal classifiers
- The number of false positive detections is kept to a minimum
- It allows the handling of missing data and uncertainties
- The resulting classifiers are faithful with respect to the data i.e., the dependencies and independencies of the data are represented correctly
- Medical background knowledge of the breast cancer domain is incorporated

Compare the performance of the resulting Bayesian networks or classifiers with the existing technique within UMCN, Support Vector Machines.

2
Background

2.1 Breast anatomy

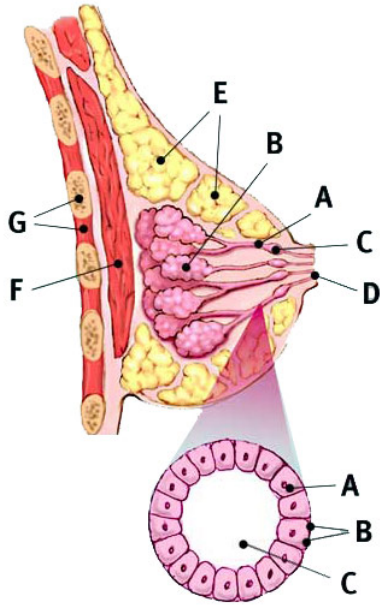
The anatomy of the breast is quite complex, Figure 2.1 shows the most important structures of the breast. To give an understanding of where and how different breast tumors may develop, we will shortly describe the structure of the breast. Each breast contains between 15 and 25 lobes that are connected to the nipple [D] through converging ducts [A]. Each lobe is made up of many smaller lobules [B]. Each lobule consists of 10 to 100 terminal duct lobular units (TDLU) where milk is produced. The most common area where breast cancer originates is in the TDLU.

2.2 Breast tumors

We can distinguish three types of breast tumors: benign breast tumors, in situ cancers, and invasive cancers.

2.2.1 Benign breast diseases

The majority of breast tumors detected by mammography are benign. They are non-cancerous growths and cannot spread outside of the breast to other organs. In some cases it is difficult to distinguish certain benign masses from malignant lesions with mammography.



	Breast profile:
A	ducts
B	lobules
C	dilated section of duct
D	nipple
E	fat
F	pectoralis major muscle
G	chest wall/rib cage

	Enlargement:
A	normal duct cells
B	basement membrane
C	lumen (center of duct)

Figure 2.1: Breast anatomy: image from www.breastcancer.org

2.2.2 In situ cancer

If the malignant cells have not gone through the basal membrane but is completely contained in the lobule or the ducts the cancer is called *in situ* or noninvasive. It does not spread to the surrounding tissues in the breast or other parts of the body. However, it can develop into a more serious invasive cancer. There are two forms of non-invasive breast cancer: ductal carcinoma in situ (DCIS) and lobular carcinoma in situ (LCIS). The location of breast carcinoma, a cancer that arises from tissue composed of a layer of cells, determines whether a lesion is classified as ductal or lobular. DCIS is often characterized in mammograms by the presence of micro calcifications. LCIS is more difficult to detect with mammography and is usually being discovered incidentally when taking a biopsy for another abnormality.

2.2.3 Invasive cancer

If the cancer has broken through the basal membrane and spread into the surrounding tissue it is called invasive. The chances on metastases (spreading of cancer from one part of the body to another) increase significantly. The success of the treatment of breast cancer largely depends on the stage of a tumor at the time of detection. There are two

features which determine the stage of a tumor: its size and whether metastases have been found in lymph nodes or distant areas. Invasive cancers vary in size from less than 10 mm to over 80 mm in diameter. If the size is smaller than 20 mm and if no metastases are found, chances of successful treatment are high. Therefore, early detection of breast cancer is essential.

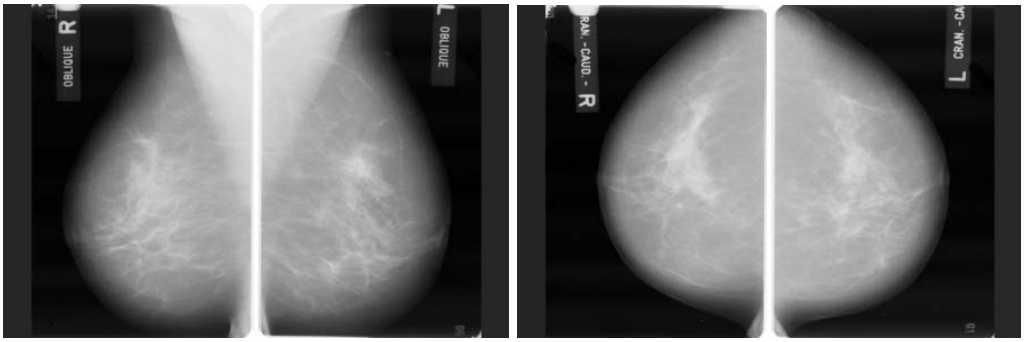
2.3 Breast cancer screening

Several researches [OFL+03, TYV+03, DTC+02] show that breast screening programs in many countries are an effective way to reduce mortality from breast cancer. The aim of breast cancer screening is to detect breast cancer as early as possible. Mammographic findings are, however, non-specific in some cases and some lesions may be indistinguishable from normal tissue.

2.4 Breast imaging modalities

2.4.1 Mammography

Mammography is the technique of choice to detect breast cancer and it is based on the difference in absorption of X-rays between the various tissue components of the breast such as fat, tumor tissue, and calcifications. If mammography is not sufficient, other techniques can be used such as ultrasonography and MRI. This project will focus on mammography only. Mammography has high sensitivity and specificity, even small tumors and micro calcifications can be detected on mammograms. The projection of the breast can be made from different angles. The two most common projections are medio-lateral oblique (side view taken at an angle) and cranio-caudal (top to bottom view), as shown in Figure 2.2. The advantage of the medio-lateral oblique projection is that almost the whole breast is visible, often including lymph nodes. Part of the pectoral muscle will be shown in upper part of the image. The cranio-caudal view is taken from above, resulting in an image that sometimes does not show the area close to the chest wall.



(a) Medio-lateral oblique (b) Cranio-caudal

Figure 2.2: The two most common projections of the breast

The two most important signs of breast cancer that can be seen on a mammogram are focal masses and micro calcifications. Other signs are architectural distortions and asymmetric breast tissue. In this project we are mainly interested in focal masses. When a mass is present in a breast, a radiologist will estimate its malignancy by looking at the appearance of the lesion and the surrounding tissue. The most important sign of malignancy is the presence of spiculation (spiky lines radiating in all directions from a central region extending into surrounding tissue). Also the borders of a mass may give additional information about the nature of the mass. Benign masses have sharp, circumscribed borders where malignant masses have slightly jagged or spiculated borders.

2.4.2 Other imaging modalities

Although mammography still remains the gold standard for breast cancer screening and diagnosis, it typically cannot differentiate benign from malignant tumors and is less accurate in patients with dense glandular breasts. Therefore other imaging modalities as Ultrasound and Magnetic Resonance Imaging can be used to further evaluate mammographic abnormalities in the breast or to distinguish between cystic and solid masses [Jac90]. It uses transmission of high frequency sound waves and the evaluation of returning sound to recognize abnormalities in the breast tissue. Because ultrasound has low sensitivity and specificity, it is not useful for screening.

Magnetic Resonance Imaging is able to differentiate between cancerous and noncancerous tissue because of differing water content and blood flow and can detect tumors missed by other modalities [GBC01]. For screening MRI is not a useful method, because of its low specificity and relatively high cost.

2.5 Computer aided detection

Although a lot of attention has been directed at technical quality assurance to guarantee optimal mammographic image quality, the quality of mammographic interpretation seems to be the weakest link in the process. Several review studies have revealed that observer errors are frequent in breast cancer screening [KOR+04]. Sometimes the radiologist is not aware of the abnormality or misinterprets the significance of an abnormality. It is estimated that 20% - 30% of the cancers could be detected in an earlier screening without an unacceptable increase in the recall rate (i.e., the rate at which mammographically screened women are recalled for additional assessment) [OKH+05, BWD+00].

Screening for breast cancer is a difficult task, especially due to the high number of normal cases: less than 1% of the screened women has breast cancer. To help radiologists in detecting signs of cancer, software has been developed for marking suspicious areas on mammograms that may indicate the presence of breast cancer. These systems act only as a second reader and the final decision is made by the radiologist. By using computer aided detection (CAD) software the number of errors might decrease, both false negatives (malignant cases that were not recalled) and false positives (cases that are recalled unnecessarily).

The most commonly used CAD systems detect mass lesions and micro calcifications by analyzing a single view of the breast. Most of the CAD programs have a two step procedure to accomplish this. The first step detects suspicious locations inside the breast area. In the second step the image at these locations is segmented into regions and several features are calculated for each region. These features are being used to determine whether a lesion is benign or malignant. They are also used to eliminate false positive detections.

More advanced CAD systems which currently are under development, are incorporating information from multiple views. They make use of multiple projections of the breast and/or views obtained from consecutive screening rounds for modeling the tumor behavior over time. Generally this results in better performance because sometimes a tumor can be seen on just one projection. Also using views obtained at different time moments can help to determine if a mass is benign or malign because benign masses tend to change slowly opposed to malignant masses which may change considerably. Such a CAD system has been developed at the UMCN and combines single view and temporal features into a single malignancy score using a Support Vector Machine classifier [Tim06].

2.6 Support vector machines

Support Vector Machines (SVMs) have been introduced by Cortes and Vapnik [CV95] for solving classification tasks and have been successfully applied in various areas of research. The basic idea of SVM is that it projects datapoints from a given two-class training set in a higher dimensional space and attempts to find a maximum-margin separating hyperplane between the data points of these two classes.

The training data for SVMs should be represented as labeled vectors in a high dimensional space where each vector is a set of features that describes one case. This representation is constructed to preserve as much information as possible about features needed for the correct classification of samples. Features in the case of breast tumor classification are characteristics such as size, shape, and contrast that are mapped to real numbers. The type of labels depends on the task. If the task of the SVM is to correctly predict benign versus malign tumors, labels can be chosen to be -1 for benign and $+1$ for malign.

In its simplest form, a SVM attempts to find a linear separator. In practice however, there may be no good linear separator of the data. In that case, SVMs can project the dataset to a significant higher dimensional feature space to make the separation easier, using a kernel function to produce separators that are non-linear.

More formally, using the notation from Burges [Bur98]: Let the datapoints of the dataset be vectors $x_1, ..., x_n$ that belong to the feature space $\mathbb{F} \subseteq \mathbb{R}^d$, associated with their labels $y_i \in \{-1, 1\}$, where $i = 1, \ldots, n$. Let Φ be a nonlinear function that maps a datapoint into a higher dimensional feature space $\mathbb{H}$:

$$\Phi : \mathbb{F} \mapsto \mathbb{H}$$

More specifically, $\mathbb{H}$ is a Hilbert space which is a real or complex vector space of infinite dimension with an inner product $\langle \cdot, \cdot \rangle$ such that $\mathbb{H}$ is complete with respect to the norm $|x| = \sqrt{\langle x, x \rangle}$. Completeness in this context means that every Cauchy sequence of elements in the space converges to an element in the space. A Cauchy sequence is an infinite sequence $x_1, x_2, x_3, \ldots$ such that for every real number $\epsilon > 0$ there is a positive integer N such that for integers $m, n > N$ one has that $|x_m - x_n| < \epsilon$. You can think of a Hilbert space as a generalization of Euclidean space that is complete, separable and infinite-dimensional. Instead of mapping our data via Φ and computing the inner product, we can do it in one operation, leaving the mapping completely implicit. Moreover, the kernel function is usually less computationally complex than the mapping function Φ which saves a lot of computation. In the literature this is known as the kernel trick [CV95]. It is called a trick because we do not need to know how the feature space really looks like, we just need the kernel function as a measure of similarity. The relationship between the kernel function K and the mapping Φ is defined as follows:

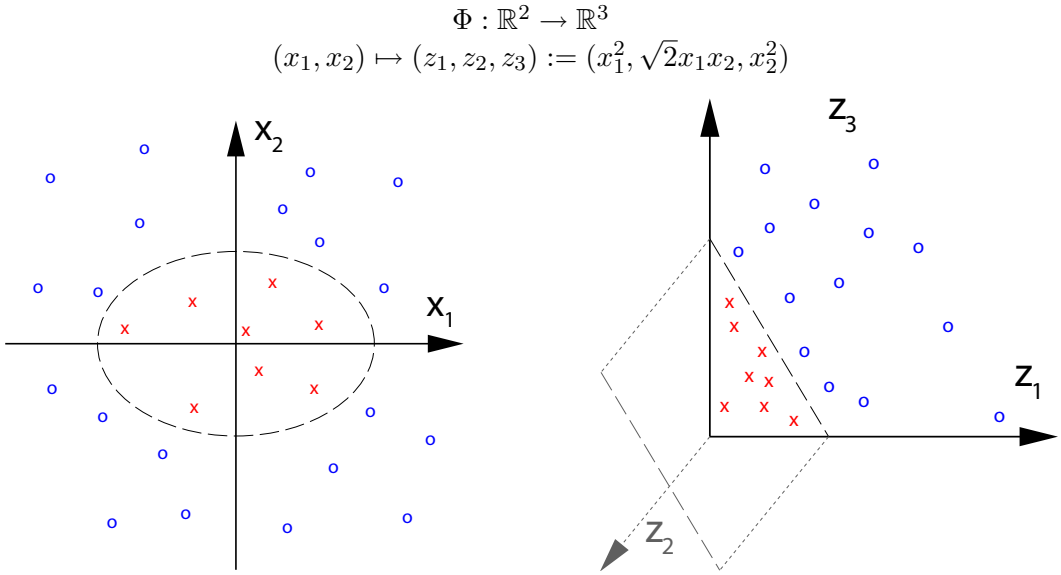


Figure 2.3: The data is elevated into a higher dimensional space by using a polynomial kernel function where the data can be discriminated with a hyperplane

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \tag{2.1}$$

In practice, we specify K, thereby specifying Φ indirectly, instead of choosing Φ. The value of $K(x_i, x_j)$ can be thought of the value of the inner product between x_i and x_j *after* they have been transformed into the higher dimensional feature space.

Although new kernels are constantly being developed by researchers, most SVM books introduce the following four basic kernels:

$$\begin{aligned}
\textit{Linear}: \quad & K(x_i, x_j) &&= \langle x_i, x_j \rangle \\
\textit{Polynomial}: \quad & K(x_i, x_j) &&= (\langle \gamma x_i, x_j \rangle + r)^d \quad \textit{where } \gamma > 0 \\
\textit{Radial Basis}: \quad & K(x_i, x_j) &&= e^{-\gamma |x_i - x_j|^2} \quad \textit{where } \gamma > 0 \\
\textit{Sigmoid}: \quad & K(x_i, x_j) &&= \tanh(\langle \gamma x_i, x_j \rangle + r)
\end{aligned}$$

Here, γ, r, and d are kernel parameters. The optimal value of the kernel parameters can be found using a parameter search which will be explained in Section 4.6.

For the linear kernel, the feature space is exactly the same as the input space. A small extension to the linear kernel is the polynomial kernel. If $d = \gamma = 1$ and $r = 0$ this

reduces to the linear kernel. Setting $d = 2$ results (nearly) in the $\mathbb{R}$ to $\mathbb{R}^2$ mapping $\Phi(x) = (x, x^2)$. The radial basis function was derived from the work in the neural networks community and the corresponding feature space is a Hilbert space of infinite dimension. It can be thought of as drawing 'balls' around the training vectors. One has to supply only one parameter, γ, to the radial basis kernel which is the size of these 'balls'.

Unfortunately there is no theory about deciding which kernel is the best [SS04, Era01], but a reasonable choice would be to first try a linear kernel and if that does not produce satisfying results, one could try the radial basis kernel. The radial basis kernel has only one parameter that needs to be set, unlike the polynomial kernel which has 3. Furthermore, the linear kernel is a special case of the radial kernel with some parameter γ [KL03]. Additionally, the sigmoid kernel behaves like the radial kernel for certain parameters [LL03].

With appropriate nonlinear mapping datapoints into the higher dimension space, and through use of such kernel functions, SVMs try to identify the optimal hyperplane that separates the two classes. For a specific projection of a dataset, there can be more than one separating hyperplane. The optimal one is the one that separates the data with the maximal margin in order to increase generalization to new data.

SVMs identify the datapoints near the optimal separating hyperplane which are called support vectors. The distance between the separating hyperplane and the nearest of the positive and negative datapoints is called the margin of the SVM classifier.

The separating hyperplane is defined as

$$D(x) = (w \cdot x) + b \tag{2.2}$$

where x is a vector of the dataset mapped to a high dimensional space, and w and b are parameters of the hyperplane that the SVM will estimate.

The nearest datapoints to the maximum margin hyperplane lie on the planes

$$(w \cdot x) + b = +1 \quad \textit{for } y = +1 \tag{2.3}$$

$$(w \cdot x) + b = -1 \quad \textit{for } y = -1 \tag{2.4}$$

Therefore, the width of the margin is given by $m = \frac{1}{||w||}$. Computing w and x is then the problem of finding the minimum of a function with the following constraints:

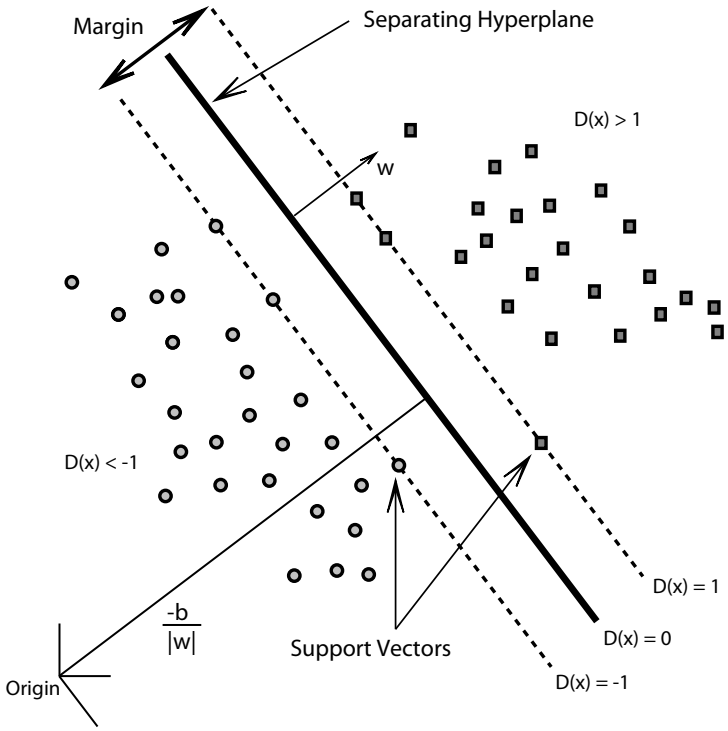


Figure 2.4: Linear separating hyperplanes for the separable case.

$$\textit{minimize} \qquad m(w) = \frac{1}{2}(w \cdot w) \tag{2.5}$$

$$\textit{subject to constraints} \qquad y_i[w \cdot x_i + b] \geq 1 \tag{2.6}$$

2.7 Bayesian networks

Bayesian networks are example of so-called probabilistic graphical models [Luc04a, LvdGAH04, Nea03, Pea88]. A bayesian network $\mathcal{B} = (G, \Theta)$ represents a joint probability distribution on a set of random variables X, which consists of two parts: (1) a qualitative part, represented as a directed acyclic graph (DAG) $G = (\mathcal{V}, \mathcal{A})$, with vertex set $\mathcal{V}$ which correspond to the random variables in X, and arc set $\mathcal{A}$ which represent the conditional dependencies between variables; (2) a quantitative part Θ which is a joint probability distribution defined on random variables X, where there is a one-to-one correspondence between the vertices in $\mathcal{V}$ and random variables in X. This is denoted by $X_{\mathcal{V}}$, where $X_V \in X_{\mathcal{V}}$ is the variable that corresponds to $V \in \mathcal{V}$.

2.7.1 Independence

Let $X_{\mathcal{A}}, X_{\mathcal{B}}, X_{\mathcal{C}} \subseteq X_{\mathcal{V}}$ be disjoint sets of random variables, and let P be a joint probability distribution defined on $X_{\mathcal{V}}$. If $P(X_{\mathcal{A}}|X_{\mathcal{B}}, X_{\mathcal{C}}) = P(X_{\mathcal{A}}|X_{\mathcal{B}})$, where $P(X_{\mathcal{B}}, X_{\mathcal{C}}) > 0$, then $X_{\mathcal{A}}$ and $X_{\mathcal{C}}$ are said to be conditionally independent given $X_{\mathcal{B}}$, which is denoted logically as

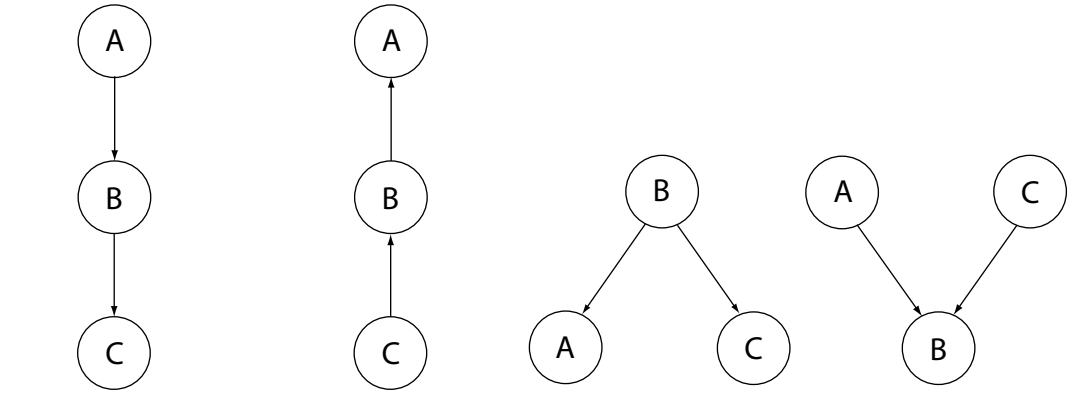
$$X_{\mathcal{A}} \perp\!\!\!\perp_P X_{\mathcal{C}} \mid X_{\mathcal{B}} \tag{2.7}$$

The independence relation can also be represented as a graphical model, where the arcs represent the dependencies, and absence of arcs represents the (conditional) independencies. Such graphical models can be understood in terms of subgraphs consisting of three vertices. There are four subgraphs of three vertices A, B, C possible when the direction of the arcs between A, B and B, C is unspecified and A and C are non-adjacent. These four possible subgraphs offer the basis for the representation of conditional dependence and independence in DAGs as illustrated in Figure 2.5. The common cause subgraph, shown in Figure 2.5(c), illustrates the situation where random variables A and C are initially dependent, but become independent once random variable B is instantiated.

$$A \not\perp\!\!\!\perp_G C \mid \emptyset \text{ and } A \perp\!\!\!\perp_G C \mid B \tag{2.8}$$

The two causal chain subgraphs shown in Figure 2.5(a) and Figure 2.5(b) represent exactly the same independence information: A and C are conditionally independent given B which means that given evidence on the value of B, additional evidence on the value of A does not longer influence the value of C and vice versa. The common effect subgraph represented in Figure 2.5(d), illustrates the situation where random variables A and C are initially independent, but become dependent once variable B is instantiated.

$$A \perp\!\!\!\perp_G C \mid \emptyset \text{ and } A \not\perp\!\!\!\perp_G C \mid B \tag{2.9}$$



(a) Causal chain 1 (b) Causal chain 2 (c) Common cause (d) Common effect

Figure 2.5: The four possible connections for acyclic directed graph $G = (\mathcal{V}, \mathcal{A})$ given vertices $A, B, C \in \mathcal{V}$ with arcs $(A \cdots B), (B \cdots C) \in \mathcal{A}$ where vertices A and C are non-adjacent.

The independence relation between a set of vertices can be determined with the d-separation procedure. Before giving the definition of d-separation, we have to define when a path between two vertices is blocked.

Definition 1 (*blocked*) Let $\mathcal{S} \subseteq \mathcal{V}$, and $A, B \in (\mathcal{V} \setminus \mathcal{S})$ be distinct vertices, which are connected to each other by the trail τ.[1] Then τ is said to be *blocked* by $\mathcal{S}$ if one of the following conditions is satisfied [FL04]:

- $K \in \mathcal{S}$ appears on trail τ, and the arcs of τ meeting at K constitute a causal chain or common cause connection;
- $K \notin \mathcal{S}$, none of K's descendants are in S, and the arcs meeting at K on trail τ constitute a common effect connection, i.e., if K appears on the trail τ then neither K nor any of its descendants occur in $\mathcal{S}$.

