Predicting outcome of renal transplantation using artificial neural networks

Steffen Arnold

September 30, 2009

Contents

1 Introduction

1.1 General Information

Due to demographic changes, end stage renal disease (ESRD) is a growing problem in western industrial societies. Therapeutic options are dialysis and kidney transplantation (KTX). Transplantation is considered the superior option in patients with acceptable surgical risk and improves life expectancy and quality of life.

Using an international allocation system (Eurotransplant in most European countries - section 1.2), organs and tissue of (brain-)dead donors are distributed to patients with terminal renal insufficiency.

A scoring system is used, mostly depending on the HLA (human leukocyte antigen - see section 1.3) match and the time on the waiting list, to allocate the organs. HLA (mis-)match defines the histocompatibility of tissue donor and recipient. If the histocompatibility is marginal then the transplanted tissue is rejected by the recipients immune system - which in medical terms is called acute rejection. The goal of the Eurotransplant allocation system is to provide a fair and transparent distribution of organs.

With an increasing number of patients on the waiting list and a nearly constant number of donors per year, we are facing a shortage of organs today (Eurotransplant annual report 2008[1]), and the situation gets worse every year. And due to a change in the donor demographics (more and more old donors, and fewer young ones) there also is an increase in the number of marginal organs. (See section 1.4.)

But at the same time the development of new, highly effective immunosuppressive agents allow to transplant HLA mismatched tissue with an acceptable risk of acute rejection. So nowadays non-immunological factors for transplant outcome become more and more important.

Non-immunological factors such as cold ischemia time (CIT), donor age, donor BMI (body mass index) and others. They seem to have an important influence on the graft function and the graft survival as well (outcome). Many publications describe these factors but in kidney allocation systems they only play a minor role or no role at all. (See section 1.5.)

Based upon evidence of immunological and non-immunological factors for transplant outcome, we designed a method using artificial neural networks (ANN) to predict the transplantation outcome.

1.2 Eurotransplant

Starting 1967 with simply registering renal transplant candidates and trying to find the optimal HLA matching, the *Eurotransplant International Foundation* now is the organization responsible for mediation and allocation of organ donations all over Europe. Countries taking part are Austria, Belgium, Croatia, Germany, Luxemburg, the Netherlands and Slovenia.

The Eurotransplant Kidney Allocation System (ETKAS) was developed in 1996 to achieve different goals. One major goal was to shorten the average and especially maximum waiting time. Anther one was a reasonably balanced kidney exchangerate among the participating countries. [21]

ETKAS follows a point scoring system to find matching recipient and donor pairs. Variables influencing the rank of the recipient on the waiting list are the urgency status, the HLA match grade, the mismatch probability, the waiting time, a distance factor and the national balance. [21].

1.3 The human leukocyte antigen

The human leukocyte antigen (HLA) is the name of the major histocompatibility complex (MHC) in humans. This protein complex is located on the outer membrane of cells. Two classes of HLA proteins can be distinguished. The class 1 HLA antigens present peptides from inside the cell (including viral peptides, if present). Class 2 HLA antigens present antigens from outside of the cell to T-lymphocytes.

HLA-antigens are used to determine the histocompatibility which is considered the indicator of the success of a transplantation. The more alike the HLA type of donor and recipient are, the better the chance, that the transplanted tissue will not be rejected by the donors immune system. Identical HLA is only to be found in enzygotic twins and clones.

If the HLA proteins of donor and recipient are not alike the immune system starts developing antibodies against the alien HLA and therefore against the alien tissue. All cells that contain the alien HLA are rejected by the hosts immune system just like a virus or a bacterial infection.

