Support Vector Machines for Risk Stratification of Childhood Leukaemia

JULIO SOTO

KTH Computer Science and Communication

Master of Science Thesis
Stockholm, Sweden 2009
Support Vector Machines for Risk Stratification of Childhood Leukaemia

JULIO SOTO

Master’s Thesis in Biomedical Engineering (30 ECTS credits)
at the School of Computer Science and Engineering
Royal Institute of Technology year 2009
Supervisor at CSC was Örjan Ekeberg
Examiner was Anders Lansner

TRITA-CSC-E 2009:018
ISRN-KTH/CSC/E--09/018--SE
ISSN-1653-5715
Support vector machines for risk stratification of childhood leukaemia

Abstract
This document describes the methods that were used when implementing support vector machines for classification and stratification of children with acute lymphoblastic leukaemia (ALL), and the underlying variable interaction used by these machines during training.

ALL stands for a third of all patients with cancer and is demonstrated with an overpopulation of immature B- and T-lymphocytes (components of the white blood cells).

In order to give the right treatment, researchers use clinical information of great importance when stratifying patients (prognostic factors). However, with the advance of research new factors emerge and following this, new treatment is developed.

The investigated dichotomous classifier could make prognosis of whether patients were going to die within a five year period with a rather high performance. Further, investigating underlying patterns with structure detection, we were able to find interrelationship between the factors. Parameter search and backward feature extraction resulted in significant improvement of the performance during training whilst principal component analysis in SPSS application program did not influence the performance. Finally, a multi-class support vector machine stratified patients into five risk groups and managed to reclassify four patients into different groups than their original classification.

The procedure of implementation, methods and result are described in detail in the different sections of this document.

Supportvektormaskiner för riskstratifiering av barnleukemi

Sammanfattning
Den här rapporten redovisar de metoder som använts vid konstruktionen av supportvektormaskiner för klassificering och stratifiering av barn med akut lymfatisk leukemi (ALL), såväl som den underliggande variabelinteraktionen som klassificeraren använder vid träning.

ALL står för en tredje del av alla patienter med barncancer och påvisas genom en ökning av omogna B- och T-lymfocytter (komponenter av de vita blodkropparna).

För att varje patient ska få rätt behandling använder forskare klinisk information som är av stor betydelse för stratifiering av patienter (prognostiska faktorer). Denna stratifiering förfinas hela tiden och leder till att behandlingen av patienterna gradvis förbättras.

I det här arbetet, kunde den dikotomiska klassificeraren prognostisera vilka patienter som var döda respektive levande efter fem år av diagnosen med hög prestanda. Vidare, i letandet av underliggande mönster med strukturdetektering, kunde vi hitta inbördes samband mellan de prognostiska faktorerna. Parametersökning och variabelextrahering vid träning av nätet ledde till en betydande förbättring av prestandan medan principalkomponentanalys i applikationsprogrammet SPSS, inte påverkade klassificeringen. Slutligen, en supportvektormaskin för multi-klassproblem stratifierade patienter i fem olika riskgrupper och kunde omklassificera fyra av dem till annan riskgrupp än deras tilldelade.

Implementering, metoder och resultat redovisas utförligt i de olika avsnitten av detta dokument.
Contents

1 Introduction................................................................................................................................ 1
  1.1 Background ......................................................................................................................... 1
  1.2 Problem specification .......................................................................................................... 2
  1.3 Milestones ........................................................................................................................... 2
2 Theory ........................................................................................................................................ 3
  2.1 Acute lymphoblastic leukaemia .......................................................................................... 3
  2.2 Stratification ........................................................................................................................ 4
  2.3 The significance of risk factors ........................................................................................... 5
  2.4 Previous stratification work ............................................................................................... 6
  2.5 Artificial neural networks ................................................................................................... 7
  2.6 Support vector machines ................................................................................................... 9
    2.6.1 Definition ..................................................................................................................... 9
    2.6.2 Primal formulation ..................................................................................................... 10
    2.6.3 Dual formulation ........................................................................................................ 10
    2.6.4 Non-separable patterns ............................................................................................... 10
  2.7 Regression analysis ........................................................................................................... 11
    2.7.1 Simple linear regression ............................................................................................. 11
    2.7.2 Multiple linear regression .......................................................................................... 11
    2.7.3 Logistic regression ..................................................................................................... 11
  2.8 Handling missing values ................................................................................................... 12
    2.8.1 Estimation methods .................................................................................................... 12
    2.8.2 Types of missing values ............................................................................................. 12
3 Data preparation....................................................................................................................... 13
  3.1 Preparing the datasets ........................................................................................................ 13
  3.2 Variable selection .............................................................................................................. 13
  3.3 Missing values ................................................................................................................... 14
  3.4 Further reduction ............................................................................................................... 15
  3.5 Data reduction (PCA) ........................................................................................................ 16
4 SVM implementation............................................................................................................... 19
  4.1 Overall process diagram .................................................................................................... 19
  4.2 Naive support vector machine ........................................................................................... 20
    4.2.1 Scaling ........................................................................................................................ 20
    4.2.2 Using means of performance ..................................................................................... 20
  4.3 Optimizing the SVM ......................................................................................................... 20
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Acute Lymphoblastic Leukaemia</td>
</tr>
<tr>
<td>AML</td>
<td>Acute Myeloid Leukaemia</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>BCFE</td>
<td>Childhood Cancer Research Unit</td>
</tr>
<tr>
<td>BM</td>
<td>Bone Marrow</td>
</tr>
<tr>
<td>CNS</td>
<td>Central Nervous System</td>
</tr>
<tr>
<td>EI</td>
<td>Extra Intensive</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>Hb</td>
<td>Haemoglobin count</td>
</tr>
<tr>
<td>II</td>
<td>Intermediate Intensive</td>
</tr>
<tr>
<td>I</td>
<td>Intensive</td>
</tr>
<tr>
<td>LR</td>
<td>Logistic Regression</td>
</tr>
<tr>
<td>LW</td>
<td>List-wise deletion</td>
</tr>
<tr>
<td>M2</td>
<td>Between 5% and 25% leukemic blasts in the Bone Marrow</td>
</tr>
<tr>
<td>M3</td>
<td>More than 25% leukemic blasts in the Bone Marrow</td>
</tr>
<tr>
<td>MRD</td>
<td>Minimal Residual Disease</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>MCAR</td>
<td>Missing Completely at Random</td>
</tr>
<tr>
<td>MAR</td>
<td>Missing at Random</td>
</tr>
<tr>
<td>MVA</td>
<td>Missing Value Analysis</td>
</tr>
<tr>
<td>MI</td>
<td>Multiple Imputation</td>
</tr>
<tr>
<td>NOPHO</td>
<td>Nordic Society of Pediatric Haematology and Oncology</td>
</tr>
<tr>
<td>OSU-SVM</td>
<td>Support Vector Machine toolbox for MATLAB</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>SI</td>
<td>Standard Intensive</td>
</tr>
<tr>
<td>Sig.</td>
<td>Significance</td>
</tr>
<tr>
<td>SPSS</td>
<td>Statistical Package for the Social Sciences</td>
</tr>
<tr>
<td>Testis ALL</td>
<td>Testicle Acute Lymphoblastic Leukaemia</td>
</tr>
<tr>
<td>Tpk</td>
<td>Thrombocytes count</td>
</tr>
<tr>
<td>VC</td>
<td>Vapnik-Chervonenkis</td>
</tr>
<tr>
<td>VI</td>
<td>Very Intensive</td>
</tr>
<tr>
<td>WBC</td>
<td>White Blood Cells Count</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background

The epidemiologic group of the childhood cancer research unit at the Karolinska Institutet (BCFE) has the task of gathering information on leukaemia and solid tumours patients in Sweden and the other Nordic countries: Norway, Denmark, Iceland and Finland. The data is then used as a basis for analyses aiming to improve the treatment of the disease in terms of increased survival and decreased toxicity.

Leukaemias constitute over one-third of childhood cancer cases and is thus the largest group of cancers in childhood. The most common diagnoses within childhood leukaemia are acute lymphoblastic leukaemia (ALL) and acute myeloid leukaemia (AML), where ALL stands for the majority of the children with the disease. As shown in figure 1.1, within the last decades the survival of patients with ALL has improved prominently, and the Nordic countries nowadays have the highest survival in this diagnosis [Gustafsson, 2000]. This progress has in part been possible because all patients have been treated with the same protocol and then followed up in a timely and systematic manner thanks to the centralized data collection by the BCFE. Other reasons for high survival in Sweden include homogeneity of treatment.

In the ALL treatment protocol there are several risk groups which are selected with respect to different disease properties. The groups are: Standard Intensive (SI), Intermediate Intensive (II), Intensive (I), Extra Intensive (EI) and Very Intensive (VI) [Nopho, 2001]. One important aspect of the treatment is to decide in which risk group a patient is to be included.

The ALL database at BCFE has nearly 4800 patients registered and are divided into treatment regimens named after the risk groups. For each patient there are about 1300 variables stored in the database, from early diagnostic variables to variables used to follow each patient over time.
Within the protocol, after the diagnosis the patient is further stratified according to its outcome risk. The decision of this risk based stratification is made with the help of an algorithm that involves parameters like age, type of ALL, white blood cell count (WBC) at diagnosis, treatment response, etc [Nopho, 2001].

There is a need to correctly identify patients at an early stage of the treatment with respect to stratification properties so that they are given the correct treatment. A lot of effort has been put into making the risk group stratification optimal, but there are still more to be done.

The main problem within child cancer research, especially in ALL, is that there is a large group of patients who has a good prognosis for survival (mostly SIs) [Gustafsson, 1998]. But there is also a minor group with poor prognosis. Which treatment should this minor group receive in order to improve their prognosis? Can we classify patients so that they come into the most likely risk group and in such way get the patients from the poor prognosis group to move to the group with a good prognosis?

The main task of this Master’s project was to build a system using support vector machines that identify a patient’s risk based on previous experiences (training) and bring information about variable interactions used in the process of stratification. Further, using this knowledge, the system would stratify patients into the most likely risk groups.

### 1.2 Problem specification

In order to build a system that correctly classifies patients and to find variable interrelationships, a classification model that uses relevant variables (significant factors) should be elected.

*Can the support vector machine reveal variable relationships in dead/alive classification of patients with ALL after five years of diagnosis?*

If we manage to correct classify patients, and thereafter find variable interactions, the next important question is if the classifier is capable of stratifying patients into multiple risk groups.

*Can a multi-class support vector machine stratify patients using the previously learned methods, parameters, kernels and data?*

### 1.3 Milestones

The system was built using a machine learning technique called support vector machines (SVM), and to make the work possible, the following milestones were completed:

- **Data preparation.** A naive SVM was trained with patients that belong to the NOPHO-92; the data had to be cleaned up from noise and irrelevant variables. Clinical decision was made to reduce the number of variables from 1300 to nearly 100 and an attempt to use principal component analysis (PCA) on the statistical application SPSS for additional reduction was done to find redundant variables.

- **Network optimization.** The classifier was constructed using SVM in MatLab with the package *osu-svm*. The performance of the naive classifier was further improved using parameter search and backward feature extraction with cross-validation.