The notion of d-separation, where the 'd' stands for dependence, uses this notion of blocking taking into account that vertices can be connected by more than one trail [FL04]:

Definition 2 (*d-separation*) Let $G = (\mathcal{V}, \mathcal{A})$ be a directed acyclic graph, and let $\mathcal{A}, \mathcal{B}, \mathcal{S} \subseteq \mathcal{V}$ be disjoint sets of vertices. Then $\mathcal{A}$ and $\mathcal{B}$ are said to be *d-separated* by $\mathcal{S}$, denoted by $\mathcal{A} \perp\!\!\!\perp_G^d \mathcal{B} \mid \mathcal{S}$, if each trail τ in G between each $A \in \mathcal{A}$ and each $B \in \mathcal{B}$ is blocked by $\mathcal{S}$; otherwise, $\mathcal{A}$ and $\mathcal{B}$ are said to be *d-connected* by $\mathcal{S}$, denoted by $\mathcal{A} \not\perp\!\!\!\perp_G^d \mathcal{B} \mid \mathcal{S}$.

[1] A trail in a graph is a sequence of edges such that any two successive edges in the sequence share a vertex and where all edges and vertices are distinct.

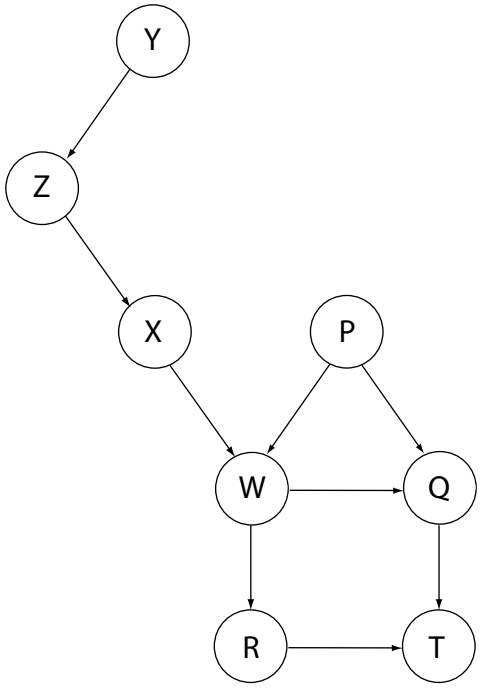


Figure 2.6: Schematic illustration of d-separation in a Bayesian network

We use an example taken from [FL04] to give a notion of d-separation. The vertices Z and P are connected by the following three trails:

- $\tau_1 = \textcircled{Z} \rightarrow \textcircled{X} \rightarrow \textcircled{W} \leftarrow \textcircled{P}$,
- $\tau_2 = \textcircled{Z} \rightarrow \textcircled{X} \rightarrow \textcircled{W} \rightarrow \textcircled{Q} \leftarrow \textcircled{P}$, and
- $\tau_3 = \textcircled{Z} \rightarrow \textcircled{X} \rightarrow \textcircled{W} \rightarrow \textcircled{R} \rightarrow \textcircled{T} \leftarrow \textcircled{Q} \leftarrow \textcircled{P}$.

The trail τ_1 is blocked by $\mathcal{S} = \{\text{X, Y}\}$ since Y does appear on this trail and the arcs on τ_1 meeting at X form a causal chain. Because X blocks τ_2 and τ_3, we conclude that $\mathcal{S}$ *d-separates* Z and P.

However, neither $\mathcal{S}' = \{\text{Y, W}\}$ nor $\mathcal{S}'' = \{\text{Y, T}\}$ block trail τ_1, because $\text{X} \rightarrow \text{W} \leftarrow \text{P}$ is a common effect connection, $\text{W} \in \mathcal{S}'$ and T is a descendent of vertex W which occurs in $\mathcal{S}''$; it also participates in a common effect connection with respect to τ_3. Therefore not every trail between Z and P in graph G is blocked by $\mathcal{S}'$ and $\mathcal{S}''$ which consequently means that Z and P are *d-connected* by $\mathcal{S}'$ or $\mathcal{S}''$.

It is not always the case that in a graphical model all independence information is represented, and it may also not be the case that all dependence information is represented.

Let $\perp\!\!\!\perp_P$ be an independence relation defined on $X_{\mathcal{V}}$ for joint probability distribution P, then for each $X_{\mathcal{A}}, X_{\mathcal{B}}, X_{\mathcal{C}} \subseteq X_{\mathcal{V}}$, where $X_{\mathcal{A}}, X_{\mathcal{B}}, X_{\mathcal{C}}$ are disjoint, we say that:

- G is an undirected dependence map, *D-map* for short, if
  $X_{\mathcal{A}} \perp\!\!\!\perp_P X_{\mathcal{B}} \mid X_{\mathcal{C}} \;\Rightarrow\; \mathcal{A} \perp\!\!\!\perp_G \mathcal{B} \mid \mathcal{C}$
- G is an undirected independence map, *I-map* for short, if
  $\mathcal{A} \perp\!\!\!\perp_G \mathcal{B} \mid \mathcal{C} \;\Rightarrow\; X_{\mathcal{A}} \perp\!\!\!\perp_P X_{\mathcal{B}} \mid X_{\mathcal{C}}$
- G is an undirected perfect map, *P-map* for short, if
  $\mathcal{A} \perp\!\!\!\perp_G \mathcal{B} \mid \mathcal{C} \;\Leftrightarrow\; X_{\mathcal{A}} \perp\!\!\!\perp_P X_{\mathcal{B}} \mid X_{\mathcal{C}}$

This means for example that in a D-map each independence encoded in the joint probability distribution P has to be represented in graph G. Also each dependence encoded by graph G has to represented in the joint probability distribution P, because it also holds that $X_{\mathcal{A}} \not\perp\!\!\!\perp_P X_{\mathcal{B}} \mid X_{\mathcal{C}} \;\Rightarrow\; \mathcal{A} \not\perp\!\!\!\perp_G \mathcal{B} \mid \mathcal{C}$ for D-maps.

In I-maps, each independence in graph G has to be consistent with the joint probability distribution P. Also each dependence relationship encoded in the joint probability distribution P has to be present in graph G. Clearly, a perfect map is just a combination of a D-map and I-map. By definition, Bayesian networks are directed I-maps. Since the complexity of conditional probability distributions is heavily dependent on the number of

parents a variable has, sparseness allows for a factorized and thus compact representation of a joint probability distribution.

As mentioned earlier, the set of arcs $\mathcal{A}$ describes the dependence and independence relationships between groups of vertices in $\mathcal{V}$ corresponding to random variables $X_{\mathcal{V}}$. If a joint probability distribution P admits a recursive factorization then P can be defined on the set of random variables $X_{\mathcal{V}}$ as follows:

$$P(X_{\mathcal{V}}) = \prod_{V \in \mathcal{V}} P(X_V | X_{\pi(V)}) \tag{2.10}$$

where $X_{\pi(V)}$ denotes the set of parents of X_V in graph G. Equation 2.10 implies that a joint probability distribution over a set of random variables can be defined in terms of local joint probability distributions $P(X_V|X_{\pi(V)})$.

We use the example [Hus04] shown in Figure 2.7 with the set of vertices $\mathcal{V} = \{A, B, C, D, E\}$ and the set of arcs $\mathcal{A} = \{(A, B), (A, C), (B, D), (C, D), (D, E)\}$ to explain the factorization rule 2.10.

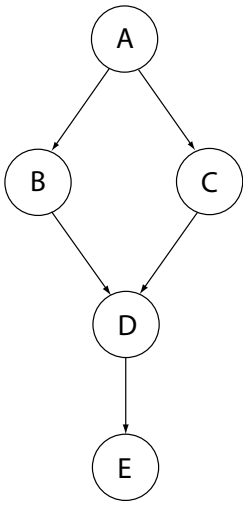


Figure 2.7: A simple Bayesian network

Vertex A does not have any parents, vertices B and C are children of vertex A, and the parent of vertex D. Vertex D has one child, vertex E. Applying Formula 2.10 will then lead to the following factorization

$$P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D|B, C)P(E|D)$$

Less formally, the arcs go from a parent node to a child node which intuitively indicates that the parent directly influences the child, and that these influences are quantified by conditional probabilities. To capture the joint probability distribution, one must specify a conditional probability distribution at each node in a Bayesian network. If the variables are discrete, this can be represented as a conditional probability table, which lists the probability that the child node takes on each of its different values for each combination of values of its parents.

2.7.2 Bayesian inference

To infer means to make a prediction based on knowledge and experience. Suppose we have a bag with thousand balls that are either red or blue, but we have no idea what percentage of the balls are which color. We are interested in how likely it is that we will pull a red ball out of the bag. In order to do that, we have to take a substantial sample

from the bag and count how many balls are red and how many balls are blue. We take hundred balls out of the bag, and come to the conclusion that 28% of the balls were red and 72% were blue. Without having to count all thousand balls, we can *infer* that 28% of the balls will be red. With Bayesian inference, we also can use prior knowledge. If we, for example, know from qualitative sources that 25% of the balls are red, we can incorporate that knowledge into the model.

When we have a Bayesian network with the associated conditional probability tables and observed nodes in the network (i.e., *feature* or *evidence* nodes), we want the ability to infer the probabilities of values for a certain node. This problem is NP-hard. However, there are several exact and approximate inference algorithms available to accomplish that task. If you have $P(X_{\mathcal{V}})$ then every probability can be calculated with the marginalization rule as follows:

$$\begin{aligned} P(X_{\mathcal{V}'}) &= \sum_{\mathcal{V}\setminus\mathcal{V}'} P(X_{\mathcal{V}}) \\ &= \sum_{\mathcal{V}\setminus\mathcal{V}'} \prod_{V\in\mathcal{V}} P(X_V|X_{\pi(V)}) \end{aligned} \tag{2.11}$$

2.7.3 Practical example

We continue by giving a very well known and more practical example from [LS88]. He introduced a fictitious expert system representing the diagnosis of a patient presenting to a chest clinic, having just come back from a trip to Asia and showing dyspnoea (shortness of breath). The doctor considers that possible causes are tuberculosis, lung cancer, and bronchitis, including the possibility that none of them or more than one of them is the cause for dyspnoea. Additional relevant information include whether the patient has recently visited Asia (where tuberculosis is more prevalent) and whether or not the patient is a smoker (which increases the chances of lung cancer and bronchitis). A positive X-ray would indicate either tuberculosis or lung cancer. A graphical model for the underlying process is shown in the Figure 2.8. Each node in a Bayesian network has an associated conditional probability table, of which one is shown partially to the left of node Dyspnoea.

If we learn the fact that a patient is a smoker, we will adjust our beliefs regarding lung cancer and bronchitis (i.e., the risks have increased). However, our beliefs regarding tuberculosis will be unchanged, because tuberculosis is conditionally independent of smoking given the empty set of variables. A positive X-ray result will affect our beliefs regarding tuberculosis and lung cancer, but not our beliefs regarding bronchitis (i.e., `bronchitis` is conditionally independent of `X-ray` given `smoking`). However, had we also known that the patient suffers from shortness-of-breath, the X-ray result would also have affected our beliefs regarding bronchitis (i.e., `bronchitis` is not conditionally

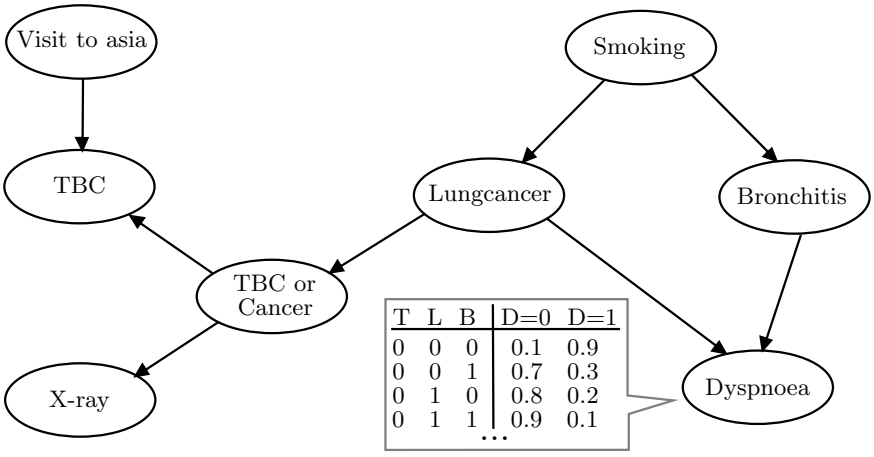


Figure 2.8: An example Bayesian network 'Asia'

independent of `X-ray` given `smoking` and `dyspnoea`).

2.7.4 Learning Bayesian networks

Many of the Bayesian networks developed in the medical environment have been constructed by hand, based on medical background knowledge. Much help is needed of medical experts to manually construct a Bayesian network and turns out to be very time consuming in practice. A lot of data has been collected and maintained in the breast screening programme. This data collection contains highly valuable information about the relationships between measured variables which can be used to learn the structure and the parameters of a Bayesian network. The quality of the learned Bayesian Network depends on the quality of the dataset because any bias introduced in the dataset will have impact on the resulting Bayesian network. To allow for reliable identification of independencies among the variables, a large amount of cases are needed in the dataset. To further increase the performance of the network, medical background knowledge of the breast cancer domain can be incorporated.

Learning a Bayesian network from data involves two steps: learning the graphical structure and learning the parameters [CBL97, LB94, Luc04a, Luc04b]. Structure learning algorithms are explained in Section 4.7.1. After we learned the structure of the Bayesian network, we have to determine the associated conditional probability distributions. An important distinction is whether all the variables are observed, or whether some of them are unavailable. If all the variables are observed the goal of learning with a Maximum Likelihood Estimator is to find the parameter values of each CPD which maximize the likelihood of the training data. The likelihood value is a measure of goodness, i.e.,

how well the distribution fits the observed data. Given a training set $D = \{d_1, \ldots, d_n\}$ where each d_i assigns values to all the variables $\{x_1, \ldots, x_k\}$ in X and a Bayesian network $\mathcal{B} = (G, \Theta)$. We further assume that the instances in the dataset are independent given Θ. This is

$$P(D|\Theta) = \prod_{i=1}^{n} P(d_i|\Theta) \tag{2.12}$$

and that the instances are identically distributed. We use the notation x_p^q to denote the variable x_p in instance d_q. The *log-likelihood* of Θ given D can be defined as

$$LL(\Theta|D) = \sum_{i=1}^{n} \log(P(d_i|\Theta)) \tag{2.13}$$

$$= \sum_{j=1}^{n} \sum_{i=1}^{k} \log P(x_i^j|\pi(x_i^j), \Theta_i) \tag{2.14}$$

where $\pi(x_i^j)$ are the parents of x_i in instance d_j. This criterion measures the likelihood that the dataset D was generated from the given model $\mathcal{B}$. The higher this value is, the closer $\mathcal{B}$ is to modeling the probability distribution in dataset D. The parameters that maximizes the log-likelihood for a given network structure can then be defined as

$$\Theta_{ML} = \arg\max_{\Theta} LL(\Theta|D) \tag{2.15}$$

If we have unobserved nodes or hidden nodes, we can rely on the expectation maximization algorithm (EM). In short, it calculates the expected value of the hidden node and uses that value for further calculations. Informally, the algorithm starts with randomly assigning values to all the parameters to be estimated. It then iteratively alternates between two steps: an expectation (E) step, and a maximization (M) step.

In the E-step, it computes the expected likelihood value for the complete data where the expectation is taken with respect to the estimated conditional distribution of the hidden variables given the most recent settings of the parameters and the observed data. In the M-step the parameters are updated by maximizing the expectation of the distribution obtained in the E-step.

We can repeatedly do the E-step and M-step until the likelihood converges, i.e., reaches a local maxima. It is proven that the distance between the real distribution and the estimated distribution decreases with every step. The whole expectation maximization (EM) procedure is explained in detail in [MK97].

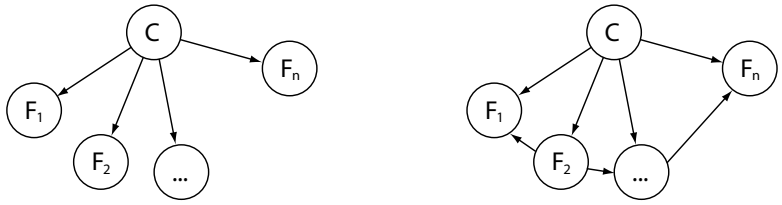


Figure 2.9: (a) Naïve Bayesian network and (b) tree-augmented Bayesian network

2.8 Bayesian classifiers

In this section, $P(X = x)$, where the uppercase X is a random variable and the lowercase x is the instantiation of that random variable, is abbreviated to $P(x)$. Using the learning methods explained in the previous section we can generate a Bayesian network $\mathcal{B}$, with an arbitrarily complex topology. We can then use the generated model in a way that given a set of features $\{f_1, f_2, \ldots, f_n\}$, the Bayesian network $\mathcal{B}$ returns label c that maximizes the posterior probability $P_{\mathcal{B}}(c|f_1, f_2, \ldots, f_n)$. It is important to note that the learning methods explained in the previous section do not distinguish the class variable from other attributes. The learning methods do not know we are evaluating the learned network on predictive performance of the class variable.

Although general Bayesian network structures may be used for classification tasks this may be computationally inefficient since the classification node is not explicitly identified and not all of the structure may be relevant for classification, since parts of the structure lie outside of the classification node's Markov blanket.

As described by Pearl [Pea88], a Markov blanket of a vertex M is the set of M's children, M's parents and the parents of the M's children in a given network structure G. In the example shown in Figure 2.7, the Markov blanket of vertex A is the set B, C, the Markov blanket of vertex B is A, C, D, the Markov blanket of vertex C is A, B, D, and so on. This set has the property that, conditioned on M's Markov blanket, M is independent of all other variables in the network.

So, if one does not care about the quality of the underlying probability distribution and only want to classify with Bayesian networks, often networks of limited topology are being used. These topologies are shown in Figure 2.9 where a distinction is made between *feature variables* f_i and a *class variable* c. Normally this kind of Bayesian networks have better classifying performance, because the quality of the network is only based on $P_{\mathcal{B}}(c|f_1, f_2, \ldots, f_n)$ (i.e., its predictive accuracy). Another reason why these networks are popular for classification is that learning naïve Bayesian classifiers (see Figure 2.9(a)) can be done in linear time which is far less computationally expensive

than learning complex Bayesian networks.

Given a set of feature variables $\{f_1, f_2, \ldots, f_n\}$, we construct the posterior probability for the event c.

Using Bayes' rule:

$$P(c|f_1, f_2, \ldots, f_n) = \frac{P(c)P(\bigwedge_{i=1}^{n} f_i|c)}{P(\bigwedge_{i=1}^{n} f_i)}$$

where $P(c|\bigwedge_{i=1}^{n} f_i)$ is the posterior probability that $\mathcal{F}$ belongs to c. The denominator which is the marginal probability of $\bigwedge_{i=1}^{n} f_i$ can be defined as

$$P(\bigwedge_{i=1}^{n} f_i) = \sum_{j=1}^{k} P(\bigwedge_{i=1}^{n} f_i|c_j)P(c_j)$$

Alternatively, $P(\bigwedge_{i=1}^{n} f_i)$ can be seen as a normalizing constant α which can be calculated, realizing that

$$\sum_{j=1}^{k} P(c_j|\bigwedge_{i=1}^{n} f_i) = \sum_{j=1}^{k} \alpha P(c_j)P(\bigwedge_{i=1}^{n} f_i|c_j) = 1$$

where $P(c_j)$ and $P(\bigwedge_{i=1}^{n} f_i)$ is known.

To dramatically simplify the classification task we can use the following simplifying assumption: each feature f_i is conditionally independent of every other feature f_j for $i \neq j$. This fairly strong assumption of independence leads to the name naïve Bayes, with the assumption often being naïve in that, by making this assumption, the algorithm does not take into account dependencies that may exist. Two events A and B are said to be independent if the occurrence of event A makes it neither more probable nor less probable that event B occurs and vice versa. When A and B are independent, learning the value of B gives us no information about A and vice versa. Formally, we can denote this as $P(A|B) = P(A)$ and $P(B|A) = P(B)$.

By using the conditionally independence assumptions we can express the joint probability model as

$$P(\bigwedge_{i=1}^{n} f_i, c) = P(c)\prod_{i=1}^{n} P(f_i|c)$$

The model in this form is much more manageable, since it factors into a so-called *class prior probability* $P(c)$ and independent probability distributions $P(f_i|c)$. These class conditional probabilities $P(f_i|c)$ can be calculated separately for each variable which reduces complexity enormously.

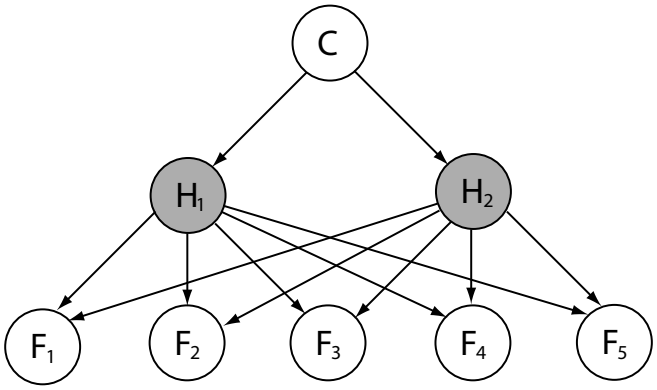


Figure 2.10: Schematic illustration of a latent classification model with 5 features and 2 latent variables.

Classification using this Bayes' probability model is done by picking the most probable hypothesis which is also known as the *maximum a posteriori*. The corresponding classifier function can be defined as follows:

$$\texttt{classify}(f_1, f_2, \ldots, f_n) = \arg\max_c \; P(c|f_1, f_2, \ldots, f_n)$$

Even with such strong simplifying assumptions, it does not seem to greatly affect the posterior probabilities, especially in regions near the decision boundaries which leaves the classification task unaffected. [DP97] shows that such naïve Bayesian classifiers yield surprisingly powerful classifiers. An extension to the naïve Bayes model is the tree-augmented naïve Bayes model where each feature node can have one correlation edge pointing to it as shown in Figure 2.9(b). [FGG97] shows that this network again could outperform a naïve Bayesian network (see Figure 2.9(a)). There are also other extensions possible, one of them is a forest-augmented Bayesian network (FAN) where arcs are allowed between feature variables as long as they form a forest of trees [Luc04b]. Recently, a new set of models for classification have been introduced termed latent classification models. They can be seen as a combination of the naïve Bayes model with latent (i.e., hidden) variables that encode the conditional dependencies among features [SSGS06, LN05]. Standard algorithms such as structural EM [Fri98] can be used to discover the structure of a latent model and the parameters of the latent variables can be learned by the EM algorithm. Other, more sophisticated latent models integrate a mixture of factor analyzers into the naïve Bayes model to relax the conditional independence assumptions of the original naïve Bayes model. A graphical example of a latent classification model is shown in Figure 2.10.

Although the variables in a Bayesian network are often assumed to be discrete, a network

may also include continuous variables that adopt a value from a range of real values [Lau92, Ole93]. Often, the conditional probability distributions for such continuous variables are assumed to be Gaussian, or normal, distributions. These distributions then are specified in terms of a limited number of parameters, such as their means and variances. Unfortunately, many real world features are not normal distributed and therefore we have to transform such variables to an approximately normal distribution when possible.

2.9 Bias-variance decomposition

If a model is constructed by a learning method using a sample taken from a given domain, and the model is being used to make predictions then some predictions are false. Bias-variance decomposition [DKS95] is an useful method for the analysis of learning problems because it distinguishes between different kind of prediction errors:

1. the bias error, a systematic component in the error associated with the learning method and the domain
2. the variance error, a component associated with differences in models between samples
3. an intrinsic error component associated with the inherent uncertainty in the domain

If the bias is high, the model is underfitting the data which means that it is not complex enough to capture the underlying structure of the data. It is known that naïve Bayes can underfit the data when using it for highly complex datasets. High variance error indicates varying, unstable predictions and is associated with overfitting. If a classification method overfits the data, the predictions for a single instance will vary between samples. This is a serious problem of support vector machines and Bayesian networks and occurs when the models describe the instances in the training set better and better but get worse and worse on new instances of the same phenomenon, i.e., the model will fit the noise in the training data which means poor generalization to new data. This can render the whole learning process worthless.