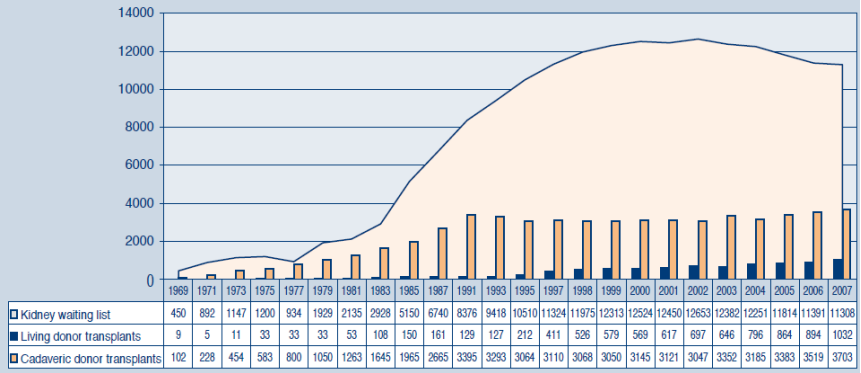
The surface of these two MHC class molecules is well documented. For MHC class I there are five antigens and for MHC class II. For the transplantation relevant are the protein loci HLA-A, HLA-B and HLA-DR. M. Bartels *et al.* [5] describe the outstanding beneficial effect of HLA-A and HLA-DR matches on the rejection free time where most people rely on the similarity of all three loci.

The loci with the most variability - loci with dozens or more allele-groups - are the loci responsible for antigen presentation (HLA-A, HAL-B, HLA-C, HLA-DP, HLA-DQ and HLA-DR). For each there are hundreds to thousands different variants documented in the IMGT/HLA Database[1].

1.4 Donor shortage - Expanding the donor pool

As mentioned before there is a increasing number of patients with end stage renal diseases, but the number of available donors is nearly constant, as seen in the Eurotransplant Annual report [1]. One way of dealing with this shortage is to increase the number of living transplantations. But this alone cannot be the ultimate resolution considering the speed in which the number of patients on the transplantation waiting list is rising (Figure 1).

[1] http://www.ebi.ac.uk/imgt/hla/stats.html



	1969	1971	1973	1975	1977	1979	1981	1983	1985	1987	1991	1993	1995	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
Kidney waiting list	450	892	1147	1200	934	1929	2135	2928	5150	6740	8376	9418	10510	11324	11975	12313	12524	12450	12653	12382	12251	11814	11391	11308
Living donor transplants	9	5	11	33	33	33	53	108	150	161	129	127	212	411	526	579	569	617	697	646	796	864	894	1032
Cadaveric donor transplants	102	228	454	583	800	1050	1263	1645	1965	2665	3395	3293	3064	3110	3068	3050	3145	3121	3047	3352	3185	3383	3519	3703

Figure 1: *Dynamics of the Eurotransplant kidney transplant waiting list and transplants between 1969 and 2008 as seen in the Eurotransplant annual report [1].*

Considering the allocation system ETKAS and the publications about factors influencing the transplantation we come to ask if the factors used to determine donor and recipient match are still in the patients best interest.

One example where recent studies are successfully included in the allocation system is advanced donor age (> 65). L. Resende *et al.* [25] show that it can be recognized as a factor for renal graft failure after transplantation. But the life span of an organ from an elderly donor, even worse than a organ from a younger donor, can still have an expected organ life span of about five to ten years. This can be sufficient if the expected life span of the recipient is within the same range.

The Eurotransplant Senior Program (ESP) allocates kidneys from +65-year-old deceased donors to +65-year-old recipients. To save time, there is not even a HLA typing performed [21]. Within this (sub-)program the HLA type is considered even less important than the time between the removal and the transplantation of the organ (the CIT). The 3-year graft survival rates for age-matched patients were nearly as good as the HLA-matched transplants (64% vs. 67%).

As one can see a balance must be achieved between optimal use of all the few available organs, including those from marginal donors, and optimal life expectancy of all transplant candidates as shown by B. Cohen *et al.* [11]. To achieve this goal we have to take a close look into the factors that influence the transplantation find out which role they play during transplantation.