- **Network evaluation.** For evaluation of the network, logistic regression was done using the same training and testing data.

- **Stratification network.** A multi-class SVM was built in order to stratify the patients into five different risk groups.
2 Theory

This chapter describes the ALL disease, the comparison of methods used in previous automatic stratification works and the definition of the methods used for classification in this thesis. The focus of this theory is the recognition of the most significant risk factors that will be used as the input to our machine.

2.1 Acute lymphoblastic leukaemia

Aetiology

In paediatric cancers, acute leukaemia is the most common malignancy, representing approximately one-third of all patients. It is slightly more common in boys than girls [Gustafsson, 2007]. There might be a genetic factor to why children develop leukaemia, but also environmental factors, such as pollution, chemical contamination of groundwater, maternal abuse of alcohol and cigarettes whilst in the womb etc, but there are no certain evidence to the development of ALL [Samuel, 2005].

Pathophysiology

ALL is an aggressive type of cancer that is manifested by the overproduction of cells in the blood or the bone marrow. In the ALL, the overproduction occurs especially in the bone marrow and the cells that are affected are the lymphoid progenitors.

There are multipotent stem cells in the bone marrow which differentiate into haemopoetic cells. Haemopoetic cells, including the lymphoid progenitors, differentiate further into mature cells. These mature cells (T-cell, - B-cell, erythrocytes, granulocytes, monocytes, and platelets) are found later in blood and the immunological system [Voûte, 1998, Samuel, 2005].

The leukaemia arises when a haemapoetic cell suffers a mutation and is arrested on maturation, but not on the capacity of self renewal [Voûte, 1998]. This accidental situation explains the overproduction of blast cells.

Symptoms

Children with ALL show symptoms that are related to the infiltration of blasts into bone marrow, extramedullary sites and the lymphoid system, which includes the central nervous system (CNS). Symptoms connected to CNS involvement are vomiting, sixth-nerve palsy, papilledema and headache. However, CNS involvement can be found in less than 5% of the children. There are also general symptoms such as fever, fatigue, pallor, weight loss and pain in the bones. Blood samples shows thrombocytopenia with low platelet count, neutropenia and low haemoglobin levels [Samuel, 2005].

Diagnosis

The diagnosis of the ALL includes different clinical tests: morphology, immunophenotyping, bone marrow aspiration or biopsy (also lumbar punction and examination of the cerebrospinal fluid is included) and genetic characteristics. Besides the finding on these tests, there are other important parameters like WBC, sex, age and CNS involvement [Ng, 2000]. All these parameters are important when the clinician has to discriminate ALL from other leukaemia such as AML [Voûte, 1998, Samuel, 2005].

Therapy

Thirty years ago, almost all children diagnosed with ALL died. Thanks to the advances in treatment of childhood ALL (especially chemotherapy) the survival rate today is about 80%
[Gustafsson, 2007]. At present, there is no specific therapy that is standardized over the world. However, treatment in all countries has the same four basic phases: induction, consolidation, maintenance and CNS-directed therapy. In the induction phase, the goal is to bring patient to remission (absence of active cancer) with three or four different drugs. In this phase as many as 95% succeed in achieving remission of the leukaemia. The consolidation phase is used to eliminate the remains of the disease. The maintenance phase is used to prevent relapses and lasts for 18 to 24 months. The CNS-directed therapy phase is given to patients who have CNS-leukaemia at diagnosis and to prevent relapses at the spinal fluid [Voûte, 1998].

Prognosis

The overall most important factors indicating favourable prognosis are age at diagnosis and WBC. At present, three different prognostic risk groups are used: standard risk (SR), which includes the age of the patient when getting diagnosed (children between age 2 and 10 years have a better prognosis and infants with ALL suffer a higher risk of relapse despite chemotherapy treatment). Also included in the SR-group are patients with WBC less than 10 x 10⁹ cells/l. The high risk (HR) group includes patients with WBC greater than 50 x 10⁹ cells/l. or other diagnosed symptoms such as T-cell leukaemia, mediastinal tumour etc. The third group, intermediate risk (IR), includes patients that have all other symptoms and diagnosis.

Some studies report differences between the sexes, where boys seem to have a poorer prognosis than girls, while others report no difference. It is unknown whether sex matters when it comes to prognostic value and in that case, why? [Voûte, 1998]. There also seems to be a prognostic factor of race of the patient, where African American children have a poorer prognosis than Caucasians, but they also have lower incidence of childhood ALL [Samuel, 2005].

2.2 Stratification

After a child is diagnosed with ALL, the next important step is to decide which treatment should be given. This procedure is done by risk stratifying the patient with a set of rules that combines risk factors according to its predictive outcome. Each risk group then has a pre-determined treatment regimen to follow.

Stratification algorithm

In the Nordic countries the stratification is done by using an algorithm which divides the patients into two groups: patients without unfavourable features and patients with unfavourable features.

The unfavourable features are decided to be WBC ≥ 50.1 x 10⁹ cells/l and:

- 11q23/MLL-rearrangements (chromosome 11 rearrangements) or
- t(9;22)(q34;q11)/BCR-ABL (translocation from chromosome 9 to 22) or
- t(1;19)(q23;p13)/E2A-PBX1 (translocation from chromosome 1 to 19) or
- Hypodiploidy (less then 45 chromosomes) or
- T-cell ALL or
- CNS-ALL or
- Testis ALL or
- Poor Response (BM day 15:M3 and/or BM day 29:M2 or M3)

With the help of these features, the patient can further be stratified into five different risk protocols. The group without unfavourable features is divided into two therapies: Standard Intensive and Intermediate Intensive. Here, new criteria are added:

- B-precursoe ALL for both
• For SI: 1 to <10 years old and WBC ≤ 10.0 x 10^9 cells/l
• For II: (age 1 to < 10 years and WBC 10.1 to < 50.0 x 10^9 cells/l) or (age ≥ 10 years old and WBC < 50 x 10^9 cells/l)

The group with unfavourable features is further divided into three therapy groups: Extra intensive, Very intensive and Intensive. New criteria are added here.

For extra intensive therapy, at least one of the following criteria has to be fulfilled:
• WBC ≥ 200.1 x 10^9 cells/l
• Very slow response (BM day 29: M3)
• 11q23 (NB. Age-adjusted)
• t(9;22)
• Hyperhaploidy (less then 34 chromosomes)

For the very intensive therapy, patients that are 5 years old at diagnosis and at least one of these features:
• Patients with T-cell ALL and a mediastinal mass
• Patients with initial WBC of 100 to 200 x 10^9 cells/l.
• Patients with CNS involvement at diagnosis.

For the intensive therapy, the rest of the patients with WBC ≤ 100.0 x 10^9 cells/l are selected [NOPHO, 2000].

As we can see, in the classification of patients the rules are constructed by a previous decision of variable cutpoints. These cutpoints are defined arbitrary by the different child cancer groups and the variation of these cutpoints might lead to a bias [Donadieu, 2000]. The most affected seem to be the continuous variables (i.e. WBC and age).

Risk stratification methods have been deployed by distinct areas of health care and medical research. In some of these areas the stratification is done manually whereas other areas use automatic systems deployed with Bayesian analysis, logistic regression, other statistics techniques and sometimes artificial neural networks. A few examples are APACHE, PRISM, NYS, Parsonnet, Ontario and EuroSCORE [Nilsson, 2005, Bigi, 2004, Rowan, 2007, Chun, 2006, Nashef, 1999]. However, in other studies, some of these statistical automated systems failed to predict mortality when applied to paediatric cancer and they were found inappropriate as a decision system for a single patient [Meyer, 2005]. If a system manages to give a good prognosis, it should be examined and compared with standard methods.

2.3 The significance of risk factors

The most essential requirement to build a stratifying system is the experience gained from databases containing all patients and its principal variables. The extraction of the risk factors from this type of medical databases has been studied since many years ago. In 1983 and with respect to the ALL, D. Miller [Miller, 1983] used multivariate analysis to study patients from 1975 to 1978, and he could identify subsets of patients at low and high probability of treatment failure detecting risk factors. Modern studies have discovered several risk factors and more will arise when factor-detection tools become more powerful. These factors can be classified in three classes: host-related, disease-related and treatment-related [Felix, 2000, Hoelzer, 2005]. A few examples of factors with considerable statistical significance are the following:
• For host-related: age, gender and race
• For disease-related: WBC, T-cell lineage, B-lineage, karyotype, hyperdiploidy, hypodiploidy, t(9,22), t(1,19) and TEL/AML1
• For treatment-related: Protocol, Early response, minimal residual disease (MRD), Day 7 marrow, Day 15 marrow and Day 29 marrow.

Of all these risk factors, WBC and age are the most statistically significant and are related with poor outcome and relapses.

The discovery of new genetic factors might reclassify or delete old factors (i.e. the question if age should be disease-related instead of host related) [Forestier, 2006]. Other factors are changing through the therapy and hence influence the outcome, i.e. MRD could be monitored through the treatment to decide more or less intensive therapy [Schmiegelow, 2001]. Factors that were reported previously can change their role; i.e. endoglin, which was reported before as a potential lineage marker, is related to identification of patients with poor outcome [Catchpoole, 2007]. In fact, placing the whole representation of the human genome on micro-arrays shall provide the mapping of the entire genome-expression profile of the leukaemic cell and help to understand the underlying biological alteration of the leukaemic clone. This can help to improve diagnostic accuracy and therapy protocols [Ross, 2003].

Interrelationship
The question if age should be a disease-related factor can be seen as relation to other factors (as cytogenetically factors). There is a need to consider possible interrelationship between the three classes of risk factors. Variables are receiving statistical significance but some of them are missing explanation of why they are favourable or unfavourable [Felix, 2000]. A clear example is that a recent study has found that there is a strong interaction between birth weight, sex and the risk of developing ALL [Dorak, 2007].

2.4 Previous stratification work

There was none previous automatic stratification work for ALL in the literature. However automatic stratifications were experimented in other areas and are explained here.

Statistics is the standard method for doing research in medicine and health care, but there are several studies where artificial neural networks have been used with equal and sometimes better performance than statistical methods.

In risk stratification of patients after acute myocardial infarction artificial neural network (ANN) was compared to robust Bayesian classifiers (RBC) where the accuracy of the ANN was 70% against 81% for the RBC, with the conclusion that ANN did not improve the prognostic classification of these patients [Bigi, 2004]. In prediction of prostate cancer, a nomogram (two-dimensional diagram designed to allow the approximate graphical computation of a function) gave better accuracy compared to ANN with a result of (70.6% with Mantel Haenszel test, p<0.001) for the nomogram and 67.0% for the ANN [Chun, 2006]. Another study compared ANN and logistic regression on prediction of mortality in head trauma. Here, the ANN outperformed in fields of discrimination and calibration but under performed in accuracy. In 68%, the accuracy of the logistic model was superior to the neural network model [Eftekhar, 2005].

But ANN was better than logistic regression in prediction of gallbladder disease among obese patients. The average correct classification rate of ANNs was better than that of the traditional logistic regression approach (97.14% versus 88.2%). Conclusions stated that ANN might be useful tool to predict risk factors of gallbladder disease on the basis of multiple variables related to laboratory and pathology features [Liew, 2007].
In other health studies, several datasets were used to compare multilayer perceptrons with logistic regression. In these studies, ANN outperformed logistic regression with \( \approx 4\% \) better accuracy for all datasets. Results for one of those datasets were 87\% for the ANN and 82\% for the logistic regression. The suggestion here was that the nature of the data was of primary importance rather than the learning technique [Song, 2004].