Each type of error requires a different strategy of error reduction. To reduce bias one could increase the representational power of the learning algorithm. Using a smaller fraction of the training data can decrease the variance error [DK95].

One of the simplest and most widely used means of avoiding overfitting and thus decreasing the variance error is to divide the data into two sets: a training set and a test

set. In order to avoid wasting data and to eliminate the possibility that the test set with randomly chosen instances could be just lucky (e.g. contain much 'easy' instances), we use the cross-validation technique explained in the next section.

Note that there is often a *bias-variance tradeoff*. Usually if one increases the number of degrees of freedom in the learning algorithm, the bias shrinks but the variance increases which leads to overfitting. The optimal number of degrees of freedom is then the number of degrees of freedom that optimizes this trade off between bias and variance.

2.10 Cross Validation

The performance of the developed support vector machines and Bayesian networks will be measured by using cross-validation where a set of available feature measurements and output classifier is divided into two parts: one part for training and one part for testing. In this way several different Bayesian networks, all trained on the training set, can be compared on the test set. The basic steps of cross-validating are as follows:

- Divide the data into N sets
- Make N Bayesian networks, each one trained on $N-1$ of the sets
- Test the Bayesian network on the remaining set

This is called N-fold cross validation. A schematic illustration of a 5-fold cross-validation is given in Figure 2.11. The idea behind it is that averaging the test error of all N Bayesian networks will give a good estimate of the true error on any randomly chosen Bayesian network [Koh95].

One of the clear advantage of cross-validation is that all data is used for training and testing, but the disadvantage is that it takes much computational work to make so many networks. An extreme variation of cross validation is the leave-one-out method, where one case is taken out for testing and the rest of the data is used to learn the Bayesian network. However, one of the dangers of this leave-one-out method is the chance of overfitting because one uses a very large amount of cases to train the network. After the cross-validation one selects the best performing network and restarts the learning process.

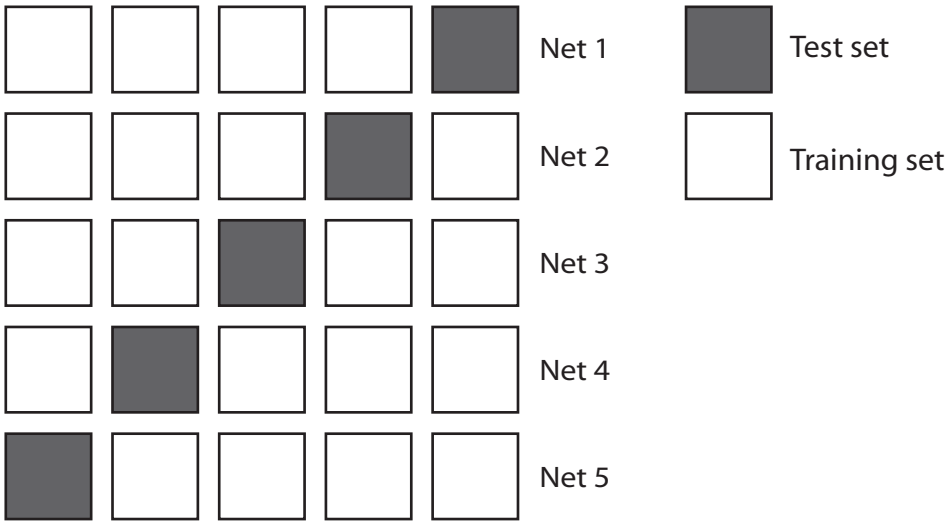


Figure 2.11: Schematic illustration of a 5-fold cross-validation

2.11 ROC Analysis

To evaluate the constructed systems, the classification performance of each system has to be measured. Often the performance of a system cannot be described by a single value. A good example of this is given by Gilbert [Gil84] 100 years ago, when he explained the exceptionally high "accuracy" a fellow meteorologist claimed in the prediction of tornados. He pointed out that because the actual frequency of tornados was so low, this high accuracy could be achieved by simply saying that there is no tornado each day. Therefore it is crucial to describe the performance by two or more values. Often these values are complementary, which means that if one value is being optimized the other one will become worse. In a Receiver Operator Characteristics (ROC) curve the sensitivity, which in this study is the share of malign tumors that is correctly classified, is plotted against 1-specificity, the share of benign tumors that is falsely classified, for different cut values.

Often the ROC analysis is used to find an optimal cut value, sometimes referred to as criterion, for use in decision-making. By changing the cut value of the system it is possible to achieve the optimal balance between sensitivity and specificity that is needed for a certain purpose. If the cost of not detecting a particular disease is very high to society, for example a highly contagious disease, one could change the cut value to achieve a very high sensitivity, but consequently lower specificity.

This technique is now widely used in the field of biomedical research and has become a

	Tumor marked as malign	Tumor marked as **not** malign
Tumor classified as malign	True Positives (TP)	False Positives (FP)
Tumor classified as **not** malign	False Negatives (FN)	True Negatives (TN)

Table 2.1: Relationship between TP, TN, FP, and FN

golden standard in performance measuring.

The following four values are calculated when comparing the classifier output of the constructed systems with the real labels that were determined by biopsy:

- True Positives (TP): Tumors marked as malign which were also classified as tumor.
- True Negatives (TN): Tumors which were not marked as malign, and that were also not classified as malign.
- False Positives (FP): Tumors which were not marked as tumor, but were classified as tumor.
- False Negatives (FN): Tumors which were marked as tumor, but which were not classified as tumor.

The relationship between these four values is shown in Table 2.1.

Based on these four values, relative measurements can be calculated:

Sensitivity is the *ratio* of tumors which were marked and classified as tumor, to all marked tumors

$$SE = \frac{TP}{TP + FN} \tag{2.16}$$

Specificity is the *ratio* of tumors which were not marked and also not classified as tumor, to all unmarked tumors

$$SP = \frac{TN}{FP + TN} \tag{2.17}$$

The total area under the ROC-curve, often referred to as the A_z value, is a measure of the classification performance since it reflects the test performance at all possible cut-off levels. The area lies in the interval $[0.5, 1]$ and the larger this area, the better the performance of the classification. In this work the A_z value will be used to compare the results of the SVM classifiers and Bayesian classifiers.

In experiments, there is usually only a finite set of points on the ROC-curve. Therefore it is only possible to find a good approximation of the area under the curve. It is obvious that the more points there are, the better estimate of the curve and area we get. There

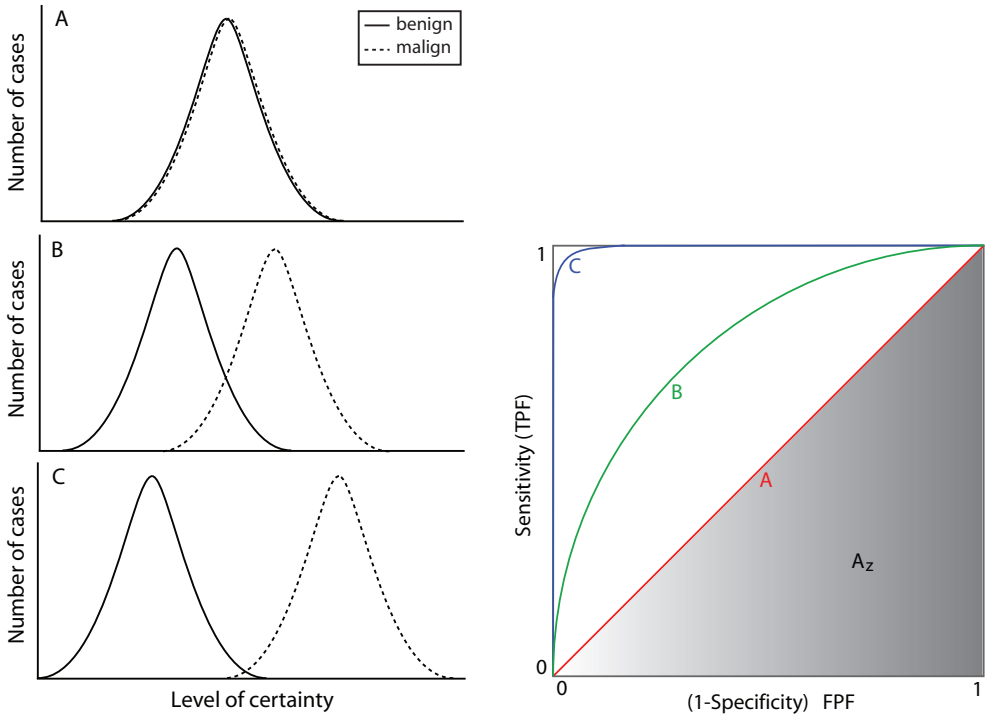


Figure 2.12: (a) Likelihood of a tumor being benign relative to malign and (b) their ROC curves

are several ways to calculate the area under a ROC curve. First, the trapezoidal rule can be used but gives an underestimation of the area. Second, it is possible to get a better approximation of the curve by fitting the data to a binormal model using curve-fitting software with maximum-likelihood estimates. After that it is possible to get a good estimate of the area. Because we have a considerable amount of points we will use the first method to estimate the area under the curve.

The two normal distributions in Figure 2.12(a) show the benign tumor and the malign tumor distributions. The horizontal axis represents the level of certainty that the tumor is malign. When a system has difficulty detecting whether a tumor is benign or malign, the two distributions will overlap considerably, see curve A in Figure 2.12(b). The A_z value of curve A is 0.5 which is the worst performance one can get. Curve C has the smallest overlap which results in a near perfect performance with an A_z value of almost 1.0.

3
Dataset

In the UMCN there are huge quantities of clinical data available. The digitized mammograms that are going to be used in this project have been obtained from the Dutch Breast Cancer Screening Program. In this program two mammographic views of each breast were obtained in the initial screening: the medio-lateral oblique (MLO) view, which is a side view taken at an angle, and a cranio caudal (CC) view, which is a top to bottom view. At subsequent screenings only a MLO was obtained, unless there was an indication that CC views could be beneficial. In Figure 3.1 we summarize the information about the dataset. In total we had 536 cases, 238 benign and 227 malignant. In about one half (265) of the cases there was only a MLO or CC view available. In the other half (271) there was both a MLO and CC view available. In some cases however the mass was not visible on the CC view. Reasons include location near the chest, obscuration of the mass lesion due to dense tissue and very subtle lesions. In our experiments, we will only use the MLO/CC pairs including the ones where the mass was not visible on the CC view.

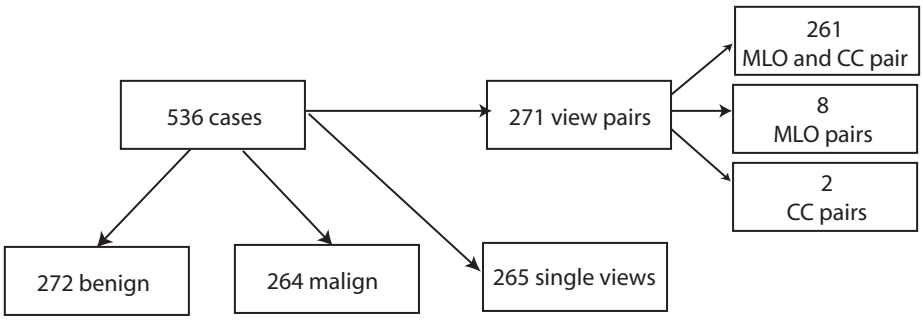


Figure 3.1: Description of the dataset

3.1 Shape and texture based features

For every digitized mammogram, a certain number of suspected regions have been indicated. For every region, specific features such as size, shape, and spiculation have been calculated. In total there are 81 different features calculated for each region. In this project only a subset of 12 features out of 81 features is being used. Each of these 12 features is being described in the following sections.

3.1.1 Stellate patterns

Malignancies tend to have a greater density than that of normal breast tissue. Generally, malignant mammographic densities are often surrounded by a radiating pattern of linear spicules.

For the detection of these stellate patterns of straight lines directed toward the center pixel of a lesion, two features have been designed by Karssemeijer and te Brake [KtB96]. The idea is that if an increase of pixels pointing to a given region is found then this region may be suspicious, especially if, viewed from the that region, such an increase is found in many directions.

The first feature $f1$ is a normalized measure for the fraction of pixels with a line orientation directed towards the center pixel. We call this set of pixels F. For calculating the second feature $f2$ the circular neighborhood is divided into 24 angular sections. This feature measures to what extent the pixels in set F are uniformly distributed among all angular sections. Also the mean values of $f1$ and $f2$ inside the region are included in the subset.

3.1.2 Region Size

Most breast tumors are about 2 cm^2 in size. Regions with a similar size are more likely to represent mass lesions than regions with a much smaller or larger size. This feature captures this difference.

3.1.3 Compactness

Compactness represents the roughness of an object's boundary relative to its area. Compactness (C) is defined as the ratio of the squared perimeter (P) to the area (A), i.e.,

$$C = \frac{P^2}{A}$$

The smallest value of compactness is $C = \frac{(2\pi r)^2}{\pi r^2} = 4\pi = 12.5664$ which is for a circle. As the circle deviates towards a more complicated shape, the compactness becomes larger.

In our dataset this feature is normalized by dividing the compactness by 4π, which results in the following simple formula:

$$C' = \frac{P^2}{4\pi A}$$

3.1.4 Linear Texture

Normal breast tissue often has different texture characteristics than tumour tissue. Therefore Karssemeijer and te Brake [KtB96] have developed a texture feature that tries to find linear structures inside the segmented area because they often indicate the presence of normal breast tissue.

3.1.5 Relative Location

The relative location of a lesion is important since most malignancies (45%) develop in the upper outer quadrant [CAAB98] of the breast toward the armpit. Therefore some features have been constructed that represent the relative location of a lesion using a new coordinate system [VTK06] (see Figure 3.2). This new coordinate system is different for MLO and CC views. In MLO views the pectoral edge is used as the y-axis. The x-axis is determined by drawing a perpendicular line on the y-axis where the distance between the y-axis and the breast boundary is maximum. We assume that at the end of this line the nipple is located. In CC views the chest wall is used as y-axis. A point is selected on the breast boundary that is most distant to the chest wall. We assume that the nipple is located at this point. Then a perpendicular line to the y-axis which passes through the nipple is defined as x-axis. In this new coordinate system we calculate the x- and y-location of the selected peak and normalize with the effective radius of the breast $r = \sqrt{\frac{A}{\pi}}$, where A is the size of the segmented breast area to allow the known positions of the cancers on the mammograms to be compared.

3.1.6 Maximum Second Order Derivate Correlation

This border feature indicates the smoothness of the contour and is especially useful to discriminate between benign and malignant lesions. Most benign lesions have a well-defined contour and the margins of these lesions are sharply confined with a sharp

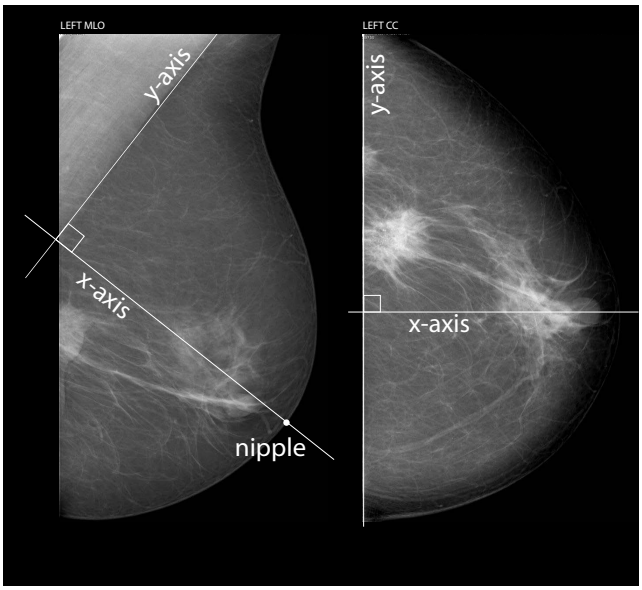


Figure 3.2: New coordinate system

transition between the lesion and the surrounding tissue which indicates that there is no infiltration [VTK06].

3.1.7 Contrast

Regions with high contrast or a higher intensity than other similar structures in the image is likely to be a mass. According to te Brake [tB00] it is an useful feature to remove false positive signals.

3.1.8 Number of Calcifications

The presence of clustered micro calcifications is one of the most important signs of cancer on a mammogram and occur in about 90% of the non-invasive cancers, see also Section 2.2.2. Therefore this feature represents the number of calcifications.

3.2 Statistical analysis

For every feature global statistics have been calculated which can be seen in Table 3.1. Besides the fundamental statistical characteristics mean and standard deviation, two other characteristics have been calculated: skewness and kurtosis. Skewness is a measure of the lack of symmetry. A data set is symmetric if it looks the same to the left and right of the center point. The skewness for a normal distribution would be zero, and any symmetric data should have a skewness near zero. Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. The kurtosis for a standard normal distribution is three.

	Mean	Std dev	Min	Max	Skewness	Kurtosis
Benign (cases: 258)						
Stellate Patterns 1	1.1256	0.1710	0.7800	2.1400	2.3002	13.4307
Stellate Patterns 2	1.0241	0.1160	0.8300	2.1900	4.7815	44.8670
Stellate Patterns 1 Mean	1.1189	0.1316	0.8600	1.5630	0.8565	3.6986
Stellate Patterns 2 Mean	1.0215	0.0713	0.8380	1.2990	0.5482	3.6256
Region Size	0.4070	0.3915	0.0200	3.4510	3.0272	17.9799
Contrast	0.5502	0.2558	0.1260	2.0110	1.9986	9.8575
Compactness	1.2141	0.0906	1.0470	1.5600	0.9308	3.8448
Linear Texture	0.1750	0.1444	0.0130	1.0240	2.2365	10.1391
Relative Location X	0.6705	0.3024	-0.0670	1.5470	0.0470	2.7819
Relative Location Y	0.2160	0.4262	-0.9680	1.2990	-0.2289	2.4769
Max. 2nd order Drv Corr.	0.6800	0.1008	0.4520	0.9060	0.0436	2.3011
Number of Calcifications	0.7871	2.6723	0.0000	19.0000	3.8831	19.2635
Malignant (cases: 274)						
Stellate Patterns 1	1.2273	0.1730	0.8200	1.7300	0.5060	3.0005
Stellate Patterns 2	1.0827	0.0965	0.7900	1.3500	0.1468	2.8634
Stellate Patterns 1 Mean	1.2357	0.1736	0.8290	1.7740	0.6844	3.1281
Stellate Patterns 2 Mean	1.0868	0.0946	0.8530	1.4140	0.4533	3.0175
Region Size	0.4471	0.3272	0.0160	1.8040	1.2728	4.4259
Contrast	0.6272	0.2777	0.0110	1.5090	0.7688	3.2074
Compactness	1.2111	0.0983	1.0410	1.7080	1.5022	6.3482
Linear Texture	0.1578	0.1161	0.0040	0.9490	2.2258	11.5829
Relative Location X	0.6130	0.3046	-0.0710	1.3080	0.0140	2.3298
Relative Location Y	0.2080	0.4449	-0.9770	1.2180	-0.2483	2.7594
Max. 2nd order Drv Corr.	0.6354	0.0951	0.4040	0.9320	0.1608	2.9336
Number of Calcifications	2.0645	6.7471	0.0000	50.0000	4.4524	25.7707

Table 3.1: Statistics of benign and malign cases in the dataset

4 Methods

4.1 Equipment and Software

The Bayesian inference and learning algorithms described in this report were implemented in Matlab version 7.1 (Mathworks Inc) using the functions of the open-source Bayes Net Toolbox (BNT) written by Kevin Murphy [Mur01]. Additionally, functions from the BNT Structure Learning Package from Philippe Leray [Ler04] were used to extend BNT's structure learning functionality. Some of these algorithms were modified and extended by the author of this report. The support vector machine experiments were implemented in R 2.2.0 [GI05], a free software environment for statistical computing and graphics which is similar to the S language and environment which was developed at Bell Laboratories. The test runs were performed on a Athlon64 2.2 GHz machine with operating system Windows XP equipped with 1,5 GB RAM.

4.2 Preprocessing

Many Bayesian learning algorithms that deal with continuous nodes are based on the assumption that the features are gaussian distributed. Unfortunately, some of the features do not follow a normal distribution, as can be seen in Table 3.1. A strategy to make non-normal data resemble normal data is by using appropriate transformations. We will follow the commonly used two-stage transformation scheme introduced by Harris and DeMets [HD72]: first remove skewness, then adjust for remaining non-gaussian kurtosis. One particularly useful transformation algorithm to remove skewness is the Box-Cox power transformation [BC64].

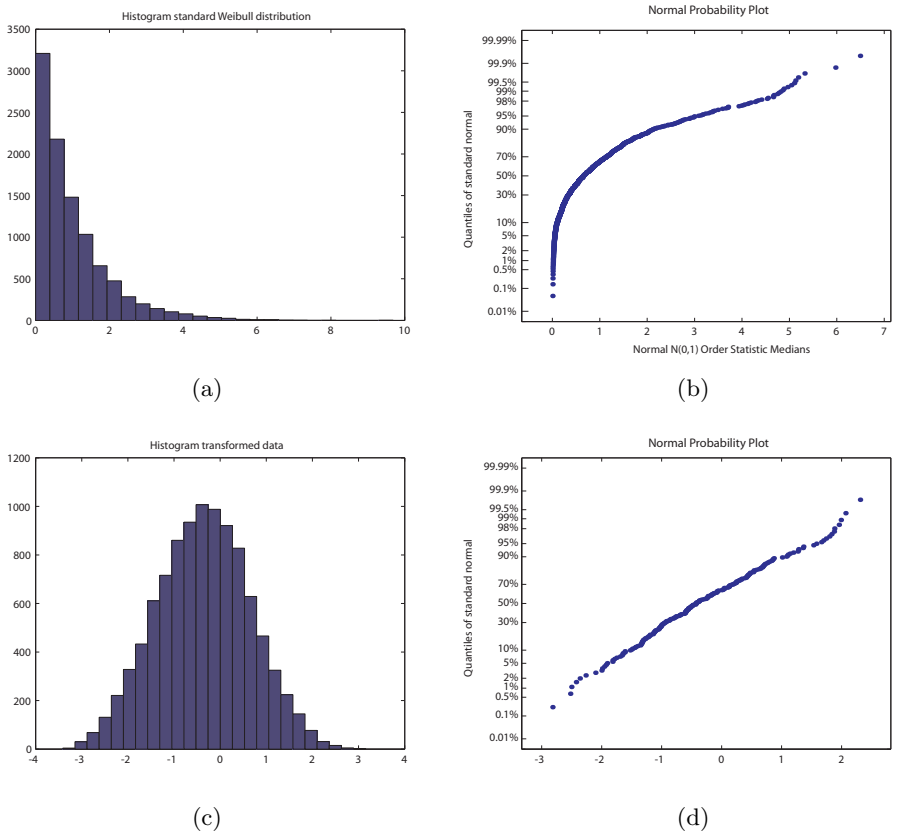


Figure 4.1: An example Box-Cox transformation: (a) histogram of a feature that is Weibull distributed, (b) normality plot of the feature, (c) histogram of the transformed feature, and (d) normality plot of the transformed feature

4.2.1 Box-Cox transformation

The Box-Cox power transformation is a transformation from y to $y^{(\lambda)}$ with parameter λ and especially works if the probability distribution of a feature can be described as a function which contains powers, logarithms, or exponentials:

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln y & \text{if } \lambda = 0 \end{cases} \tag{4.1}$$

The assumption made by this transformation is that $y^{(\lambda)}$ follows a normal linear model with parameters β and σ^2 for some value of λ.

Notice that this transformation is essentially y^λ for $\lambda \neq 0$ and $\ln y$ for $\lambda = 0$, but has been scaled to be continuous at $\lambda = 0$. Useful values of λ can often be found in the range $[-2, 2]$. If we do not consider the scaling factors, -1 is the complement, 0 is the logarithm, 0.5 is the square root, 1 is the identity and 2 is the square.