1.5 Factors influencing the transplantation

Dealing with transplantations, not only the blood group (which has to match) and the HLA match (which should match as good as possible) are crucial. Several studies demonstrate, that a broad range of other donor and recipient characteristics influence transplantations outcome. The Cold ischemia time, for example, is a long known indicator of the transplantation outcome (Hall, Sansom *et al.* [16]). The term is defined as the time between perfusion of the organ with conservation fluid and the release of the arterial clamp after vascular anastomosis. Metaphorically speaking it is the time the kidney is "on ice" after being separated from the donors body and before being transplanted into the recipients body. A prolonged cold ischemia time increases the ischemia-reperfusion damage, a combination of hypoxic cell death during the transport of organs and additional tissue injury. This damage to the kidney additionally influences the transplant outcome and the grafts survival marginal. After a failed transplantation it is possible for the recipient to have another graft transplanted. It is considered that with the height of the number of retransplants the chance of graft survival decreases. (K. Ahmed *et al.* [3])

Advanced donor age is, as mentioned before, also considered a factor influencing the transplantation. It is shown that there is an increased risk but acceptable risk for DGF (Delayed Graft Function) using graft from the elderly (L. Resende *et al.* [26]). Due to the demographic changes in western industrial civilization, incidence of hypertension and diabetes mellitus is growing in the donor population. As shown by Y.W. Cho *et al.* [10] a history of Hypertension (HTN) affects the graft survival if its duration was longer than 10 years. Grafts from diabetes mellitus (DMT) positive donors also lead to inferior graft survival, but also with an acceptable risk. (M. Ahmad *et al.* [2]) Another disease of western industrial civilization is obesity. Beside the increase of surgical complications it can directly influence allograft outcome. In a recently published study, K.A. Armstrong *et al.* [4] consider a recipients body size greater than the relative donor body size, a risk factor. All these patients could be used to increase the donor pool. Although these influencing factors are known and topic of several studies they are not yet considered in the ETKAS.

1.6 The problem

What we try to show during our experiments is that it is possible to predict a reliable outcome of renal transplantation using artificial neural networks. We will try to train the network on patient-data that includes the factors described above that

take influence on the transplantation. Those were taken after considering multiple publications and of course the experience of the transplantation center of the *Charite Virchow Klinikum Berlin* and considering the available data.

All these factors are well documented and are used as reliable indicator in deciding whether a donors graft is accepted in a patient. So we aspect the network to be able to find a pattern within the data. We will try to show a ranking of the features that influence the classifiers outcome. It should show that even taking influence on the transplantations outcome the HLA is not the single most important factor.

To compare the results of the neural network we will do the experiments on other classifiers as well. We choose the logistic regression and support vector machines. A problem that was noticed at the very beginning of the task was that even after taking every single patient that was transplanted in the *Charite Virchow Klinikum Berlin* it still might not be enough to train the network properly.

2 Related work

Neural networks as well as other classifiers (i.e. logistic regression) were used before to estimate the transplantation outcome. The attempt usually is similar but the results vary widely.

The best results we found during the research was the paper of Nikolai Petrovsky *et al.* [23] where the authors describe the ANN to predict up to 85% of successful and 72% of failed transplants correctly. Features used were on donor-side (age, sex,graft source) for the recipient (age, CMV and EBV antibody status, other organ transplants, earlier transfusions). They also included the institutions (referring, donor, transplant hospital, state), the graft (total ischemia, kidney preservation), and the HLA matching (HLA-A, -B, -DR, and DQ). The net topology they achieved the best results with had eight nodes in the hidden layer. The data is trained in a similar way we performed the training. The size of the data they trained their network on is the major difference between our work and theirs. They had data on 1542 kidney transplants which is more than twice the size of our training-data. Probably this is the reason why their results are this much better than ours.

E. Michael *et al.* [9] use a combined attempt of neural networks and logistic regression to estimate the probability of delayed graft function incidence. prediction of delayed renal allograft function They had data of 304 transplants and reached a

63.5% sensitivity and 64.8% specificity of the neural network. The Logistic regression was 36.5% sensitive and 90.7% specific. Of course they used another outcome to train their classifiers but the data still is the same. Their features were recipients age, height, weight, body surface area,gender and race and the donors gender and race. They used the HLA match as well and the cold ischaemic time.

3 Material and methods

3.1 The data

3.1.1 Feature overview

The data on the recipients side is the patient data provided by the transplant-center of the *Virchow Klinikum Berlin* of the last ten years and on the equivalent donor side the data we conceived from Eurotransplant. Altogether the data spans over 800 patients. We excluded ten percent (80) of these for testing. These patients were never used during the coding, experimenting and testing of the methods. So in the end we had a test group no classifier was tested or trained on before. The rest of the data was used for experiments.

