Survival Prognosis
in
Ovarian Cancer

BRUNO VINCENT

MSc by Research
in
Pattern Analysis and Neural Networks

Supervisor: Professor David Lowe

ASTON UNIVERSITY

September 1999

ASTON UNIVERSITY

Survival Prognosis in Ovarian Cancer

BRUNO VINCENT

MSc by Research
in
Pattern Analysis and Neural Networks, 1999

Thesis Summary

In collaboration with Birmingham's City Hospital we want to attempt a study of likely factors which can provide the medical professionals with better prognosis of ovarian cancer. Current data analysis methods have concentrated upon linear factor analysis to try and identify the most useful prognostic indicators. This project researches and develops advanced pattern processing techniques to try and estimate the likely survival probabilities.

In the first part of the project a certain number of methods have been researched to cope with missing data. Then the neural networks approach was introduced: it deals with both regression problems such as estimating how many months a patient is going to live, and classification problems such as finding the probability a patient will die before a given number of months.

In the third part confidence in the results obtained is discussed through the analysis of bayesian error bars and the plotting of ROC curves. The conclusions derived from these analysis are discussed at the end of the thesis.

Keywords: Cancer, Neural Networks, Error Bars, ROC curves, Missing Data

This thesis is dedicated to Lance Armstrong and his victory in the Tour de France 1999 after his determined fight against testicular cancer

Acknowledgements

I would like to thank my supervisor Professor David Lowe for his constant help, guidance and patience. Many thanks to Sean Kehoe[1] and Judy Powell[2], whose collaboration and medical explanations were of great help to me in carrying on that project.

My very sincere and deepest thanks to the British Council for paying my tuition fees and therefore helping me studying in the United Kingdom. As well, I wish to thank the IIE[3] for giving me the opportunity to go to Aston University as an exchange student.

I would also like to thank the MSc PANN students for their good humour and support, and in general all the Post Docs and PhD students. Some special thanks to the French team, Mehdi Azzouzi, Hassen Bouchekif, Frédéric Viot, Franck Mertzweiller, Christophe Tremblin and Pierre-Alain Marzio for joining me through very late nights in the computers lab.

[1] Department of Obstetrics & Gynaecology, Birmingham City Hospital, UK

[2] West Midlands Regional Children's Tumour Research Group, Birmingham, UK

[3] Institut d'Informatique d'Entreprise, Paris, France

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Each life makes its own imitation of immortality.

Stephen King

1.1 Ovarian cancer

Background

According to the National Ovarian Cancer Coalition[1], ovarian cancer is the fifth leading cause of new cancer cases in the Unites States, and accounts for 4 percent of all cancers in women. In women age 35-74, ovarian cancer is the fourth leading cause of cancer related deaths. An estimated one woman in 55 will develop ovarian cancer during her lifetime.

The American Cancer Society estimates that each year approximately 25,500 new cases of ovarian cancer are diagnosed and 14,500 women die of ovarian cancer in the United States. In the European Community 26,000 new cases of ovarian cancer are reported each year and only 10 percent of those diagnosed with advanced ovarian cancer survive a year.

[1] NOCC: http://www.ovarian.org

Symptoms and potential signs

While the symptoms of ovarian cancer (particularly in the early stages) are often not acute or intense, they are not always silent if you know what to look for. Some symptoms of ovarian cancer include weight loss, abdominal pain or discomfort, abnormal vaginal bleeding, sometimes breathlessness and clots in the legs.

Risk factors

The relationship between the number of ovulatory cycles and ovarian cancer risk has been described for many years, and this basic concept has withstood the test of time. Women with no pregnancies specially have a greater risk than those with (the former constitute 50 percent of ovarian cancer sufferers). When ovulation is stopped by either pregnancy of the use of the pill, then the relative risk of ovarian cancer is reduced. In addition, this disease affects mainly postmenopausal women. Thus, there is a peak incidence in the sixth decade of life. Besides there are other risk factors such as family history or history of breast colonic uterine cancer.

Treatment options for ovarian cancer

Optimal treatment vary depending on the stage of disease, the woman's age, and the overall condition of her health. There are three main types of treatments for ovarian cancer: surgery to remove the cancerous growth is the primary method for diagnosis and therapy for ovarian cancer. Chemotherapy relies on the use of drugs that travel through the bloodstream to kill cancerous cells both in and outside of the ovaries. It is used in the majority of cases as a follow-up therapy to surgery. Finally, radiation therapy uses high-energy x-rays to kill cancer cells and shrink tumors.

Most of the time these treatments are very painful and include a lot of side-effects that reduce the patients' quality of life. In some cases, the disease is even preferable to the treatment to prolong life a bit.

Outlook for women diagnosed with ovarian cancer

Because each woman diagnosed with ovarian cancer has a different profile, it is often difficult to give a general prognosis. However such a general approach is currently all that Medical Doctors can give. If diagnosed and treated early, when the cancer is confined to the ovary, the 5-year survival rate approaches 93 percent (78-98 percent depending upon tumor type, stage, and grade; Sean Kehoe, 1999). Unfortunately, due to ovarian cancers' quiet symptoms, only 24 percent of all cases are found at this early stage. Because many ovarian cancers are not detected early, the overall 5-year survival rate for women with ovarian cancer is only between 35 percent and 47 percent, depending upon the type of tumor.

Neural networks in ovarian cancer prognosis

Neural Networks may allow the gynecologist to give a better predictor of outcome for an individual. Presently medical professionals use general survival prognosis techniques such as Multivariate Regression Analysis[1] which naturally encompass a lot of variables not specific to individual cases and take only linear relationships into account.

Moreover such a medical problem involves high levels of noise and as shown below can often have a lot of missing data. Those problems can generally not be handled by linear methods and might be well confronted with the use of neural networks[6].

1.2 Description of the data set

The original dataset was an Excel file containing 1426 patients (rows) and 35 variables (columns) provided in by Dr Sean Kehoe of Birmingham's City Hospital. The provided data was the result of extensive data extraction from patients records taken over a 7-year period (1985-1992). The exact description of the variables is given in A.1. The nature of these variables might be very different: surgical procedures, type of surgeon, tumour grade etc. Some of the original variables had values that could be used directly (e.g AGE) but many of them were also non scaled codes or non-numeric. For example, the coding of HISTO goes from 1 to 14 and then up to 19. This might be logical and understood by any medical professional but it is not accurate at all for mathematical methods nor for neural networks.

Therefore the first thing to do was to recode the data in order to have proper scalings ready for computational techniques. AGE, AGP, COD, STAGE, ADEQ, GRADE, the various surgical procedures, SURGEON, RESDIS, PREVHYST, IDS, OPTYPE, PM, OTMALIG and INTERVAL have been kept unchanged since they had a mathematical meaning and values in a scaled range.

In addition, the following changes have been made:

- HADSURG: "1-No, 2-Yes, 3-Laparotomy" was changed into "1-No, 2-Laparatomy, 3-Yes" so as to make this variable better scaled[2].
- DAN and DLAST have been combined to form the output variable Prognosis $(DLAST - DAN)$.
- ID, DIST, DHA and ICDO-M had to be discarded. The last one was indeed impossible to recode in a scaled range and the other ones deemed to be irrelevant for this study.

[2] Laparotomy is a type of surgery

Chapter 2

General overview

Do what you can, with what you have, where you are.

Theodore Roosevelt

2.1 Some statistical plots

According to medical professionals and most of the literature dealing with the topic(cf [18] and [4]), three of the variables discussed are *a priori* crucial when trying to make any survival prognosis in ovarian cancer. These are:

- Residual disease
- Age
- Stage

Generally, if disease is present but the maximum diameter of any nodule is less than 2 cm in maximum diameter, these patients have a better survival compared with those with larger residual disease. As for age, the older the patients are, the poorer their survival pattern.

Stage is a strong (maybe the strongest) prognosticator. Medical doctors usually give the following statistic about this variable (Sean Kehoe, 1999):

Stage I	90%
Stage II	70%
Stage III	30%
Stage IV	10%

Table 2.1: Rates of 5 year survival in stages I to IV

Note that 75% of women with ovarian cancer present stage III and IV disease.

Here are some statistical plots using the original data set that confirm these general prognoses. Figure 2.1 indicates the survival time (given in a number of days) versus age for each of the 1426 patients in the original data set. In addition, STAGE is used as a color indicator. The figure shows that most of the patients diagnosed with ovarian cancer are under eighty and beyond 60 years old. Moreover the earlier the stage is, the longer the patients survive, what is confirmed by the bar graph on the top of figure 2.2. As for residual disease, the bar graph at the bottom of figure 2.2 reveals that patients with no residual disease have much better survival. In addition this graph confirms that patients with nodule diameter larger than 2 cm have worse survival than those with smaller residual disease. On the other hand, according to this figure patients with seedlings have almost the same survival than those with nodule diameter larger than 2 cm.