In a psychopharmacology study, ANN approach was found to be as valid as traditional multivariate techniques for the analysis. With the original outcome definition (responders/non-responders), ANN performed better than logistic regression (90\% of correct classifications in the training sample vs. 77\%). However, only 62\% of new patients were correctly predicted by ANN for their outcome class [Serretti, 2001].

When using artificial neural network to stratify the length of the stay of cardiac patients based on preoperative and initial postoperative factors, the study demonstrated the suitability of ANN to this outcome prediction task. An individual ANN with the highest discriminating ability produced an accuracy of 81.9\%. The use of ensamples of networks improved the accuracy to 90\% [Rowan, 2007].

In other studies they were winning every other time when using different databases [Sargent, 2001]. This implies that the performance of either ANN or statistical methods depends on the type of database used and its properties.

Support vector machines (SVM) were also tested against linear regression and multiple linear regressions in several databases, expecting SVM as a powerful tool for prediction and pointing out its nonlinearity [Pochet, 2006, Zhao, 2005].

### 2.5 Artificial neural networks

The previous studies showed that ANN and SVM are suitable and powerful tools for stratification of patients. In this work we used SVM as a tool to classify patients diagnosed with ALL. As an introduction to the SVM and to give a basic understanding of supervised learning methods, we start giving a short description of artificial neural networks (ANN).

**Definition**

Our brain is a very complex, nonlinear and parallel computer that processes information (e.g. human vision) sometimes faster than a modern computer. The information is passed from one neuron to the other through links called synapses as shown in figure 2.1. In the similar manner, artificial neural network (ANN) makes its computation through its interconnected units, called neurons. In that way, ANN can solve sophisticated problems, such as pattern recognition, perception and motor control, problems that are impossible or very difficult to solve with conventional programming techniques [Haykin, 1999].

![Figure 2.1. Biological neurons. The signal travels from one neuron through its Axon and reaches the other at their connection called synapse.](image)
ANN was inspired by the way our brain works. Figure 2.2 shows a schematic view of an artificial neuron where the input signals \((x_1, x_2, \ldots, x_m)\) arrive from other neurons; thereafter they are multiplied with their synaptic weights \((w_{k1}, w_{k2}, \ldots, w_{km})\) and processed by the summation and the activation function to finally deliver the output signal. This output signal is further used as an input to the next receiving unit.

\begin{align}
  u_k &= \sum_{j=1}^{m} w_{kj} x_j \\
  v_k &= u_k + b_k \\
  y_k &= \varphi(v_k)
\end{align}

The linear combiner of the model \(u\) (equation 2.1) were \(x\) is the input signal, \(w\) the synaptic weight and \(m\) the number of neurons connected to neuron \(k\), is affined by the bias \(b\), giving the resulting activation potential \(v\) (equation 2.2). This is further modified by the activation function \(\varphi(\cdot)\) (equation 2.3) to finally end with the output of the neuron \(y\).

**Learning**

Unlike the biological neuron that is a compound of about 10,000 signal inputs and many outputs to other neurons, the simple artificial neuron in figure 2.2 (that is called the perceptron) has less inputs and only a single output signal. The learning of this simple model consists in continuously updating the synaptic weights. The output \(y\) is compared with the real outcome, and the discrepancy between the output of the machine and the actual outcome, called the error, is used to update the synaptic weights. This learning method is called the perceptron rule [Haykin, 1999].

The neuron that was described in this section can together with several other neurons, build the so called feed forward multilayer perceptrons networks. The signal of these feed forward networks flows in one direction, from one layer to the next and can be trained with a training rule called backpropagation. As the name of this rule implies, the updating process reaches all the layers in the network, propagating back in the network [Haykin, 1999].
2.6 Support vector machines

2.6.1 Definition
The support vector machine (SVM) is another universal approximator based on statistical learning theory pioneered by Vladimir Vapnik in [Vapnik, 1995]. Like multilayer perceptrons, SVM can be used for pattern classification and nonlinear regression. The main idea of the SVM is to construct a (N-1)-hyperplane to separate samples of N-dimensional vectors into two classes. In figure 2.3, a simple example with 2-dimensional vectors is drawn and the optimal hyperplane results in a single line. The support vectors are those samples that maximize the margin of separation (p) between the positive and negative examples and lay over two parallel hyperplanes. The optimal hyperplane is further found in the middle if these two parallel hyperplanes as in figure 2.3.

In order to make the classification more generalized to new data, the SVM strives to minimize not only the empirical risk (mean of error over training data) but also the structural risk which implies minimization of the Vapnik-Chervonenkis (VC) dimensions (the number of solutions) and becoming more generalized to new data [Haykin, 1999].

![Figure 2.3. The optimal hyperplane and support vectors are found maximizing the margin of separation of the parallel hyperplanes.](image)

If we have \( N \) linearly separable points in space described by expression 2.4 where \( x \) is the input vector and \( d \) is the corresponding desired target output vector and \( d \in \{-1,+1\} \)

\[
\{(x_i, d_i)\}_{i=1}^{N}
\]

(2.4)

the optimal hyperplane can be described by:

\[
w^T x + b = 0
\]

(2.5)

where \( w \) is the weight vector, \( x \) the input vector and \( b \) is the bias, and the closest samples to the optimal hyperplane lay on the following parallel hyperplanes:

\[
w^T x + b = 1
\]

\[
w^T x + b = -1
\]

(2.6)
2.6.2 Primal formulation

The idea is to maximize the distance between the closest sample and the optimal hyperplane (the margin). In the primal formulation, maximizing the margin implies:

Minimize: \[ w^T w, \]
subject to: \[ d_i (w^T x_i + b) \geq 1, \quad i \in \{1..n\} \]

(2.7)

2.6.3 Dual formulation

The dual formulation arises with the introduction of Lagrange multipliers \( \alpha \). These multipliers come to be the support vectors. In the dual formulation, the following equation needs to be maximized:

\[
Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^T x_j
\]

subject to the constraints

(1) \[ \sum_{j=1}^{N} \alpha_j d_j = 0 \]

(2) \[ \alpha_i \geq 0 \quad \text{for } i = 1,2,...,N \]

(2.8)

(2.9)

2.6.4 Non-separable patterns

For the optimal SVM with non-separable patterns, the introduction of inner-product kernels \( K \) in equation 2.10 permits the patterns to be linearly separable in a higher dimensional feature space. The problem becomes to find the Lagrange multipliers that maximize:

\[
Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(x_i, x_j)
\]

subject to the constraints

(1) \[ \sum_{j=1}^{N} \alpha_j d_j = 0 \]

(2) \[ 0 \leq \alpha_i \leq C \quad \text{for } i = 1,2,...,N \]

(2.10)

(2.11)

C is a user parameter and the optimum weight matrix \( w \) with the bias \( b \) as the first component is defined by:

\[ w_o = \sum_{i=1}^{N} \alpha_i d_i \varphi(x_i) \]

(2.12)

where \( \varphi(x_i) \) is the image in the feature space. Two examples of inner-product kernels are showed in equation 2.13:
Polynomial \( K(x, x_i) = (x^T x_i + 1)^p \)

Radial-basis \( K(x, x_i) = \exp\left(-\frac{1}{2\sigma^2}\|x - x_i\|^2\right) \) \hspace{1cm} (2.13)

for \( i = 1, 2, ..., N \)

where power \( p \) for the polynomial kernel is specified \textit{a priori} by the user. The width \( \sigma^2 \), common to all the radial kernels is also specified \textit{a priori} by the user.

### 2.7 Regression analysis

The performance of the SVM was evaluated comparing it with a logistic regression model described with the following introduction.

Regression analysis is a common statistical method that is used in various fields of medicine, especially in risk estimations. The analysis has the aim to investigate the relationship between dependent and independent variables. The regression equation contains the dependent variable (the one which will be predicted), the independent variables (predictors) and the parameters which will be calculated by training with a set of samples, as in ANN. The dependent variables can be continuous, often modeled with linear regression, or dichotomous that can be modeled with logistic regression. Independent variables that are found to have a statistically significant effect on the outcome can be used for further stratification [SPSS, 1999].

#### 2.7.1 Simple linear regression

The simplest equation that has the goal to find the relationship and predict the value of one variable given the value of another has the form:

\[
y = \beta_0 + \beta_1 x + \epsilon \hspace{1cm} (2.14)
\]

Where, \( \beta_0 \) is the intercept (the constant where the line intercepts the vertical axis at \( x=0 \)), \( \beta_i \) is the slope (the ratio between the vertical change and the horizontal change along the line) and \( \epsilon \) is the error that is added because the points do not all fall on the line.

#### 2.7.2 Multiple linear regression

When the independent variables are two or more, the model becomes multidimensional as in equation (2.15). In multiple linear regressions, the new variables can be added separately in order to estimate the effect of each of them. With this method, the simultaneous impact of each of several factors can be estimated, so as to enable identification of a subset of variables which is most effective for estimating the dependent variable [SPSS, 1999]

\[
y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \epsilon \hspace{1cm} (2.15)
\]

Finding a good model and the decision of which variable to include in it can be done in various ways determined by knowledge of the relationship under examination and is a rather complex task.

#### 2.7.3 Logistic regression

Logistic regression is another regression model that is useful in situations when it is necessary to predict the presence or absence of a characteristic or outcome based on values of a set of predictor variables. It works in the similar manner as the linear regression but the difference is that the outcome variable is binary or dichotomous. Logistic regression is applicable to a broader range of research situation than other methods and is widely used in medical and social science [SPSS, 1999].
2.8 Handling missing values

Given that the database which was used in this thesis had too many missing values, it was necessary to find a good method to handle this issue. These methods are explained in this section.

Handling missing values is important in order to gain a good analysis of the data. The values can either be deleted (especially when the number of cases is less than 5%) or imputed. Deleting cases with missing values or imputing the values with any imputation method have the risk of resulting in a biased analysis depending of the type of value.

2.8.1 Estimation methods

There are several methods for handling missing values either for deletion or imputation, the most common are:

- **Listwise deletion** drops the cases that have missing values on all variables in the current analysis. This method is the most recommended.
- **Parwise deletion** drops variables using sub-sample of data and therefore resulting in different calculation parameters (ex. Correlation matrix, mean, covariance matrix).
- **Mean substitution** replaces the missing value with the simple mean. This method was very popular before but is not longer preferred. Imputation of means may reduce the variance of the variable giving a bias to the analysis.
- **Multiple regressions** predict the missing values using non-missing data. A problem with this method is that the prediction may be over-fitted. A solution to this problem can be found by adding a random error to the predicted values.
- **Maximum likelihood estimation (EM)** uses the mean and variance of the non-missing data to calculate the most likely values than other methods for missing values. The method performs an iteration of two steps: finding the expected value (E) and maximizing the expectation (M). This method is widely used.
- **Multiple imputations** generate multiple predictions of the missing values, giving different datasets. The datasets are then iterated comparing the results and giving the average as the estimated values [SPSS, 2007].