Due to missing values or problems estimating the outcome of each patient (i.e. some transplants were performed just a few months ago) we reached a total number of 696 patients for experiments. Using all available data out of the medical files we reached a total number of 83 features. These included the patients history (i.e. of being smoker, ultrasound results, the number of days the patient spent on the ICU (intensive care unit) and others). Due to the quality of the data the count of the features was reduced to 44 (See section 3.2.2). Those were the features that had the least lack of data or were complete from the beginning.

basic donor and recipient data

The basic donor data (the sex, height, weight and age) was complete in every patient. For the recipients there were three percent missing for the patients height and six percent for the recipients weight.

virology

Virology results also were part of our data. Hepatitis B (HBV) , Hepatitis C (HCV) and Cytomegalovirus (CMV) were complete on the donor side, and missing in between four to eight percent on the recipient side. So they were chosen as features.

For each type of virus the general rule is that transplanting graft of a infected donor leads to the worst results and might lead to the recipients death. Mostly because the recipients immune system is weakened because of the surgical intervention and the immunosuppressive agents. If both parties are infected the risk is medium.

laboratory findings

Four features of laboratory findings (glucose, sodium, creatinine and urea) were also added to our training-data. High glucose values i.e. are a indicator of DMT metabolism. Sodium levels $> 155\ \mu mol \cdot l^{-1}$ is considered a risk factor on the transplant outcome. creatininee is found in muscles and usually is filtered out of the blood nearly completely by the kidneys and only small amounts are secreted into the urine. Urea plays a major role in creating hyperosmotic urine which means that the kidney is able to produce higher concentration of substances in urine than in the blood plasma. So creatinine and urea concentration both point to the current kidney function of the donor.

causes of the donors death

Our data also contains nine common causes of the donors (brain) death. These features were documented well and were completely available. One of them is the brain tumor (occurred in 0.718 percent of our data) which is an abnormal growth of braincells (or other cells) within the scull. If the growth is cancerous the brain tumor is lethal.

The cerebro vascular accident (CVA) or stroke (61.78 percent) is caused by sealed blood vessels which lead to a rapid loss of blood supply to the brain. A rapid loss of brain functionality is the result of that. Other features are i.e. the death by circulational causes (1.29 percent). It in general implies death by circulatory induced hypoxia. A possible death could be bleeding to death or failed reanimation that led to brain damage. [31]

Hypoxic-Anoxic Brain Injury (HAI) (5.74 percent) is a disruption of the constant flow of oxygen which is required for the brains normal function. Brain

cells die within minutes if they are not provided with oxygen. The cause of oxygen under-provision can be a disease called hypoxicischemic injury which basicly means that body internal processes prevent enough oxygen rich blood to reach the brain. The lack of oxygen rich blood may also be caused by a lack of respiration (0.86 percent), i.e. if the person was drowning. The cerebral oeadmea (2.15 percent) is an accumulation of fluid in or in between the cells of the brain causing the brain to swell. [19]

Polytrauma (1.58 percent) describe a number of traumatic injuries to the body which happened at the same time, i.e. an car accident. The body is heavly damaged and can't maintain the basic body functions and life support is mostly necessary, to keep the body alive. [19] Another case of physical trauma is the trauma capitis (23.41 percent) which is an physical injury on the head of the patient. The cause of brain death i.e. would be severe damage to the higher brain functions. [31]

The Sub Dural Hematoma (2.5 percent) is the last cause of death we included. It describes a bleeding that occurs underneath the dura mater but above the pia mater ($\rightarrow$ between the hard and the soft cerebral membrane). Death usually is induced by an increase of cerebral compression. This compression leads to damaged nerve tissue and a pressure-linked decrease of blood circulation leading to death. [19]

other features

As part of the recipients data the number of previous transplants was also added. As discussed in factors influencing the transplantation (section 1.5) a increased number of previous transplants worsens the outcome.

To determine the HLA section 1.3) match of donor and recipient we added the broad and split mismatch for three different HLA loci (-A,-B, -DR) and the major antigen. A split mismatch is a variation of a major antigen with structural differences in the antigen binding regions. A good HLA match matches in each split value. The broad mismatch describes the mismatch between major antigen groups. The blood-group also has to match for the similar imunological reasons as the HLA. On the surface of of blood cells are proteins that are the same for each blood group, but are rejected when different.[24].