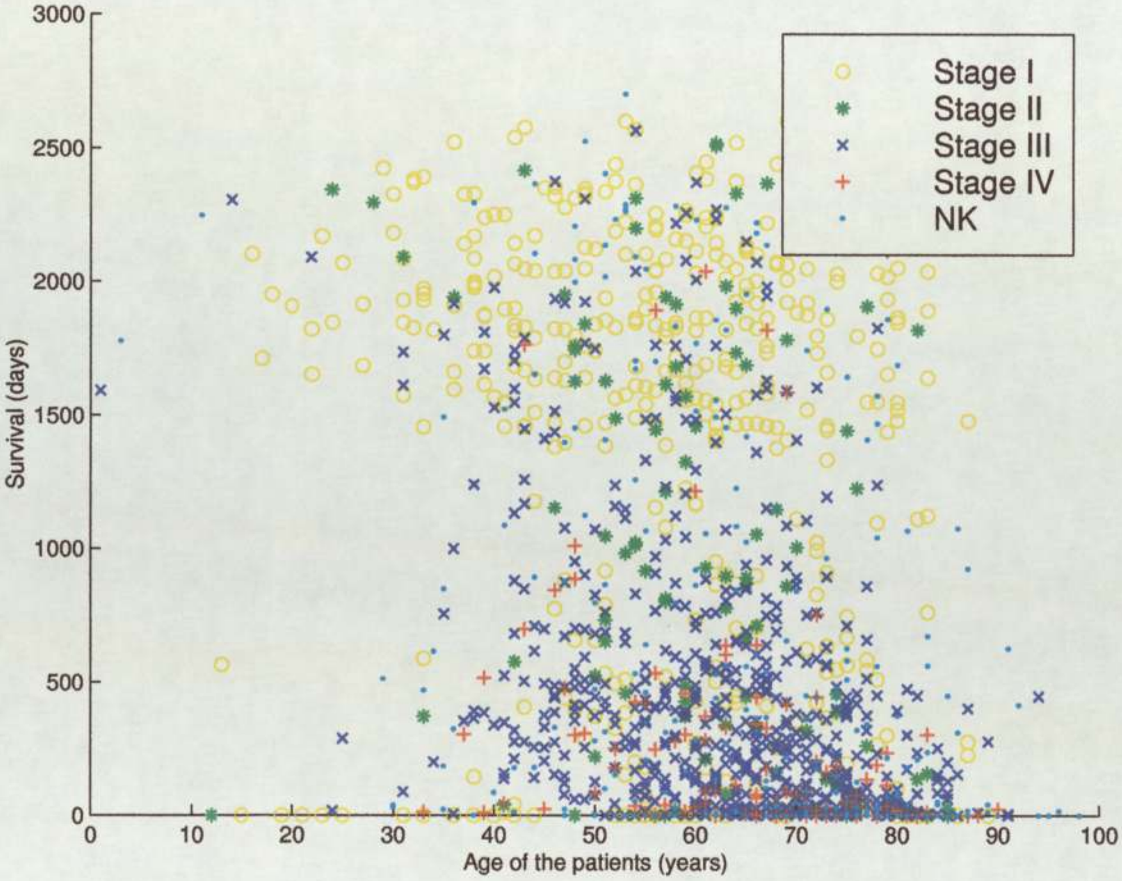


Figure 2.1: Survival versus Age

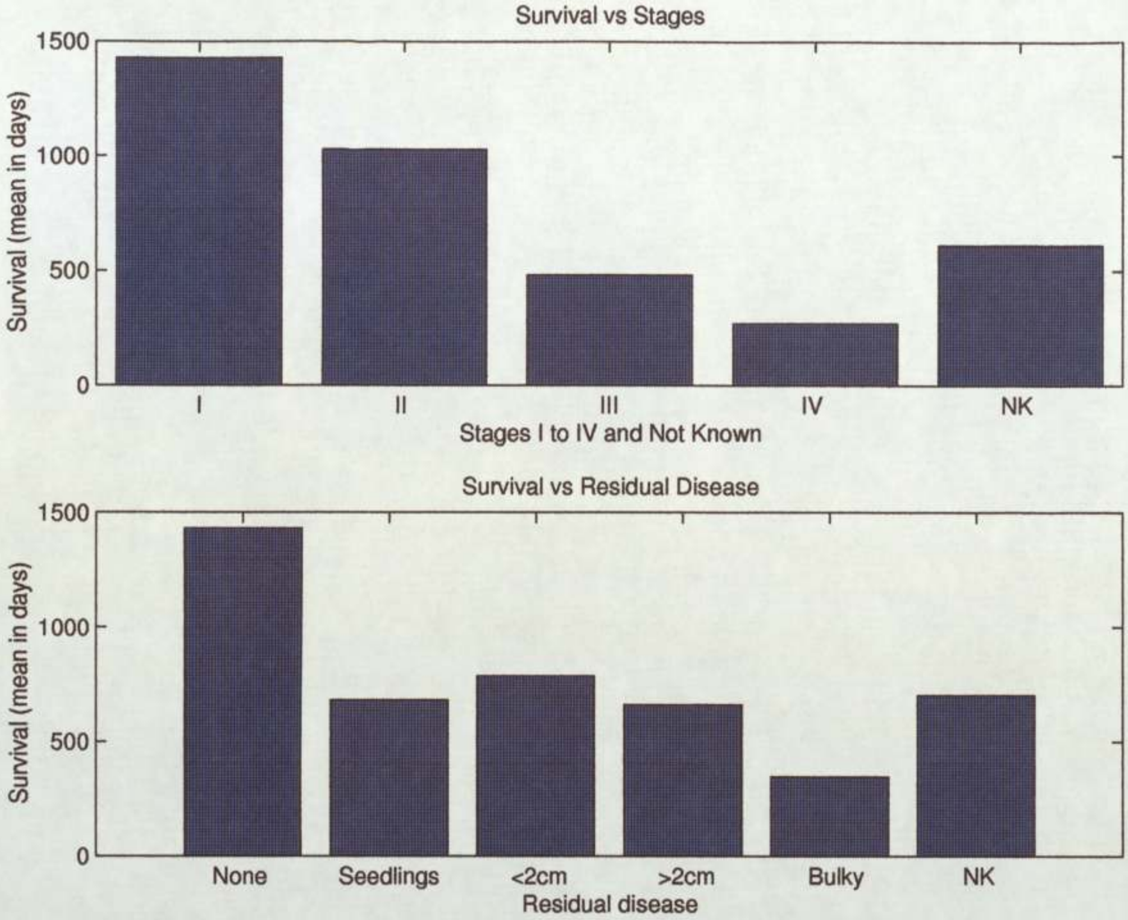


Figure 2.2: Survival versus Stage and Residual Disease

2.2 Variable selection and PCA

One of the first tasks was to determine whether the data supported the notion that prognosis was dominated by the three factors of Residual Disease, Age and Stage, by data visualization. The question was: is it possible to find some linear functions linking those three important variables to the other ones ? A Principal Component Analysis (PCA)[17] is a classical statistical method of data analysis for reducing the dimensionality of the data. The purpose is to find a set of n orthogonal vectors in data space that account for as much as possible of the data variance. In terms of linear algebra, this problem consists of finding a new basis for the data so that if we drop some components in the new basis, the reconstruction error is as small as possible.

Doing a Principal Component Analysis unfortunately requires a complete input data set without any missing value. For each of the variables, the following table gives the number of rows where the data is missing.

STAT	419
COD	22
STAGE	257
ADEQ	346
GRADE	714
HADSURG	76
SURGEON	139
RESDIS	551
PREVHYST	403
OPTYPE	6

Table 2.2: Number of rows with a missing data

If we discard them all, 255 rows out of the original 1426 ones remain. Such results are quite bad news since RESDIS is a priori supposed to be an important factor. Also, half of the tumor grade data is unknown. If we remove this latter variable we obtain 407 remaining rows, 473 if we also remove PREVHYST and 533 if we remove ADEQ.

However, we can exploit expert domain knowledge from medical background (cf A.2) which enables to replace missing data in GRADE using the values of ICDO-B[1]. Finally we can perform a Principal Component Analysis on a data set composed of 542 rows and 10 variables (cf B.1). This set will be called $D_{missing}$ through the rest of this thesis. Note that the 10 remaining variables will be the ones used in the NN approach too.

[1] there is a direct relationship to GRADE for some values of ICDO-B

Here are the results obtained:

Number of principal components	Cumulative Percentage of variance explained
1	26.57
2	45.94
3	63.32
4	73.01
5	82.34
6	91.06
7	96.40
8	98.50
9	99.93
10	100

Table 2.3: Cumulative percentage of variance explained by the principal components

Nearly 85% of the whole variance is explained by 5 of the Principal Components variables. However there is no obvious knee on the plot of the percentage of variance (cf B.1).

The plots of the first three principal components (cf B.2, B.3 and B.4) show that:

- RESDIS is strongly correlated to STAGE
- OPTYPE is strongly anticorrelated to a cluster of STAGE, AGE, AGP, RESDIS and PM.

Thus, we can conclude that according to a Principal Component Analysis:

- the earlier the patients' stages are, the smaller the residual diseases.
- the later the patient's stage are and the older these patients are, the less often Medical Doctors use complex surgery.

2.3 Coping with missing data

An obvious problem, which is common to many medical data problems is the issue of missing data. Simply discarding patients with missing data is wasteful of information. Hence we need to consider methods which compensate for missing data.

2.3.1 Using the mean value

Various solutions have been proposed for coping with missing data in the field of neural networks. The most common one is probably the use of the mean value:

$$\forall j \text{ if } i \in I_j \text{ then } X_{ij} = \frac{1}{\mid I_j \mid} \sum_{k \in I_j} X_{kj} \tag{2.1}$$

where I_j stands for the indices of missing values in column j. Such a method can lead to very poor results[10]. However it has been used as well in some parts of the project since it gave a data set containing 985 rows. From now on the corresponding data set will be called D_{mean}. Note that the distribution of D_{mean} is given in C.3.

2.3.2 Using Bayes' theorem

A more elaborate approach is to express any variable which has missing values in terms of a regression over the other variables using the available data, and then to use the regression function to fill in the missing values. Bayes' theorem (2.3) allows us to implement this strategy quite easily:

$$p(a) = \sum_i p(a|b_i)p(b_i) \tag{2.2}$$

or more generally

$$p(a) = \int_B p(a|b)p(b)\, db \tag{2.3}$$

Here is the algorithm used to fill in the blanks:

Let M_1 be the matrix without any blanks and M_2 be the matrix with missing data.

$$\mathbf{M}_2 = \begin{pmatrix} M_{11} & \cdots & \cdots & \cdots & M_{1m} \\ \vdots & \ddots & & & \\ M_{i1}^* & \cdots & M_{ij}^* & \cdots & M_{im} \\ \vdots & & & \ddots & \\ M_{n1} & \cdots & \cdots & \cdots & M_{nm} \end{pmatrix} \tag{2.4}$$

where any star stands for a missing data.

1. we compute the correlation coefficients of M_1
2. for each variable M_j the other variables are sorted st: $M_j \to M_{ji_1}, M_{ji_2}, \cdots M_{ji_{m-1}}$ where M_{ji_1} is the most correlated variable with M_j and $M_{ji_{m-1}}$ the least one.
3. a scalar N is chosen and wherever there is a star Bayes' theorem is applied:

   $$p(a) = \int_B p(a|b)p(b)\,db$$

   where a is the missing value and B is the vector space described by the N first correlated variables *ie* $p(b)$ is truncated to the N components chosen and thus $p(b)$ and $p(a|b)$ are computed from the empirical distribution of M_1.

The corresponding new data set will be called D_{bayes} in the rest of this thesis. Note that as previously, this new data set contains 985 rows (the distribution of D_{bayes} is given in C.3.

2.3.3 Using mixtures of Gaussians

In the last two sections the methods applied try to fill in the blanks in an input data set. Unfortunately such methods sometimes lead to very poor results. In 1995, Tresp, Neuneier and Ahmad[21] proposed a new method for dealing with missing data in the field of neural networks. The main idea of this method is that instead of trying to fill in the blanks in the input data, we should try to find the output of a trained neural network even if the input vector is incomplete.

Let's assume that a neural network $NN(x)$ has been trained to predict $E(y|x)$, the expectation of $y \in \Re^D$. During recall we would like to know the network's prediction based on an incomplete input vector $x = (x^c, x^u)$ where x^c denotes the known inputs and x^u the unknown inputs. The optimal prediction given the known features can be written as

$$E(y|x^c) = \int E(y|x^c, x^u)P(x^u|x^c)\, dx^u \approx \frac{1}{P(x^c)} \int NN(x^c, x^u)P(x^c, x^u)\, dx^u. \tag{2.5}$$

The integrals in the last equation can be problematic since the computation is exponential in the number of missing inputs. Tresp recommends the use Parzen windows to approximate densities. Given N training data $\{(x^k, y^k)|k = 1, \ldots, N\}$, we can approximate

$$P(x) \approx \frac{1}{N} \sum_{k=1}^{N} G(x; x^k, \sigma) \tag{2.6}$$

where

$$G(x; x^k, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}} \exp(-\frac{1}{2\sigma^2} \parallel x - x^k \parallel^2)$$

is a multidimensional properly normalised Gaussian centered at data x^k with variance σ^2.

Using Parzen windows and equation (2.6)we may write

$$E(y|x^c) \approx \frac{1}{\sum_{k=1}^{N} G(x^c; x^{c,k}, \sigma)} \sum_{k=1}^{N} [\int NN(x^c, x^u)G(x^c, x^u; x^k, \sigma)\, dx^u] \tag{2.7}$$

where $G(x^c; x^{c,k}, \sigma)$ is a Gaussian projected onto the known input dimensions. $x^{c,k}$ are the components of the training data corresponding to the known input.

Now if we assume that the network prediction is approximately constant over the "width"of the Gaussians σ, we can approximate

$$\int NN(x^c, x^u)G(x^c, x^u; x^k, \sigma)\, dx^u \approx NN(x^c, x^{u,k})G(x^c; x^{c,k}, \sigma) \tag{2.8}$$

where $NN(x^c, x^{u,k})$ is the network prediction which we obtain if we substituted the corresponding components of the training data for the unknown inputs. With this approximation,

$$E(y|x^c) \approx \frac{\sum_{k=1}^{N} \alpha_k G(x^c; x^{c,k}, \sigma)}{\sum_{k=1}^{N} G(x^c; x^{c,k}, \sigma)} \tag{2.9}$$

where

$$\alpha_k = NN(x^c, x^{u,k}) \tag{2.10}$$

Chapter 3

The Neural Networks approach

Do not worry about your difficulties in Mathematics.

I can assure you mine are still greater.

Albert Einstein

3.1 Different types of problem

3.1.1 Regression problems

We are basically interested in predicting how long a particular patient is going to live. This prognosis is actually the output of the neural networks we use. Such a prognosis can obviously take different values but can also have very different natures. In an ideal situation we probably would like to predict an exact number of days. However the more precise we want the neural networks to be, the smaller the confidence in the results we get.

In the case of regression we consider problems where the output is a single real or discrete value. This way three types of survival output have been taken into account in that project:

- numbers of days
- numbers of months
- class labels

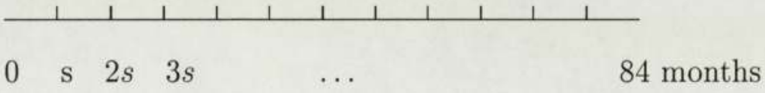
where class labels are indices standing for a category of survival. For example, we might want to predict if a patient is going to die before 3 months, 6 months or a year. We then have a class label for each of theses categories (1: $0 \leq$ survival ≤ 3 months, 2: $4 \leq$ survival ≤ 6 months, 3: 7 months $\leq$ survival $\leq$ a year).