2.8.2 Types of missing values

Missing completely at random (MCAR) is the type when the missing values are not dependent on any other variable. Though, the values are distributed randomly over the whole data. The analyzer might use listwise or parwise deletion when the values are of this type. Otherwise, the data should be imputed.

Missing at random (MAR) occurs when the values are not randomly distributed over the whole dataset but are randomly distributed over any subset. When missing values are of this type, the cases might be deleted but more usually imputed.
3 Data preparation

This chapter gives a description of the preparation of the data to be inputted in the classification machine. The database for this work had many parameters and missing values needed to be handled properly.

3.1 Preparing the datasets

As mentioned before, the database that was used for this analysis contains patients that belong to the ALL-92 and ALL-2000 protocols. Together, these databases include 2524 patients diagnosed from 1992 until 2007. In order to recognize rows at latter stage, the first column of the selected dataset contains the identification of patient \( \text{nophonr} \) and the last column was the outcome variable \( \text{dead} \) for the two-class alive/dead prediction and \( \text{protocolslu} \) for the multi-class prediction/stratification). All other variables play the role of independents or predictors. The protocol ALL-92 is, for a while ago, closed and the patients have a follow-up time of more than five years. Hence, it was properly to partition the whole database into two new databases, one for each protocol. After making this partition, we ended with 1648 patients in the ALL-92 dataset and 876 in the ALL-2000. We train and tested the classifier with patients of the ALL-92 protocol.

3.2 Variable selection

With the help of a physician, the dataset was checked for changes, errors, and inconsistencies. In this manner, the number of variables was reduced from 1300 to 86. Further, a reduction was done following the literature concerning the most significant outcome predictors [Miller, 1983, Ng, 2000, Ross, 2003, Samuel, 2005]. Most studies have shown that age, sex, WBC and early response are highly correlated with outcome while other studies discovered additional clinical and biological variables that revealed significance [Catchpoole, 2007, Donadieu, 2000, Ng, 2000]. The dataset was now reduced to 20 most relevant and significant variables. Table 3.1 shows a complete list of the reduced variables and their definition.

<table>
<thead>
<tr>
<th>No</th>
<th>Variable</th>
<th>Description</th>
<th>No</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sex</td>
<td>Male/Female</td>
<td>12</td>
<td>lymfoall</td>
<td>Lymphomatous leukaemia? Yes/No</td>
</tr>
<tr>
<td>2</td>
<td>agear</td>
<td>Age in years</td>
<td>13</td>
<td>mbdown</td>
<td>Down’s syndrom? Yes/No</td>
</tr>
<tr>
<td>3</td>
<td>WBC</td>
<td>White blood cell count</td>
<td>14</td>
<td>cns</td>
<td>CNS disease? Yes/No</td>
</tr>
<tr>
<td>4</td>
<td>Hb</td>
<td>Haemoglobin count (g/l)</td>
<td>15</td>
<td>immkod</td>
<td>Immunofenotype</td>
</tr>
<tr>
<td>5</td>
<td>Tpk</td>
<td>Trombicocites count</td>
<td>16</td>
<td>genkod</td>
<td>Genetic abnormality</td>
</tr>
<tr>
<td>6</td>
<td>hepar</td>
<td>Liver size (cm)</td>
<td>17</td>
<td>protocolpri</td>
<td>Primary treatment protocol</td>
</tr>
<tr>
<td>7</td>
<td>lien</td>
<td>Spleen size (cm)</td>
<td>18</td>
<td>protocolslu</td>
<td>Final treatment protocol</td>
</tr>
<tr>
<td>8</td>
<td>medm</td>
<td>Mediastinal tumour? Yes/No</td>
<td>19</td>
<td>bm15blko</td>
<td>Day 15 number of blast</td>
</tr>
<tr>
<td>9</td>
<td>testis</td>
<td>Testis leukaemia? Yes/No</td>
<td>20</td>
<td>bm29blko</td>
<td>Day 29 number of blast</td>
</tr>
<tr>
<td>10</td>
<td>spleno</td>
<td>Splanomegaly? Yes/No</td>
<td>21</td>
<td>dead</td>
<td>Dead after five years? Yes/No</td>
</tr>
<tr>
<td>11</td>
<td>lgl3cm</td>
<td>LGL greater than 3 cm? Yes/No</td>
<td>22</td>
<td>nophonr</td>
<td>Identification number</td>
</tr>
</tbody>
</table>
3.3 Missing values

Having this reduced dataset, we came into the problem of missing values and which method to use in order to get a complete dataset. Looking at figure 3.1, we can see that the variables containing missing values are: Hb, Tpk, hepar and lien where hepar and lien have more than 5% missing. These variables have numerical values that represent either the amount of molecules or the size of an organ and might be easily imputed using any proper method. The categorical variables with missing values (not shown here) were: medm, testis, spleno, mbdown and cns, and they were replaced with an extra alternative called “Missing”.

In case of using list-wise deletion (deletion of cases that have at least one missing variable), the dataset should decrease with more than 20% and probably give us less improvement to the learning machine. Hence, imputation should be the best option for the missing values. We applied Missing Value Analysis in SPSS version 16.0 to identify if the missing values were MCAR or not. A standard procedure in SPSS states that deleting cases missing up to 5% do not affect the overall classification considerably [SPSS, 2007]. Cases should strictly be deleted only if they are MCAR, otherwise, they should be imputed.

Figure 3.2 shows another output table from the MVA. Variables with less than 5% are not displayed (in this case, Hb and Tpk). From this table we can establish that the missing values are missing complete at random. For example, the mean of agear when the values of hepar are present (5.21), is almost equal to that when the values are missing (5.73). The means do not depend on the missing values and there are similar relations with WBC-hepar, agear-lien and WBC-lien. The complete output of this analysis demonstrates the relation of the missing with all other variables and, there, we could see that there were not significant variations of the means.
Given that the missing values were MCAR, we could easily drop these cases (list-wise deletion). Nevertheless, in order to make comparisons, we also retained these cases, imputed them and built two new dataset instead of one. The imputation of missing values was done using the EM-algorithm.

- **DeadLW.** This dataset comes up after deleting all cases with missing values and setting the outcome or independent variable to the dichotomous `dead(Yes/No)`. Deleting the cases with missing values, the dataset decreases to 1492 rows.

- **DeadEM.** This dataset comes up after imputing the cases with missing values with the Expectation Maximisation (EM) method and setting the independent variable to be the dichotomous `dead(Yes/No)`.

Using the naive classifiers (Linear, Polynomial and Radial Basis with default values), we tested the accuracy of this datasets and achieved the performances described in table 3.2.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Datasets</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DeadLW</td>
<td>DeadEM</td>
<td></td>
</tr>
<tr>
<td>LinearNaiveBinSVM</td>
<td>0.84598</td>
<td>0.84729</td>
<td></td>
</tr>
<tr>
<td>RadialNaiveBinSVM</td>
<td>0.84375</td>
<td>0.84283</td>
<td></td>
</tr>
<tr>
<td>PolyNaiveBinSVM</td>
<td>0.80134</td>
<td>0.79555</td>
<td></td>
</tr>
<tr>
<td>Log. Regression</td>
<td>0.859,</td>
<td>0.867,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.857</td>
<td>0.869</td>
<td></td>
</tr>
</tbody>
</table>

Concerning the two-class problem, the performances of the naive SVM-classifiers listed in table 3.2 shows that the SVM with the linear kernel appears to achieve a better performance compared to the one with radial basis function. The polynomial kernel SVM seems to have difficulties to learn compared to the others. Logistic regression gave better performance because it performed a stepwise variable inclusion (Forward-LR) and extraction (Backward-LR). Both results are displayed in last row of table 3.2.

### 3.4 Further reduction

When we look at the output of the SPSS logistic regression, we observe that the classifier achieved, in the best case, a 100% of correct predicted negative cases and 7% of the positives. The reason for this imbalance might be explained by the extremely large difference in amount of positive and negative cases. Figure 3.3 shows a descriptive analysis of variable `dead`. There are around 85% negative examples in the case of Dead/Alive outcome. This issue may lead to a biased classification making the classifier more likely to predicting negative than the positives cases.

![Figure 3.3. Outcome frequencies. The figure shows an imbalance in the negative and positive training patterns (84.5% negatives and 15.5% positives).](image)
We assumed that the prediction of the SVM-classifiers listed in table 3.2 behaved in the same manner as logistic regression and found that, precisely, the prediction on the negative cases were 100% and on the positives 0%. To solve this problem, we make a simple search loop to find the adequate portion of negative samples to be used in the training sample. Figure 3.4 shows that a reduction of at least 50% of the negative cases should improve the prediction of positive cases. At each step, the performance was measured as a mean of 25 iterations.

![Figure 3.4. Finding a subset of negative samples that increases prediction of positive samples.](image)

After this observation, we made a random reduction of the negative examples to be 50% of the original size. This gave us a modified group of the previous datasets. We reduced the datasets and tested again with the naive SVMs and logistic regression. The overall performance decreased to nearly 75% but we can safely trust that the classifier can predict both classes.

### 3.5 Data reduction (PCA)

The poor prediction accuracy of these classifiers might be influenced by the existence of highly correlated variables (redundant variables). In order to find any correlation between the variables and then reduce the number of predictors, we performed a factor analysis using Principal Component Analysis (PCA) in SPSS. The output of the PCA includes several description tables displaying the eigenvalues, variance and the relation between the variance of each factor and the rotated components. We choose two descriptive tables. The components listed in figure 3.5 are the result of linear combinations of the original variables and are sorted descending on the amount of variance they represent on the data. We choose to exclude the components with eigenvalues less than 0.5 and get 16 principal components. Looking at the rotation sums, this should give us a cumulative variance of 95.9% which implies that we have a loss of only 4.1% of the original information and this is a reasonable amount to lose. Depending on how much variance we are allowed to lose, we could drop more components.
Figure 3.5. Principal component variances. Dropping components with eigenvalues up to 5% we could maintain a cumulative variance of 95.9%.

The matrix on figure 3.6 explains the correlation of the rotated components with the original variables. We can see that the first component is most highly correlated with protocol primary and protocol final. However, protocol final is a better representation of the first component. The same procedure is made on the 16 components. The components 7 to 16 are not displayed here but are found in the complete output of the PCA analysis.

Figure 3.6. Principal component correlation matrix. The figure shows the first six rotated components and their correlation with the original variables.
The variables that are not correlated with any of the components contribute with less variance to the data. In this case, they are Spleen-size, Testis, Lymfo ALL and protocol primary. If we look at the second component, we can see that Testis and Sex are highly correlated with the component and both contain almost the same information.

This analysis is made on both datasets listed in table 3.2 and we obtain two additional datasets with PCA-reduced variables. The four datasets were tested with the naive classifiers resulting in a non-significant variation of performances of ≈0.1%.
4 SVM implementation

In this chapter we describe the construction and optimization of the SVM for binary classification. In section 4.5 and 4.6 we attempt to identify the relationship between variables found by the machine. As an introduction, we present in figure 4.1 a state diagram of the overall process of the work. The diagram covers the main chapters: data preparation, implementation and stratification.