3.1.2 The outcome variable

Several methods were discussed to produce the outcome variable. This variable is the target vector all the classifiers are trained on. Of course if this variable is not designed properly the result of each classifier used fails.

We chose the GFR (Glomerular filtration rate) as indicator for the transplant outcome. Next to the cases with failed transplantation or graft loss or patients dying during or shortly after the transplantation the usual outcome is less extreme and less obvious. One year after the transplantation the GFR is used to reveal the state of the renal graft function.

The GFR is the volume of fluid filtered from the renal glomerular capillaries into the Bowman's capsule per unit time. [13]

$$GFR = \frac{\text{Urine Concentration} \cdot \text{Urine Flow}}{\text{Plasma Concentration}} \text{ ml/minute}$$

To measure the filtration rate any substance can be used that has a nearly constant level in the blood, can be freely filtered but is neither reabsorbed nor secreted by the kidneys. So the filtrate of this substance is always the same with no respect to labor or stress or other factors. creatinine fulfills these conditions and is used during our experiments.

Normal reference ranges of creatininee clearance (GFR estimated using creatininee in respect of the body surface) are 55 to 146 ml/minute/1.73m^2 in male (52 to 13 ml/minute/1.73m^2 in female) [15].

We considered a GFR < 45 one year after the transplantation a negative outcome. Everything else was used as positive outcome. Using this 53.8 percent of transplant outcome is positive.

3.2 Methods

3.2.1 Pre-processing of the data

The Data contains numeric (age, height, weight, sodium, potassium, glucose, creatininee, urea, # of previous transplants, HLA) as well as discrete features (sex, HBV, HCV, CMV, various causes of death). During the preprocessing of the data

all the discrete values had to be converted into binary values. For example the feature containing the death-code was a single feature before with a range of possible values. Each value out of this range now is a feature with a ”1” at positions it stood in the original feature and a ”0” else. Now it is possible to write each value as a binary value.

To finish the preprocessing of the data each variable was normalized, i.e. the mean was subtracted from every value and the result was divided by the standard deviation. This was necessary to compare the distribution of the features.

3.2.2 Missing values

Dealing with clinical data

Dealing with clinical data one most certainly has to face the problem, that the data set is not complete. During the as-semblance of the data you can't presume that every case (here: patient) is the same, and is treated the same way. In our example the data was assembled over a time span of over ten years. During that time the check-in-routine and the standard examinations during the patients stay in the hospitals have changed several times. So for patients that had their transplant less than three years ago you might have new or other variables.

Also the routines change from hospital to hospital. So there is a chronological and a geographical difference between the way the data was gathered. Sometimes the amount and the consistency of the data changes even from station to station, and between different clinical personnel. And Sometimes you got empty fields among the patients files and you can't say whether this means that i.e. the patient is not a smoker or the patient never was asked. All of this results in huge gaps in the dataset.

The problem of missing values

Dealing with these gaps is necessary for training a classifier it is necessary to be trained on a big (at best infinite) amount of data that represents all possible derivations of the features. Missing values can't be used for training, so the data either has to be excluded or to be filled. Excluding the data results in either a loss of features or patients. To keep the feature/data-point ratio on the small, and therefore increase the probability of learning something out of the data, the better way would be to exclude features than patients. (See B.M.Marlin [20, chapter 6.1.2]) We did this for features with more than 100 values missing. For the other features we

found ways to fill the gaps.

Three ways of filling in the missing values

To predict single features the data has to be missing at random, meaning that the probability of missing values in a feature f_i is can be estimated through f_j with $j \in [1 \dots n, \setminus \{i\}]$ as described by Little and Rubin in 1987[18]. For features that are not missing at random other attempts have to be used. During our experiments because of multiple missing values, we had to exclude each variable that was not missing at random.

We chose three ways to deal with this problem and will provide results based on all three different ways.