3.1.2 Classification problem

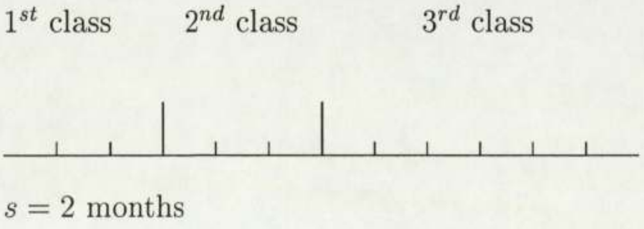
This problem is quite similar to the latter one since our goal is still to predict the belonging of a patient to a specific category. The difference is that we do not have a single output anymore. Assuming we have N distinct classes, the neural networks we use have N outputs which would ideally be the N probabilities $p(C_i|x)$.

A question is immediately arising from that definition. As a matter of fact, the problem we face is not like a handwritten recognition problem where the classes (e.g: the digits from '0' to '9') are already known. How many classes and what periods of time should we *a priori* take for these classes ?

The first approach is to consider time of survival as a line scaled into segments. Thus, we divide time into segments of size s months



Then, we put *flags* on some of the extremities of the segments as shown in the next figure. If we put one flag, we define 2 classes, two flags, three classes and so on:



In the example above, the first class is defined by patients who will die before 6 months, the second by those who will die after 6 months but before a year, and the last class by the others.

Generally speaking, if we consider we want N classes for k segments of time then we get $\mathcal{C}_{k-1}^{N-1}$ possibilities for positioning the flags, where $\mathcal{C}_a^b$ is the probabilistic $\mathcal{C}$ operator defined by $\frac{a!}{b!(a-b)!}$. Thus, if we want to train and test neural networks for each of these sets of classes and then keep the ones for which the networks have the best performance we need quite a large amount of processing time.

In addition such an approach is not optimal. As a matter of fact, feed-forward layered networks trained on a pattern classification task in which the number of training patterns in each class is non-uniform, exhibit a strong classification bias towards those classes with largest membership[12]. This is an unfortunate property of networks when the relative importance of classes with smaller membership is much greater than that of classes with many training patterns. Therefore we should concentrate on classes with relatively equal densities. Such classes can be found by analyzing the distribution functions (cf C.1, C.2 and C.3) of the different data sets. Here are the results obtained for $D_{missing}$. The other data sets' results are given in C.1 and C.2.

Number of classes	corresponding periods of time (months)
5	0 to 2 3 to 7 8 to 13 14 to 24 24 to 84
4	0 to 3 4 to 10 11 to 20 21 to 84
3	0 to 5 6 to 16 17 to 84
2	0 to 10 11 to 84

Table 3.1: Classes with equal densities, $D_{missing}$

3.2 Different types of networks

3.2.1 Multilayer Perceptrons

Multilayer perceptrons (MLPs) and radial basis function (RBF) networks are the two most commonly-used types of feedforward network. They have much more in common than most of the NN literature would suggest. The only fundamental difference is the way in which hidden units combine values coming from preceding layers in the network: MLPs use inner products, while RBFs use Euclidean distance. There are also differences in the customary methods for training MLPs and RBF networks, although most methods for training MLPs can also be applied to RBF networks.

A multilayer perceptron has one or more hidden layers for which the combination function is the inner product of the inputs and weights, plus a bias. The activation function is usually a *logistic* or *tanh* function. The MLP architecture is the most popular one in practical applications. Each layer uses a linear combination function. The inputs are fully connected to the first hidden layer, each hidden layer is fully connected to the next, and the last hidden layer is fully connected to the outputs. In that project, the linear output unit activation function has been chosen in all the cases involving MLPs since it gave the best results with the provided data set. Besides, we used the scaled conjugate gradients algorithm to train MLPs.

3.2.2 Radial Basis Functions Networks

Radial basis function (RBF) networks usually have only one hidden layer for which the combination function is based on the Euclidean distance between the input vector and the weight vector. RBF networks do not have anything that's exactly the same as the bias term in an MLP. But some types of RBFs have a "width"associated with each hidden unit or with the the entire hidden layer; instead of adding it in the combination function like a bias, you divide the Euclidean distance by the width.

3.2.3 Preprocessing

Since the nature of the input variables is very different, these latter need to be normalised before being used by any neural network. For each variable x_i we calculate its mean $\overline{x_i}$ and variance σ_i^2 with respect to the training set and we apply the following transformation:

$$\widetilde{x_i^n} = \frac{x_i^n - \overline{x_i}}{\sigma_i} \tag{3.1}$$

what can also be written in a matrix form:

$$\widetilde{X} = (I - \frac{1}{n}\mathbf{11})X\Sigma \tag{3.2}$$

where

$$\Sigma = \begin{pmatrix} \frac{1}{\sigma_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_p} \end{pmatrix} \tag{3.3}$$

3.3 Learning and generalization

3.3.1 The hold out method

We want to find the network having the best generalization performance ie the best performance on data sets different from the training set. The simplest approach to reach that goal is to compare different networks by evaluating their error functions using data which is separate from that used for training. Various networks are trained by minimisation of an appropriate error function defined with respect to a *training* data set, as shown in figure 3.1. The performance of the networks is then compared by evaluating the error function using an independent *validation* set. We then select the network having the smallest error with respect to the validation set.

This method is called the *hold out approach.* However such a method can lead to some over-fitting to the training set. Therefore the performance of the selected network should be confirmed by measuring its performance on a third independent set of data called a *test* set.

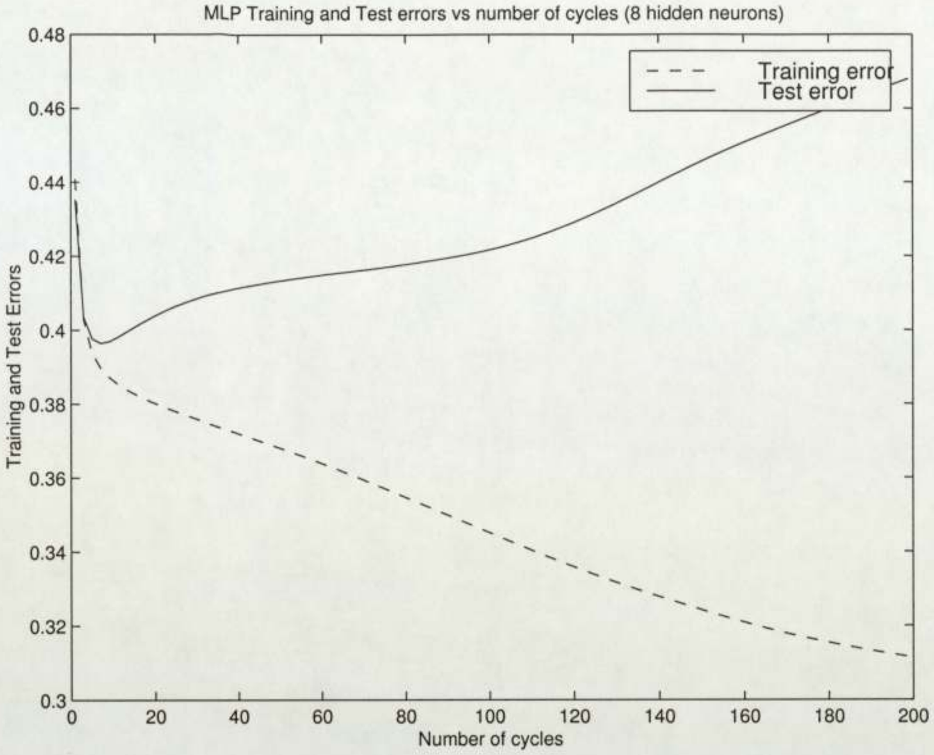


Figure 3.1: An implementation example of the hold out method on $D_{missing}$ using a MLP with 8 hidden neurons. Note that training should be stopped when the error on the validation set reaches a minimum

3.3.2 Cross validation

In this method the training set is randomly divided into N distinct segments. Each network is then trained using data from $N-1$ of the segments and test its performance by evaluating the error function on the remaining segment. We repeat this process for each of the N possible choices for the test segment and the test errors are averaged over all N results. Such a procedure allows us to use a high proportion of the available data (a fraction $1-\frac{1}{N}$) to train the networks, while also making use of all data points in evaluating the cross-validation error. The drawback of such an approach is that the training process must be repeated N times which can lead to a requirement for large amounts of processing time.

3.3.3 Choice of the test error function

Regression problems

When validating and testing the networks, instead of using the sum-of-squares error, an appropriate choice would be the normalised error function given by

$$\widetilde{E} = \sqrt{\frac{\sum_n (y(x^n) - t^n)^2}{\sum_n (\bar{t} - t^n)^2}} \tag{3.4}$$

where $\bar{t}$ is the mean of the target data over the test set.
This normalised error has the advantage that its value does not grow with the size of the data set. If it has a value of unity then the network is predicting the test data "in the mean"while a value of zero means perfect prediction of the test data.

Classification problems

Let C be the confusion matrix. In the confusion matrix, the rows represent the true classes and the columns the predicted classes.

$$\mathbf{C} = \begin{pmatrix} c_{11} & \cdots & c_{1N} \\ \vdots & \ddots & \\ c_{N1} & \cdots & c_{NN} \end{pmatrix}$$

The normalised error is still a good error function since we want a good classification rate. On the other hand, we also aim at having a confusion matrix as diagonal as possible and values not too spread inside it. That means that for each line i we would also like to minimize the distance

$$d_i = \sqrt{\sum_{j=1}^{n} c_{ij}(j-i)^2} \tag{3.5}$$

and so globally minimize

$$d = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}(j-i)^2} \tag{3.6}$$

This can be seen as a second criterion for validating models in classification problems. Thus, in the case of Multilayer Perceptrons we can choose the right number of training cycles in regard of this distance instead of stopping the training process as soon as the normalised error has reached its minimum.

Chapter 4

Results

Errors using inadequate data are much less than those using no data at all.

Charles Babbage

4.1 Outputs of the Neural Networks

In this part we give outputs of survival prediction by neural networks. The errors on validation or test sets we give are mostly obtained in regard of $D_{missing}$. As a matter of fact the D_{bayes} data set has most of the time given the same results than the former one. This result could besides be further explored in another study. As a matter of fact this might mean that some of the variables are less important than others since the neural networks adapted their weights in that way.

As expected[10] the use of D_{mean} has unfortunately led to very poor results (in that case the networks often give the same output whatever the input is). Finally the Mixture of Gaussians method was implemented in the ROC curves part.

4.1.1 Regression Problems

Number of days

The "best"results have been obtained with Multilayer Perceptrons. The hidden units use the *tanh* activation function:

$$tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{4.1}$$

Tables 4.1 and 4.2 indicate the normalised errors on the test sets versus the number of hidden neurons. Both tables show that the number of hidden neurons has very little influence on the results. Thus, we do not obtain a classical minimum for the error functions. These latter globally all take values around 0.6 which is not a very good result for a normalised error (cf 3.3.3).

Number of hidden neurons	2	3	4	5	6	7	8	9
Normalised error	0.588	0.590	0.605	0.586	0.597	0.596	0.610	0.590

Table 4.1: Normalised error given by MLPs predicting survival as a number of days and using the hold-out method averaged over 10 different random sets (25 patients in the validation set, 25 patients in the test set, the rest is used in the training part)

Number of hidden neurons	2	3	4	5	6	7	8	9
Normalised error	0.616	0.637	0.653	0.610	0.605	0.633	0.625	0.585

Table 4.2: Normalised error given by MLPs predicting survival as a number of days and using cross validation (S, the number of segments is equal to 10)

We can have a quite accurate idea of the kind of predictions we get with such errors by looking at figure 4.1. The circles stand for the predicted outputs of the network while the line across the figure is the plot of the identity function $y = x$. Thus, we would like to have the circles quite close to this line. Unfortunately figure 4.1 shows that even if one of the outputs was perfectly predicted, most of the circles are quite far from the line.