4.1 Overall process diagram

![Diagram of the overall process of building SVMs for classification and stratification of patients with ALL. The input was the single database containing the patients and the outputs are knowledge about underlying variable relationships and stratification of patients.](image)
4.2 Naive support vector machine

The osu-svm package contains predefined functions for training and testing different kernels of a SVM. In order to test each step in the data preparation, we constructed a naive classifier using the linear, polynomial and the radial basis kernel. All of those are listed in table 4.1 with their default parameters.

*Table 4.1 SVM Default parameters. These parameters are used by default in the package osu-svm.*

<table>
<thead>
<tr>
<th>Kernel</th>
<th>C (cost error)</th>
<th>Gamma</th>
<th>Degree</th>
<th>Coeff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radial</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polynomial</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

4.2.1 Scaling

Before the data was passed to the training classifier, it was scaled and normalized to eliminate the predominance of great variances. The scaling was done using the method called Scale in the package. The targets were separated from the patterns to prevent any influence when scaling the patterns. The positive and negative examples were also separated because there were much more negative than positives. The examples were then added proportionally to the train and test patterns. This procedure gave us a better stability in the performance than previous naive training.

4.2.2 Using means of performance

The performance can be measured either by the error or the rate (number of correct predicted samples). We measured the rate of the classifiers after an iteration of 25 times. The results were then given as means of performance with a 95% (±0.01) confidence interval.

The performances of the naive classifiers on each dataset are described in the diverse tables of chapter 6 and the evaluation method used is explained in chapter 5.

4.3 Optimizing the SVM

4.3.1 Parameter search

To find the best parameters for each kernel listed in table 4.1, we implemented a search procedure where the values that yielded best performance were selected. The performance was approximated by means of the 5-fold cross validation method. Given that the datasets were randomly reduced at each run, we assumed that each of them used different parameters. Hence, we applied several iterations of this search procedure to all of them to detect the convergence of the highest performance and find the parameters used. The MatLab-classes *linearXval.m, radialXval.m* and *polyXval.m* gave us the results for each kernel and dataset during the parameter search and are described with the following pseudo-code:
Pseudo code finding parameter values

```
folds ← 5
alldata ← Load data from file
alldata ← Scale(alldata)
for i ← -10 to 5 do
    C = 2^i
    for j ← 1 to folds do
        [train, test] = fold(alldata, folds, f)
        parameters = SVMtrain(train, C)
        ClassRate = SVMtest(test, parameters)
        Rates = [Rates ClassRates]
        MeanRates = [MeanRates mean(Rates)]
        vectorC = [vectorC C]
    plot(vectorC, MeanRates)
```

The pseudo code shows the case of the linear kernel, the parameter C was searched first in a wide interval (C = 2^i in the code) between nearly 0 and 32, as shown in figure 4.2(a), followed by a smooth search nearly its maximum performance figure 4.2(b). The same procedure was done to find the parameters for the remaining kernels (Radial and Polynomial) applied to the datasets: DeadLW, DeadLWPCA, DeadEM and DeadEMPCA. The mean of performances were calculated using a 5-fold cross validation and stored in a vector to be plotted as in figure 4.2.

![Figure 4.2.](image) (a) Groove cross-validation search of parameter C with linear kernel on list-wise deleted and PCA-reduced data. (b) A smooth search of the parameter on the same dataset.
4.3.2 Backward feature extraction

The parameter search in the previous section brought an improvement of about 1% and in order to further improve the performance of the classifiers, a backward feature extraction was added. This procedure might still detect some noisy data and probably give us the best subset of variables for each dataset. Given that the training and testing set were still selected randomly from the entire dataset, we experienced different subsets and performances on different runs. However, the performances differed only in ±1% and we saved the best results to a file for later classification. The MatLab-files in charge of this optimization were LinearXvalFeatureBW, RadialXvalFeatureBW and PolyXvalFeatureBW using their best parameters found in the previous section (C, Gamma and Degree). These classes can be described with the following pseudo-code:

Pseudo code finding best subset of variables

\[
\begin{align*}
\text{alldata} & \leftarrow \text{Load data from file} \\
\text{subset} & \leftarrow \text{All column names} \\
\text{alldata} & \leftarrow \text{Scale(alldata)} \\
\text{folds} & \leftarrow 5 \\
\text{for } i & \leftarrow \text{cols}-1 \text{ to } 1 \text{ do} \\
& \text{tmpsubset } = \text{subset } - i \\
& \text{tmpdata } = \text{alldata(tmpsubset)} \\
& \text{for } j \leftarrow 1 \text{ to } \text{folds} \text{ do} \\
& \quad \text{[train, test] } = \text{fold(tmpdata, folds, f)} \\
& \quad \text{parameters } = \text{SVMtrain(train, C)} \\
& \quad \text{ClassRate } = \text{SVMtest(test, parameters)} \\
& \quad \text{Rates } = [\text{Rates ClassRates}] \\
& \quad \text{if mean(Rates) } >= \text{BestRate then} \\
& \quad & \text{subset } = \text{tmpsubset} \\
& \quad & \text{alldata } = \text{alldata(subset)} \\
& \quad & \text{parameters } = \text{SVMtrain(train, C)} \\
& \quad & \text{ClassRate } = \text{SVMtest(test, parameters)}
\end{align*}
\]

Running these classes, the subsets and performance were not equal at every time. There were variables that were presented more often than others. Other variables appear very rarely and some times not at all. After several runs we could find subsets that brought performance improvement to the classifiers and in order to find the variables most often selected, we made a simple iteration and displayed their frequency in figure 4.3. The variables that were selected less frequently (variables 9, 12, 14, 18 and 20) were testis, lymfoall, cns, protslu and bm29blko.
Figure 4.3. Ranking features, summation of variables selected by the feature selection optimizer after 50 iterations.

Using the best parameters found in the parameter search section and the subsets selected by the feature selection algorithm, we trained and tested each dataset with each kernel, and got the results shown in table 4.2. These SVM performances are the ones who resulted with the highest values during the iteration. In order to compare these results, the datasets were also classified with the logistic regression using forward inclusion and backward exclusion variables and the results are displayed in the last column of the table.

Table 4.2. Subsets of features selected by the feature selection classifier at their best performances.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Kernel</th>
<th>Subset</th>
<th>SVM Performance</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeadLW</td>
<td>Linear</td>
<td>1, 2, 3, 5, 7, 8, 12, 13, 15, 17, 21</td>
<td>0.776</td>
<td>0.753, 0.753</td>
</tr>
<tr>
<td></td>
<td>Radial</td>
<td>2, 3, 8, 11, 12, 13, 15, 17, 18, 19, 20, 21</td>
<td>0.764</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poly.</td>
<td>1, 2, 3, 4, 5, 8, 10, 13, 14, 15, 17, 18, 21</td>
<td>0.76</td>
<td></td>
</tr>
<tr>
<td>DeadLWPCA</td>
<td>Linear</td>
<td>3, 4, 6, 7, 10, 11, 17</td>
<td>0.772</td>
<td>0.753, 0.753</td>
</tr>
<tr>
<td></td>
<td>Radial</td>
<td>2, 3, 6, 7, 10, 11, 14, 17</td>
<td>0.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poly.</td>
<td>2, 3, 7, 8, 9, 10, 11, 12, 13, 17</td>
<td>0.765</td>
<td></td>
</tr>
<tr>
<td>DeadEM</td>
<td>Linear</td>
<td>1, 2, 3, 5, 7, 8, 12, 13, 15, 17, 21</td>
<td>0.785</td>
<td>0.801, 0.794</td>
</tr>
<tr>
<td></td>
<td>Radial</td>
<td>2, 3, 4, 7, 8, 11, 13, 15, 16, 17, 18, 19, 20, 21</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poly.</td>
<td>1, 2, 3, 6, 7, 8, 9, 11, 12, 13, 15, 21</td>
<td>0.757</td>
<td></td>
</tr>
<tr>
<td>DeadEMPCA</td>
<td>Linear</td>
<td>1, 2, 3, 6, 7, 8, 10, 12, 15, 17</td>
<td>0.775</td>
<td>0.801, 0.794</td>
</tr>
<tr>
<td></td>
<td>Radial</td>
<td>2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17</td>
<td>0.763</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poly.</td>
<td>3, 7, 10, 12, 15, 16, 17</td>
<td>0.779</td>
<td></td>
</tr>
</tbody>
</table>


4.4 Extracting the support vectors

The best performance in table 4.2 applying backward feature extraction is the one using the DeadEM dataset with the linear kernel SVM. Although the logistic regression was still better, we were satisfied with this result and used it to analyse the support vectors behind the classification.

Analysing the support vectors can help us to find a semantic explanation of the role they play in the classification and detect their distribution or structure. We assume that they have hidden information about the cutpoints of factors used in the stratification procedure described in section 2.2.

There were 347 support vectors, 171 positives and 176 negatives. The support vectors, their performances and all parameters used in the training moment were saved into a file which we, using their identification number (nophomr), imported and rebuilt two complete SPSS-datasets, one with the positive and the other with the negative support vectors.

Understanding that the support vectors are those patients that lie very close to the decision surface, we knew that these points are difficult to classify. The SVM managed to correctly classify 11 of these 171 points. Nevertheless, we know that these patients reveal hidden variable interactions to our machine. Mathematically, to the SVM, these patients have a common distance to the optimal hyperplane and the discriminate-function that finds this distance with respect to the linear separable problem is:

$$g(x) = w^T_x + b \quad (4.1)$$

Using equation 4.1, patients that have the same distance as the support vectors to the optimal hyperplane, are hard to classify but they carry important information about the variables interaction that helps to decide their outcome risk.

4.5 Structure detection of the support vectors

In order to understand the patterns of the support vectors, we chose the positives and applied a factor analysis for structure detection (a factor analysis method for finding relationships between variables). The principal components extracted from the factor analysis method in SPSS helps us to understand the underlying relationships between the variables. Figure 4.4 shows that the extracted factors with eigenvalues grater than 0.6, represents only 31.5 % of the data. However, the factor analysis extracts the most significant factors and we used these components to analyse the correlations.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Extraction Sums of Squared Loadings</th>
<th>Rotation Sums of Squared Loadings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>% of Variance</td>
</tr>
<tr>
<td>2</td>
<td>0.820</td>
<td>7.451</td>
</tr>
<tr>
<td>3</td>
<td>0.646</td>
<td>5.872</td>
</tr>
<tr>
<td>4</td>
<td>0.546</td>
<td>4.757</td>
</tr>
</tbody>
</table>

*Figure 4.4. Total variance of extracted factor using factor analysis for structure detection in SPSS.*

The rotated components in figure 4.5 help us to determine what the extracted component represents. The first component is most correlated with protocol primary, WBC and immunology-type (group 1), the second component is highly correlated with protocol primary, age at diagnosis, WBC and bone marrow (BM) day-15 blast (group 2) and the third component is correlated with hepar-size and spleen-size (group 3).
Because of the moderately high correlation with the first and second component, protocol primary, age at diagnosis and WBC bridge the groups 1 and 2. WBC is also correlated with the third component, so it serves as a bridge for all three groups. Spleen-size is moderately correlated with the second and third component, thus it bridges group 2 and 3.