Given a value of λ, we can estimate the linear model parameters β and σ^2 as usual, except that we work with the transformed variable $y^{(\lambda)}$ instead of y. To select an appropriate transformation we need to try values of λ in a suitable range. We did this by using the Box-Cox normality plot which underlying technique is based on the normal probability plot. The normal probability plot is a graphical technique to determine whether data is approximately normally distributed. The data is plotted against a theoretical normal distribution in such a way that the points should form an approximate straight line as shown in Figure 4.1(d). Deviations of this straight line means that the data is less normally distributed as shown in Figure 4.1(b). In the Box-Cox normality plot we use that property: the correlation coefficient of the normality plot is plotted against a range of λ's. The lambda resulting in the largest correlation coefficient is the optimal one, in Figure 4.2 the optimal λ is 0.2726. Instead of trying each λ in a certain range, we use the well-known *divide and conquer* technique [Man89] to search much more efficiently through the search space.

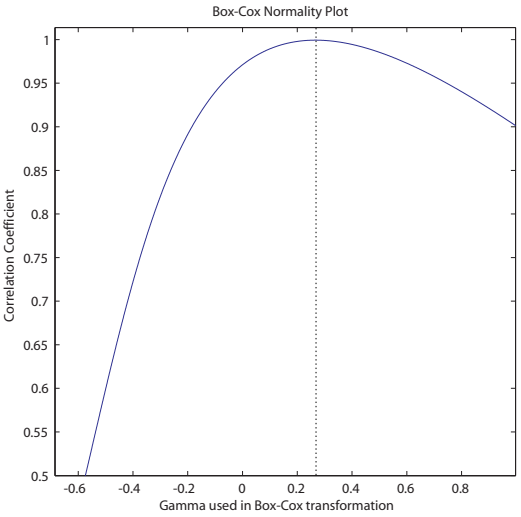


Figure 4.2: Box-Cox normality plot for choosing optimal λ

4.2.2 Manly transformation

Manly [Man76] proposed a modification of the Box-Cox transformation which also allows negative values:

$$y^{(\lambda)} = \begin{cases} \frac{e^{\lambda y}-1}{\lambda} & \text{if } \lambda \neq 0 \\ y & \text{if } \lambda = 0 \end{cases} \tag{4.2}$$

It was reported successful in transforming unimodal distributions and should not be used for bimodal or U-shaped distributions. Because our dataset contains negative values and

the experimental results of Manly's exponential transformation were slightly better than the Box-Cox transformation, we use this particular method to remove skewness.

4.2.3 John and Draper modulus function

To adjust the remaining non-gaussian kurtosis on symmetric data we use the John and Draper modulus function [JD80]:

$$y(\lambda) = \begin{cases} \text{sign}(y)\frac{(|y|+1)^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \text{sign}(y)\ln(|\,y\,|+1) & \text{if } \lambda = 0 \end{cases} \tag{4.3}$$

where

$$\text{sign}(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ -1 & \text{if } y < 0 \end{cases} \tag{4.4}$$

which is a modified power transformation applied to each tail separately. Non-negative powers λ less than 1 reduce kurtosis, while powers greater than 1 increase kurtosis. Again, we can use a divide and conquer approach for estimating the optimal λ. If y is symmetric around 0, then the modulus transformation will change the kurtosis without introducing skew. If y is not centered at 0, we add a constant before applying the modulus transformation.

4.2.4 Transformation Results

A small subset of the features did not perform well when transformed. The *Stellate Pattern Mean* features and the *Maximum Second Order Derivate Correlation* feature are approximately normal distributed in their original form and therefore not transformed with above functions. Also the *Number of Calcifications* feature was not an useful candidate to transform, because of its discrete nature. Statistical information about the transformed dataset is found in Table 4.1.

	Mean	**Std dev**	**Min**	**Max**	**Skewness**	**Kurtosis**
All cases (cases: 532)						
Stellate Patterns 1	0.5159	0.0358	0.4213	0.6297	0.2462	2.4515
Stellate Patterns 2	0.4702	0.0942	0.3497	0.6307	0.0765	1.1293
Stellate Patterns 1 Mean	1.1790	0.1655	0.8290	1.7740	0.8638	3.5949
Stellate Patterns 2 Mean	1.0551	0.0903	0.8380	1.4140	0.6548	3.4349
Region Size	0.2148	0.0883	0.0156	0.3891	0.0256	1.9484
Contrast	0.3608	0.0929	0.0109	0.6034	-0.0056	2.7369
Compactness	0.2079	0.0046	0.2026	0.2132	0.0031	1.0123
Linear Texture	0.0957	0.0421	0.0040	0.2052	0.2835	2.5912
Relative Location X	0.6451	0.3107	-0.0709	1.6145	0.0832	2.6490
Relative Location Y	0.2497	0.4623	-0.8520	1.6288	0.0676	2.6064
Max. 2nd order Drv Corr.	0.6571	0.1005	0.4040	0.9320	0.1290	2.5924
Number of Calcifications	1.4446	5.2303	0.0000	50.0000	5.4429	39.8079
Benign (cases: 258)						
Stellate Patterns 1	0.5083	0.0354	0.4213	0.6297	0.3459	2.5813
Stellate Patterns 2	0.4655	0.0922	0.3552	0.6307	0.0729	1.1252
Stellate Patterns 1 Mean	1.1166	0.1300	0.8600	1.5630	0.8740	3.8136
Stellate Patterns 2 Mean	1.0209	0.0718	0.8380	1.2990	0.5485	3.6036
Region Size	0.2064	0.0881	0.0195	0.3891	0.1922	1.9876
Contrast	0.3472	0.0873	0.1137	0.6034	0.2267	2.8038
Compactness	0.2079	0.0046	0.2028	0.2132	0.0035	1.0120
Linear Texture	0.0970	0.0432	0.0125	0.1982	0.1914	2.3855
Relative Location X	0.6665	0.3151	-0.0669	1.6145	0.1278	2.8950
Relative Location Y	0.2488	0.4574	-0.8452	1.6288	0.0391	2.4780
Max. 2nd order Drv Corr.	0.6774	0.0988	0.4520	0.9050	0.0325	2.3345
Number of Calcifications	0.7901	2.6820	0.0000	19.0000	3.8744	19.1856
Malignant (cases: 274)						
Stellate Patterns 1	0.5231	0.0347	0.4309	0.6065	0.2068	2.3569
Stellate Patterns 2	0.4747	0.0960	0.3497	0.6188	0.0691	1.1125
Stellate Patterns 1 Mean	1.2375	0.1737	0.8290	1.7740	0.6660	3.1044
Stellate Patterns 2 Mean	1.0871	0.0941	0.8530	1.4140	0.4737	3.0384
Region Size	0.2227	0.0879	0.0156	0.3817	-0.1293	1.9925
Contrast	0.3736	0.0962	0.0109	0.5793	-0.2382	2.8535
Compactness	0.2079	0.0046	0.2026	0.2132	0.0028	1.0126
Linear Texture	0.0945	0.0411	0.0040	0.2052	0.3745	2.8285
Relative Location X	0.6252	0.3057	-0.0709	1.3560	0.0262	2.3481
Relative Location Y	0.2505	0.4677	-0.8520	1.4580	0.0923	2.7135
Max. 2nd order Drv Corr.	0.6380	0.0986	0.4040	0.9320	0.2313	2.9313
Number of Calcifications	2.0571	6.7482	0.0000	50.0000	4.4613	25.8653

Table 4.1: Statistics of benign and malign cases after transformation

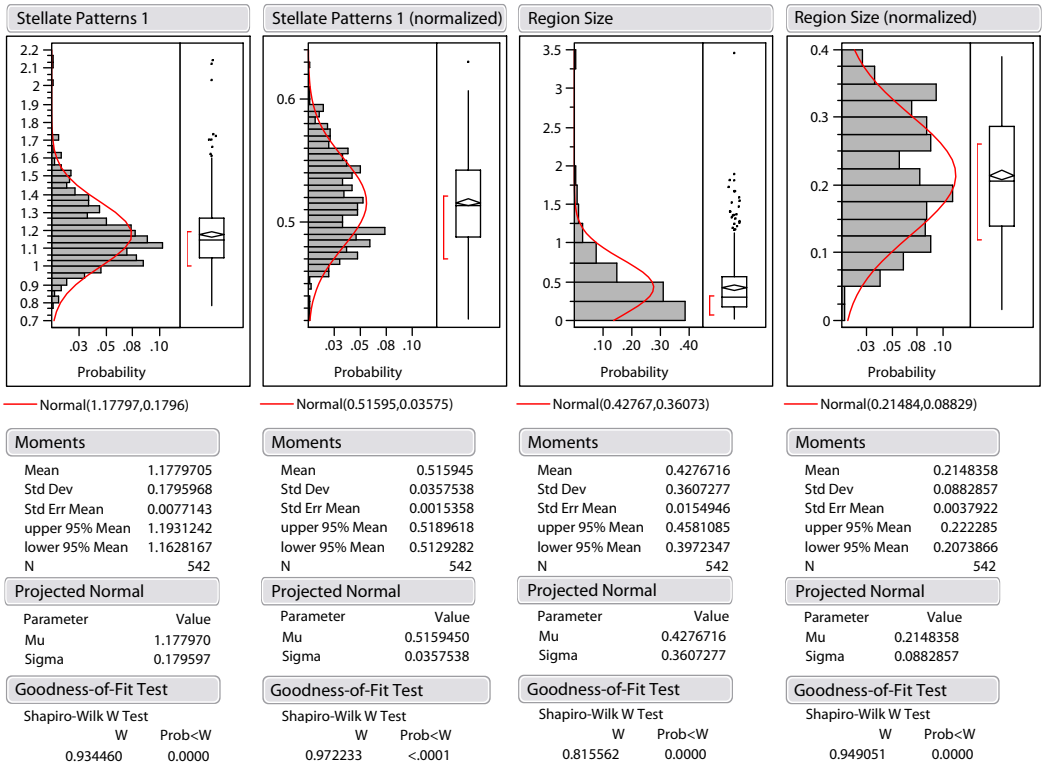


Figure 4.3: The posterior marginal distribution of two features before and after the normality transform

In Figure 4.3 we show the posterior marginal distribution for the *Stellate Pattern 1* and the *Region Size* feature, before and after the normality transformation. The histogram displays the actual distribution of the observed feature. The expected normal distribution with the calculated mean and variance is projected over this histogram as a blue curve. The expected mean and variance of the original *Stellate Pattern 1* feature are calculated as 1.1780 and 0.1796, respectively. After transformation they are calculated as 0.5159 and 0.0358. We use the Shapiro-Wilk W Test in the JMP statistical package [Inc05] to measure the goodness of the fit. When transformed, the goodness of the expected normal distribution increases with 4%, from 93% to 97%. The *Region Size* feature follows an approximately exponential distribution. The expected mean and variance of the original feature are calculated as 0.4277 and 0.3607, respectively. When transforming this feature to a normal distribution, the expected mean and variance are calculated as 0.2148 and 0.0833 and the goodness of the fit increases with 13%, from 82% to 95%.

4.3 Discretizing

As discussed in the previous section, our continuous features are not normal distributed. In order to overcome this problem, another approach would be discretizing the features. Although we will loose some information in the process, it is shown in several papers that naïve Bayes performs well with discretized data [DKS95, YW03a, YW02a]. A number of discretization methods have been developed and each have their advantages and disadvantages. A selection of the available methods have been implemented and their strategy will be explained in the following subsections.

4.3.1 Equal Width Discretization (EWD)

EWD [DKS95] is one of the simplest discretization techniques. This method sorts the values v of each feature into ascending order from v_{min} to v_{max} into k equally sized intervals. Each interval has width $w = (v_{min} - v_{max})/k$ and the cutpoints are at $v_{min} + w$, $v_{min} + 2w$, $\cdots$, $v_{min} + (k-1)w$. k is a parameter supplied by the user.

Let us now show how this works on the continuous *stellate pattern 1* feature in the non-discretized dataset depicted in Table 4.2.

Instance	*Features*		*Class*
	stellate pattern 1	stellate pattern 2	
1	1.23	1.03	benign
2	1.22	1.11	malign
3	1.60	1.28	malign
4	1.04	1.00	malign
5	1.11	0.97	malign
6	1.19	1.07	malign
7	1.06	1.01	benign
8	1.12	1.04	benign
9	1.17	1.08	malign
10	1.14	1.05	malign
11	0.80	0.83	benign
12	1.12	0.98	malign
13	1.11	1.03	malign
14	1.32	1.13	malign
15	1.20	1.03	benign

Table 4.2: Small part of the UMCN dataset with two continuous attributes

If we choose $k = 5$ then $v_{min} = 0.80$, $v_{max} = 1.60$, and $w = (1.60 - 0.80)/5 = 0.16$. The resulting intervals (often called bins) are shown in Table 4.3.

Intervals	[0.80,0.96]	(0.96,1.12]	(1.12, 1.28]	(1.28, 1.44]	(1.44,1.60]
SP1 feat.	0.80	1.04 1.06 1.11 1.11 1.12 1.12 1.14 1.20	1.17 1.19 1.22 1.23 1.32		1.60

Table 4.3: Equal Width Discretization of the *Stellate Pattern 1 (SP1)* feature

4.3.2 Equal Frequency Discretization (EFD)

EFD [DKS95] divides the sorted values into k intervals containing approximately the same number of training instances. Thus each interval contains n/k adjacent (possibly identical) values where k is a parameter supplied by the user.

As an example let us go back to the *stellate pattern 1* feature from Table 4.2. If we choose $k = 5$ and the number of instances is $n = 15$ then $n/k = 15/5 = 3$. The resulting intervals can be found in Table 4.4.

Intervals	[0.80,1.06]	[1.11,1.12]	[1.14,1.17]	[1.19,1.22]	[1.23,1.62]
SP1 feat.	0.80 1.04 1.06	1.11 1.11 1.12 1.12	1.14 1.17	1.19 1.20 1.22	1.23 1.32 1.62
Instances	3	4	2	3	3

Table 4.4: Equal Frequency Discretization of the *Stellate Pattern 1 (SP1)* feature

Although EWD and EFD are rather simplistic discretization methods, they are often used and work surprisingly well for naïve Bayes classifiers according to [HHW00].

4.3.3 Proportional k-Interval Discretization (PKID)

PKID [YW01] adjusts the size of the intervals (i.e., the number of instances in an interval) and therefore also the number of them proportional to the number of training instances. The idea behind that strategy is to adjust the discretization bias and variance to achieve a lower classification error. Discretization bias is the discretization error that results from the use of a particular discretization strategy. Variance measures how sensitive the discretization strategy is to changes in the data. Discretization bias and variance are directly related to interval size and number. The larger the interval size, the smaller the interval number, the lower the variance but the higher the bias. The opposite is also true: the smaller the interval size, the larger the interval number, the lower the bias but the higher the variance [YW02a].

The inverse relationship between interval size s and interval number t is trivial and can

be calculated as follows:

$$\begin{aligned} s \times t &= n \\ s &= t \end{aligned} \tag{4.5}$$

where n is the number of instances in the dataset.

As you can see, PKID gives equal weight to discretization bias and variance reduction by setting the interval size equal to the interval number ($s = t \approx \sqrt{n}$). Furthermore, the interval size and number are both proportional to the training data size. One flaw of the PKID method is that for small training sets it forms intervals small in size which might not present enough data for reliable probability estimation, hence resulting in high variance and poorer performance of the naïve Bayes classifier.

4.3.4 Non-Disjoint Discretization (NDD)

The idea behind NDD is that it works with overlapping intervals. [YW02b,YW02a] show that calculating the probability estimation of a continuous value v_i which is assigned to an interval $(a_i, b_i]$ is more reliable if the v_i falls towards the middle of the interval instead of close to either a_i or b_i.

Given a continuous feature for which there are n training instances with known values, the desired interval size s and the desired interval number t are calculated in the same way as the PKID strategy (see Equation 4.5), NDD forms t' *atomic intervals* of the form $(a'_1, b'_1], (a'_2, b'_2], \cdots, (a'_{t'}, b'_{t'}]$ each with frequency equal to s', so that

$$\begin{aligned} s' &= \frac{s}{\alpha} \\ s' \times t' &= n \end{aligned} \tag{4.6}$$

where α is any odd number and does not vary. For simplicity, we will take $\alpha = 3$.

One interval is formed for each set of three consecutive *atomic intervals*, such that the kth ($1 \leq k \leq t' - 2$) interval $(a_k, b_k]$ satisfies $a_k = a'_k$ and $b_k = b'_{k+2}$. An illustration of this can be found in Figure 4.4.

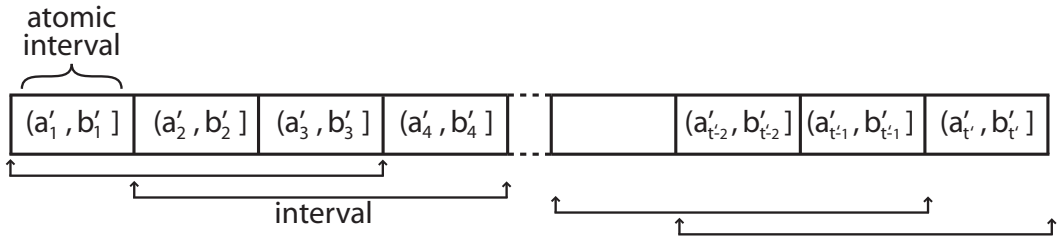


Figure 4.4: The actual intervals are formed out of three consecutive *atomic intervals*. For example, a value that falls into the *atomic interval* $(a_3, b_3]$ will be assigned to the interval $(a'_2, b'_4]$.

A value v will then be assigned to the interval $(a'_{i-1}, a'_{i+1}]$ where i is the index of the *atomic interval* which contains v. Using this method ensures that v always falls towards the middle of the interval, except when $i = 1$ in which case v is assigned to the interval $(a'_1, b'_3]$, and when $i = t'$ in which case v is assigned to $(a'_{t'-2}, b'_{t'}]$.

4.3.5 Weighted Proportional k-Interval Discretization (WPKID)

WPKID [YW03b] is the improved version of PKID and provides a solution for the PKID problem of the possible insufficient data in a single interval. For smaller datasets, discretization variance reduction has a bigger impact on naïve Bayes performance than discretization bias [Fri97]. This strategy weights discretization variance reduction more than bias for small training sets by setting a minimum interval size to make the probability estimation more reliable. It calculates s and t in a slightly different way than PKID, where s is the interval size and t is the number of intervals:

$$\begin{aligned} s \times t &= n \\ s - m &= t \\ m &= 30 \end{aligned} \tag{4.7}$$

The minimum interval size m is set to 30 because this is the minimum sample from which one should draw statistical inferences [YW02a].

4.4 Dimensionality Reduction

One might think that the use of more features will automatically improve the classification power of the classifier. However the number of samples needed per feature increases exponentially with the number of features to maintain a certain level of accuracy. This is better known as the *curse of dimensionality* which is a significant obstacle in machine learning problems that involve learning from few data samples in a high-dimensional feature space [Fri97].

For example, let the feature space be 24 dimensional, i.e., $\mathcal{F} = \{f_1, f_2, \ldots, f_{24}\}$ and every feature f_i has a domain D_i. Suppose that each domain $D_i \subseteq \mathbb{R}$ is discretized into 6 intervals $D_{i,1}, \ldots, D_{i,6}$ then the domain

$$\Omega = \prod_{\substack{1 \leq i \leq 24 \\ j \in \{1,\ldots,6\}}} D_{i,j}$$

has 6^{24} cells which is, in general, much more than the available training samples. Consequently most cells do not contain observations. It is therefore a good choice to use dimensionality reduction to overcome this problem.

The general philosophy behind dimensionality reduction techniques is that many real life datasets contain (linear) redundancies and noise. The following techniques will produce a lower-dimensional representation that approximates the original high-dimensional features and suppress the noise and remove redundancies. These methods will be used as a preprocessing step for classification.

4.4.1 Principal Component Analysis (PCA)

One of the most well-known dimension reduction techniques is Principal Component Analysis (PCA) [DHS01]. The success of PCA is partially due to its simplicity. The assumption made in PCA is that most of the information is carried in the variance of the features: the higher the variance in one dimension (feature), the more information is carried by that feature. The general idea is therefore to preserve the most variance in the data using the least number of dimensions.

We will explain the steps that have to be taken for doing PCA analysis using the covariance method using a small example dataset shown in Figure 4.5(a):

1. Organize the dataset into column vectors, so you end up with a $m \times n$ matrix, where m is the number of dimensions (features) and n is the number of cases.

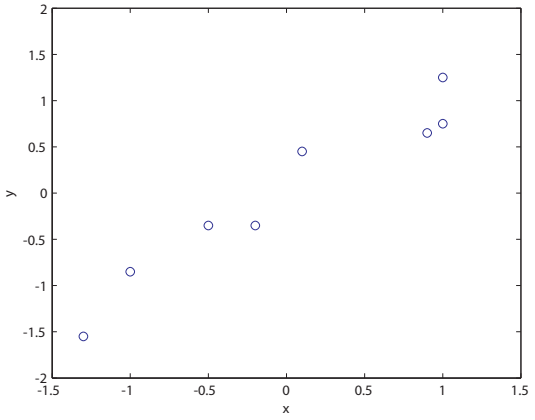
2. Subtract the mean from each of the dataset dimensions, so that each dimension has zero mean. We call the resulting dataset D.

x	y
0.5	0.2
2.8	3.0
1.3	1.4
2.8	2.5
2.7	2.4
0.8	0.9
1.6	1.4
1.9	2.2

(a)

x	y
-1.30	-1.55
1.00	1.25
-0.50	-0.35
1.00	0.75
0.90	0.65
-1.00	-0.85
-0.20	-0.35
0.10	0.45

(b)



(c)

Figure 4.5: First step of the PCA algorithm: (a) original data, (b) data with the means subtracted, and (c) plot of the means subtracted data.

3. Calculate the covariance matrix from D. Since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variable increase together.

$$cov = \begin{pmatrix} 0.8286 & 0.8200 \\ 0.8200 & 0.8743 \end{pmatrix}$$

4. Calculate the eigenvectors and eigenvalues of the covariance matrix. Since the covariance matrix is square, the eigenvectors and eigenvalues can be calculated.

$$eigenvalues = \begin{pmatrix} 0.0311 \\ 1.6717 \end{pmatrix} eigenvectors = \begin{pmatrix} -0.7169 & -0.6972 \\ 0.6972 & -0.7169 \end{pmatrix}$$

5. Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives you the components in order of significance, in our case the second column of the eigenvectors is the one with the highest corresponding eigenvalue (1.6717). The eigenvector with the *highest* eigenvalue is the *principle component* of the dataset. See also Figure 4.6.

6. Select the desired number of components (one can decide to ignore the components of lesser significance to reduce dimensionality).

$$component = \begin{pmatrix} -0.6972 \\ -0.7169 \end{pmatrix}$$

7. Construct the new dataset by transposing the selected components vector and multiplying it with the mean-adjusted dataset, transposed.

$$FinalDataset = RowEigenVectors \times RowMeanAdjustedDataset$$

$$= \begin{pmatrix} -0.6972 & -0.7169 \end{pmatrix} \times \begin{pmatrix} -1.3000 & 1.0000 & -0.5000 & 1.0000 & 0.9000 & -1.0000 & -0.2000 & 0.1000 \\ -1.5500 & 1.2500 & -0.3500 & 0.7500 & 0.6500 & -0.8500 & -0.3500 & 0.4500 \end{pmatrix}$$

$$= \begin{pmatrix} 2.0175 & -1.5933 & 0.5995 & -1.2349 & -1.0934 & 1.3065 & 0.3903 & -0.3923 \end{pmatrix}$$

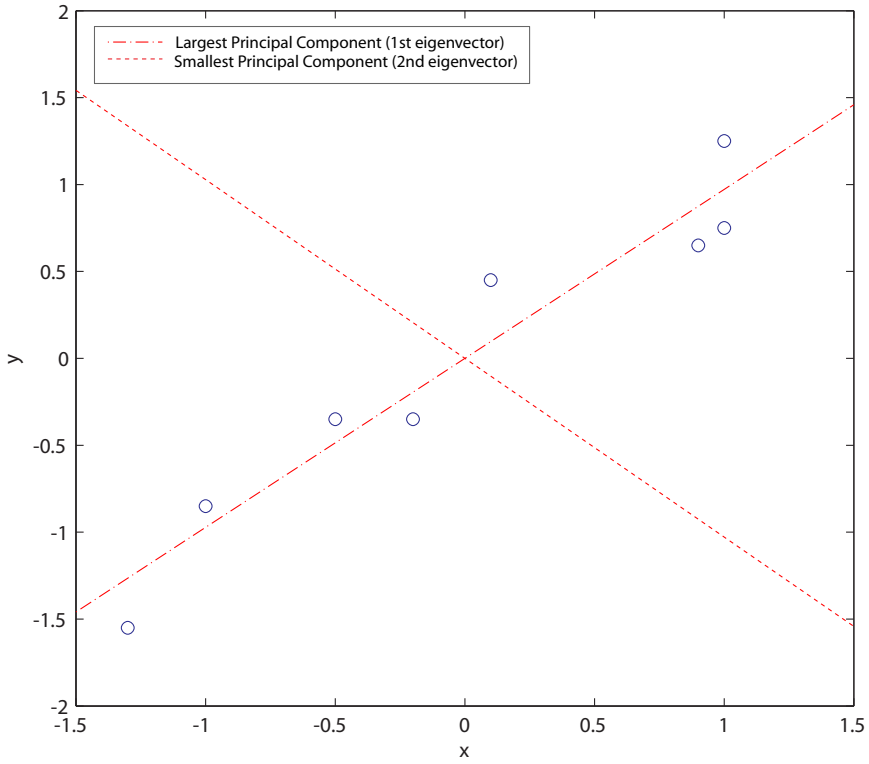


Figure 4.6: Normalized data (means subtracted) with the eigenvectors of the covariance matrix overlayed.