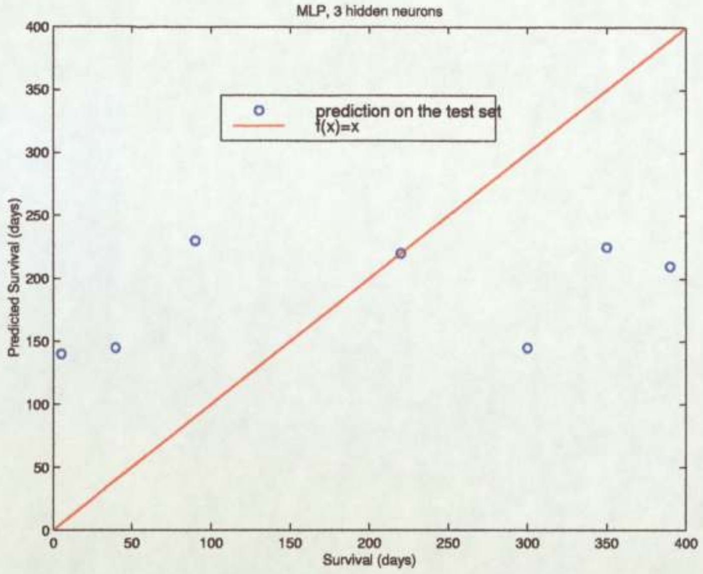


Figure 4.1: Survival predictions given by a 3 hidden neurons MLP

Number of months

The second regression problem deals with networks predicting survival as a number of months. As previously, table 4.3 shows the results we get with MLPs (*tanh* activation function). Once again, these error values are not good at all. Actually they are quite similar to the ones obtained with days. Such a result is logical since the same networks were trained with data which are almost linear transformations of the data used before. As a matter of fact,

$$D^{new} = Floor(\frac{D * 12}{365}) \tag{4.2}$$

where $Floor(X)$ rounds the elements of X to the nearest integers towards minus infinity.

Number of hidden neurons	2	3	4	5	6	7	8	9
Normalised error	0.593	0.599	0.598	0.597	0.592	0.607	0.632	0.560

Table 4.3: Normalised error given by MLPs predicting survival as a number of months and using the hold-out method averaged over 10 different random sets (25 patients in the validation set, 25 patients in the test set, the rest is used in the training part)

Class labels

Since this problem is quite similar to real classification (cf 3.1), confusion matrices and classification rates have been used throughout this part. The classes that had been chosen here are the ones given in table 3.1. Once again, the best results were obtained by using MLPs with *tanh* activation functions, as shown in table 4.4.

	number of classes			
number of hidden neurons	2	3	4	5
2	84%	72%	45%	52%
3	87%	71%	50%	51%
4	79%	67%	55%	50%
5	76%	65%	52%	47%
6	75%	67%	51%	47%
7	73%	67%	51%	47%
8	72%	61%	45%	45%
9	72%	61%	46%	45%

Table 4.4: Classification rate in the class labels regression problem given by MLP

As expected, the more classes we have ie the more precise we want to be, the less good the results. Even if these latter are not that great, they are quite encouraging since one can see that we can predict in 87 percent of the cases if a patient is going to die before or after 11 months.

In addition the "bad"results in the last two problems are due to the fact that the networks seem not to be able to differentiate patients in class 4 and in class 5. Here is an example of the kind of confusion matrices we get as outputs:

$$\mathbf{C} = \begin{pmatrix} 9 & 1 & 0 & 0 & 0 \\ 1 & 7 & 2 & 0 & 0 \\ 0 & 1 & 5 & 1 & 2 \\ 0 & 0 & 1 & 5 & 6 \\ 0 & 0 & 1 & 7 & 4 \end{pmatrix}$$

$$classification\ rate = 56\%$$

4.1.2 Classification problem

Here the best results have been obtained with RBF networks using the thin plate spline activation function which is defined by:

$$tps(r) = r^2 \log(r) \tag{4.3}$$

where r is the squared distance to the centers of the RBF.

Figure 4.2 shows the classification rate for each of the five classes defined in 3.1. At first sight, one can see that the results for the two and three classes problems are a bit less good than the ones we got with the class labels regression problem. However it is interesting to note that the corresponding curves for these problems are quite symmetric and reach their maxima with the same number of hidden neurons (cf 4.5). On the other hand the results for the four classes problem are now better than with class labels but unfortunately the five classes problem is still very difficult to handle since the best classification rate is now only 46%.

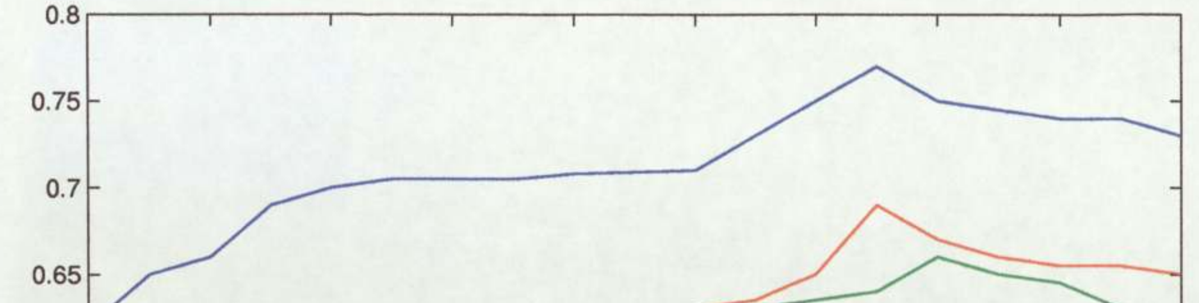


Figure 4.2: Classification rate versus number of hidden neurons given by RBFs using the thin plate spline activation function in the hidden units layer

number of classes	maximum classification rate	number of hidden neurons
2	77%	15
3	69%	15
4	66%	16
5	46%	11

Table 4.5: Maximum classification rates given by RBF networks

4.2 Committees of networks

In appendix D we show that the use of committees of models can lead to significant improvements in the predictions those models make. Therefore it was quite pertinent to try improving the rather bad results we obtained in the days and months regression problems. The neural networks we considered here are still MLPs whose number of hidden neurons in the hidden layer vary in the range [2 ... 9]. The simplest approach of committing networks ie averaging the output such that

$$y(x) = \frac{1}{(L-1)} \sum_{i=2}^{L} y_i(x), \ L \leq 9 \tag{4.4}$$

has not led to any significant decrease (by using the hold-out method) of the normalised errors in both days and months regression problems.

Using the general calculus of the α_i described in appendix D, we get quite better results since the errors are now around 0.5 when combining a 4 hidden neurons network with a 5 one and a 6 one.

On one hand decreasing an error by 10% is an interesting result. On the other hand 0.5 is still a large normalised error value and we can hardly be very confident in an exact number of months or days predicted by these committees. However, it can give a scaled idea of the result that should then be confirmed by a classification or a class labels regression problem.

4.3 Confidence

4.3.1 Error bars

How confident can we be in the neural networks predictions ? In other words, what is the accuracy of such neural networks we used. Bayesian error bars allow us to partially answer this question.

Consider the regression problem with a set of inputs $x^n = x_1, \ldots, x_N$ (where x is a vector) and a corresponding set of target $t^n = t_1, \ldots, t_N$. Here $D = \{t^n, x^n\}$ forms a data set from which inference about the relationship between t and x can be made.

If the targets are related to the inputs through some deterministic function $f(x)$ with additive noise

$$t = f(x) + \epsilon \tag{4.5}$$

where ϵ is a Gaussian ($\mathcal{N}(0, \sigma_t^2)$), the probability density of t^* given some new input x^* is:

$$p(t^*|x^*) \propto \exp[-\frac{1}{2\sigma_t^2}(f(x^* - t^*)^2)] \tag{4.6}$$

Given that the regression can be undertaken by a neural network with an output, $y(x^*; w)$, which depends on the new input and a set of model weights w then using the Laplace approximation[14], the predictive distribution may be approximated as:

$$p(t^*|x^*, D) = \frac{1}{(2\pi\sigma_d^2)^{\frac{1}{2}}} \exp(-\frac{\{t^* - y(x^*; w_{MP})\}^2}{2\sigma_d^2}). \tag{4.7}$$

Here w_{MP} is the most probable weight vector obtained through the expansion[1]and minimisation of

$$S(w) = -\frac{\beta}{2}\sum_{i=1}^{N}\{y(x_i; w) - t_i\}^2 - \frac{\alpha}{2} \parallel w \parallel^2 \tag{4.8}$$

with respect to w, $\beta = \frac{1}{\sigma_t^2}$ and similarly α is the inverse of the variance in w. It has been assumed that $p(w) \propto \exp(-\frac{\alpha}{2} \parallel w \parallel^2)$. The variance

$$\sigma_d^2 = \frac{1}{\beta} + g^T A^{-1} g, \tag{4.9}$$

where $g = \nabla_w y(x^*; w)|_{w=w_{MP}}$ and A is the Hessian $A = \nabla_w \nabla_w S(w_{MP})$, provides an unbiased estimate of the uncertainty (or "error bar")about the predicted mean $y_{MP} = E[t^*|x^*, D]$.

As a summary, there are in the Bayesian approach two sources of error. The first is concerned with the intrinsic noise on the target data and the second is a consequence of the error on the weight themselves. The overall output variance can then be written as following:

$$\sigma_d^2(x) = \sigma_t^2 + \sigma_w^2(x) \tag{4.10}$$

[1] A quadratic regularisation function for the form of the prior in the weights has been used here.

where

$$\sigma_w^2(x) = g^T(x)A^{-1}g(x) \tag{4.11}$$

σ_t^2 can be approximated ([11] and [13]) by

$$\sigma_t^2 = \frac{2E_D}{N-\gamma} \tag{4.12}$$

where N is the number of training examples, γ and E_D the error measured on the training set. The parameter γ can either be approximated by the number of weights k in the model or, according to the full Bayesian treatment of the error bars, it could be set to[11]

$$\gamma = k - \alpha Trace(A^{-1}) \tag{4.13}$$

Also, when implementing the full Bayesian approach, the training is iterated until the values of hyperparameters α and β converge.

Figure 4.3 shows the outputs of such an implementation on the "best"network for the two classes labels regression problem (cf 4.4). The wider the separation between true points and error bars on each side of the figure is, the more confident we are. One can see in figure 4.3 that a few error bars points are very close to true points but globally speaking they are quite well differentiated. Thus, according to a Bayesian approach we should be quite confident in such a network's predictions.

However it is important to note that most of the error found is due to the noisy aspect of the data set. Table 4.6 indicates the first five predictions of the patients used in figure 4.3. It shows that σ_t^2 is 100 times larger than σ_w^2.

In table 4.7 we give the average error bars for each of the class labels problem when keeping 3 neurons in the hidden layer of the MLP. It shows that the error bar logically increases when the complexity of the problem is higher. The error bar for the three classes problem is still good so we should be quite confident in such predictions. The

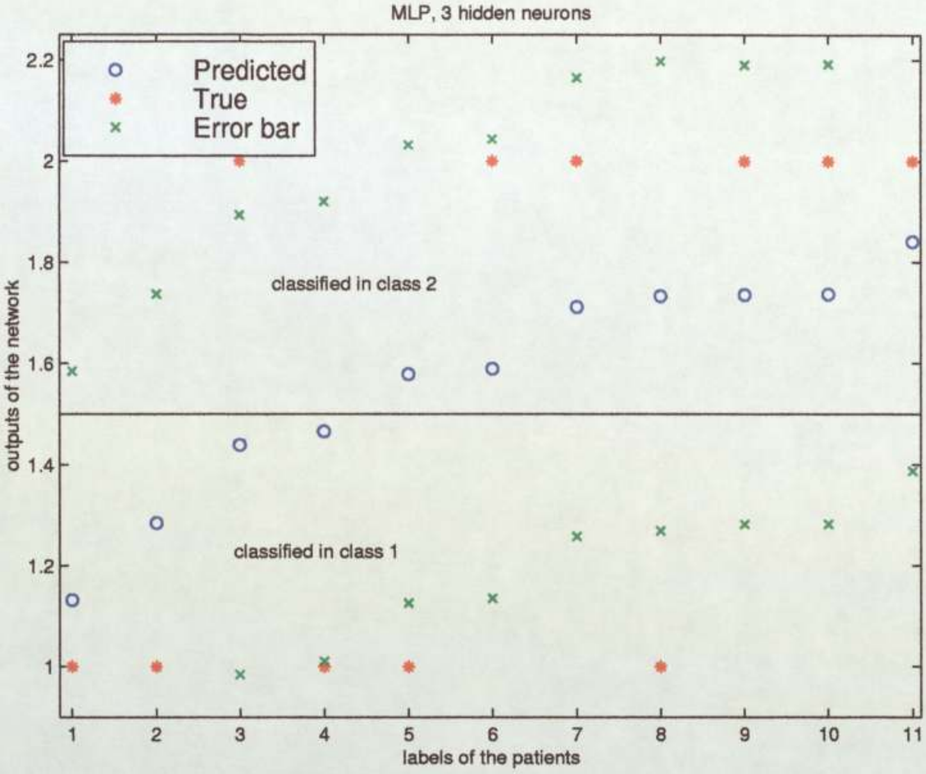


Figure 4.3: Error bars of a 3 hidden neurons MLP on the two classes labels regression problem

$y_{predicted}$	σ_d^2	σ_w^2
1.4388	0.2049	0.0020
1.2844	0.2049	0.0020
1.7359	0.2066	0.0037
1.7114	0.2065	0.0035
1.132	0.2049	0.0026

Table 4.6: Contribution of σ_w^2 to the total error bar

Number of classes	2	3	4	5
Average error bar	0.45	0.78	0.83	1.45

Table 4.7: Average error bar versus number of classes in the class labels regression problem

error bar of the four classes problem has obtained a good average value but it is not really relevant since the classification obtained in that case was only 50%. In addition, this table indicates that we should have a very mitigate confidence in the five classes problem's predictions since the error bar is quite high.