<table>
<thead>
<tr>
<th>Rotated Factor Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td><strong>Sex</strong></td>
</tr>
<tr>
<td>protocol primary short</td>
</tr>
<tr>
<td>Age at diagnosis-years</td>
</tr>
<tr>
<td>White blood cells</td>
</tr>
<tr>
<td>Haemoglobin</td>
</tr>
<tr>
<td>Hep-siz</td>
</tr>
<tr>
<td>Spleen-size</td>
</tr>
<tr>
<td>BM day 15-blast code</td>
</tr>
<tr>
<td>Med mass ym</td>
</tr>
<tr>
<td>Mth Down ym</td>
</tr>
<tr>
<td>Immunology-Type</td>
</tr>
</tbody>
</table>

Extraction Method: Principal Axis Factoring.
Rotation Method: Varimax with Kaiser Normalization.

Figure 4.5. Rotated factors vs. variables in Principal Component Analysis for structure detection. High values represent correlations between variables and rotated components.

4.6 Analysing correctly classified test cases

Finally, we extracted the correct classified examples to find additional patterns that were used by the SVM doing the classification.

The test set contains 285 patients (209 negatives and 76 positives). From the negative examples, 203 were classified correctly and from the positive only 18. It means that our machine could recognize exactly 18 of the positive samples but there are still many unknown patients.

Correctly classified negative cases

Observing these recognized negative patients in a SPSS datasheet, we can see that in the negative examples (not dead) 100 percent had No Down syndrome, 94.1 percent were B-precursor ALL and 68 percent had less than 5% blast in the bone marrow. The remaining categorical and continuous variables were almost evenly distributed. In the case of patients being T-cell, the majority were boys, had mediastinal mass and high haemoglobin value. Age, WBC, Hepar and Lien, were growing together.

Correctly classified positive cases

On the 18 recognized dead patients, there were clear combinations of variables. Patients, who were B-precursor and had relative low WBC presented Down syndrome and some augmented size of liver and spleen. T-cell patients had No down but very high WBC, augmented size of liver, spleen and mediastinum. All these 18 dead patients had relative high Haemoglobin.
5 Evaluation method

5.1 Analysing with logistic regression

The statistical classification method that best fitted our datasets was the logistic regression analysis. This method classifies the data in the similar manner as the SVM does. The dataset was divided randomly into training and testing data using the Bernoulli distribution where 70% of the examples were used as the training set and the remaining set was aimed to test the classification. SPSS has a variety of options for tuning the regression and two of them (Forward LR and Backward LR) were used to classify our data.

5.2 The logistic function

The mathematics behind the graphical logistic regression in SPSS is described with the following equations:

\[ f(z) = \frac{1}{1 + e^{-z}} \]  
\[ z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \]

\( f(z) \) in equation 5.1 is called the logistic function and \( z \) is calculated in formula 5.2. Vector \( \beta \) contains the regression coefficients to be calculated, \( x \) is the input vector and \( n \) is the number of predictors. Function \( f \) is almost exactly the sigmoid function used for the activation of the neurons in artificial neural networks with the difference that, in the ANN, exponent \( z \) is multiplied with a slope parameter. The first term \( (\beta_0) \) can be seen as the bias and in that way we find a homology in the classifications.

5.3 Logistic regression optimization

Probability for stepwise

The stepwise procedure allows controlling the criteria of entering or removing variables during the classification. The variables are entered to the model if the probability of their score statistic is less than the Entry value and removed if the probability is greater than the Removal value. The last iteration on Backward LR variable removal using SPSS’s logistic regression is shown in figure 5.1. We can compare this procedure with the Forward/Backward feature selection on the SVM in section 4.3.2.

![Figure 5.1. Logistic Regression Backward LR Variable removals (three last iterations). Using the DeadLW dataset the logistic regression reduced the variables in 17 iterations removing less significant variable at each iteration.](image)
The goodness of prediction is achieved by calculating the regression coefficients using the maximum likelihood estimation on each step.

The logistic regression classifications made on our datasets showed in the previous tables (3.2 and 4.2) and tables in chapter 7, are done using Forward Selection (Likelihood Ratio) and Backward Elimination (Likelihood Ratio). That is why we show two values for each dataset. The first values are the result using Forward LR and the second values using Backward LR. Using the Backward LR, the removal is based on the probability of the likelihood ratio statistic based on the maximum partial likelihood estimates. The Forward LR enters variables based on the significance of the score statistic and remove in the same way as the Backward LR.

<table>
<thead>
<tr>
<th>Step</th>
<th>probfu</th>
<th>B</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>probfu</td>
<td>-.407</td>
<td>.000</td>
</tr>
<tr>
<td>1</td>
<td>constant</td>
<td>-1.928</td>
<td>.000</td>
</tr>
<tr>
<td>2</td>
<td>mbrown</td>
<td>-2.741</td>
<td>.001</td>
</tr>
<tr>
<td>2</td>
<td>protbu</td>
<td>.422</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td>constant</td>
<td>3.469</td>
<td>.029</td>
</tr>
<tr>
<td>3</td>
<td>WBC</td>
<td>.063</td>
<td>.004</td>
</tr>
<tr>
<td>3</td>
<td>mbrown</td>
<td>-2.726</td>
<td>.001</td>
</tr>
<tr>
<td>3</td>
<td>protbu</td>
<td>.323</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td>constant</td>
<td>3.610</td>
<td>.027</td>
</tr>
<tr>
<td>4</td>
<td>age</td>
<td>.008</td>
<td>.028</td>
</tr>
<tr>
<td>4</td>
<td>WBC</td>
<td>.003</td>
<td>.002</td>
</tr>
<tr>
<td>4</td>
<td>mbrown</td>
<td>-2.781</td>
<td>.001</td>
</tr>
<tr>
<td>4</td>
<td>protbu</td>
<td>.249</td>
<td>.002</td>
</tr>
<tr>
<td></td>
<td>constant</td>
<td>3.464</td>
<td>.029</td>
</tr>
</tbody>
</table>

**Figure 5.2. Variable inclusion with logistic regression and Forward LR**

The output of the Forward LR gives us information about variables included at each step, the regression coefficients stored in B and the significance of each entered variable (Sig.) as shown in figure 5.2. A predictor is useful to the model if its significance is lower than 0.05. This value is computed in SPSS from a chi-square distribution with 8 degrees of freedom to test the fit of the logistic model.

Figure 5.3 shows a table containing the performance of the regression on each step. We can see that the best performance (80.1) was obtained on step 3 but the whole classification ended with a performance of 78.4.

**Figure 5.3. Logistic regression performances at each step.**
6 Stratifying with SVM

In this chapter we try to use the support vector machine as a tool for stratification of patients into a treatment protocol.

6.1 Implementation

In order to teach the classifier to stratify patients we modify the datasets and let the outcome variable (Dead Yes/No) be part of the predictors and the final protocol variable (protslu) be the outcome variable. The outcome variable contains now five alternatives labelled from 1 to 5 (SI, II, I, VI and EI) and we created a multi-class problem.

The dataset was then trained and tested with a multi-class SVM. The machine was built with the same methods as in the previous dichotomous problem finding the parameters with a groove and a smooth search algorithm and selecting a subset of features with a backward feature extraction.

For this purpose we used the dataset that yielded best performance in the previous classification (the dataset with the imputed values DeadEM) and all three kernels (LinearXvalFeatureMulti.m, RadialXvalFeatureMulti.m and PolyXvalFeatureMulti.m). The classifier with polynomial kernel achieved best performance and the results (patterns, targets, svm-parameters and classification results) were saved into the files train.mat and test.mat for later analysis.

6.2 Analysing correctly classified patients

Following the procedure in the previous section, we loaded the results from the polynomial kernel stored in the files train.mat and test.mat and extracted the information needed.

There were total 317 cases in the test dataset where 259 were alive and 58 dead patients, the number of features were 10 (with column indexes 1, 2, 3, 4, 8, 9, 12, 14, 15 and 19). From these 58 dead, 42 were classified correctly by our machine and thereafter selected to make a test classification.

Test classification

In order to answer the question about these 42 dead patients (What treatment would these patients have had in order to be alive?), we changed the feature number 19 (Dead Yes = 1) for all these patients to be (Dead No = -1) and made a new classification using the class SVMClass.m in the osu-svm package using the stored parameters. The output from this class is a matrix holding the information about the original class, predicted class and the predicted class if the patient were alive. The SVM stratified four patients to other therapies than their original.
7 Results

7.1 Data preparation

The final set of variables after variable selection in chapter 3 consisted of 20 entries including variables with missing values. Running the Missing Value Analysis we could construct datasets with deleted and imputed cases and compare their behaviour with the SVM. At first glance, testing with the naive classifier, it seemed that both datasets will end the same accuracy but in the end of the data preparation, the imputed dataset showed best performance.

Table 7.1 shows the results after the reduction of negative examples. The performance ended with a value of 75.8% using the linear kernel on the dataset with deleted values. This performance was better than the one reached by the logistic regression (75.3%). The second best performance was obtained using the EM-imputed dataset testing with the linear kernel (75.4%).

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Datasets</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DeadLW</td>
<td>DeadEM</td>
<td></td>
</tr>
<tr>
<td>LinearNaiveBinSVM</td>
<td>0.75799</td>
<td>0.75439</td>
<td></td>
</tr>
<tr>
<td>RadialNaiveBinSVM</td>
<td>0.71676</td>
<td>0.71761</td>
<td></td>
</tr>
<tr>
<td>PolyNaiveBinSVM</td>
<td>0.6525</td>
<td>0.6807</td>
<td></td>
</tr>
<tr>
<td>Log. Regression</td>
<td>0.753, 0.753</td>
<td>0.801, 0.794</td>
<td></td>
</tr>
</tbody>
</table>

Using principal component analysis, we found variable correlations/redundancy and some variables could be excluded from the datasets resulting in two additional datasets. Running our native SVM on these datasets, we achieve performance values displayed in table 7.2. The table contains now both the previous and the new results. Unfortunately, we did not gain significant accuracy of performance at this point (±0.2%), but the polynomial kernel improved its result from 65.3% to 71.4% on the list-wise deleted dataset.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Datasets</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DeadLW</td>
<td>DeadLWPCA</td>
<td>DeadEM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeadEMPCA</td>
</tr>
<tr>
<td>LinearNaiveBinSVM</td>
<td>0.75799</td>
<td>0.75676</td>
<td>0.75439</td>
</tr>
<tr>
<td>RadialNaiveBinSVM</td>
<td>0.71676</td>
<td>0.7129</td>
<td>0.71761</td>
</tr>
<tr>
<td>PolyNaiveBinSVM</td>
<td>0.6525</td>
<td>0.71429</td>
<td>0.6807</td>
</tr>
<tr>
<td>Log. Regression</td>
<td>0.753, 0.753</td>
<td>0.753, 0.753</td>
<td>0.801, 0.794</td>
</tr>
</tbody>
</table>
7.2 Training
Given that the default values used in the package osu-svm were the best values for some datasets and specially using the linear kernel, the naive classifier yielded good performance (75.8% with linear kernel during data preparation).

7.2.1 Parameter search
The 5-fold cross validation parameter search in section 4.3.1 provided the results presented in this section.