One major drawback of PCA is that it can eliminate the dimension that is best for discriminating positive cases from negative cases, because it is an unsupervised algorithm. Suppose the data are spread parallel on each side of the linear separator, then it is easy to see that the discriminating dimension will be eliminated.

4.4.2 Fisher Discriminant Analysis (FDA)

For a classification task FDA is often preferred above PCA because it incorporates class information. FDA tries to find a mapping from the high-dimensional space to a low-dimensional space (the so called Fisher space) in which the most discriminant features are preserved. It accomplishes this by minimizing the variation within the same class and maximizing the variation between classes [DHS01]. This can be expressed in mathematical terms as follows.

Consider that each case in the learning set belongs to one of n classes $(C_1, C_2, \ldots, C_n)$. The between-class scatter matrix S_B and within-class scatter matrix S_W can be defined as:

$$S_B = \sum_{i=1}^{n} m_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{4.8}$$

and

$$S_W = \sum_{i=1}^{n} \sum_{x_k \in C_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{4.9}$$

where μ is the reference class mean, μ_i is the mean of class i, m_i is the number of cases and the superscript T indicates a transpose action.

The objective of FDA is then to find W_{opt} maximizing the ratio of the between-class scatter to the within-class scatter:

$$W_{opt} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} \tag{4.10}$$

Finding the maximum W_{opt} could be tricky, but fortunately it is known that the solution can be found relatively simple: W_{opt} is the solution of the following conventional eigenvalue problem:

$$S_B W - \lambda S_W W = 0 \tag{4.11}$$

where λ is a diagonal matrix whose elements are the eigenvalues. The column vectors $w_i (i = 1, \ldots, m)$ of matrix W are eigenvectors corresponding to the eigenvalues in λ.

If FDA is used as a preprocessing step to reduce the dimension of the initial feature space, the dimension of the resulting subspace can only be reduced to no more than $c - 1$, where c is the number of classes. Because we have only 2 classes, i.e., *benign* and

malign, the resulting dataset can only be a one-dimensional space, known as the Fisher linear discriminant, which is heavily inadequate for our classification problem.

Therefore Duchene and Leclercq [DL88] has proposed some tricks to effectively use this dimension reduction technique which was implemented in Matlab by Roger Jang. The whole DCPR (Data Clustering and Pattern Recognition) Toolbox by the same author is available at [Jan06].

4.5 Scaling

Scaling the features before applying SVM is of significant importance [ER04]. One of the main advantages of scaling is that features in greater numeric ranges do not dominate those in smaller numeric ranges. Because SVM kernels usually depend on the inner products of feature vectors, large values can cause numerical problems. By scaling, numerical difficulties will be avoided. We will calculate the scaling parameters of the training dataset and scale the training and test dataset with the same scaling parameters. The following paragraphs will discuss centering and the three scaling methods we have used [EJKW$^+$01].

Centering

The most basic, but important preprocessing step is to center the multidimensional feature vector $\hat{x}$. For every column $x_i \in \hat{x}$, the column mean $\frac{1}{n}\sum_{i=1}^{n} x_i$ is subtracted from every value in that column to make $\hat{x}$ a zero-mean variable.

Standardizing

Standardizing is scaling based on the variability of the values. This is done by dividing the columns by their sample standard deviation to obtain a standard deviation of 1. The sample standard deviation for a column is obtained by Equation 4.12. If the columns are centered, dividing the columns by their sample RMS (Equation 4.13) gives the same result because the mean is zero.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4.12}$$

$$RMS_{sample} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} x_i^2} \tag{4.13}$$

Range scaling

An other approach of scaling is transforming the range of each feature in the training set to $[-1, +1]$ range. The range of the scaled test set can however be slightly different, because the training set does not have to contain the actual minimum and maximum value of the column vector. This is not a big problem, as there is no requirement that the input data for the support vector machine should be within that range. The scaling is done with the following formula:

$$x_i = \left(\frac{2}{\max(\hat{x}) - \min(\hat{x})} \cdot \hat{x}\right) - \left(\frac{\max(\hat{x}) + \min(\hat{x})}{\max(\hat{x}) - \min(\hat{x})}\right) \text{ for every } x_i \in \hat{x} \tag{4.14}$$

In our specific application, this scaling method gave an average decrease of 5% in performance (AUC) in comparison with the scaling based on variance.

Class-Specific scaling

The last method we used is the *Class-Specific* scaling which incorporates class label information. It attempts to increase the influence of features that are likely to be predictive by increasing the range of its data points. A feature is predictive when that feature has small variance but significantly different means in the positive and negative classes. This scaling is accomplished by applying the following formula:

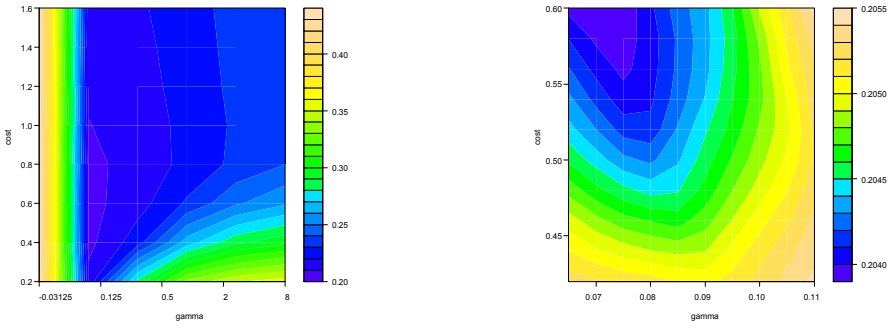
$$x_i = \frac{\text{mean}(\hat{x}_+) - \text{mean}(\hat{x}_-)}{\text{var}(\hat{x}_+) + \text{var}(\hat{x}_-)} \text{ for every } x_i \in \hat{x} \tag{4.15}$$

where $\hat{x}_+$ and $\hat{x}_-$ represent the feature vector for the positive and negative class instances respectively. Using this scaling method as a preprocessing step on our dataset did improve the support vector machine classifier accuracy, but not as much as the standardizing method.

4.6 SVM Model Selection

In Section 2.6 there are four common kernel functions mentioned from which we have to choose one for the classification task. Generally, the radial kernel is a reasonable choice when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel with a cost parameter C is a special case of the radial kernel with some parameters (C, γ) [KL03]. Additionally, the sigmoid kernel behaves like the radial kernel for certain parameters [LL03]. Because the radial kernel has less parameters than the polynomial kernel, the complexity of model selection is significantly lower.

When using the radial kernel, two parameters have to be provided: misclassification cost C and kernel width γ. Which C and γ are best for a certain problem is not known beforehand, therefore some model selection (i.e., parameter search) has to be done. In the e1071 library [DHL+05], there exists a function *tune.svm* which does a grid-search on these parameters using 10-fold cross-validation where all pairs (C, γ) are tried and the one with the best cross-validation accuracy is selected. Obviously this greedy search has a high computational cost. Consequently several heuristic methods have been designed to lower computational cost by, for example, approximating the cross-validation rate. However, we chose to do a grid-search without heuristics and reduce the time by first using a coarse grid as can be seen in Figure 4.7(a). After identifying the best region on the grid, we will search that region with a finer grid (Figure 4.7(b)) to finetune the values.



(a) Coarse grid search

(b) Fine grid search

Figure 4.7: Kernel parameter tuning

4.7 Building the Bayesian Networks

Manually constructing a Bayesian network turns out to be very time consuming in practice, therefore several techniques have been developed for automatically learning Bayesian networks from clinical data. Learning a Bayesian network consists of two tasks, namely learning the *structure* (i.e., identifying the topology of the network) and learning the *parameters* (i.e., determining the conditional probability distributions). There are several existing learning algorithms available for Bayesian networks that offer a good starting point [CH91, CBL97, LB94].

Besides fully automatic learning, fragments of causal background knowledge can be used to guide the learning process. The causal background knowledge could be provided by the Radiology department as it already has a lot of experience in building classifiers for this problem domain.

4.7.1 Structure Learning

The first but rather naïve idea to find the best network structure is to choose the structure that has the best score of all possible graphs. The number of different structures for a Bayesian network with n nodes is super exponential and can be calculated with the following formula [Rob77]:

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} r(n-i) = n^{2^{O(n)}}$$

This gives $r(1) = 1$, $r(2) = 3$, $r(3) = 25$, $r(5) = 29,281$ and $r(10) \backsimeq 4.2 \cdot 10^{18}$.

If the number of nodes exceeds 8, such exhaustive search can't be done in a reasonable time and therefore structure learning methods often use search heuristics. They often make use of operators like arc-insertion and arc-deletion and compare resulting graphs by their score (a calculated value for how well a graph explains the data) in order to choose the next best step.

The two most popular scorings algorithms are the Bayesian score [CH91, Hec95] which integrates out the parameters, i.e., it is the marginal likelihood of the model, and the BIC score [Nea03], which is the sum of a likelihood term and a penalty term which penalize complex networks.

For this project we will evaluate several structure learning algorithms to learn complex topologies and structures with a very limited topology such as naïve Bayes (NB) and the tree augmented network (TAN). In the next sections we will briefly explain the used algorithms.

- NB (Naïve Bayes)
- TAN (Tree Augmented Network)
- MWST (Maximum Weighted Spanning Tree)
- K2 (initialized with a random ordering of the nodes)
- K2+T (initialized with ordering returned by MWST)
- K2-T (initialized with reverse ordering returned by MWST)
- MCMC (Markov Chain Monte Carlo)

Naïve Bayes

The naïve Bayesian classifier [LIT92] is a simple Bayesian classification algorithm. It assumes that every feature is independent from the rest of the features given the state of the class variable. As a result its structure only contains edges from the class node to the other features in order to simplify the joint distribution.

Tree Augmented Network

Unlike the naïve Bayes network, the tree augmented Bayesian network (TAN) [Gei92] also allows edges between evidence variables as long as they form a tree. This approach approximates the interactions between attributes and could theoretically lead to better classification performance. The best tree relying all the observations can be obtained with the MWST algorithm.

Maximum Weight Spanning Tree

[CL68] have proposed a method based on the maximum weight spanning tree algorithm (MWST). This method associates a weight to each edge. This weight can be either the mutual information between the two variables or the score variation when one node becomes a parent of the other. When the weight matrix is created, an usual MWST algorithm gives an undirected tree that can be oriented with the choice of a root.

K2

The main idea of the K2 algorithm is to maximize the structure probability given the data. It assumes there is an ordering available on the variables which means that if X_a comes before X_b, then X_b cannot be a parent of X_a, which is used to reduce the size of the search space. The search space becomes the subspace of all the directed acyclic

graphs admitting this order as topological order. It further assumes that, a priori, all structures are equally likely. In the next paragraph we explain how the algorithm works.

The parent set for a node X_a is initially set to the empty set. Then the algorithm greedily adds the node, from among the predecessors in the ordering, to the parent set that increases the probability of the resultant network by the largest amount. It stops when there are no more parents to add or if no parent addition improves the network score.

The main problem with K2 is that it is a greedy algorithm. The K2 algorithm requires a good ordering on the nodes to be given as input to perform well. In this study we also use the MWST algorithm first to find a good ordering on the nodes, as this tends to lead to better results than just giving a random ordering as input.

MCMC

Markov chain Monte Carlo methods, hereafter called MCMC, are a class of algorithms for sampling networks from the posterior distribution. The Metropolis-Hastings algorithm is the MCMC algorithm that is implemented in the BNT toolbox [Mur01] to search the space of all DAGs. The basic idea is to use the Metropolis-Hastings algorithm to draw samples from $P(D|G)$ (see Section 2.7.4) after a chosen burn-in time. Then a new graph G' is kept if the Bayes factor $\frac{P(D|G')}{P(D|G)}$ (i.e., the ratio between the marginal likelihood of the new model to the marginal likelihood of the previous model) increases. The quality of the sample improves as a function of the number of steps, until the distribution of simulated values converges to the true posterior distribution. A more detailed explanation of the Metropolis-Hastings algorithm is given in [CG95].

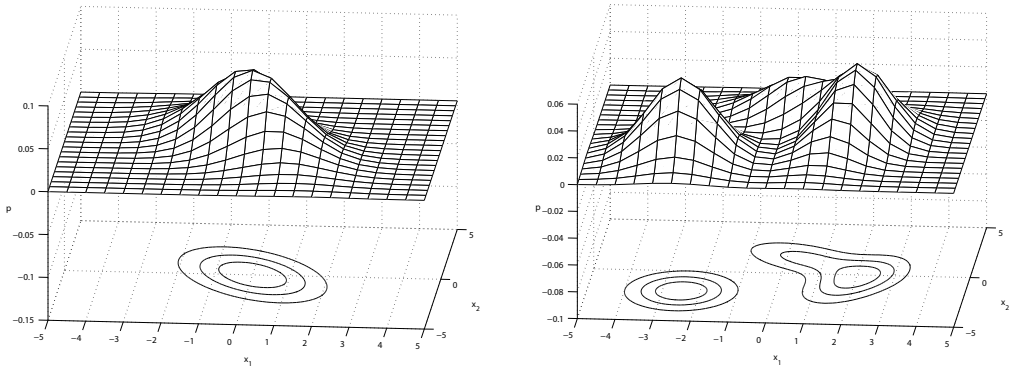
4.7.2 Gaussian Mixture Model (GMM)

Gaussian mixture models [Ver04] belong to the class of pattern recognition systems. They are easy to implement and model the probability density function[1] of observed variables using a multivariate Gaussian mixture density and have been applied to medical image classification problems before [PG04, PCH+00, TS05].

The Gaussian probability density function in one dimension is a bell shaped curve defined by two parameters, mean μ and variance σ^2. In a d-dimensional space it is defined as

$$p(\bar{x}|\bar{\mu}, \Sigma) = \frac{\exp\left(-\frac{1}{2}(\bar{x}-\bar{\mu})^T\Sigma^{-1}(\bar{x}-\bar{\mu})\right)}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} \tag{4.16}$$

[1] The probability density function (or pdf) of a random variable is the relative frequency of occurrence of that random variable. The area under the pdf is exactly one.



(a) (b)

Figure 4.8: (a) An example surface of a two-dimensional Gaussian *pdf* with $\bar{\mu} = [0; 0]$ and $\Sigma = [1.56, -0.97, -0.97, 2.68]$ and (b) is an example surface of a two-dimensional Gaussian mixture pdf with three components: $\alpha_1 = 0.40, \bar{\mu_1} = [-2.5; -2], \Sigma_1 = [0.81, 0; 0, 1.44], \alpha_2 = 0.25, \bar{\mu_2} = [0.5; 1.5], \Sigma_2 = [1.30, -0.66; -0.66, 1.30]$ and $\alpha_3 = 0.35, \bar{\mu_3} = [2.0; -0.5], \Sigma_3 = [0.69, 0.61; 0.61, 2.36]$.

where $\bar{\mu}$ is the mean vector and Σ the covariance matrix. See Figure 4.8(a) for an example (taken from [DHS01]).

In some cases approximation of the posterior probability by a single Gaussian might be too simple; a better approach would be using a Gaussian mixture model which is a mixture of several Gaussian distributions and assume that the entire data set can be modeled by a M-gaussian mixture probability density function. This probability density function is defined as a weighted sum of Gaussians:

$$p(\bar{x}|\Theta) = \sum_{m=1}^{M} \alpha_m \, p(\bar{x}|\bar{\mu}_m, \Sigma_m) \tag{4.17}$$

where M is the number of Gaussian components, α_m is the prior probability (i.e., mixing weight) of component m, $0 < \alpha_m < 1$ for all components, and $\sum_{m=1}^{M} \alpha_m = 1$. Θ is the parameter list $\{\alpha_1, \bar{\mu}_1, \Sigma_1, \ldots, \alpha_M, \bar{\mu}_M, \Sigma_M\}$. An example of a Gaussian mixture model is shown in Figure 4.8(b) (taken from [DHS01]).

During the training phase, we select all vectors that belong to a given class and learn the parameters of the Gaussian mixture, such as the mixing weights, the mean vectors, and the diagonal covariance matrices that maximizes the likelihood function. The standard algorithm to find the optimal Θ is the expectation maximization (EM) procedure, explained in detail in [MK97].

Recall that the posterior probability can be computed with the Bayes rule

$$P(C_k|\bar{x}) = \frac{p(\bar{x}|C_k)P(c_k)}{p(x)} \tag{4.18}$$

where $P(C_k|\bar{x})$ is the probability density function of class C_k in the feature space, and $P(c_k)$ is the *a priori* probability which is the probability of the class before measuring any features, and $p(x)$ is merely a scaling factor to assure that posterior probabilities are really probabilities lying between 0 and 1.

Once the Gaussian mixture parameters have been found for each class, assigning a test vector to a class is straightforward. A test vector $\bar{x}$ is assigned to the class that maximizes $P(C|\bar{x})$.

5 Results

In this chapter, we evaluate the algorithms and techniques described in the previous chapters to learn Support Vector Machines and Bayesian Networks and use them to classify breast tumors.

5.1 Image based performance

In this section we use our real-world dataset described in Section 3 with 522 instances, each with 12 continuous features. Every MLO view has a corresponding CC view in the dataset. In the following experiments we will be measuring the image based performance and therefore do not make a distinction between the views, which means we use the same Bayesian network or support vector machine for classifying the MLO and CC view. The process is shown schematically in Figure 5.1.

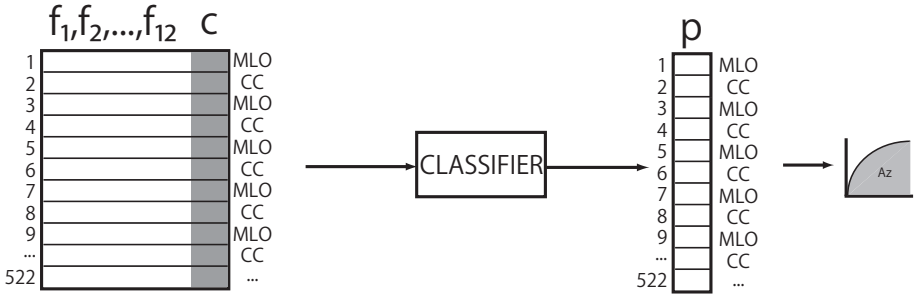


Figure 5.1: Schematic illustration of testing the image based performance

In the first experiment we constructed a number of Bayesian networks with different structure learning algorithms. We classified with the constructed Bayesian networks using a 10-fold cross validation scheme. The classification was performed 5 times and averaged to compute the final classification accuracy. The learning time per fold is denoted in the fourth column, and the BIC score (see Section 4.7.1) in the fifth column gives an indication of the quality of the learned network where higher is better. Both Bayesian networks with limited topology, Naive Bayes (NB) and Tree Augmented Networks (TAN), performed very well compared to the more sophisticated Bayesian networks in terms of classification performance and they are considerably faster than K2 and MCMC. The tree search method (MWST) gives the worst performance for our dataset. The different initializations for the K2 algorithm has a noticeable impact on the performance. K2 uses a random ordering on the nodes, K2+T uses the ordering from the MWST algorithm, and K2-T uses the reverse ordering from the MWST algorithm. For learning with the MCMC algorithm we used a burn-in of 10% of the desired amount of samples. We have not put any restriction on the maximum fan in and fan out (incoming and outgoing arcs). The overall quality of the Bayesian networks learned with the MCMC algorithm is better when using a considerable amount of samples, but the achieved classification performance does not surpass the simple networks NB and TAN. Furthermore the MCMC algorithm is the slowest structure learning algorithm, especially when using 500 or more samples. We also see that a high Bayesian network quality does not necessarily guarantee a good classification performance, which is throughout this thesis the most important evaluation measure.

Structure Algo.	A_z value	T	BIC
NB	0.7315 ± 0.0008 [0.7301;0.7316;0.7318;0.7319;0.7320]	3	-313.767
TAN	0.7413 ± 0.0023 [0.7390;0.7393;0.7412;0.7422;0.7447]	18	552.469
MWST	0.7092 ± 0.0035 [0.7037;0.7077;0.7110;0.7115;0.7121]	20	685.025
K2	0.7303 ± 0.0046 [0.7235;0.7280;0.7310;0.7339;0.7349]	101	737.863
K2+T	0.7208 ± 0.0015 [0.7182;0.7205;0.7215;0.7217;0.7220]	134	829.274
K2-T	0.7345 ± 0.0081 [0.7411;0.7310;0.7250;0.7309;0.7446]	122	736.032
MCMC (n=100)	0.7266 ± 0.0074 [0.7154;0.7238;0.7274;0.7326;0.7337]	58	466.582
MCMC (n=500)	0.7299 ± 0.0040 [0.7260;0.7274;0.7276;0.7341;0.7343]	205	728.434
MCMC (n=1000)	0.7202 ± 0.0112 [0.7104;0.7158;0.7163;0.7188;0.7394]	330	872.148

Table 5.1: Area under ROC curve results for different structure learning methods using 10-fold cross-validation, averaged over five times. T is the structure learning time per fold in seconds.

5.2 Image based performance SVM kernels

In the next experiment we test the different SVM kernels as described in Section 2.6. For every kernel we used the default parameters and compared them to the tuned parameters. The tuning of the SVM kernel parameters using a grid search is discussed in Section 4.6. Except for the sigmoid kernel, the default parameters perform well and do not differ significantly from the tuned parameters. The results are shown in Table 5.2. Due to the computational complexity of searching the four optimal kernel parameters for the polynomial kernel, we have not been able to find these parameters. As expected, the radial kernel achieved the best classification performance and is chosen for the experiments hereafter, using the default kernel parameters.

Kernel	A_z value
Linear (default)	0.7346 ± 0.0010 [0.7329;0.7346;0.7346;0.7352;0.7356]
Linear (tuned)	0.7401 ± 0.0031 [0.7369;0.7374;0.7396;0.7425;0.7439]
Polynomial (default)	0.7103 ± 0.0026 [0.7078;0.7089;0.7099;0.7102;0.7145]
Polynomial (tuned)	n/a
Radial (default)	0.7729 ± 0.0008 [0.7719;0.7721;0.7732;0.7736;0.7736]
Radial (tuned)	0.7779 ± 0.0009 [0.7762;0.7771;0.7777;0.7779;0.7784]
Sigmoid (default)	0.5113 ± 0.0029 [0.5090;0.5092;0.5099;0.5125;0.5159]
Sigmoid (tuned)	0.7338 ± 0.0009 [0.7327;0.7334;0.7334;0.7347;0.7348]

Table 5.2: Area under ROC curve results for different kernels with and without parameter tuning using 10-fold cross validation, averaged over 5 times. The default kernel parameters are $C = 1$, $\gamma = 1/(\text{data dimension})$, $r = 0$, and $d = 3$. The cost parameter of the tuned linear kernel is $C = 0.03125$, the kernel parameters of the tuned radial kernel are $C = 0.6, \gamma = 0.075$, and the kernel parameters of the tuned sigmoid kernel are $C = 0.0078125, \gamma = 0.5$.