4.3.2 ROC curves

It is quite common in the field of medical applications to use ROC curves to determine the accuracy of diagnostic tests (cf E). ROC curves can be used with classifiers [2] in an interesting way.

Let's consider the question "Will these patients live beyond x months ?". The classifiers we use here are the networks given in 4.4. ROC curves might be good validation tests for these models.

The thresholds used here are successively 0.1, 0.2 and so on until 0.9. The next three figures show ROC curves obtained by MLP with various hidden neurons on the specific question "Will these patients live beyond a year ?"

Figure 4.4 shows that the curve climbs quite rapidly. The area under the curve is equal to 0.84 which is quite good. Therefore we can consider that this diagnostic test is well validated.

Figure 4.5 shows a curve that climbs a bit less rapidly than the former one, what is quite coherent with the results given by table 4.4 since the classification rate was less good with a 5 hidden neurons network than with a 3 hidden neurons one. This is confirmed by figure 4.6 with a 0.79 wide area under the curve.

[2] A classifier is a neural network with one single output node typically designed to answer 'Yes'(1) or 'No'(0). Thus, its outputs are normalised and ranged in [0 ... 1].

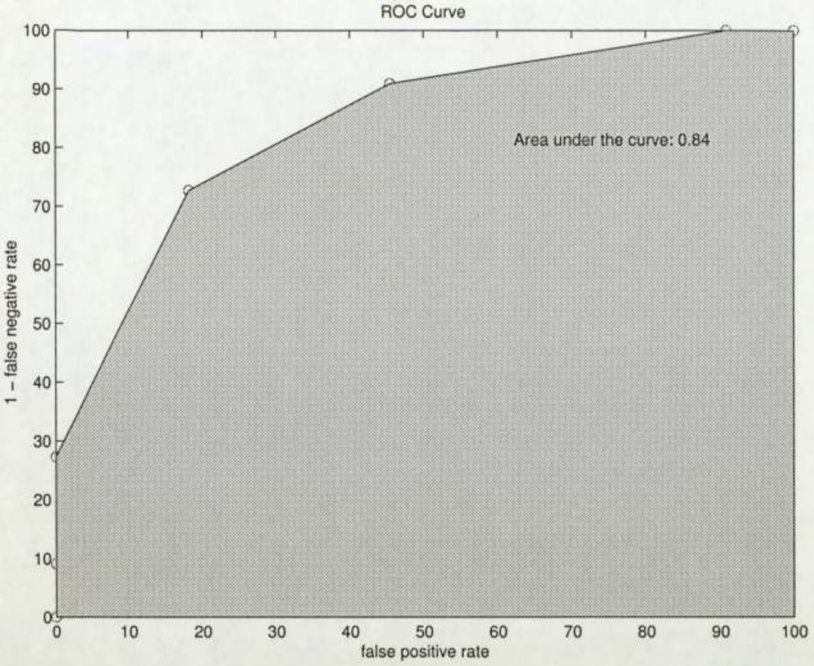


Figure 4.4: ROC curve obtained by a 3 hidden neurons MLP, "positive"patients are the ones living more than a year

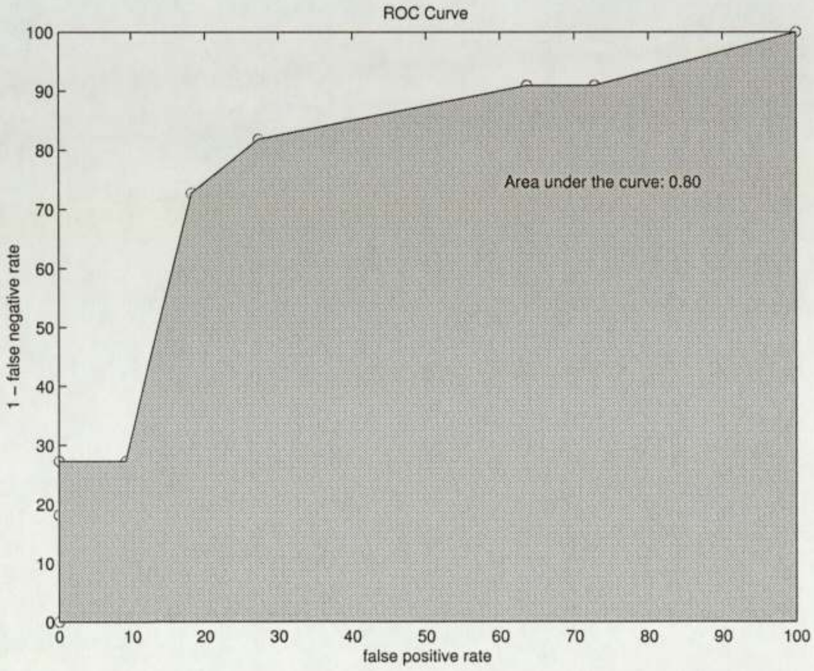


Figure 4.5: ROC curve obtained by a 5 hidden neurons MLP, "negative"patients are the ones living less than a year

Finally another diagnostic test is considered by the ROC curve in figure 4.7. The step is now 6 months and the network is still a MLP with 3 hidden neurons in the hidden layer. We can notice that the slope of the curve is not very huge but regular. Thus the top of the graph is reached quickly and the diagnostic test can be considered as quite decisive what is coherent with the 3 class labels regression problem whose result is given in table 4.4.

However, the more we decrease the step of the question the less wide the area under the curve. The symmetric problem appears when we increase the step to a high value (greater than 20 months). The models are indeed not able to differentiate two patients who are on extreme parts of the survival scale (e.g 1 month and 3 months, the step being equal to 2 months).

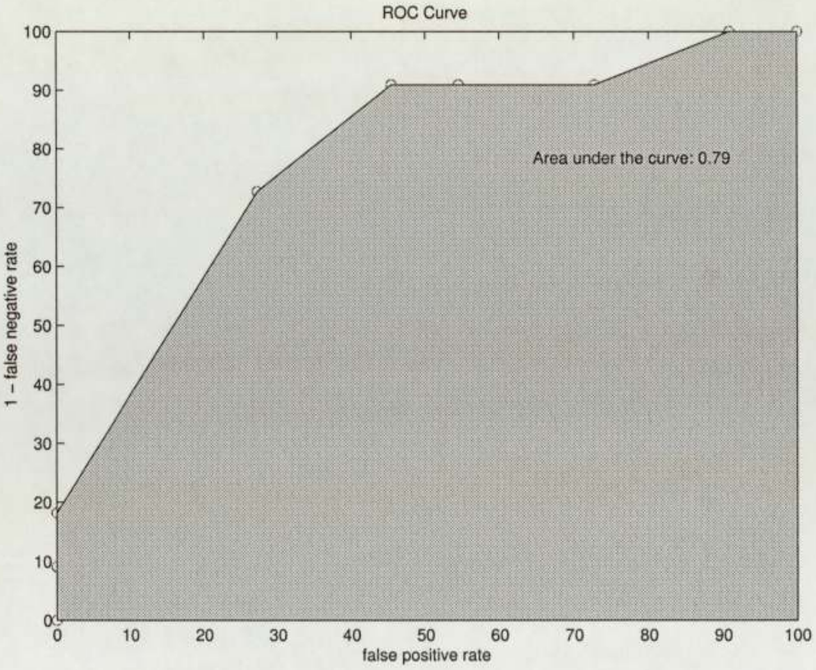


Figure 4.6: ROC curve obtained by a 9 hidden neurons MLP, “positive”patients are the ones living more than a year

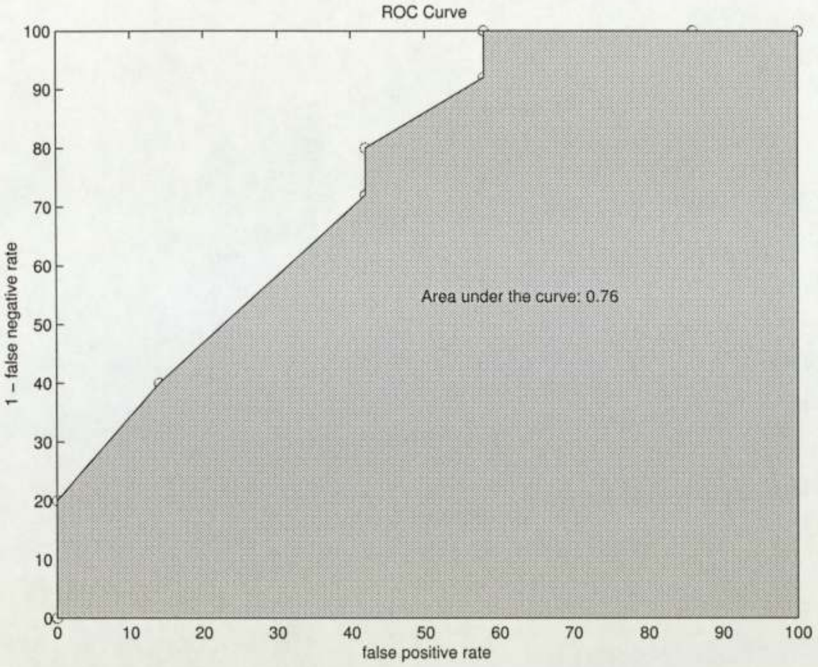


Figure 4.7: ROC curve obtained by a 3 hidden neurons MLP, "positive"patients are the ones living more than 6 months

Chapter 5

Conclusion

Computers are useless. They can only give you answers.

Pablo Picasso

In this thesis we have tried to present a general methodology to implement a medical problem such as survival in ovarian cancer prognosis in the field of neural networks. First we gave a general overview of the data set provided by plotting connections between *a priori* important medical variables and by doing a Principal Component Analysis. We highlighted the problem of missing values which is definitely emphasized in the data set we used and then tried to give some solutions to cope with it.

The neural networks approach was then introduced by listing the different accurate kinds of problems and outputs linked to survival prognosis. We presented several neural networks and methods used when validating such models and then test them on our data set.

The goodness of the results obtained depends on the type of problem involved. Whenever we aimed at making very precise predictions like a number of days or months we faced large errors on the test sets. On the other hand, the implementation of the class labels problem enabled us to give quite good predictions on classes with equal priors. Thus we have been able to correctly classify 87% of test patients and say if they would live less or more than 11 months by using a simple Multilayer Perceptron with 3 hidden neurons. Such regression problems involving three classes have also been able to give quite good results.

"Real"classification tasks with RBF networks were also implemented and allowed us to get a better differentiation on the fourth class than the one given by MLPs on the class label regression problem.

Afterwards, we tried to have an idea of the confidence we could have in the former networks we used by testing a bayesian error bar approach which was quiet decisive on the class labels regression problem. Finally we crossed the frontier to medical diagnostic tests by plotting ROC curves on classifiers' predictions whose results also validate the work done before.

However this project has underlined the important and crucial gaps brought by missing values. An account of time delays and dead lines, the Mixture of Gaussians method was only implemented throughout the ROC curves part. It would probably be accurate to carry on experiments on the data set by using this method on the other parts of the project. As a matter of fact, neither the mean value method nor Bayes' theorem have allowed to give significant improvements in validating the models.