In the case of the SVM with a linear kernel, the parameter C at 0.5 begins to converge to the highest performance (76%) for the reduced datasets DeadLW and DeadLWPCA, as shown in figure 7.1(a) and (b). For the imputed datasets, the highest performances were stabilized first at C = 2, see figure 7.1(c) and (d).

![Figure 7.1. Finding cost parameter C with linear kernel on (a) List-Wise case deleted, (b) List-Wise case deletion and PCA variable reduction, (c) Expectation Maximization (EM) value imputation and (d) EM value imputation and PCA variable reduction dataset.](image-url)
In the case of radial kernel, we were alternating between searching the parameters \( C \) and Gamma. One parameter with the best result of the other and vice versa. We see clearly, in figure 7.2 that \( \text{Gamma} \) tends to converge into a quite low value for all datasets (≈0.05) and \( C \) gets a value close to 2, see figure 7.2(b). The values of \( C \) for the remaining datasets (not plotted here) were also greater than 2 and the complete results are listed in table 7.3.

![Figure 7.2](image_url)

*Figure 7.2. Finding parameters \( C \) and \( \text{Gamma} \) on the radial kernel for (a) LW case deletion, (b) LW case deletion and PCA variable reduction, (c) EM value imputation and (d) EM value imputation and PCA variable reduction dataset.*

Searching several times with the polynomial kernel and its parameters, we discovered that the parameters that had influence on the performance were only \( C \) and \( \text{Degree} \). Further, we found that the best value for \( \text{Degree} \) was 1. Finally, the smooth search of \( C \) on every dataset was performed showing the best performance (≈76%) at values of \( C \) nearly 2 in figures 7.3(a) and 7.3(b). The EM-imputed datasets in combination with this polynomial kernel machine showed lower results. See figures 7.3(c) and 7.3(d)
Figure 7.3. Finding cost parameter C on polynomial kernel on (a) LW-deletion, (b) LW-deletion with PCA-reduction, (c) EM-imputation and (d) EM-imputation with PCA-reduction datasets.

As a summary of this phase, we managed to find the best parameters using a groove search followed by a soft search and got the values and performances listed in Table 7.3. We see that all kernels converged to their best parameter values and their performances were stabilized in the range of 0.753 to 0.76. However, the logistic regression was still better on the imputed datasets.

Table 7.3. Finding parameter values on the four final pre-processed datasets. The parameters C and Gamma were searched using the Linear, Radial and Polynomial kernels. Performances are displayed inside the parenthesis and Logistic Regression was done with FW variable inclusion and BW variable extraction.

<table>
<thead>
<tr>
<th>Value</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DeadLW</td>
</tr>
<tr>
<td>Linear C</td>
<td>1 (0.758)</td>
</tr>
<tr>
<td>Radial C</td>
<td>2.5 (0.75351)</td>
</tr>
<tr>
<td>Radial Gamma</td>
<td>0.03, C=2.5</td>
</tr>
<tr>
<td>Poly. C</td>
<td>2 (0.757)</td>
</tr>
<tr>
<td>Log. Regression</td>
<td>0.753, 0.753</td>
</tr>
</tbody>
</table>

We can also observe that almost all performance values had an improvement of around 1% during this search procedure. For the row of Radial Gamma, we used the best C-value when searching the Gamma-value. The logistic regression is also taken in the last row and the two values shown are the results from the forward inclusion and backward variable extraction optimization methods.


7.2.2 Feature extraction

In section 4.3.2 we added an optimization method to our machine (backward feature extraction) and improved the results further. In the results displayed in table 7.4, all machines raised their performance with an amount of about 2 percent. The best of all was the linear kernel on imputed non-PCA applied dataset with a performance of 78.5%.

Table 7.4. Performance of the SVM during backward feature extraction optimization. The results were averaged by means of 5-fold cross validation.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Datasets</th>
<th>DeadLW</th>
<th>DeadLWPCA</th>
<th>DeadEM</th>
<th>DeadEMPCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearXvalBW</td>
<td>0.776</td>
<td>0.772</td>
<td>0.785</td>
<td>0.775</td>
<td></td>
</tr>
<tr>
<td>RadialXvalBW</td>
<td>0.764</td>
<td>0.76</td>
<td>0.77</td>
<td>0.763</td>
<td></td>
</tr>
<tr>
<td>PolyXvalBW</td>
<td>0.76</td>
<td>0.765</td>
<td>0.757</td>
<td>0.779</td>
<td></td>
</tr>
<tr>
<td>Log. Regression</td>
<td>0.753, 0.753</td>
<td>0.753, 0.753</td>
<td>0.801, 0.794</td>
<td>0.801, 0.794</td>
<td></td>
</tr>
</tbody>
</table>

7.2.3 Structure detection of support vectors

The distance of the support vectors to the optimal hyperplane did not tell us much about the nature of the patients represented by them except that these support vectors hide variables interaction in order to compute the distance. Anyway, the structure detection of those support vectors performed in section 4.3 reveals underlying patterns and correlation between variables. There were three groups of correlated variables: group 1 included protocol primary, WBC and immulogy type, group 2 included protocol primary, age, WBC and BM day 15 blast, and group 3 included hepar-size and spleen size. Besides, there were variables that bridged all three groups. Plotting these groups of variables we see three-dimensional clusters of support vectors showed in figures 7.4 and 7.5.

Figure 7.4 A simple plot of the correlated variables in group 1 (protocol primary, immunology type and WBC). Support vectors cluster in higher protocol, low WBC and immunologic code (B-precursor=1).
In group 2 in figure 7.5, the cluster seems to be more evenly distributed over the age but, in fact, age and protocol are growing together. Group 3 (liver-size and spleen-size) also showed a two-dimensional clustering with a radio of 2 near the origin.

### 7.2.4 Observing correctly classified test cases

Observations of the correctly classified positive and negative examples demonstrate clearly that there were additional patterns to the ones found by the structure detection that the SVM used in order to make the recognition.

<table>
<thead>
<tr>
<th>Patients</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>Had No Down syndrome</td>
</tr>
<tr>
<td>94.1%</td>
<td>Were B-precursor patients</td>
</tr>
<tr>
<td>68%</td>
<td>Had less than 5% blast in the bone marrow</td>
</tr>
<tr>
<td>T-cell patients</td>
<td>were boys, had mediastinal-mass and high haemoglobin value</td>
</tr>
</tbody>
</table>

Table 7.5 shows the properties of the correctly classified negative data by our classifier. Down syndrome is a clear factor to our machine, B precursor in combination with low values of WBC, age, hepar and lien is also a feature that the classifier uses to recognize good responding patients.
In the case of positive cases (see table 7.6), patients that seem to belong to a low-risk group (B-precursor and low WBC) died if they had Down syndrome. T-cell in combination with high WBC, augmented liver, spleen and mediastinum size also were properties of dead patients.

Table 7.6. Features of correctly classified positive (dead) test data.

<table>
<thead>
<tr>
<th>Patients</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-B patients</td>
<td>relative low WBC, Down syndrome and some augmented size of liver and spleen</td>
</tr>
<tr>
<td>T-cell patients</td>
<td>No down but very high WBC, augmented size of liver, spleen and mediastinum</td>
</tr>
<tr>
<td>All patients</td>
<td>relative high Haemoglobin</td>
</tr>
</tbody>
</table>

7.3 Evaluation

The classifiers were evaluated and compared with the logistic regression analysis using the four final pre-processed datasets created during the data preparation stage.

Using logistic regression was the best option for classification of our datasets and rapidly reached high performance. The high performances shown in table 7.7 were obtained because logistic regression was optimized using variables inclusion or extraction through an iterated procedure.

Table 7.7. Performance values on Forward variable inclusion and Backward variable extraction for Logistic Regression in SPSS.

<table>
<thead>
<tr>
<th>Method</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DeadLW</td>
</tr>
<tr>
<td>Forward ML</td>
<td>0.753</td>
</tr>
<tr>
<td>Backward ML</td>
<td>0.753</td>
</tr>
</tbody>
</table>

The logistic regression was outperformed by the linear kernel on the missing value deleted dataset, but it was better than the others on the imputed datasets with the performance of 80.1% using the forward variable inclusion.

In SPSS, there are several variable extraction and inclusion methods that can be chosen through the iterations. We used forward ML and backward ML and both methods where getting best performance independently of each other but dependent of the randomly chosen training set. Consequently, the performance of the whole regression increased when these methods where applied.

7.4 Stratification

Using the osu-svm and the optimization steps already learned in the previous section, we could construct a multi-class SVM that gave us the performance displayed on tables 7.8 and 7.9.

7.4.1 Parameter search

All kernels reached a rather good performance during the parameter search procedure, but the linear and radial kernel used very high cost values, see table 7.8.
Table 7.8. Parameter search on the multi-class SVM.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Performance</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearXvalMulti</td>
<td>0.7768</td>
<td>C: 32</td>
</tr>
<tr>
<td>RadialXvalMulti</td>
<td>0.6937</td>
<td>C: 32, Gamma: 0.125</td>
</tr>
<tr>
<td>PolyXvalMulti</td>
<td>0.74</td>
<td>C: 2, Degree: 2</td>
</tr>
</tbody>
</table>

The best performance (77.7%) in this step was obtained by the linear kernel but with a high cost value C = 32, and the second best performance (74%) was running the polynomial kernel using a low cost value C = 2.

7.4.2 Feature extraction

The results in table 7.9 show that the best performance value was achieved by the polynomial kernel (82.6%) when we applied the feature extraction procedure. All parameters, training and testing samples, and performance were stored in a file for the further classification.

Table 7.9. Multiclass performance during feature extraction.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearXvalFeatureMulti</td>
<td>0.7949</td>
</tr>
<tr>
<td>RadialXvalFeatureMulti</td>
<td>0.7618</td>
</tr>
<tr>
<td>PolyXvalFeatureMulti</td>
<td>0.8265</td>
</tr>
</tbody>
</table>

7.4.3 Test sample classification

Reading the positive samples (dead patients) from the file saved in the previous section and trying to classify them as if they were alive, we manage to identify 4 patients who should belong to different treatment group. Looking at table 7.10, patient (a) should belong to intermediate (2) therapy instead of standard (1), patient (b) to extra intensive (5) instead of very intensive (4), patient (c) to very intensive (4) instead of extra intensive (5) and patient (d) to intensive (3) instead of extra intensive (5).

Table 7.10. Test classification. Patients (a, b, c and d) where classified to other therapy. Original class is the original classification of the patient. Predicted class is classification made in the training moment.

<table>
<thead>
<tr>
<th>Target type</th>
<th>Patients</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original class</td>
<td>a</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Predicted class</td>
<td>b</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Predicted class (if alive)</td>
<td>c</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Predicted class (if alive)</td>
<td>d</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
From the 42 samples used in this classification, only 4 patients showed that they should have had other therapy than the original (last row in table 7.10). The remaining patients still obtained the same classification.
8 Discussion

8.1 Data preparation

In pattern recognition and classification problems, the nature and integrity of the data is very important for the quality of the results. For example, when we classify images we do not fall into this problem, we just recode the images into a format that is readable to the classifier and do our experiments. In registers, the problem is that variables are most likely to be represented in different formats. Some variables can have continuous values whereas other have categorical, the variance of the variables differs from each other, there might be outliers and, most importantly, there always exist missing values.