5.3 SVM (radial) vs Bayesian (NB, TAN) performance

Based on the results of the above experiments we felt that it was reasonable to use the radial kernel for learning Support Vector Machines and Naive Bayes and Tree Augmented Networks for learning Bayesian networks in the upcoming experiments.

We conduct the following three groups of experiments to evaluate the classification performance:

1. Image based performance test
2. Case based performance test using averaging
3. Case based performance test combining features

As previously explained, the image based performance test does not make a distinction between MLO and CC views in contrast to case based performance. The outcome of the image based experiments are summarized in Table 5.3. The first case based performance test we conducted averages the classifier output of the MLO view with the classifier output of the corresponding CC view, which is schematically shown in Figure 5.2. We essentially combine information from multiple images of the same patient. We have tried four ways to combine the classifier outputs: taking the average, the median, the maximum, and the minimum. We found that the average always produced an improved area under the ROC curve (A_z value) compared to the single view images. Using the minimum or maximum could potentially lead to better results in some situations, if for example the mass is seen only in the MLO or CC view, but for our images in the dataset it had a negative effect on the A_z value. This also confirms the observations of [LMJ04]. In the second case based performance we combine both views to one case. Because there are 12 features per view available this extends to 24 features per case (see Figure 5.3). Using this extended dataset with 261 instances results in the classification performances shown in Table 5.5.

5.3.1 Image based

Technique	A_z value
SVM	0.7962 ± 0.0012 [0.7945;0.7957;0.7966;0.7967;0.7975]
NB	0.7315 ± 0.0008 [0.7301;0.7316;0.7318;0.7319;0.7320]
TAN	0.7413 ± 0.0023 [0.7390;0.7393;0.7412;0.7422;0.7447]
GMM	0.7678 ± 0.0019 [0.7650;0.7669;0.7683;0.7687;0.7699]

Table 5.3: Area under ROC curve results for image based classifier, 10-fold cross validation, averaged over 5 times.

5.3.2 Case based MLO and CC averaging

As one would expect, we see a significant improvement in classification performance for all classifier types in comparison with the image based performance tests (Table 5.4).

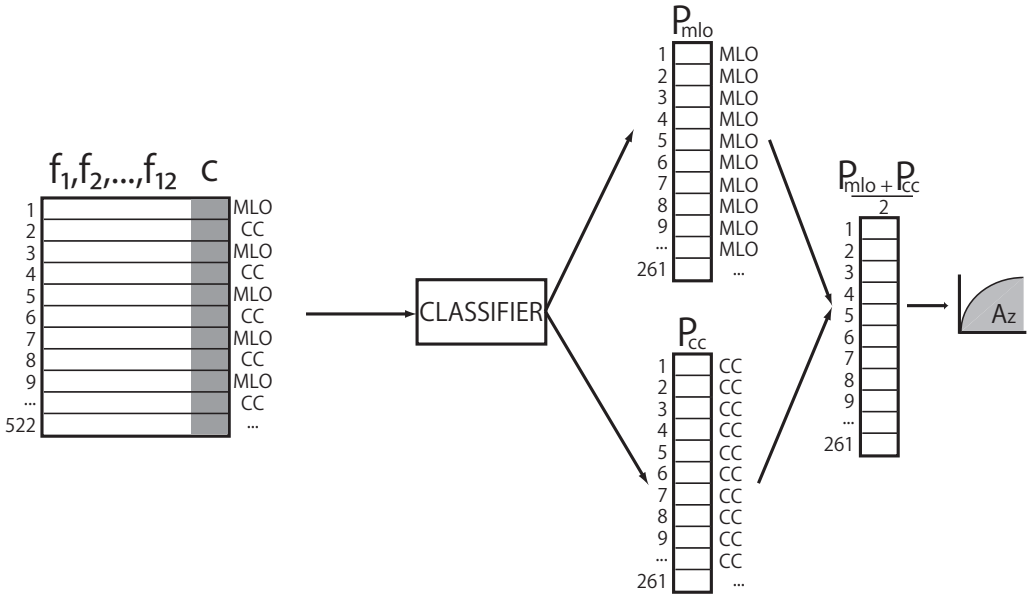


Figure 5.2: Schematic illustration of testing the case based performance by averaging the classifier output of both views

Technique	A_z value
SVM	0.7985 ± 0.0023 [0.7967;0.7968;0.7977;0.7989;0.8023]
NB	0.7667 ± 0.0032 [0.7627;0.7647;0.7666;0.7688;0.7707]
TAN	0.7617 ± 0.0043 [0.7559;0.7587;0.7628;0.7646;0.7665]
GMM	0.7685 ± 0.0104 [0.7652;0.7653;0.7785;0.7791;0.7543]

Table 5.4: Area under ROC curve results for MLO and CC averaged classifier using 10-fold cross validation, averaged over 5 times.

5.3.3 Case based MLO and CC features combined

As can be seen in Table 5.5, the performance of the SVM classifier is a bit lower compared with the averaging method. This confirms our beliefs that a larger number of features negatively effects the SVM performance. On the other side, naive Bayes performs better with the combined feature sets. Contrary to our intuition the TAN classifier gives worse results than the naive Bayes classifier in the case based performance tests.

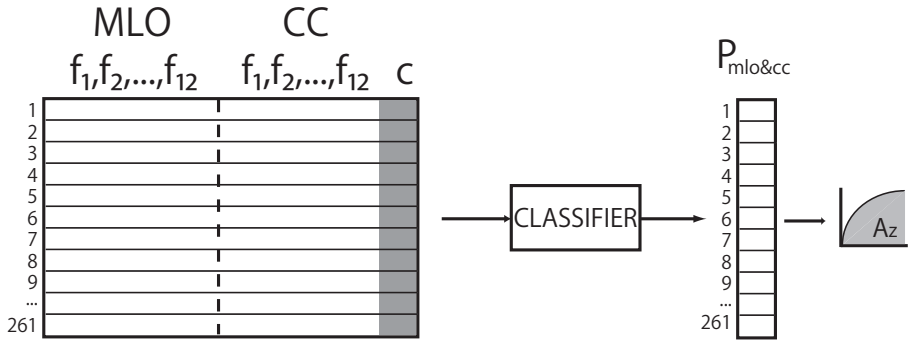


Figure 5.3: Schematic illustration of testing the case based performance by combining the features of both views

Technique	A_z value
SVM	0.7926 ± 0.0030 [0.7881;0.7915;0.7927;0.7951;0.7955]
NB	0.7672 ± 0.0027 [0.7631;0.7661;0.7678;0.7691;0.7698]
TAN	0.7428 ± 0.0067 [0.7337;0.7385;0.7452;0.7457;0.7509]
GMM	0.7499 ± 0.0312 [0.7114;0.7258;0.7540;0.7718;0.7866]

Table 5.5: Area under ROC curve results for MLO and CC combined classifier using 10-fold cross validation, averaged over 5 times.

5.4 Transformation results

We observed in Section 3.2 that the real world dataset we are using contains a lot of features that do not follow a normal distribution. Bayesian networks with continuous nodes assume that within each state of the class the observed continuous features follow a normal (Gaussian) distribution. These continuous features have two parameters, mean and variance, to describe the normal distribution. Transformation methods have been designed to transform the non-normal features closer to a normal distribution. In this

experiment we have tested three transformation methods described in Section 4.2 to determine if the case based classification performance of the naive Bayes network would increase after transformation. The Box-Cox and Manly transformation methods, which primarily remove skewness, have a significantly positive impact on the performance. Applying the John & Draper method to remove additional kurtosis introduced too much skewness and reduced the accomplished performance. The results are summarized in Table 5.6.

Technique	A_z value
Box-Cox	0.7743 ± 0.0060 [0.7650;0.7731;0.7757;0.7763;0.7814]
Manly	0.7876 ± 0.0028 [0.7851;0.7853;0.7865;0.7906;0.7906]
Manly + JohnDraper	0.7649 ± 0.0039 [0.7601;0.7628;0.7641;0.7676;0.7700]

Table 5.6: Area under ROC curve results of the naïve Bayes models when using different methods for transforming variables to normal distribution. The optimal gamma for the transformation function was found by using divide and conquer to find the lowest skewness.

5.5 Discretization

In Section 4.3 we explained five discretization algorithms which are suitable for Naive Bayes classifiers. To avoid zero probabilities in the conditional probability tables we used a uniform dirichlet prior on the discrete nodes. We expected that these methods would accomplish the same or better performance than the Naive Bayes network with continuous nodes. However, the classification performance was somewhat disappointing. Based on the results in Table 5.7 it seems that the performance decreases if the number of intervals increase above a certain level. We used all 522 instances of our dataset in this experiment, but apparently this may still be not enough to accurately define the conditional probability tables of the discrete nodes. Furthermore, we found that transforming the features to a normal distribution before discretizing them did not have any effect on the performance.

	Intervals	A_z value
EWD	10	0.7421 ± 0.0020 [0.7388;0.7418;0.7429;0.7431;0.7440]
EFD	6	0.7352 ± 0.0011 [0.7339;0.7345;0.7350;0.7356;0.7369]
NDD	65*	0.6832 ± 0.0048 [0.6791;0.6796;0.6821;0.6840;0.6910]
PKID	21	0.7403 ± 0.0030 [0.7368;0.7384;0.7394;0.7427;0.7439]
WPKID	6	0.7513 ± 0.0016 [0.7496;0.7507;0.7512;0.7513;0.7540]

Table 5.7: Area under ROC curve results of different discretization techniques and the number of intervals (image based performance). * With NDD there is an overlap between intervals.

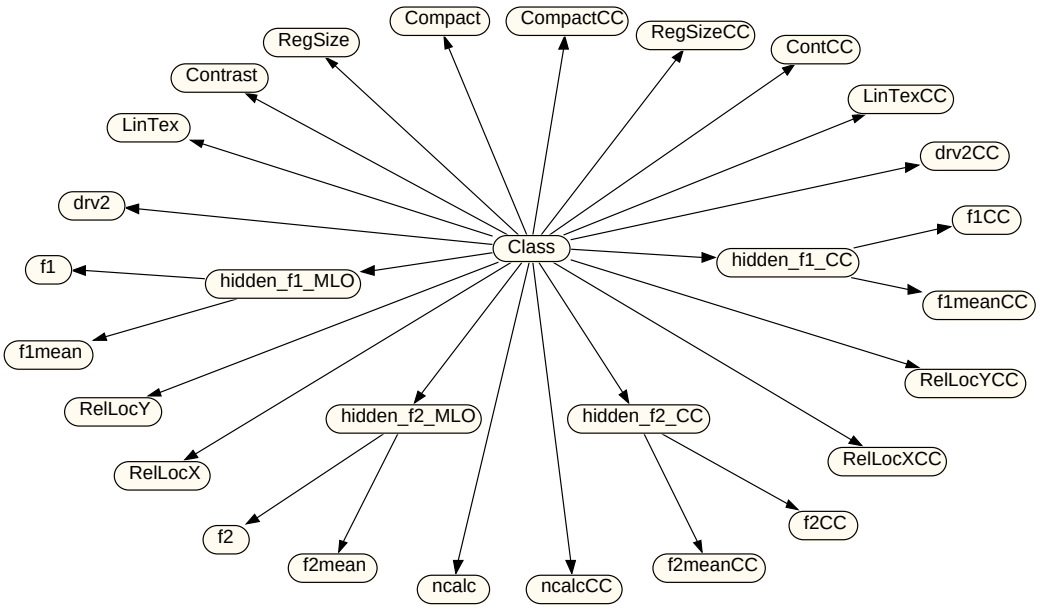


Figure 5.4: Bayesian network with hidden nodes

5.6 Hidden nodes

Additionally, we constructed a latent model (a Bayesian network with hidden, unobserved nodes). While there are more complex latent models possible, we tried a simple approach to see if the classification performance would increase if we add two hidden nodes per view between two strong correlated features. In Figure 5.5 we see that f1 and f2 are linearly correlated with respectively f1mean and f2mean. We disconnected these correlated nodes from the class node and placed a hidden node in between. The hidden node is then connected to the class node as can be seen in Figure 5.4. We have used Bayesian networks with discrete as well as with (normalized) continuous nodes, and learned the parameters with the EM algorithm. To avoid zero probabilities in the conditional probability tables we used a uniform dirichlet prior on the discrete nodes. More sophisticated latent models incorporating Factor Analyzers were outside the scope of this report, but may give much better results.

	A_z value
EWD	0.7425 ± 0.0018 [0.7398;0.7414;0.7434;0.7438;0.7440]
EFD	0.7334 ± 0.0064 [0.7278;0.7283;0.7329;0.7341;0.7437]
NDD	0.5952 ± 0.0018 [0.5925;0.5943;0.5959;0.5963;0.5972]
PKID	0.6596 ± 0.0015 [0.6585;0.6589;0.6590;0.6594;0.6623]
WPKID	0.7347 ± 0.0077 [0.7282;0.7296;0.7337;0.7345;0.7476]
Gaussian (untouched)	0.6548 ± 0.0040 [0.6499;0.6533;0.6541;0.6560;0.6609]
Gaussian (normalized)	0.6629 ± 0.0047 [0.6580;0.6599;0.6611;0.6667;0.6689]

Table 5.8: Area under ROC curve results of using a simple latent model

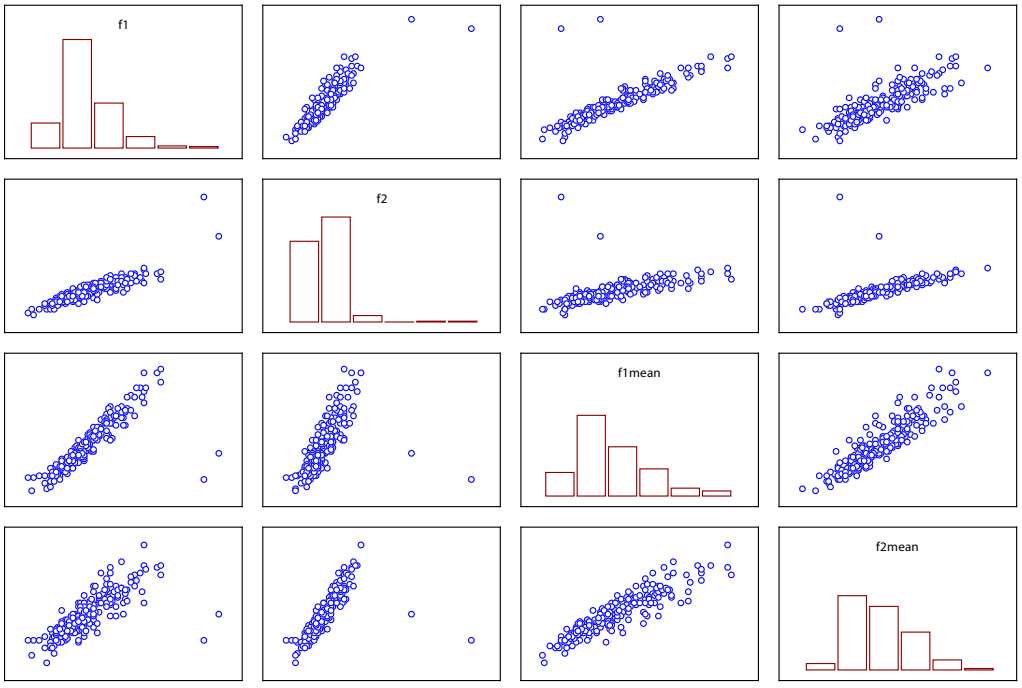


Figure 5.5: Scatter matrix plot of the strongly correlated stellate pattern features

5.7 Dimensionality Reduction

In this section we will conduct several experiments to evaluate the classification performance of the SVM and naïve Bayes classifier after applying dimensionality reduction.

5.7.1 Principal Component Analysis in combination with Naïve Bayes

The classification result of using PCA on the 24 combined MLO and CC features is presented in Figure 5.6, it shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The principal component vectors are calculated using the training set only. These PC vectors are then used to transform both the training and test set. The best result is obtained in 14 dimensions and remains almost constant when adding more dimensions.

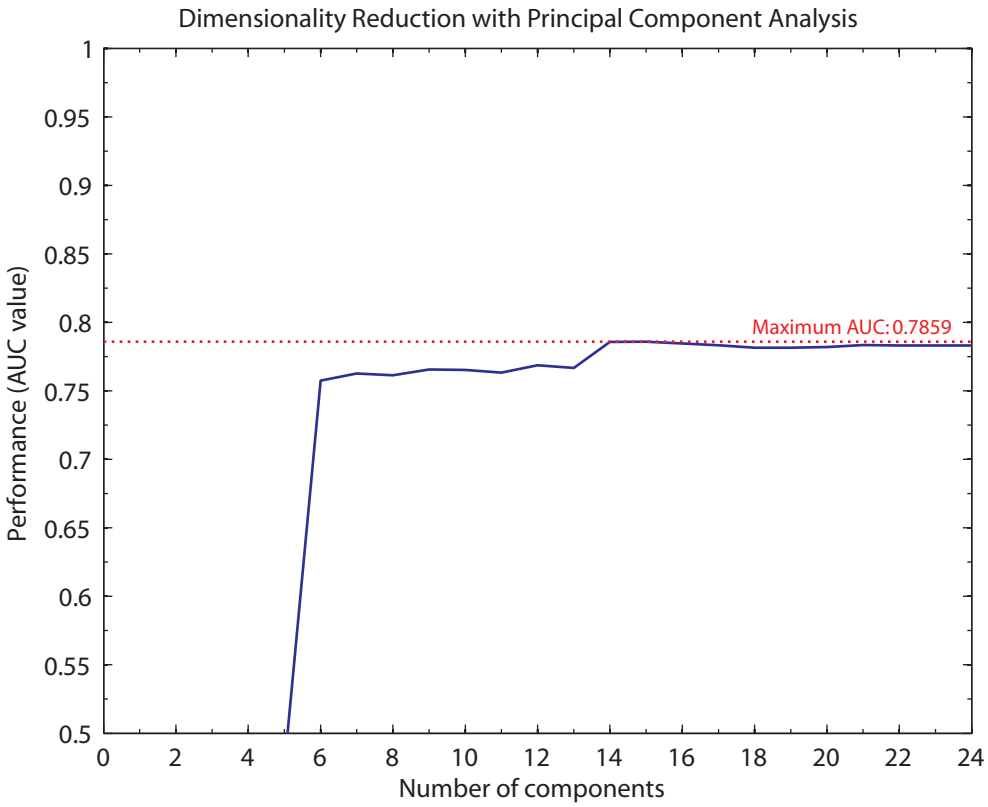


Figure 5.6: Performance naïve Bayes classifier after dimensionality reduction with PCA, averaged over 5 runs.

5.7.2 Fisher Discriminant Analysis in combination with Naïve Bayes

The classification result of using FDA on the 24 combined MLO and CC features is presented in Figure 5.7, it shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The discriminant vectors are calculated using the training set only. These discriminant vectors are then used to transform both the training and test set. The best result is obtained in 10 dimensions and remains almost constant when adding more discriminant vectors.

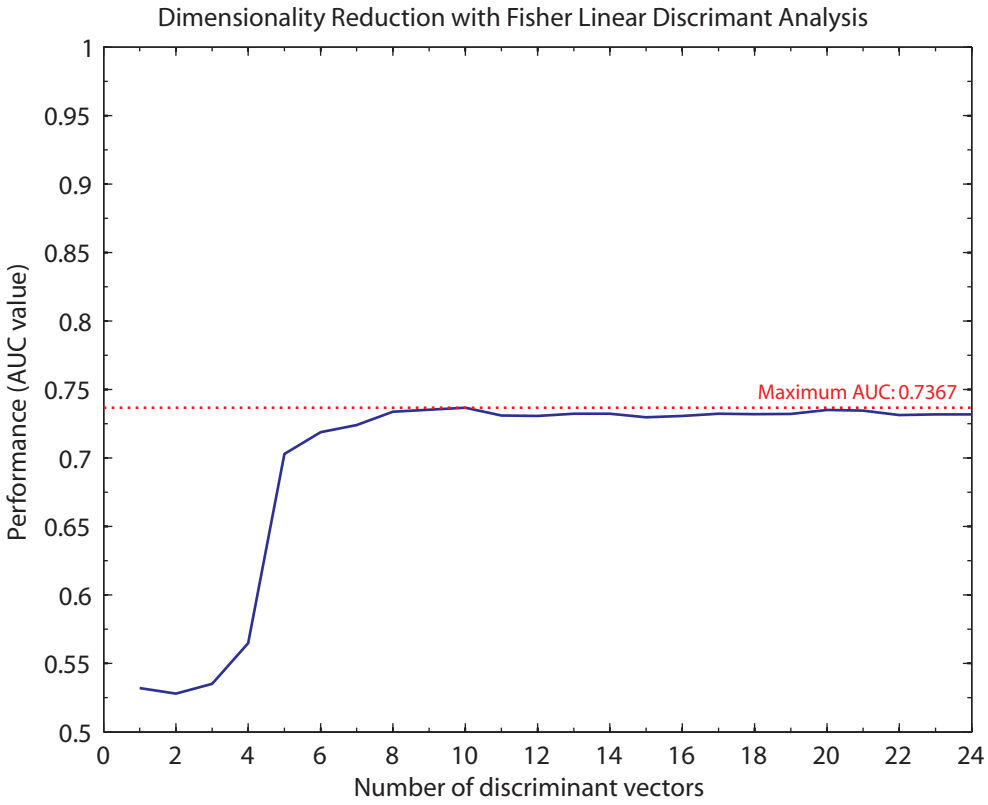


Figure 5.7: Performance naïve Bayes classifier after dimensionality reduction with FDA, averaged over 5 runs.

5.7.3 PCA followed by FDA in combination with NB

In some literature FDA is used in combination with PCA to incorporate class information [Joo03]. The classification result of using FDA on the 24 combined MLO and CC features is presented in Figure 5.8, it shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The principal components and discriminant vectors are calculated using the training set only. The best result is then obtained in 24 dimensions, the same number of features we originally began with.

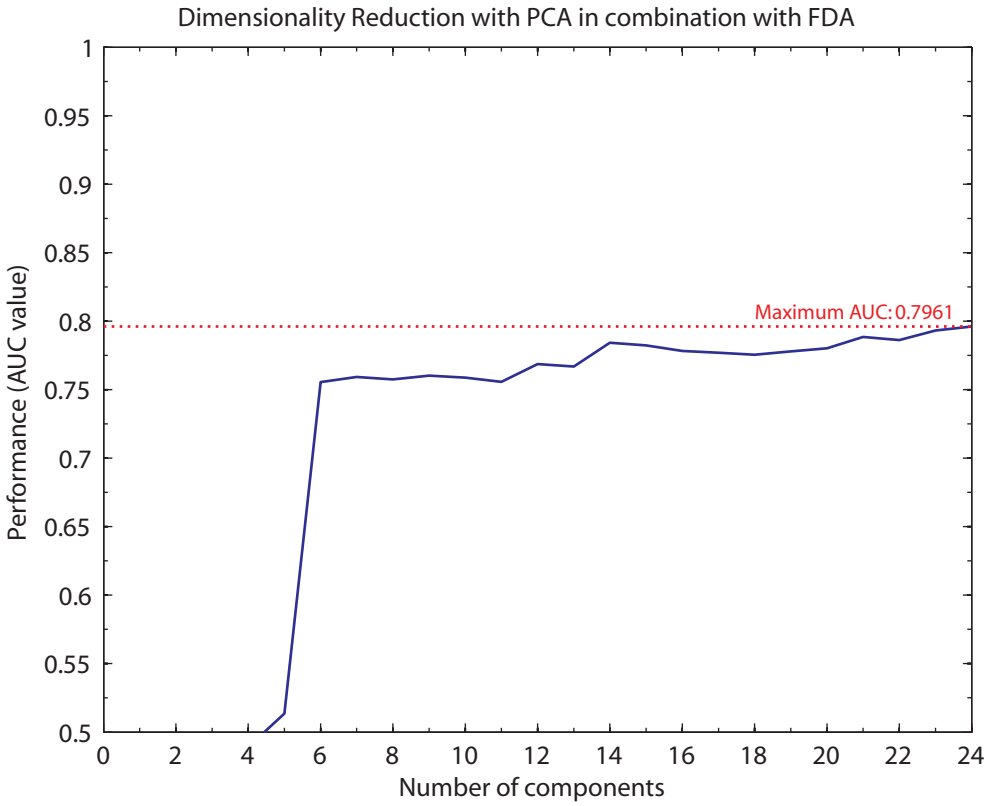


Figure 5.8: Performance naïve Bayes classifier after dimensionality reduction with PCA followed by FDA, averaged over 5 runs.