Appendix A

Variables in the original data set

NAME	DESCRIPTION	CODES
ID	Arbitrary case identifier	
AGP	quinary ageband	1=0-4 years 2=5-9 years etc.
DIST	residence: DHA subregion	e.g. 57CA
DHA	Residence: DHA	1-11, rural 12-22 within West Midlands county
ICDO-M	ICDO morphology code	8000-9110
ICDO-B	ICDO behaviour code (modified)	0 "malignant" - but no biopsy 1 borderline malignancy 3 malignant 4 malignant, mod/well differentiated 6 malignant, metastatic site 8 malignant,poorly differentiated
DAN	anniversary date (diagnosis date)	
DLAST	date last seen alive or date death	
STAT	vital status on dlast	1 alive 2 dead

Table A.1: Description of the original data set (see also next pages)

COD	cause of death from death certificate	1 cancer 2 second malignancy 3 other - cancer mentioned on DC 4 other=cancer not mentioned on DC 5 Indeterminate (2 primaries present)
STAGE	Clinical stage/substage	1 stage I 2 Ia 3 Ib 4 Ic 5 stage II 6 IIa 7 IIb 8 IIc 9 stage III 10 IIIa 11 IIIb 12 IIIc 13 Stage IV 0 or 99 NK
ADEQ	Adequacy of staging procedure	1 adequate 2 inadequate 0 or 9 NK

HISTO	Histology	1 Serous 2 Mucinous 3 Endometroid 4 Clear cell 5 Germ cell 6 Granulosa 7 Theca 8 Adenocarcinoma 9 Bordeline malignancy 10 Mixed mullerian 11 Brenner 12 Mixed mesodermal 13 Sarcoma 14 Mesonephroid 19 Borderline (serous) 29 Borderline (mucinous) 69 Borderline (granulosa) 0 or 99 NK
GRADE	Tumour grade	1 Well differentiated 2 Moderately differentiated 3 Poorly differentiated 0 or 9 NK
HADSURG	Did patient have surgery ?	1 No 2 Yes 3 Laparotomy only 0 or 9 NK

TAH	Various surgical procedures	Y or blank OOPH SUBTAH GIT BSO LAVAGE OMENT BIOPSY NODES
SURGEON	Type of surgeon	1 Gynaecological oncologist 2 Gynaecologist 3 General surgeon 4 Other surgeon or clinician 5 Surgeon outside region or NK 0 or 9 NK
RESDIS	Residual disease	1 None 2 Seedlings 3 < 2 cm 4 > 2 cm 5 Bulky 0 or 9 NK
PREVHYST	Previous hysterectomy	1 No 2 Yes 0 or 9 NK
IDS	Intervention debulking surgery	0 no 1 yes

OPTYPE	Extent of surgery	1 Biopsy only 2-5=palliative surgery 6=failed radical surgery 7-12=radical surgery 0 or 99 No surgery or NK
HADCT	Type of chemotherapy	1 single agent 2 combination chemotherapy blank not known
PM	Diagnosed at post mortem	0 No 1 Yes
OTMALIG	Other malignancy	0 No 1 Concurrent tumour 2 Previous tumour
TYPE	type of tumour	1 endometrial cancer 2 breast cancer 3 colon/rectum 4 cervical cancer 5 Skin cancer 9 miscellaneous
INTERVAL	interval between tumours (time in years)	