Handling missing values

In chapter 3 we attempt to solve these problems using SPSS tools and we manage to achieve satisfactory results, especially with the imputation method Expectation Maximization. Despite this improvement, during the process of this work we realized that there was still some missing information that could be of importance to our machine. For example, some laboratory values, that are continuous of nature, were categorically coded. Genetic, phenotype and other information as MRD were not complete due to lack of importance until nowadays. This missing information might be helpful for classification accuracy.

Data reduction

Unfortunately, the principal component analysis performed in SPSS neither decreased nor increased the performance of our datasets. Perhaps, we needed to do some data recoding (categorical variables can be recoded into dummy binary variables) or scaling before applying PCA (scaling was done only in the SVM but not in PCA). These two procedures could help to stabilize the overall variance respectively eliminate the predominance of great variances.

8.2 Implementation

The implementation of the SVM functions in MatLab includes steps that might be slightly complex. However, using the osu-svm package we have functions for scaling, training and testing the machine and we need only to focus on data preparation, parameter tuning and other optimization tasks.

Parameter search

The methods applied in chapter 4, in the optimization section, managed to raise the performance and even if the results did not reach high values (75.8% with the linear kernel), they fulfilled their assigned tasks.

The search of the parameters was a greedy search that rapidly found the global maxima of the performance using a large followed by a short region of search. This optimization never brought the performance to levels over 76% because the nature of the data discussed in the previous section (either we need more information or we need to improve the data preparation). This affirmation can be proved with the results of the Logistic Regression which had a performance of 75% on the LW-deleted datasets.

Backward feature extraction

More significant increase of performance was reached applying the feature extraction algorithm (≈2%). This implies that this optimization had impact on the accuracy of the classifier and
should be further experimented with. In this work we only applied the backward feature ex-tract-ion but the forward feature inclusion implementation could also be added. Even a simultaneous use of them could bring improvement.

**Structure detection of support vectors**

As mentioned before, the most important findings to physicians could be a semantic explanation about the underlying information in the support vectors. This information could serve as a basis for defining dynamical cutpoints in order to correctly stratify patients.

In contrast with statistical methods that give statistical results and interpretation, the SVM brings underlying patterns used during the training moment stored in the support vectors. The cutpoints used nowadays for WBC, age, response and others variables that serve to stratify patients are varying in protocols and countries and it is difficult, if not impossible, to know which are the optimal. In that way, the support vectors are those points that carry this type of information given that they lie closest to the decision surface or separating hyperplane. We can also assume that these cutpoints are not fixed points. Instead, they are depending on each other, the increase of a single or two variables might imply the increase or decrease of the other and vice versa. They have an interrelationship. In section 7.2 we described and visualized a few examples that describe part of this relationship. Finding the exact values for cutpoints of these factors is out of the scope of this thesis.

**Observing correctly classified cases**

Two more examples about patterns recognized by the classifier were found when we observed the correctly predicted positive and negative examples. The results of this observation listed in tables 7.5 for the negative samples and table 7.6 for the positives showed some clear properties (for example, *No Down syndrome* for all alive patients) and some variable interactions as for example *T-cell* in combination with very high *WBC*, augmented size of *liver*, *spleen* and *mediastinum* were unfavourable and these patients were classified as positives (Dead Yes=1). Some of these assumptions and predictor factors are already known by physicians. Nevertheless, the classifier had not previous knowledge in order to find this solution.

### 8.3 Evaluation

There are several methods that could be used for evaluation of both the binary and multi-class problem. We used the logistic regression because of its method of classification was close to our problem and the underlying mathematical computation is almost equal to that of the artificial neural network.

**Logistic regression options**

Classifying with logistic regression in SPSS has a wide range of options to optimize the results as variable recoding functions, variable inclusion or extraction and more. We used stepwise backward variable extraction and forward variable inclusion and improved the logistic regression's performance extensively. The inclusion of additional optimization functions could improve the performance further, but this optimization was out of the scope of our task.

Given that the logistic regression is widely used in medical, social and marketing applications and the performance results were high [Bigi, 2004], the option of this method was proper to compare to SVM.
8.4 Stratification

A machine that can be of interest might be the stratifier, as we can call it. Even if we know that physicians prefer to stratify patients by their methods, a stratification machine might contribute with suggestions about which therapy should be more adequate to a specific patient.

Implementation

The stratification machine that we constructed in chapter 6 is a nonlinear multi-classification machine that uses patient properties as input and therapy protocols as its output. In contrast with statistical methods, the multi-class SVM could be constructed without complexity due to the available classes implemented previously in the two-class problem.

Optimization

During the parameter search and feature extraction optimization steps, the polynomial kernel outperformed the other two and the machine reached 82.6% in performance, which was satisfactory. Even though the other kernels obtained reasonable performance (linear=79.5% and radial=76.2%) we could not find any reason for why they used a very high cost value (C=32). Perhaps, it has to do with the data preparation or machine optimization issues named in sections 8.1 and 8.2. Furthermore we used only one dataset while training and were not able to know how they would behave with the other. The crucial result to us is that we managed to get a rather high performance with the polynomial kernel and used its parameters to perform our experiment.

Classification of correctly classified samples

The experiment done in section 6.2.1 ended in that our machine could stratify four patients from the 42 correctly classified to other therapy than their original. This very low identification implies that our most important variable (Dead) was not as significant as we expected. The motivation for that is that there could be more underlying information missing. We need to include those requested variables (clinical and biological) mentioned in section 8.1. Another clear reason is that the classifier needs to learn more about patients with dead attribute (there were 137 dead and 604 non-dead patients in the training set). Once more, there is a need of more positive (Dead) samples in order to do a reasonable calibration of negative and positive examples.

8.5 Future work

Data preparation

In order to improve classification of patients, the database needs to be completed with more information about the disease, it is necessary to include additional information in the domains of host-related, disease-related and treatment-related factors. More extended information might contribute to better data classification.

While preparing the data, there is a task that could be worth to try in order to stabilize the training, recoding the categorical variables into a set of dummy binary variables (several variables that have values 0 or 1 and together represent a single categorical variable). At least, for those variables that have up to five categories.

Missing values might also be imputed with an alternative method not used in this thesis in order to compare the accuracy against the one used in section 3.3. A modern method that is widely used is the Multiple Imputation (MI). This procedure creates multiple imputed datasets which are then analysed with traditional procedures. The results are then combined to generate statistical inferences [SPSS, 2007].
It is very satisfactory that we have a low level of dead patients in the registry (255 dead against 1393 alive), but not for our classifier. We do not mean that we want more patients to die; instead, an important task could be to merge with other leukaemia registers in order to increase positive cases (dead patients) and followed gain improvement on the prediction of those. It is important that the classifier learns to classify both classes and not end into the problem as in section 3.1.3.

Implementation

When we look back at our implementation, we realize that there is still a lot optimization work to be done. For example, we can focus more in making deep search of kernel parameters through the greedy search algorithm or scatter search. Scatter search is an iterative algorithm where the best values are stored in a vector; new values are created through a linear combination of these results and used as a starting point to the new iteration [Kelly, 1996].

Given that we achieved good result in the feature extraction moment, there should be further investigation on this matter. The subsets of features were different when we used support SVM or logistic regression and even when we used the same machine several times. It was not surprising because we used a random partition of training a test sets. Hence, Forward feature inclusion should be implemented and even a combination of both could be examined in order to make an improved selection of the best subset.

Another important observation of the data which we did not take account in this thesis is that there were variables that were time-dependent. For example: response to treatment, MRD, haemoglobin, WBC, age and more. These variables have different input values at a specific time-point of the treatment and bring answers of how the treatment has behaved until this time. It is obvious that these time-dependent measurements affect the overall prediction in additional manner. Therefore, taking this in account, a future machine to detect these sudden changes could for example be a dynamically driven recurrent network.

Evaluation

Logistic regression was a proper evaluation method for the classification of the binary problem. The rapidly convergence of performance on our datasets demonstrated that it is a powerful tool. In addition, an emphasized work on its data preparation before classification could give better results. For example, SPSS has a data preparation set of procedures that could be useful:

- Metadata preparation can review the variables, and determine their valid values, labels and measurement levels. It can also identify combination of variables that are impossible but commonly miscoded and define validation rules base on this information.
- Data validation checks against defined validation rules to identify invalid cases, variables and data values.
- Model preparation identifies potential statistical outliers that can cause problems to the predictive models. These outliers are usually unidentified invalid cases.

For multi-class prediction models, the following strategies could be used as evaluation methods:

- Multiple logistic regressions which do logistic regression for each pair of categorical outcome options
- Linear regression where the categorical outcome variable can be transformed to be continuous and.
- Multiple discriminant analysis which is a technique for group classification.

However, there are some data consideration and limitation that have to be studied in order to obtain good results. In linear regression, for example, all categorical variables have to be re-coded to binary dummy variables, the distribution of the dependent variables must to be normal.
and the relationship between the dependent and each independent variable should be linear [SPSS, 2007].

8.6 Summary

Analysing the support vectors used by our implemented SVM for five years dead/alive classification of patients with childhood ALL, we could find multiple variable interactions. Further, we attempted to build an automatic stratification SVM to recognize appropriate treatment for patients. In the process of building these SVMs we dealt with the following questions:

Can the support vector machine reveal variable relationships in dead/alive classification of patients with ALL after five years of diagnosis?

Can a multi-class support vector machine stratify patients using the previously learned methods, parameters, kernels and data?

The use of the SVM brought satisfactory results in order to answer these questions, especially the first. In order to make the classification we prepared and transformed the data from the SPSS-file into a MatLab readable format in chapter 3. The implementation and optimization strategies in chapter 4 helped to find good parameters and subsets of variables which resulted in reliable performances. Trying to understand the underlying patterns used by the machines during classification, we performed a statistical structure detection of the support vectors and could find three groups of correlated variables: (protocol primary, WBC and immunology-type), (protocol primary, age at diagnosis, WBC and BM day-15 blast) and (hepar-size and spleen-size). Observing the correctly classified test samples in section 4.5 we could identify two additional relationships: (immunology-type, sex, mediastinal-mass and haemoglobin value) and (Down syndrome, immunology-type and bone marrow blasts). A deeply study of these relationships could help to understand the behaviour of the predictor factors and how they influence each other in the decision of their cutpoints prior to the stratification. The performance of this SVM was evaluated with the statistical prediction model logistic regression in SPSS.

The multi-class SVM implemented in chapter 6 could identify adequate stratification group of 4 patients. Two patients were classified to higher risk therapy than they actually received and the other two received lower therapy classification. Obviously, the best scenario should be to identify the right treatment to all dead patients. Even though the classification machine reached high performance (82.6%), more investigation about underlying interactions between variables must to be done in order to build a reliable stratification machine.
References


[Nopho, 2001] Nordic Society of Pediatric Haematology and Oncology (Nopho), *Treatment Protocol for Children with Acute Lymphoblastic Leukemia diagnosed in the Nordic Countries*


[Sargent, 2001] Sargent D: *Comparison of Artificial Neural Networks with Other Statistical Approaches: Results from Medical Data Sets*. American Cancer Society 2001:91:1636-42


44