5.7.4 Principal Component Analysis in combination with SVM

The classification result of using PCA on the 24 combined MLO and CC features is presented in Figure 5.9, it shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The principal component vectors are calculated using the training set only. These PC vectors are then used to transform both the training and test set. The best result is obtained in 6 dimensions. Adding more principal components decreases performance.

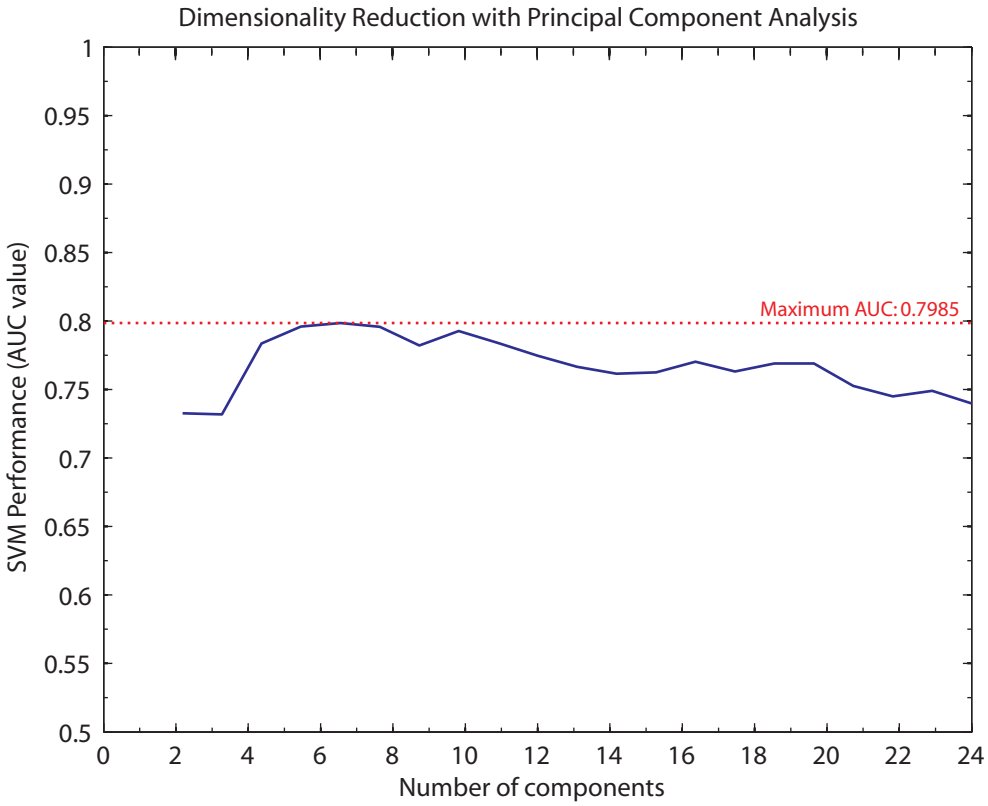


Figure 5.9: Performance SVM classifier with radial kernel function after dimensionality reduction with PCA, averaged over 5 runs.

5.7.5 Principal Component Analysis in combination with SVM using all features

Here we used all the 81 features available for a single view. Some of these features are obviously correlated with each other. We combined the MLO and CC view features in the same way as in the latter experiments, resulting in 162 features per case. Using PCA to reduce the number of dimensions led to the classification results shown in Figure 5.10. It shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The best result is obtained in 10 dimensions.

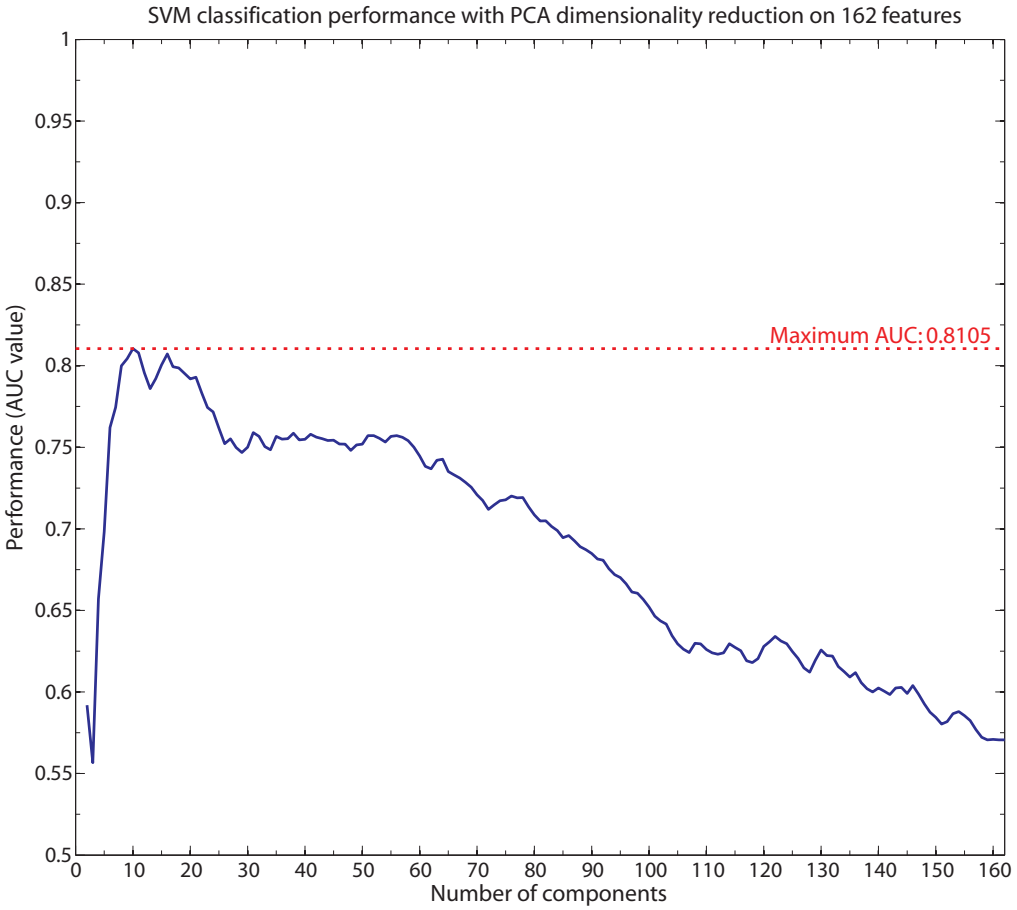


Figure 5.10: Performance SVM classifier with radial kernel function after dimensionality reduction of all features with PCA, averaged over 5 runs.

5.7.6 Fisher Discriminant Analysis in combination with SVM using all features

As with the previous experiment, we used all the 162 features available when combining the MLO and CC view into one case. In this and the following experiment we increased the number of dimensions with steps of 5 in order to decrease the computational time. Using FDA to reduce the number of dimensions led to the classification results shown in Figure 5.11. It shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The best result is obtained if we use all the 160 dimensions, which differs from using PCA where adding more dimensions when the maximum performance is reached will decrease the performance.

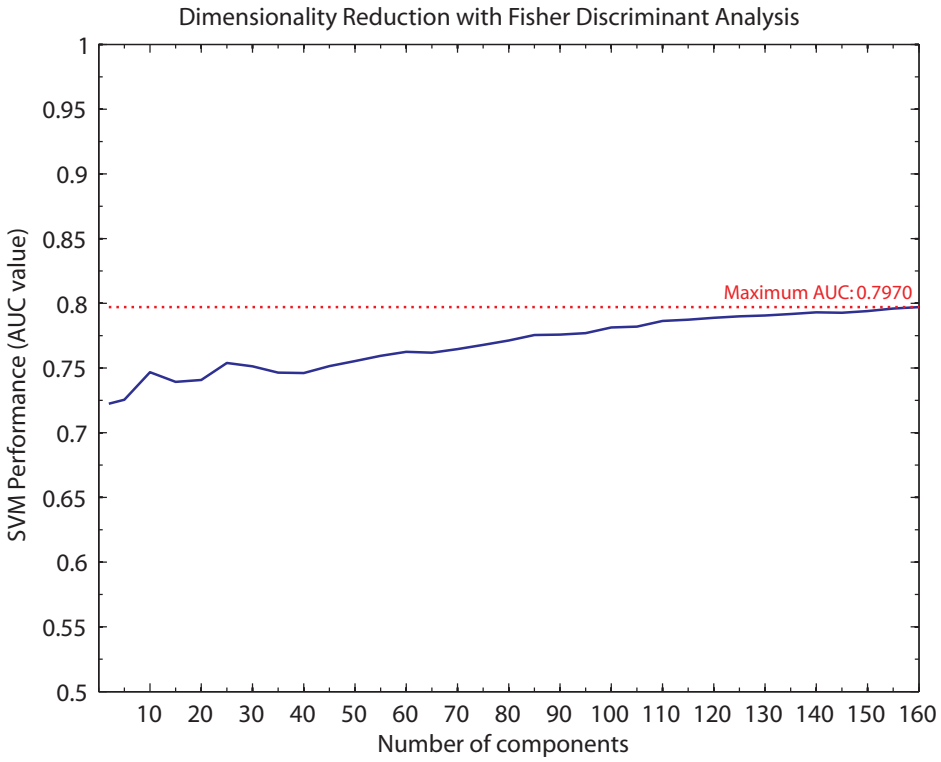


Figure 5.11: Performance SVM classifier with radial kernel function after dimensionality reduction of all features using FDA, averaged over 5 runs.

5.7.7 PCA followed by FDA in combination with SVM using all features

Because FDA is often used in combination with PCA, we have tested the performance of the SVM classifier using the combination of these two dimension reduction methods. Again we use all the 162 features per case and increase the number of dimensions with steps of 5. The results of this experiment is shown in Figure 5.12. It shows the classification performance in terms of the AUC value (y axis) in respect to the dimension reduction (x axis). The best result is obtained if we use 100 or more dimensions and remains almost steady when adding more dimensions.

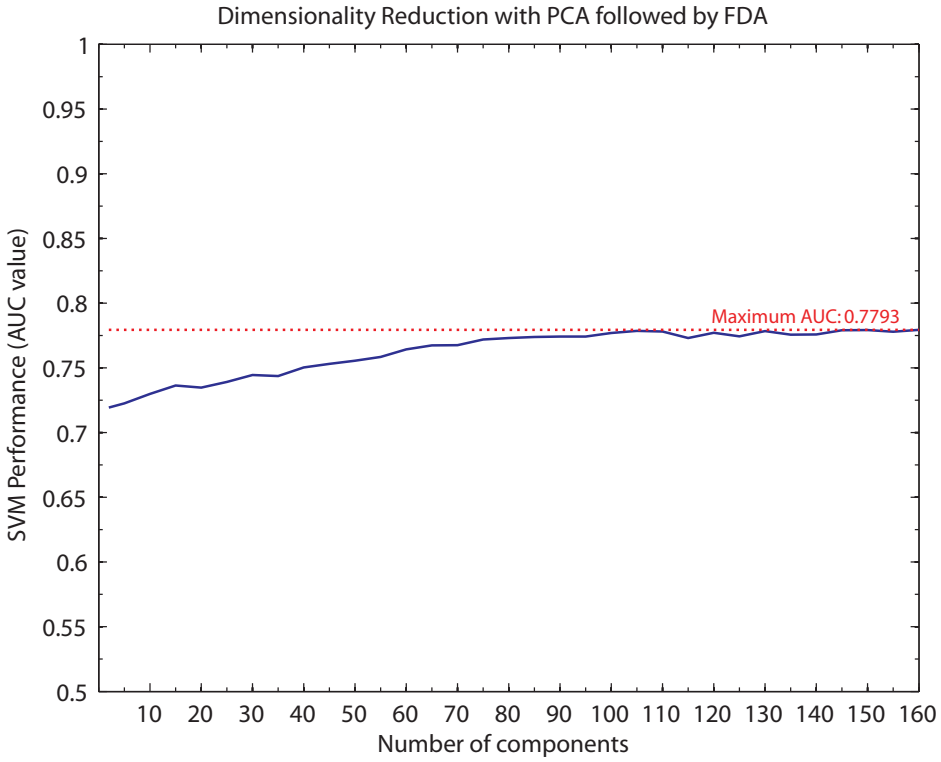


Figure 5.12: Performance SVM classifier with radial kernel function after dimensionality reduction of all features using PCA followed by FDA, averaged over 5 runs.

6 Conclusions and Discussion

6.1 Normalizing

Like most other common Bayesian software packages, BNT [Mur01] assumes that within each state of the class the observed continuous features follow a normal (Gaussian) distribution. The real world dataset we used contains features that do not follow a normal distribution. We showed that the Box-Cox and Manly transformation methods improve classification accuracy by transforming the distribution of the non-normal data closer to the normal distribution. With these methods, the area below the ROC curve increased from approximately 0.767 to 0.795, matching the performance of the SVM classifier. However, we found that the transformation does not work for all data. Transforming features that were already approximately normal even had a bad impact on the classification performance. Also discrete variables were not a useful candidate. Other sophisticated transformation methods have been developed, such as the quadratic discriminant function [UOA02], and folded exponential transformations [Pie03], which could possibly lead to better performance. Another approach for dealing with continuous variables in naïve Bayes classifiers is using the kernel method [JL95], which uses a non-parameterized approximation for a continuous variable by a sum of so called kernels. These approaches or a combination of them could be beneficial but were outside the scope of this thesis.

6.2 Scaling

Support Vector Machines are sensitive to the distribution of the input values, because it uses distances between data vectors. It turns out that if the features are centered and scaled, SVM is more reliable. Commonly, the input features are first centered by subtracting their mean, and scaled by dividing them by their standard deviation. We

used two other scaling methods, one that transform features to a $[-1, +1]$ range, and a supervised scaling method. In our specific application they did not outperform the generally used scaling method.

6.3 Dimension reduction

The most promising algorithm in this work has been the Principal Component Analysis. One major drawback of Principal Component Analysis (PCA) is that it can eliminate the dimension that is best for discriminating positive cases from negative cases, because it is an unsupervised algorithm. If the data points are spread parallel on each side of the linear separator, it is easy to see that the discriminating dimension will be eliminated. Despite this problem, it performed very well with the Support Vector Machine and the naïve Bayes classifier.

6.3.1 Naïve Bayes

Applying the PCA algorithm on the twelve MLO features and the twelve CC features increased the naïve Bayes classifier performance and reached its maximum when using 14 principal components. This is probably due to the fact that PCA components are independent of each other which conforms to the independent assumption of naïve Bayes. Adding more components did not have any noticeable effect on the performance, because the naïve Bayes classifier assigned very low weights to the added components.

The dimension reduction did however not surpass the performance gained using the transformation methods. Transforming the features to a normal distribution after PCA has been applied decreased the classification performance enormously due to the arbitrary distribution of the PCA components. This confirms the observation that these transformation methods only work for certain distributions. Furthermore, it should be noted that the underlying joint probability distribution of the model is not very useful compared to a model with real feature variables where someone could see which feature is responsible for a certain classification.

The Fisher Discriminant Analysis (FDA) is not so commonly used as PCA in this research area. For a classification task FDA is often preferred above PCA because it is a supervised algorithm, i.e., it incorporates class information. Surprisingly, FDA alone was not able to outperform PCA. However using FDA after PCA has been applied, increased the performance significantly.

6.3.2 Support Vector Machines

Applying the PCA algorithm on the twelve MLO features and the twelve CC features did not lead to a significant improvement in performance of the SVM classifier, though the number of features needed for maximum performance was reduced to 6. Adding more principal components decreased the performance enormously, confirming the observation that SVM is highly sensitive to the number of dimensions, because it can be more difficult to find a separating hyperplane when the number of dimensions increases.

The best classification performance we achieved in this work was by using Principal Component Analysis on all available 162 features (81 features per view). The best result was obtained at 10 dimensions, with an area under the ROC curve value of 0.811. Using the supervised FDA algorithm with and without PCA did not lead to better performance.

6.4 Discretization

We investigated several discretization techniques that were confirmed to provide good results when used in combination with a naïve Bayes classifier. The two most simple discretization techniques, Equal Frequency Discretization and Equal Width Discretization, performed quite well when compared to the more complex techniques which are basically variations on the EFD algorithm. Only WPKID accomplished a slight improvement of the classification performance. To avoid the empty cells in the conditional probability tables which gave problems classifying unseen test data, we had to define a uniform dirichlet prior. Lowering the number of intervals to avoid empty cells in the CPT lead only to worse classification performance, due to underfitting the data.

6.5 Latent Models

To relax the conditional independence assumptions embedded in naïve Bayesian models, we experimented with a new class of models, termed latent models. The learned hidden nodes between strongly correlated features did not have a significant impact on the resulting performance. Langseth and Nielsen [LN05] proposed new kinds of latent models, such as the non-linear latent models combining a mixture of Factor Analyzers with a naïve Bayes classifier. Unfortunately, there was no publicly available implementation which we could use in our experiments.

6.6 Combining Classifiers

One question that arises when studying the performance of both classifiers is whether a combination of SVM and NB would lead to an increase in performance. To investigate this we constructed a network with five SVM classifiers for five groups of similar features. The output of the SVM classifiers was then classified with a NB classifier. For our dataset the performance did not improve. One possible reason is that the performance using a single classifier was already high to begin with.

6.7 Receiver Operator Characteristic

The main measure used for classification performance in this thesis is the area below the Receiver-Operator Characteristic curve (i.e., A_z value). It has the advantage that it is widely used, independent of an arbitrarily selectable classification treshold, and also independent of the prior probability of the two classes. A disadvantage is, that classifiers with the same A_z value can have quite different ROC curves. If the consequences of misclassifications are not equally costly then one could prefer one classifier above the other even though they have the same A_z value. The A_z value is therefore only a global quality measure. There are two ways to calculate the A_z value. Using the first method, we have put all the labels and classifier outputs of all folds into one TPF/FPF matrix whereupon we generated the ROC curve. The second method is described in [WE05, BS00] where the final A_z value is calculated by averaging the A_z values of each fold. Using the latter approach results in a slightly higher A_z value. The difference of the resulting A_z values between both methods could be due to the small number of data points per fold and because the cut-off value in the second method can be different for each fold. Therefore we assumed that the first method is the most honest one to calculate the performance and therefore used in all our experiments described in this thesis.

6.8 Classification Performance

The overall conclusion of our thesis is that support vector machines are still the method of choice if the aim is to maximize classification performance. Although Bayesian networks are not primarily designed for classification problems, did not perform drastically lower and in some experiments even slightly better than the support vector machine classifier. However, the real potential of Bayesian networks lies in the qualitative properties of the model, the causal relationships between variables, the possibility to incorporate background information into the model, the capability to deal with missing data, and

the use of hidden variables. To the best of our knowledge, Support Vector Machines lack these features that could play an important role in the future when new datasets are being constructed and more background knowledge become available.

6.9 Future research

Although the research described in this thesis has given a extensive comparison of models and techniques, and extended the work done previously, it has not exhausted the possibilities of classification models. Some suggested extensions of the work done are:

- Capturing the temporal pattern in the sequence of mammograms by incorporating information from prior and current views to model tumor behaviour over time. Originally, this was planned to be part of this thesis but was left out due to time constraints.
- Incorporate medical background knowledge of the breast cancer domain by adding dependencies to the classification model.
- Improved Latent classification models combining a mixture of Factor Analyzers with a naïve Bayes classifier.
- Building hybrid models by combining models of different types like Hidden Markov Model with Support Vector Machines systems to obtain the best of both worlds. An other interesting extension could be Bayesian Kernel Models, which can be viewed as a combination of the Bayesian method and the kernel method. It could tackle the nonlinearly problem with kernels as is done in the SVM approach, and obtain estimation results within the classical Bayesian framework.
- Using a larger mammographic database and extract more features from the images. In addition, non-visual features attached to the images such as age, with/without children and family history could be interesting and relevant as additional attributes for classification.
- Using an entropy-based discretization algorithm, which employs the Minimum Description Length Principle as a preprocessing step to convert the continuous attributes of the data set into discrete values.
- Projecting the data onto a lower dimensional space with kernel based methods. All the non-kernel based dimension reduction algorithms we used have their kernel equivalents, known as KFD and KPCA, which could in theory lead to higher levels of accuracy.

A

Matlab Code and Functions for classifying with Bayesian Networks

The developed Matlab code which was used to classify with Bayesian networks, functions for transformation, discretization and other supporting code, is available from our website http://www.student.ru.nl/m.samulski/. Please note that the BNT toolkit [Mur01] from Kevin Murphy is required to run most of this code. We will briefly explain their purpose:

- MANLY.M
  Implements Manly's exponential transformation (which is a modification of the Box-Cox transform) to make non-normal data resemble normal data by removing skewness.

- JOHNDRAPER.M
  Implements John and Draper's modulus function to adjust non-gaussian kurtosis on symmetric data.

- SEARCHGAMMA_BOXCOX.M, SEARCHGAMMA_MANLY.M, SEARCHGAMMA_JOHNDRAPER.M
  Searches for the optimal gamma for the specific transformation algorithms by using a divide and conquer approach to find the best gamma that results in the lowest skewness or kurtosis.

- DISCRETIZING_EFD.M, DISCRETIZING_EWD.M, DISCRETIZING_NDD.M, DISCRETIZING_PKID.M, DISCRETIZING_WPKID.M
  Used for discretization of a dataset with continuous features applying the discretization algorithms described in this thesis.

- ROC.M
  Estimates the ROC (Receiver Operating Characteristic) curve and the area under the ROC curve (AUC) for a two-class classifier using the trapezoidal rule which

connect the points of the ROC curve with straight lines and sums the resulting triangular areas.

- CV.M, CVINIT.M, CVSPLITPARTITION.M
  These functions implement the cross-validation test method where a set of available feature measurements and output classifier is divided into two parts: one part for training and one part for testing. In this way several different networks, all trained on the training set, can be compared on the test set.

- CV_MLO_CC_SINGLEVIEW.M
  CV_MLO_CC_AVERAGING.M
  CV_MLO_CC_COMBINED.M
  CV_MLO_CC_COMBINED_PCA.M
  CV_MLO_CC_COMBINED_FISHER.M
  CV_MLO_CC_COMBINED_DISCRETE_EFD.M
  CV_MLO_CC_COMBINED_DISCRETE_EWD.M
  CV_MLO_CC_COMBINED_DISCRETE_NDD.M
  CV_MLO_CC_COMBINED_DISCRETE_PKID.M
  CV_MLO_CC_COMBINED_DISCRETE_WPKID.M
  CV_MLO_CC_SINGLEVIEW_HIDDENNODES_GAUSSIAN.M
  CV_MLO_CC_SINGLEVIEW_HIDDENNODES_GMM.M
  CV_MLO_CC_AVERAGING.M
  CV_MLO_CC_AVERAGING_HIDDENNODES_GAUSSIAN.M
  CV_MLO_CC_AVERAGING_HIDDENNODES_GMM.M
  CV_MLO_CC_COMBINED_HIDDENNODES_GAUSSIAN.M
  CV_MLO_CC_COMBINED_HIDDENNODES_GMM.M
  We used a certain naming convention for our Matlab implementations. The CV stands for cross validation. In an early stage of our project we used also LOO (leave-one-out) cross validation, but this code was not used for the experiments in this thesis. This was also the case for MLO_CC, we started out with MLO views only but eventually the dataset with MLO and CC was used in all our experiments. The third part of the filename gives the type of experiment (single view, averaging and combined) corresponding with the naming used in this thesis. All the implementations contain the structure learning algorithms NB, TAN, MWST, K2, K2+T, K2-T, and MCMC and also the transformation and ROC functions are incorporated. Other specific experiments, e.g., dimension reduction, that were conducted are described in the fourth part of the filename (e.g., pca, fisher, hidden nodes, discrete).

- CV_MLO_CC_MIXED_SVM_BN_DISCRETE.M
  CV_MLO_CC_MIXED_SVM_BN_GAUSSIAN.M
  This code was a prototype of a hybrid network, combining five SVM classifiers with one naive Bayes classifier, with discretized and continuous nodes.