ICDO-B	GRADE
1, 3, 4	1
6	2
8	3

Table A.2: Links between ICDO-B and GRADE

Appendix B

Principal Component Analysis

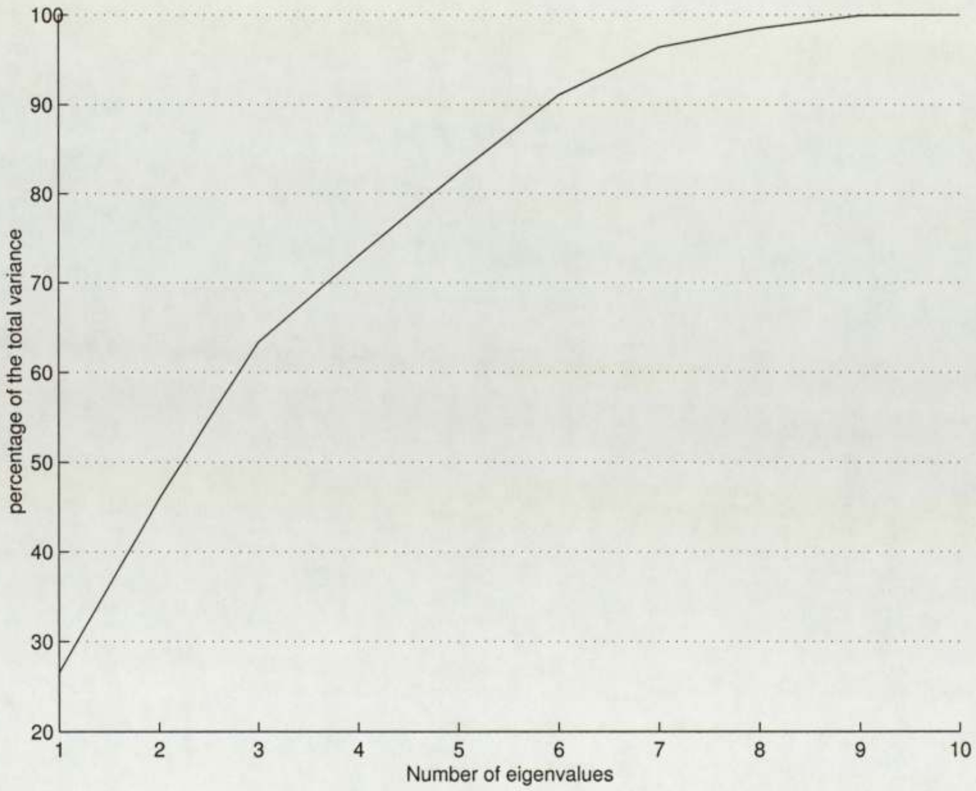


Figure B.1: Percentage of variance explained by the PCA

1	AGE
2	AGP
3	COD
4	STAGE
5	GRADE
6	HADSURG
7	RESDIS
8	IDS
9	OPTYPE
10	PM

Table B.1: Index of the variables used in the PCA and the NN approach

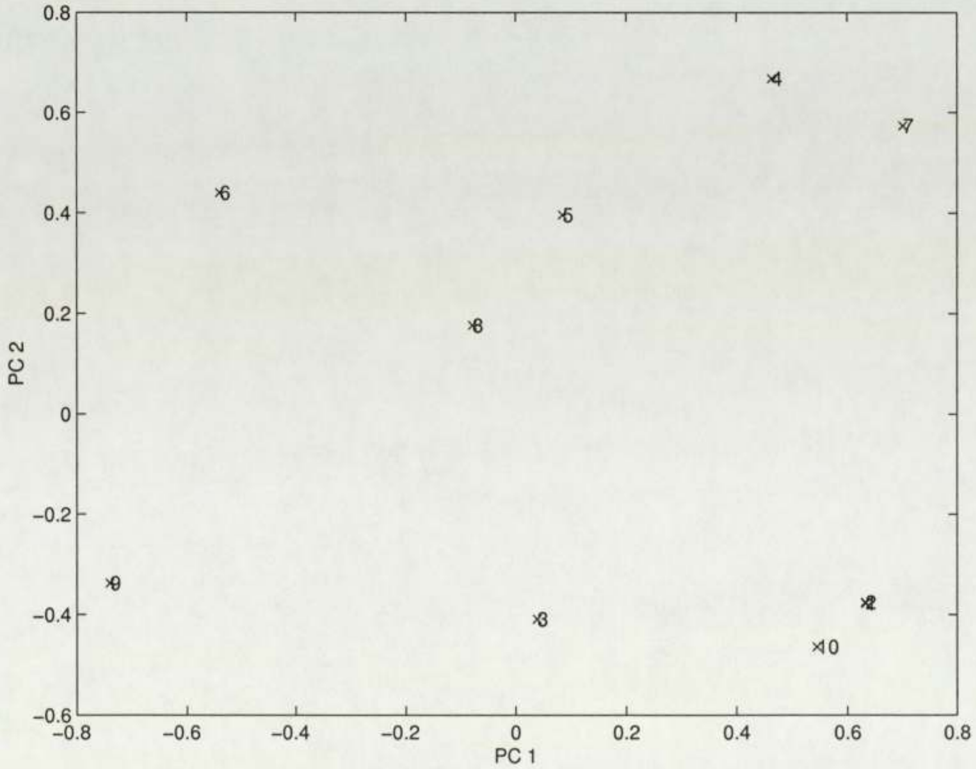


Figure B.2: Correlation of the initial variables with PCs 1 and 2

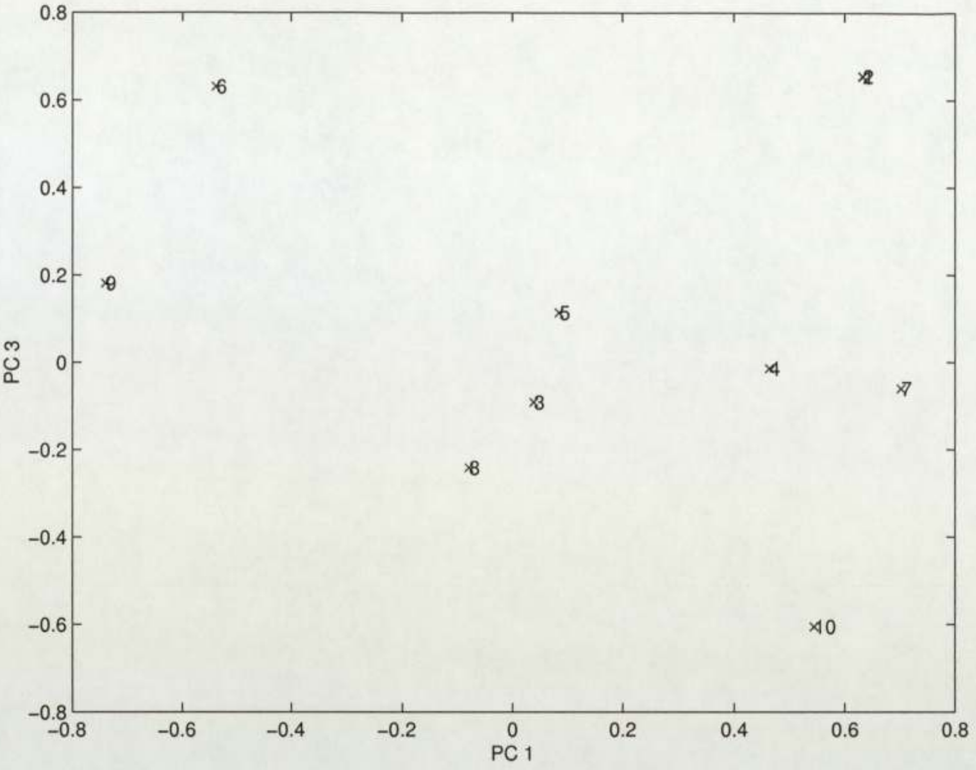


Figure B.3: Correlation of the initial variables with PCs 1 and 3

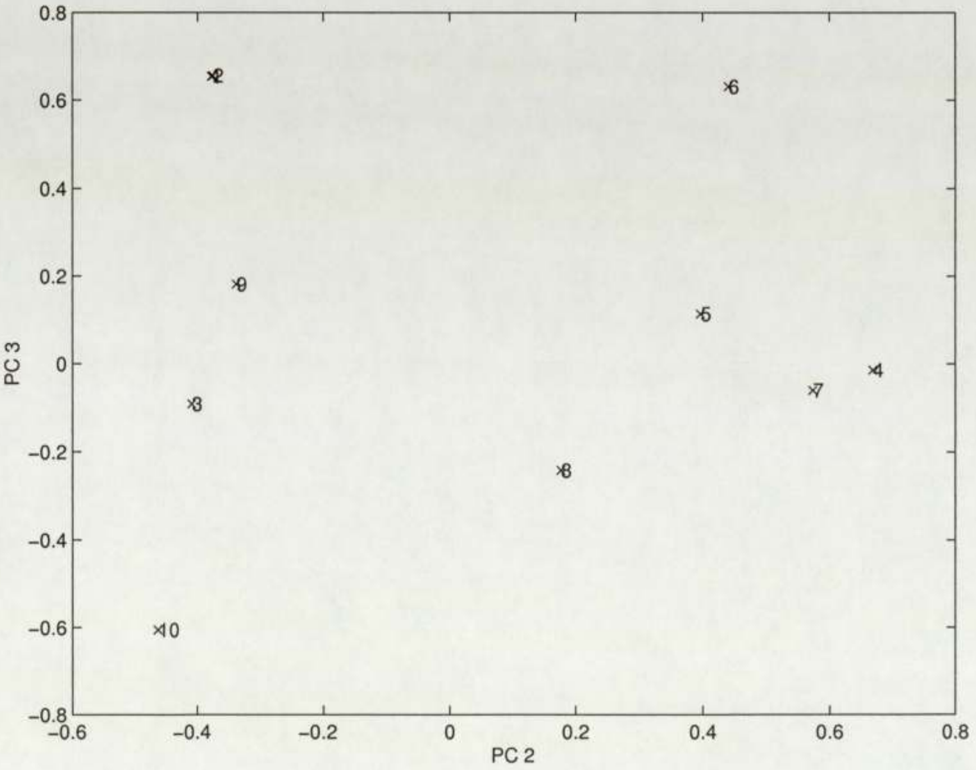


Figure B.4: Correlation of the initial variables with PCs 2 and 3

Appendix C

Distribution functions

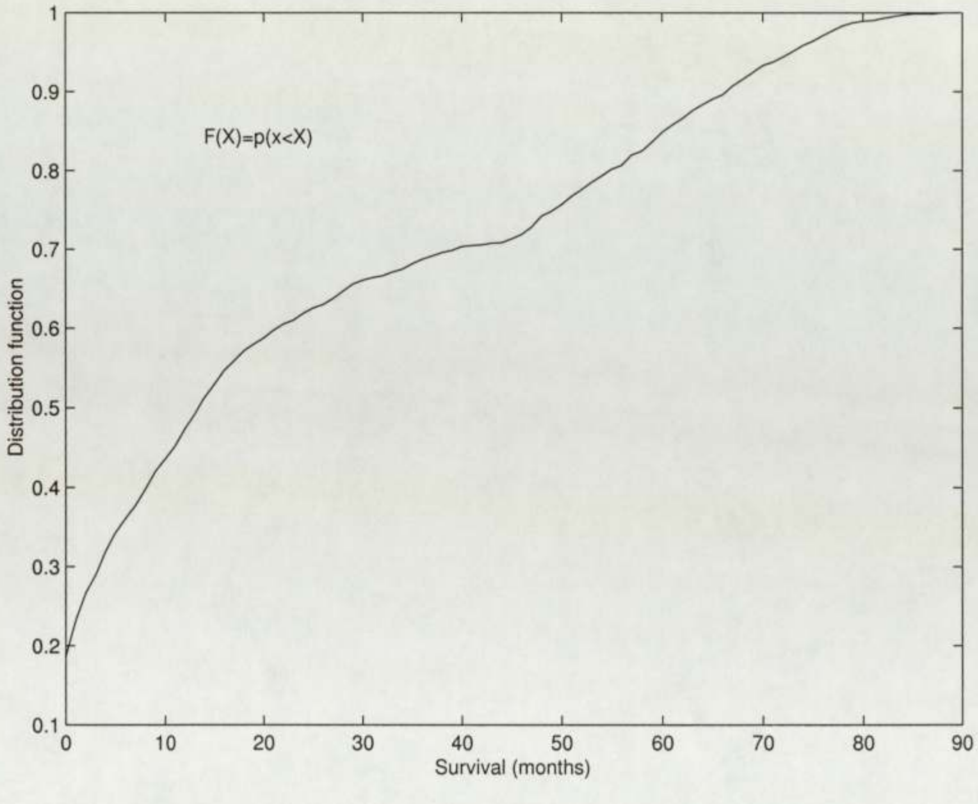


Figure C.1: Distribution function of the original data set

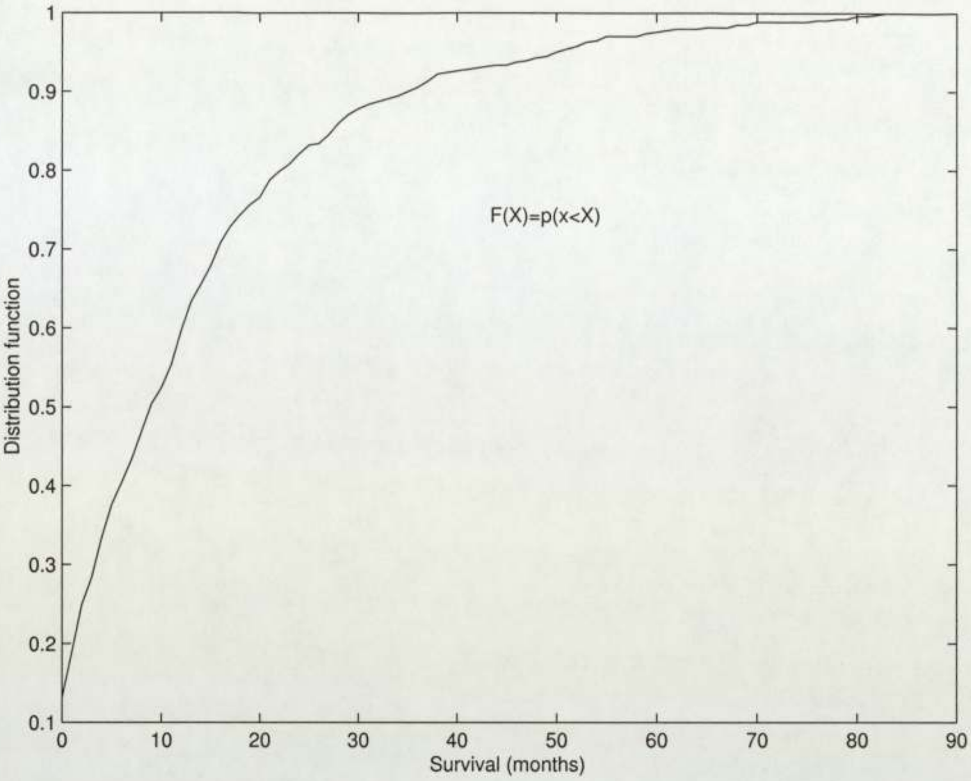


Figure C.2: Distribution function of $D_{missing}$

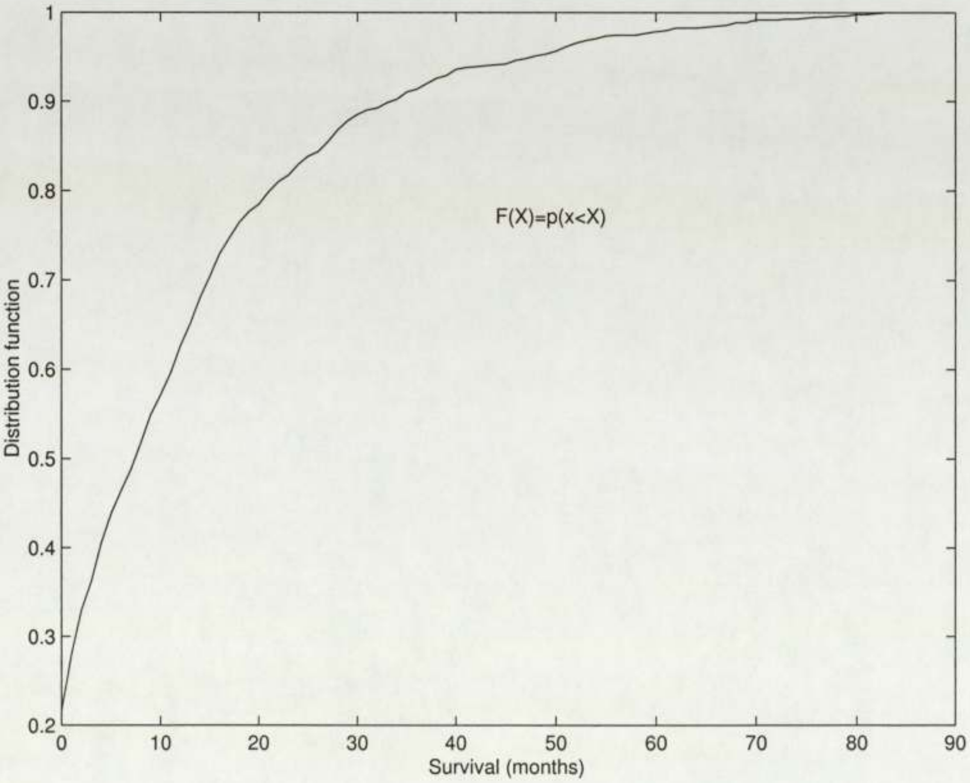


Figure C.3: Distribution function of D_{mean} and D_{bayes}

Number of classes	corresponding periods of time (months)
5	0 to 1
	2 to 5
	6 to 13
	14 to 28
	29 to 84
4	0 to 2
	3 to 9
	10 to 19
	20 to 84
3	0 to 3
	4 to 14
	15 to 84
2	0 to 9
	10 to 84

Table C.1: Classes with equal densities, D_{mean} and D_{bayes}

Number of classes	corresponding periods of time (months)
5	0 to 1
	2 to 9
	10 to 23
	24 to 56
	57 to 84
4	0 to 3
	4 to 15
	16 to 50
	51 to 84
3	0 to 6
	7 to 31
	32 to 84
2	0 to 15
	16 to 84

Table C.2: Classes with equal densities, D

Appendix D

Committees of networks

In the field of neural networks it is quite common to proceed the following way: many different candidate networks are first trained, then the best one is selected on the basis of performance on an independent validation set and all the other networks are discarded. There are basically two drawbacks with such an approach. First all the effort involved in training the remaining networks is wasted. Second, the generalization performance on the validation set has a random component due to the noise on the data (cf 4.3.1), and so the network which had best performance on the validation set might not be the one with the best performance on new test data.

These disadvantages can be overcome by combining the networks together. This is what is usually called *committee of networks.* Such an approach can lead to significant improvements in the predictions on new data, while involving little additional computational effort. As a matter of fact, the performance of a committee can be better than the performance of the best single network used isolated.

Let's consider networks with a single output y (the generalization to several outputs is straightforward) and suppose we have a set of L trained network models $y_i(x)$ where $i = 1, \ldots, L$. Let's call $h(x)$ the true regression function which we are willing to approximate. Thus, the mapping function of each network can be written as the desired function plus an error:

$$y_i(x) = h(x) + \epsilon_i(x) \tag{D.1}$$

The average sum-of-squares error for model $y_i(x)$ can be written as

$$E_i = \varepsilon[\{y_i(x) - h(x)\}^2] = \varepsilon[\epsilon_i^2] \tag{D.2}$$

where $\varepsilon[\cdot]$ denotes the expectation function.

$$\varepsilon[\epsilon_i^2] \equiv \int \epsilon_i^2(x)p(x)\mathrm{d}x. \tag{D.3}$$

From (D.2) the average error made by the networks acting individually is given by

$$E_{AV} = \frac{1}{L}\sum_{i=1}^{L} E_i = \frac{1}{L}\sum_{i=1}^{L} \varepsilon[\epsilon_i^2]. \tag{D.4}$$

Let's now introduce a simple form of committee: the output of the committee is going to be the average of the outputs of the L networks which comprise the committee. Thus, the committee prediction can be written in the form

$$y_{COM}(x) = \frac{1}{L}\sum_{i=1}^{L} y_i(x). \tag{D.5}$$

The error due to the committee can then be written as

$$E_{COM} = \varepsilon[(\frac{1}{L}\sum_{i=1}^{L} y_i(x) - h(x))^2] = \varepsilon[(\frac{1}{L}\sum_{i=1}^{L} \epsilon_i)^2]. \tag{D.6}$$

Making the assumption that the errors $\epsilon_i(x)$ have zero mean and are uncorrelated, so that

$$\varepsilon[\epsilon_i] = 0, \varepsilon[\epsilon_i\epsilon_j] = 0 if j \neq i \tag{D.7}$$

then, using (D.4), we can relate the committees error (D.6) to the average error of the networks acting separately as follows:

$$E_{COM} = \frac{1}{L^2}\sum_{i=1}^{L} \varepsilon[\epsilon_i^2] = \frac{1}{L}E_{AV}. \tag{D.8}$$

According to this result, the sum-of-squares error can be reduced by a factor of L simply by averaging the prediction of L networks! Practically speaking, the reduction in error is generally not that huge because the errors $\epsilon_i(x)$ of different models are typically highly correlated, and so assumption (D.7) becomes irrelevant. However, it can easily been shown that the committee averaging process cannot produce an increase in the expected error. As a matter of fact, if we use Cauchy's inequality we get:

$$(\sum_{i=1}^{L} \epsilon_i)^2 \leq L \sum_{i=1}^{L} \epsilon_i^2 \tag{D.9}$$

which gives the result

$$E_{COM} \leq E_{AV}. \tag{D.10}$$

Thus, some useful reduction in error is generally obtained, and the method has the advantage of being quite easy to implement.

The simple committee discussed so far involves averaging the predictions of the individual networks. However, we might expect that some members of the committee will typically make better predictions than other members. We would therefore expect to be able to reduce the error still further if we give greater weight to some committee members than others. Thus, we consider a generalized committee prediction given by a weighted combination of the predictions of the members of the form

$$\begin{aligned} y_{GEN}(x) &= \sum_{i=1}^{L} \alpha_i y_i(x) && \text{(D.11)} \\ &= h(x) + \sum_{i=1}^{L} \alpha_i \epsilon_i(x) && \text{(D.12)} \end{aligned}$$

where the parameters will be determined shortly. We now introduce the error correlation matrix C with elements given by

$$C_{ij} = \varepsilon[\epsilon_i(x)\epsilon_j(x)]. \tag{D.13}$$

This allows the error due to the generalized committee to be written as

$$\begin{aligned} E_{GEN} &= \varepsilon[\{y_{GEN}(x) - h(x)\}^2] && \text{(D.14)} \\ &= \varepsilon[(\sum_{i=1}^{L} \alpha_i \epsilon_i)(\sum_{j=1}^{L} \alpha_j \epsilon_j)] && \text{(D.15)} \\ &= \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j C_{ij}. && \text{(D.16)} \end{aligned}$$

We can now determine optimal values for the α_i by minimization of E_{GEN}. In order to find a non-trivial minimum (i.e. a solution other than $\alpha_i = 0$ for all i) we need to constraint the α_i. This is most naturally done by requiring

$$\sum_{i=1}^{L} \alpha_i = 1. \tag{D.17}$$

Using a Lagrange multiplier λ to enforce this constraint, we see that the minimum of (D.16) occurs when

$$2\sum_{j=1}^{L} \alpha_j C_{ij} + \lambda = 0 \tag{D.18}$$

which has the solution

$$\alpha_i = -\frac{\lambda}{2}\sum_{j=1}^{L}(C^{-1})_{ij}. \tag{D.19}$$

We can find the value of λ by substituting (D.19) into the constraint equation (D.17), which gives the solution for the α_i, in the form

$$\alpha_i = \frac{\sum_{j=1}^{L}(C^{-1})_{ij}}{\sum_{k=1}^{L}\sum_{j=1}^{L}(C^{-1})_{kj}}. \tag{D.20}$$

Substituting (D.20) into (D.16) we find that the value of the error at the minimum is given by

$$E_{GEN} = (\sum_{i=1}^{L}\sum_{j=1}^{L}(C^{-1})_{ij})^{-1}. \tag{D.21}$$

To sum up, to set up this generalized committee, we train L network models and then compute the correlation matrix C using a finite-sample approximation to (D.13) given by

$$C_{ij} \simeq \frac{1}{N}\sum_{n=1}^{N}(y_i(x^n) - t^n)(y_j(x^n) - t^n) \tag{D.22}$$

where t^n is the target value corresponding to input vector x^n. We then find C^{-1}, evaluate the α_i using (D.20) and then use (D.11) to make new predictions.

Since the simple average committee (D.5) is a special case of the generalized committee (D.11) we have the inequality

$$E_{GEN} \leq E_{COM}. \tag{D.23}$$

One problem with the constraint (D.17) is that it does not prevent the weighting coefficients in the committee from adopting large negative and positive values and hence giving extreme predictions from the committee even when each member of the committee might be making sensible predictions. We might therefore seek to constraint the coefficients further by insisting that, for each value of x, we have $y_{min}(x) \leq y_{GEN}(x) \leq y_{max}(x)$. This condition can be satisfied in general by requiring that $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$.

Appendix E

ROC curves

Definition

To understand an ROC curve, we first have to accept the fact that Medical Doctors often prefer turning a continuous outcome measure into a dichotomy. For example, doctors have measured the S100 protein in serum and found that higher values tend to be associated with Creutzfeldt-Jacob disease. The median value is 395 pg/ml for the 108 patients with the disease and only 109 pg/ml for the 74 patients without the disease. The doctors set a cutoff of 213 pg/ml, even though they realized that 22.2% of the diseased patients had values below the cut off and 18.9% of the disease-free patients had values above the cutoff.

The two percentages listed above are the false negative and false positive rates, respectively. If we lowered the cut off value, we would decrease the false negative rate probability, but we would also increase the false positive rate. Similarly, if we raised the cutoff, we would decrease the false positive rate, but we would increase the false negative rate.

An ROC curve is a graphical representation of the trade off between the false negative and false positive rates for every possible cutoff. By tradition, the plot shows the false positive rate on the X axis and 1 - (the false negative rate) on the Y axis.

Diagnostic tests

A good diagnostic test is one that has small false positive and false negative rates across a reasonable range of cut off values. A bad diagnostic test is one where the only cutoffs that keep the false positive rate low have a high false negative rate and vice versa.

We are usually happy when the ROC curve climbs rapidly towards upper left hand corner of the graph. This means that 1 - (false negative rate) is high and the false positive rate is low. We are less happy when the ROC curve follows a diagonal path from the lower left hand corner to the upper right hand corner. This means that every improvement in false positive rate is matched by a corresponding decline in the false positive rate.

One can quantify how quickly the ROC curve rises to the upper left hand corner by measuring the area under the curve. The larger the area, the better the diagnostic test.

An example of an ROC curve

Consider a diagnostic test that can only take on five values, −−, −, 0, +, ++. We classify diseased (D+) and healthy (D−) patients by this test and get the following results:

	−−	−	0	+	++	Total
D+	2	4	10	14	20	50
D−	28	14	5	2	1	50

It is a bit easier if we convert this table to cumulative percentages.

	−−	−	0	+	++
D+	4%	12%	32%	60%	100%
D−	56%	84%	94%	98%	100%

If we used a cutoff of −−, we would have 4% false negatives and 44% (100-56%) false positives. If we changed the cutoff to −, we would have 12% and 16% rates respectively. For a cutoff of 0, we would have 32% and 6% and for a cutoff of + we would have 60% and 2%.

There are two more cases to consider to complete the curve. If we ignored the test and called everyone healthy, we would have 0% false negatives, but 100% false positives. Conversely if we called everyone diseased, we would have 100% false negatives and 0% false positives.

Here is a summary of the relationship between false positive (FP) and false negative (FN) rates.

FN	0%	4%	12%	32%	60%	100%
FP	100%	44%	16%	6%	2%	0%

Here is what the graph would look like.

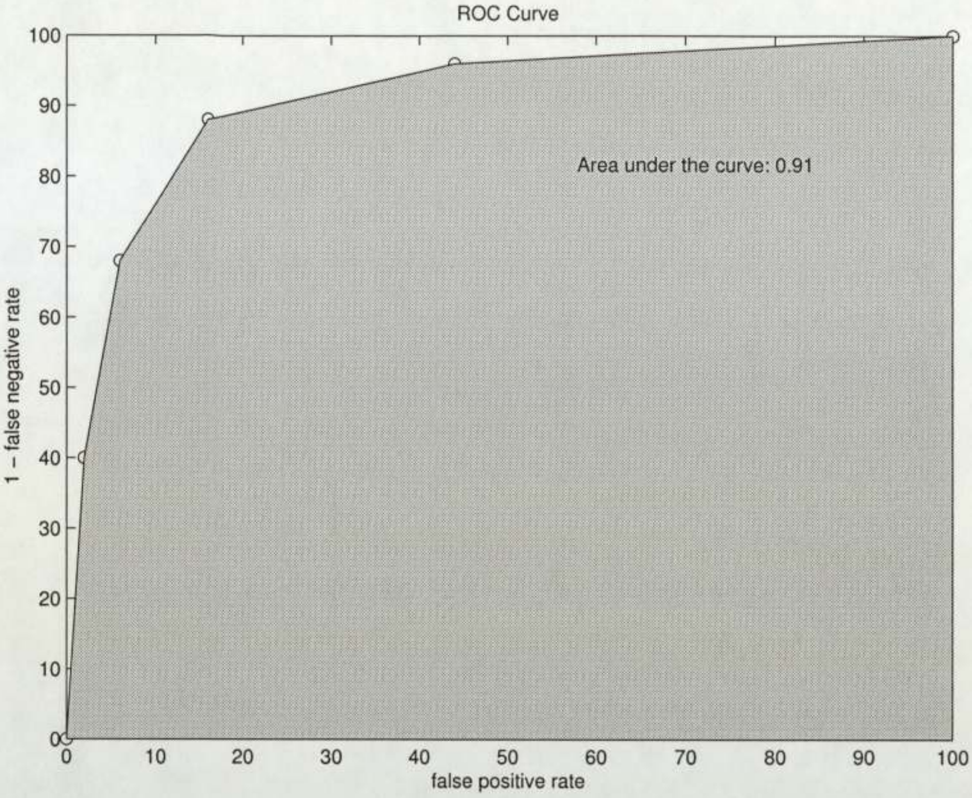


Figure E.1: The S100 protein ROC curve example

Bibliography

[1] David S. Alberts, Steve Dahlberg, Stephanie J. Green, Dava Garcia, Edward V. Hannigan, Robert O'Toole, Donna Stock-Novack, Earl A. Surwit, Vinay K. Malviya, and Christopher J. Jolles. Analysis of Patient Age as an Independent Prognostic Factor for Survival in a Phase III Study of Cisplatin-Cyclophosphamide Versus Carboplatin-Cyclophosphamide in Stage III (Suboptimal) and IV Ovarian Cancer. *Cancer Supplement*, 71, No.2, January 1993.

[2] Christopher M. Bishop. Neural networks and their applications. *Review of scientific instruments*, 65, No.6:1803–1832, June 1994.

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

[4] Mark E. Boyd. Ovarian Cancer. *The Canadian Journal of Surgery*, 28, No.2:114–118, March 1985.

[5] Harry Burke. Evaluating Artificial Neural Networks for Medical Applications. *IEEE International Conference on Neural Networks - Conference Proceedings*, 4:2494–2496, 1997.

[6] Harry Burke, Philip Goodman, and David Rosen. Neural networks for measuring cancer outcomes. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 1:157–159, 1994.

[7] Harry Burke, Philip Goodman, and David Rosen. Neural networks significantly improve cancer staging accuracy. *IEEE Symposium on Computer-Based Medical Systems*, page 200, 1994.

[8] F.M. Collet. Introduction of Neural Networks in Cox Regression. Master's thesis, Neural Computing Research Group, Aston University, September 1996.

[9] Elena Ellioti. Survival Data Analysis Using Neural Networks. Master's thesis, Neural Computing Research Group, Aston University, September 1997.

[10] Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via an EM approach. *Advances in Neural Information Processing Systems*, 6:120–127, 1994.

[11] D. Lowe and C. Zapart. Point-Wise Confidence Interval Estimation by Neural Networks: A Comparative Study based on Automotive Engine Calibration. 1999.

[12] David Lowe and Andrew R. Webb. Exploiting prior knowledge in network optimization: an illustration from medical prognosis. *Network*, 1:299–323, 1990.

[13] David J.C. MacKay. Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks.

[14] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4, No.3:415–447, 1992.

[15] R.N.G. Naguib and F.C. Hamdy. A general regression neural network analysis of prognostic markers in prostate cancer. *Neurocomputing*, 19, No.1-3:145–150, March 1998.

[16] Mackay A. E. Okure and Michael A. Peshkin. Quantitative Evaluation of Neural Networks for NDE Applications Using the ROC Curve.

[17] Gilbert Saporta. *Probabilités Analyse des Données et Statistique.* Technip, Paris, France, 1990.

[18] Tate Thigpen, Mark F. Brady, George A. Omura, William T. Creasman, William P. McGuire, William J. Hoskins, and Stephen Williams. Age as a Prognostic Factor in Ovarian Cancer Carcinoma. *Cancer Supplement*, 71, No.2, January 1993.

[19] Georgia D. Tourassi and P. Xenopoulos. An Artificial Neural Network to Predict Mortality in Patients who Undergo Percutaneous Coronary Interventions. *IEEE International Conference on Neural Networks - Conference Proceedings*, 7:2464–2467, 1997.

[20] Volker Tresp, Subutai Ahmad, and Ralph Neuneier. Training Neural Networks with Deficient Data. *Advances in Neural Information Processing Systems*, 6:128–135, 1994.

[21] Volker Tresp, Ralph Neuneier, and Subutai Ahmad. Efficient Methods for Dealing with Missing Data in Supervised Learning. *Advances in Neural Information Processing Systems*, 7, 1995.

[22] Twyla R. Willoughby, George Starkschall, Noran A. Janjan, and Isaac I. Rosen. Evaluation and scoring of radiotherapy treatment plans using an artificial neural network. *International Journal of Radiation Oncology, Biology, Physics*, 34, No.4:923, March 1996.

[23] W. A. Wright. Neural Network Regression with Input Uncertainty. *Neural Networks for Signal Processing VIII, Proceedings of the 1998 IEEE Workshop*, page 284.