B

R Code for Support Vector Machines

In this appendix we briefly explain our developed R code for classifying with support vector machines. It is primarily based on the previous work of Sheila Timp [Tim06], which kindly supplied us her source code. It requires the e1071 library of the Department of Statistics, TU Wien [DHL+05] and the ROC-R package [SSBL04] from the Max-Planck-Institute of Informatics.

- SVM_IMAGEBASED.R, SVM_AVERAGED.R, SVM_COMBINED.R
  R source code for the three types of experiments we have done with the SVM classifier:

  1. Image based, where we do not make a distinction between MLO and CC views
  2. Averaged, where we average the classifier output of the MLO view with the corresponding classifier output of the CC view
  3. Combined, where we combine the 12 features of the MLO view with the 12 features of the CC view

- SVM_IMAGEBASED_LINEAR.R,
  SVM_IMAGEBASED_POLYNOM.R,
  SVM_IMAGEBASED_RADIAL.R,
  SVM_IMAGEBASED_SIGMOID.R
  Code that was used for the evaluation of the available SVM kernels, i.e. the linear, polynomial, radial and the sigmoid kernel

- SVM_COMBINED_PCA.R,
  SVM_COMBINED_FDA_ONLY.R,
  SVM_COMBINED_PCA_FDA.R
  Code for Support Vector Machines using the dimensionality techniques Principal Component Analysis, Fisher Discriminant Analysis, and the combination of the two, using the dataset with 24 features per case (12 MLO features and 12 CC features combined).

- SVM_IMAGEBASED_ALLFEATS_PCA.R,
  SVM_IMAGEBASED_ALLFEATS_FDA_ONLY.R,
  SVM_IMAGEBASED_ALLFEATS_PCA_FDA.R
  Same as the code before, only using the dataset with all available 162 features, 81 per view.

Bibliography

[AZC01] M. Antonie, O.R. Zaiane, and A. Coman. Application of data mining techniques for medical image classification. In *Proceedings of the Second International Workshop on Multimedia Data Mining, MDM/KDD'2001*, San Francisco, CA, USA, August 2001. University of Alberta. Available from: http://www.cs.ualberta.ca/~luiza/mdmkdd01.pdf.

[BBB+00] A. Bazzani, A. Bevilacqua, D. Bollini, R. Campanini, N. Lanconelli, A. Riccardi, and D. Romani. Automatic detection of clustered micro-calcifications using a combined method and an svm classifier. In *5th International Workshop on Digital Mammography*, 2000. Available from: http://www.bo.infn.it/calma/publications/IWDM2000_full.pdf.

[BC64] G. Box and D. Cox. An analysis of transformations. *Journal of Royal Statistical Society, Series B*, 26:211–252, 1964.

[BRS00] E. Burnside, D. Rubin, and R. Schachter. A bayesian network for mammography. In *Proceedings of AMIA Annual Symposium*, pages 106–110, 2000. Available from: http://smi-web.stanford.edu/pubs/SMI_Reports/SMI-2001-0867.pdf.

[BS00] K. Bovis and S. Singh. Detection of masses in mammograms using texture features. In *ICPR00*, volume 2, pages 267–270, 2000. Available from: http://www.dcs.ex.ac.uk/people/jefields/JF_02.pdf.

[Bur98] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998. Available from: http://www.kernel-machines.org/papers/Burges98.ps.gz.

[BWD+00] L. J. W. Burhenne, S. A. Wood, C. J. D'Orsi, S. A. Feig, D. B. Kopans, K. F. O'Shaughnessy, E. A. Sickles, L. Tabar, C. J. Vyborny, and R. A. Castellino. Potential contribution of computer-aided detection to the sensitivity of screening mammography. *Radiology*, 215:554–562, July 2000. Available from: http://radiology.rsnajnls.org/cgi/ijlink?linkType=ABST&journalCode=radiology&resid=215/2/554.

[CAAB98] S. Caulkin, S. Astley, J. Asquith, and C. Boggis. *Sites of occurrence of malignancies in mammograms*, volume 13 of *Digital Mammography Nijmegen*, pages 279–282. Kluwer Academic Publishers, Dordrecht, The Netherlands, first edition, December 1998. Editors: N. Karssemeijer and M. Thijssen and J. Hendriks and L. Erning.

[CBL97] J. Cheng, D. A. Bell, and W. Liu. Learning belief networks from data: An information theory based approach. In *Proceedings of the 6th ACM International Conference on Information and Knowledge Management*, pages 325–331, 1997. Available from: http://www.cs.ualberta.ca/~jcheng/Doc/cikm97.pdf.

[CDK99] I. Christoyianni, E. Dermatas, and G. Kokkinakis. Neural classification of abnormal tissue in digital mammography using statistical features of the texture. In *The 6th IEEE International Conference on Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99*, September 1999.

[CG95] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[CH91] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, January 1991. Available from: http://smi-web.stanford.edu/pubs/SMI_Reports/SMI-91-0355.pdf.

[CL68] C. Chow and C. Lui. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 12:462–467, 1968.

[CV95] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–297, 1995. Available from: http://www.research.att.com/~corinna/papers/support.vector.ps.gz.

[DCBW95] A.P. Dhawan, Y. Chite, C. Bonasso, and K. Wheeler. Radial-basis-function based classification of mammographic microcalcifications using texture features. In *Proceedings of the 17th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 535–536, 1995.

[DHL+05] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. e1071: Misc functions of the department of statistics [online]. September 2005 [cited 1 April 2006]. Available from: http://cran.r-project.org/doc/packages/e1071.pdf.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, second edition, 2001.

[DK95] T. G. Dietterich and E. B. Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Department of Computer Science, Oregon State University, 1995. Available from: http://web.engr.oregonstate.edu/~tgd/publications/tr-bias.ps.gz.

[DKS95] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202, San Francisco, CA, 1995. Morgan Kaufmann. Available from: http://robotics.stanford.edu/users/sahami/papers-dir/disc.ps.

[DL88] J. Duchene and S. Leclercq. An optimal transformation for discriminant principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6), November 1988. Available from: http://www.student.kun.nl/m.samulski/papers/DucheneLeclercq.pdf.

[DP97] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997. Available from: http://www.ics.uci.edu/~pazzani/Publications/MLJ97.pdf.

[DTC+02] S. Duffy, L. Tabr, H. Chen, M. Holmqvist, M. Yen, S. Abdsalah, B. Epstein, E. Frodis, E. Ljungberg, C. Hedborg-Melander, A. Sundbom, M. Tholin, M. Wiege, A. kerlund, H. Wu, T. Tung, Y. Chiu, C. Chiu, C. Huang, R. A. Smith, M. Rosn, M. Stenbeck, and L. Holmberg. The impact of organized mammography service screening on breast carcinoma mortality in seven swedish counties. *Cancer*, 95:458, 2002. Available from: http://www3.interscience.wiley.com/cgi-bin/fulltext/97015455/PDFSTART.

[EJKW+01] L. Eriksson, E. Johansson, N. Kettaneh-Wold, J.Trygg, C. Wikstrm, and S. Wold. *Multi- and Megavariate Data Analysis Part I: Basic Principles and Applications*. Umetrics, second revised and enlarged edition, 2001.

[ER04] C. Edwards and B. Raskutti. The effect of attribute scaling on the performance of support vector machines. In G. I. Webb and X. Yu, editors, *Australian Conference on Artificial Intelligence*, volume 3339 of *Lecture Notes in Computer Science*, pages 500–512. Springer, January 2004. Available from: http://springerlink.metapress.com/(azgszkuhuns021mynsu01l55)/app/home/content.asp?referrer=contribution&format=2&page=1&pagecount=13.

[Era01] P. Erasto. *Support Vector Machines - Backgrounds and Practice*. Academic dissertation for the degree of licentiate of philosophy, Rolf Nevanlinna Helsinki, 2001. Available from: http://www.svms.org/tutorials/Erasto2001.pdf.

[FGG97] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997. Available from: http://www.kyriakides.net/CBCL/references/General_Articles/friedman97bayesian.pdf.

[FL04] I. Flesch and P. Lucas. Markov equivalence in bayesian networks. Technical Report NIII–R0436, Institute for Computing and Information Science, University of Nijmegen, August 2004. Available from: http://www.cs.ru.nl/~peterl/markoveq.pdf.

[Fri97] J. H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997. Available from: http://www.cse.unr.edu/~sushil/class/ml/papers/local/curseDim.pdf.

[Fri98] N. Friedman. The bayesian structural em algorithm. In *Fourteenth Conf. on Uncertainty in Artificial Intelligence (UAI)*, 1998. Available from: http://www.cs.huji.ac.il/~nir/Papers/Fr2.pdf.

[FWB+98] D.B. Fogel, E.C. Wasson, E.M. Boughton, V.W. Porto, and P.J. Angeline. Linear and neural models for classifying breast masses. In *IEEE Transactions on Medical Imaging*, volume 17, pages 485–488, 1998.

[GBC01] C. P. Goscin, C. G. Berman, and R. A. Clark. Magnetic Resonance Imaging of the breast. *Cancer Control*, 8:399–406, 2001. Available from: http://www.moffitt.usf.edu/pubs/ccj//v8n5/pdf/399.pdf.

[Gei92] D. Geiger. An entropy-based learning algorithm of bayesian conditional trees. *UAI'92*, pages 92–97, 1992.

[GI05] R. Gentleman and R. Ihaka. The r project for statistical computing [online]. 2005 [cited 1 April 2006]. Available from: http://www.r-project.org/index.html.

[Gil84] G. Gilbert. Finley's tornado predictions. *The American Meteorological Journal*, 1:166–172, 1884.

[HD72] E. Harris and D. DeMets. Estimation of normal ranges and cumulative proportions by transforming observed distributions to gaussian form. *Clinical Chemistry*, 18(7):605–612, 1972.

[Hec95] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, 1995.

[HHW00] C. Hsu, H. Huang, and T. Wong. Why discretization works for naïve bayesian classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 399–406. Morgan Kaufmann, San Francisco, CA, 2000. Available from: http://www.iis.sinica.edu.tw/~chunnan/DOWNLOADS/chunnan-icml2k.ps.

[Hus04] D. Husmeier. *Introduction to Learning Bayesian Networks from Data*, pages 17–57. Springer, London, United Kingdom, 2004. Editors: Dybowski and Roberts. Available from: http://www.springeronline.com/sgw/cda/pageitems/document/cda_downloaddocument/0,11996,0-0-45-137530-0,00.pdf.

[Inc05] SAS Institute Inc. Statistical discovery software – JMP Statistics and Graphics [online]. October 2005 [cited 1 April 2006]. Available from: http://www.jmp.com/.

[Jac90] V. P. Jackson. The role of US in breast imaging. *Radiology*, 177(2):305–311, 1990. Available from: http://radiology.rsnajnls.org/cgi/reprint/177/2/305.pdf.

[Jan06] R. Jang. DCPR (Data Clustering and Pattern Recognition) [online]. 2006 [cited 1 April 2006]. Available from: http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/.

[JD80] J. John and N. Draper. An alternative family of transformations. *Applied Statistics*, 29(2):190–197, 1980.

[JL95] G.H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, 1995. Available from: http://cll.stanford.edu/~langley/papers/flex.uai95.ps.

[Joo03] S. Joo. Face recognition using pca and fda with intensity normalization. Technical report, Department of Computer Science, University of Maryland, December 2003. Available from: http://www.cs.umd.edu/~swjoo/reports/739Q_report.pdf.

[KL03] S. Keerthi and C.-J Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, 15(7):1667–1689, 2003. Available from: http://www.csie.ntu.edu.tw/~cjlin/papers/limit.ps.gz.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, 1995. Available from: http://robotics.stanford.edu/%7Eronnyk/accEst.ps.

[KOR+04] N. Karssemeijer, J. D. Otten, T. Roelofs, S. van Woudenberg, and J. H. Hendriks. Effect of independent multiple reading of mammograms on detection performance. *Medical Imaging 2004: Image Perception, Observer Performance, and Technology Assessment*, 5372:82–89, May 2004. Available from: http://dx.doi.org/10.1117/12.535225.

[KRSH97] C. Kahn, L. Roberts, K. Shafer, and P. Haddawy. Construction of a bayesian network for mammographic diagnosis of breast cancer. *Computers in Biology and Medicine*, 27(1):19–29, 1997.

[KRW+95] C. Kahn, L. Roberts, K. Wang, D. Jenks, and P. Haddawy. Preliminary investigation of a bayesian network for mammographic diagnosis of breast cancer. In *Proceedings of American Medical Informatics Association Conference*, pages 208–212, 1995. Available from: http://www.mcw.edu/midas/papers/AMIA95-MammoNet.ps.

[KtB96] N. Karssemeijer and G. te Brake. Detection of stellate distortions in mammograms. *IEEE Transactions on Medical Imaging*, 15(5):611–619, 1996.

[Lau92] S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

[LB94] W. Lam and F. Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. *Computational Intelligence*, 10:269–293, July 1994. Available from: http://www.cs.toronto.edu/~fbacchus/Papers/LBCI94.ps.

[LCY06] X. Long, L. Cleveland, and L. Yao. Automatic detection of unstained viable cells in bright field images using a support vector machine with an improved training procedure. *Computers in Biology and Medicine*, 36(4):339–362, April 2006.

[LDP04] J. Listgarten, S. Damaraju, and B. Poulin. Predictive models for breast cancer susceptibility from multiple single nucleotide polymorphisms. *Clin Cancer Res*, 10:2725–2737, 2004.

[Ler04] P. Leray. Bnt structure learning package v1.3 [online]. November 2004 [cited 1 April 2006]. Available from: http://bnt.insa-rouen.fr/SL.html.

[LFK06] S. Li, T. Fevens, and A. Krzyzak. Automatic clinical image segmentation using pathological modeling, pca and svm. *Engineering Applications of Artificial Intelligence*, 19(4):403–410, June 2006.

[LIT92] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. *AAAI'92*, pages 223–228, 1992.

[LL03] H.-T Lin and C.-J Lin. A study on sigmoid kernels for SVM and the training of non-psd kernels by smo-type methods. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, March 2003. Available from: http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf.

[LMJ04] B. Liu, C.E. Metz, and Y. Jiang. An roc comparison of four methods of combining information from multiple images of the same patient. *Medical Physics*, 31(9):2552–2563, September 2004.

[LN05] H. Langseth and T. D. Nielsen. Latent classification models. *Machine Learning*, 59(3):237–265, 2005. Available from: http://www.idi.ntnu.no/~helgel/papers/lcm_tech.pdf.

[LS88] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:157–224, 1988.

[Luc04a] P. J. F. Lucas. Bayesian analysis, pattern analysis, and data mining in health care. *Current Opinion in Critical Care*, 10:399–403, 2004. Available from: http://www.cs.ru.nl/~peterl/current-opinion.pdf.

[Luc04b] P. J. F. Lucas. *Restricted Bayesian network structure learning*, volume 146 of *Advances in Bayesian Networks, Studies in Fuzziness and Soft Computing. Editors: G. A. Gamez, S. Moral, A. Salmeron.*, pages 217–232. Springer-Verlag, Berlin, 2004. Available from: http://www.cs.ru.nl/~peterl/pgm02-lucas.pdf.

[LvdGAH04] P. J. F. Lucas, L. C. van der Gaag, and A. Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine*, 30(3):201–214, 2004. Available from: http://osiris.cs.kun.nl/~perry/hcc/publications/biomedicine_in_health_care.pdf.

[Man76] B. Manly. Exponential data transformations. *The Statistician*, 25:37–42, 1976.

[Man89] Udi Manber. *Introduction to Algorithms: A Creative Approach*. Addison-Wesley, 1989.

[MGD+04] M. Mavroforakis, H. Georgiou, N. Dimitropoulos, D. Cavouras, and S. Theodoridis. Significance analysis of qualitative mammographic features, using linear classifiers, neural networks and support vector machines. *European Journal of Radiology*, 54(1):80–89, 2004.

[MK97] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, New York, 1997.

[Mur01] K. Murphy. The bayes net toolbox for Matlab [online]. 2001 [cited 1 April 2006]. Available from: http://bnt.sourceforge.net/.

[NAL+04] T.W. Nattkemper, B. Arnrich, O. Lichte, W. Timm, A. Degenhard, L. Pointon, C. Hayes, and M.O. Leach. Evaluation of radiological features for breast tumour classification in clinical screening with machine learning methods. *Artificial Intelligence Medical*, 34(2):129–139, 6 2004.

[Nea03] R.E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, Upper Saddle River, NJ, 2003. Available from: http://www.neiu.edu/~reneapol/.

[OFL+03] S. Otto, J. Fracheboud, C. Looman, M. Broeders, R. Boer, J. Hendriks, A. Verbeek, and H. de Koning. Initiation of population-based mammography screening in dutch municipalities and effect on breast-cancer mortality: a systematic review. *The Lancet*, 361(9367):1411–1417, April 2003. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0140673603131327.

[OKH+05] J. D. Otten, N. Karssemeijer, J. H. Hendriks, J. H. Groenewoud, J. Fracheboud, A.L. Verbeek, H. J. de Koning, and R. Holland. Effect of recall rate on earlier screen detection of breast cancers based on the dutch performance indicators. *Journal of the National Cancer Institute*, 97(10):748–754, May 2005.

[Ole93] K. G. Olesen. Causal probabilistic networks with both discrete and continuous variables. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(15):275–279, 1993. Available from: http://www.cs.auc.dk/research/DSS/papers/olesen-93.pdf.

[PCH+00] E. Pisano, E. Cole, B. Hemminger, S. Aylward M. Yaffe, A. Maidment, R. Johnston, M. Williams, M. Niklason, L. Conant, E. Fajardo, L. Kopans, D. Brown, and M. Pizer. Image processing algorithms for digital mammography: A pictorial essay. *Radiographics*, 20(5):1479–1491, 2000. Available from: http://radiographics.rsnajnls.org/cgi/reprint/20/5/1479.pdf.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988. Available from: http://www.amazon.de/exec/obidos/ASIN/1558604790.

[PG04] A. T. Pinhas and H. K. Greenspan. A continuous and probabilistic framework for medical image representation and categorization. *Medical Imaging 2004: PACS and Imaging Informatics*, 5371:230–238, April 2004.

[Pie03] P. Piepho. The folded exponential transformation for proportions. *Journal of the Royal Statistical Society: Series D (The Statistican)*, 52(4):575–589, 2003. Available from: http://dx.doi.org/10.1046/j.0039-0526.2003.00509.x.

[PSSM04] N. Pochet, F. Smet, J.A.K. Suykens, and B. De Moor. Systematic benchmarking of microarray data classification: assessing the role of nonlinearity and dimensionality reduction. *Bioinformatics*, 20(17):3185–3195, 2004.

[Rob77] R. W. Robinson. Counting unlabeled acyclic digraphs. In C. H. C. Little, editor, *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, pages 28–43, Berlin, 1977. Springer-Verlag.

[SS04] A. J. Smola and B. Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004. Available from: www.svms.org/regression/SmSc98.pdf.

[SSBL04] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. Rocr: An r package for visualizing the performance of scoring classifiers [online]. 2004 [cited 1 April 2006]. Available from: http://rocr.bioinf.mpi-sb.mpg.de.

[SSGS06] R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006. Available from: http://jmlr.csail.mit.edu/papers/volume7/silva06a/silva06a.pdf.

[Sta05] Statistics Netherlands. Deaths among the dutch population by main primary cause of death, sex and age on last birthday and sex 1996-2004 [online]. June 2005 [cited 25 September 2005]. Available from: http://statline.cbs.nl.

[tB00] G. te Brake. *Computer Aided Detection of Masses in Digital Mammograms*. PhD in medical sciences, Radboud University Nijmegen, 2000. ISBN 9090133186. Available from: http://webdoc.ubn.kun.nl/mono/b/brake_g_te/compaideo.pdf.

[The03] The Netherlands Cancer Registry. Incidence data: newly diagnosed cancer patients per year of 1989-2003 [online]. 2003 [cited 9 March 2006]. Available from: http://www.ikcnet.nl.

[Tim06] S. Timp. *Analysis of Temporal Mammogram Pairs to Detect and Characterise Mass Lesions*. PhD in medical sciences, Radboud University Nijmegen, 2006. ISBN 9090205500.

[TS05] L. Tesar and D. Smutek. Ultrasonography diagnostics using gaussian mixture model. In *Ultrasonography diagnostics using Gaussian mixture model*, page 94, Berlin, 2005. Josef Stefan Institute.

[TYV+03] L. Tabar, M. F. Yen, B. Vitak, H. H. Chen, R. A. Smith, and S. W. Duffy. Mammography service screening and mortality in breast cancer patients: 20-year follow-up before and after introduction of screening. *The Lancet*, 361(9367):1405–1410, April 2003. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0140673603131431.

[UOA02] H. Ujiie, S. Omachi, and H. Aso. A discriminant function considering normality improvement of the distribution. In *ICPR (2)*, pages 224–227, 2002.

[Ver04] J. J. Verbeek. *Mixture Models for Clustering and Dimension Reduction*. PhD in computer science, University of Amsterdam, 2004. ISBN 9057761254. Available from: http://staff.science.uva.nl/~jverbeek/pub/thesis_verbeek.pdf.

[VRRL96] J.L. Viton, M. Rasigni, G. Rasigni, and A. Llebaria. Method for characterizing masses in digital mammograms. *Optical Engineering*, 35:3453–3459, December 1996.

[VTK06] C. Varela, S. Timp, and N. Karssemeijer. Use of border information in the classification of mammographic masses. *Physics in Medicine and Biology*, January 2006.

[WBMS03] K. E. Witte, M. C. M. Busch, I. T. H. M. Maassen, and A. J. Schuit. Branchereport prevention (dutch). *Ministerie ministerie van Volksgezondheid, Welzijn en Sport*, pages 25–28, 2003. Available from: http://www.brancherapporten.minvws.nl/object_binary/minvws_branche_preventie_00_03.pdf.

[WE05] I. H. Witten and F. Eibe. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005. Available from: http://www.amazon.com/gp/product/0120884070/.

[WZG+99] X. Wang, B. Zheng, W.F. Good, J.L. King, and Y. Chang. Computer assisted diagnosis of breast cancer using a data-driven bayesian belief network. *International Journal of Medical Informatics*, 54(2):115–126, May 1999.

[YW01] Y. Yang and G. I. Webb. Proportional k-interval discretization for naive-bayes classifiers. In *Proceedings of the 12th European Conference on Machine Learning*, pages 564–575, Berlin, 2001. Springer-Verlag. Available from: http://www.csse.monash.edu.au/~webb/Files/YangWebb01.pdf.

[YW02a] Y. Yang and G. I. Webb. A comparative study of discretization methods for naive-bayes classifiers. In *Proceedings of PKAW 2002: The 2002 Pacific Rim Knowledge Acquisition Workshop*, pages 159–173, Tokyo, 2002. Available from: http://www.csse.monash.edu/~webb/Files/YangWebb02a.pdf.

[YW02b] Y. Yang and G. I. Webb. Non-disjoint discretization for naive-bayes classifiers. In *Proceedings of the 19th International Conference on Machine Learning*, pages 399–406, San Francisco, CA, 2002. Morgan Kaufmann. Available from: http://www.csse.monash.edu/~webb/Files/YangWebb02b.pdf.

[YW03a] Y. Yang and G. Webb. On why discretization works for naive-bayes classifiers. In T.D. Gedeon and L.C.C. Fung, editors, *Australian Conference on Artificial Intelligence*, volume 2903 of *Lecture Notes in Computer Science*. Springer, 2003. Available from: http://www.cs.iastate.edu/~honavar/yang-bayes.pdf.

[YW03b] Y. Yang and G. I. Webb. Weighted proportional k-interval discretization for naive-bayes classifiers. In *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 501–512, Berlin, 2003. Springer-Verlag. Available from: http://www.csse.monash.edu.au/~webb/Files/YangWebb03.pdf.

[ZAC02] O.R. Zaiane, M. Antonie, and A. Coman. Mammography classification by an association rule-based classifier. In *Proceedings of the Third International Workshop on Multimedia Data Mining, MDM/KDD'2002*, Edmonton, Alberta, Canada, July 2002. University of Alberta. Available from: http://www.cs.ualberta.ca/~zaiane/postscript/mdmkdd02.pdf.

[ZYHWG99] B. Zheng, W. Yuan-Hsiang, X. Wang, and W.F. Good. Comparison of artificial neural network and bayesian belief network in a computer-assisted diagnosis scheme for mammography. *International Joint Conference on Neural Networks (IJCNN'99)*, pages 4181–4185, July 1999.