- The first way, is simply to **set all missing values to zero.** This was done before the normalization was performed. So i.e. the age of each donor was set to zero years. Of course we aspect this method to result the worst. The distribution of the feature is ignored completely, and in the example with the donors age, the value filled in even is physiological impossible Also we don't deal with the combined effect that other variables might have on the missing one. For example the size and the age of the patient, as well as the fact that the patient is diabetes positive, might result in another weight than you'd expect if you'd just look at the standard distribution. And the mean age might be completely wrong.

- The second way is to just pick **random values out of the features distribution**, without looking at links between features. We expect this method to be more effective, than just setting missing values to zero. ([20, chapter 6.13])

- The last method we used was to **predict the missing values using linear and logistic regression**[20, chapter 6.3.4]). Linear regression was used to predict the numeric features only, whereas logistic regression was used to predict the discrete ones. An algorithm selected the feature with the least missing values of a kind and predicted the missing values. Those were combined to the original data and so on until the whole dataset was completed. We did not use features that had missing values of more than forty percent, because of mathematical reasons. It's a too small amount of data to predict the missing values.

The experiments were performed on each dataset and the results are displayed

using ROC curves. So every method of filling the missing values can easily be compared.

3.2.3 Classifiers

A classifier is a mathematical method to map discrete or numeric values x into a discrete space y. Classifiers can be either fixed or learnt using supervised or unsupervised learning. (Artificial) neural networks are one kind of non-linear classifier which simulates the structure of biological neural networks. To compare the results of the neural network all experiments were performed using logistic regression and support vector machines as well. The results of each classifier is shown using ROC curves in the following section.

To check if a classifier did learn anything out of the training data new data needs to be presented to it, for which the label is available. Now the test performance can be calculated to see if the network was able to abstract the information within the training data. During our experiments, we used the area under the curve of the associated ROC curve as a measure of performance.

Logistic regression

Is a linear classifier where the input data is fitted to a logistic curve using the logistic function

$$F(x) = \sigma(a(x)) = \frac{1}{1 + e^{-a(x)}} = p(z = 1) \tag{1}$$

Where $a(x)$ is a linear function of the parameters. Because $F(x) \in [0, 1]$, it can be viewed as the probability that x belongs to a particular class. The result may later be post processed to convert the numeric result into a binary one.

$a(x)$ can be viewed as both a variable and a function. For keeping the following equations clear, we will keep the notation a instead of $a(x)$. It represents the activity and is defined as

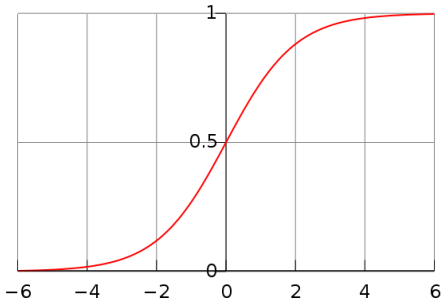


Figure 2: *Plot of a sigmoid function as used in logistic regression or as activation function in perceptrons and neural networks.*

$$a = b + w_1 x_1 + \cdots + w_k x_k \quad (2)$$
$$= \mathbf{w}^T \mathbf{x} + b \quad (3)$$

with b the intercept (or bias), and w_1 to w_k the weights or in logistic regression the regression coefficients. The intercept b is defined as the point in a coordinate system where a function f intercepts the y axis. If the data is normalized, each regression coefficient measures the weight the factor has on the outcome. Surprisingly a strong positive coefficient in our experiments was the donor age.

The perceptron

Another binary linear classifier is the preceptron. The input is directly fed into output nodes only changed by some weight functions. Evolutionary speaking the perceptron is the first stage in the development of the artificial neural networks and was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt.

The equation behind the perceptron is

$$p(y = 1) = \phi\left(a(x)\right) = \phi\left(\mathbf{w}^T \mathbf{x} + b\right) \quad (4)$$

with ϕ an activation function of the perceptrons nodes, w the weight of each

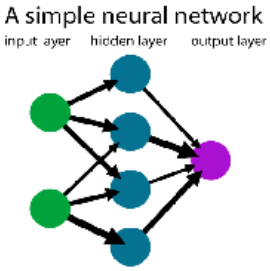


Figure 3: *Graphic showing the basic structure of a simple artifical neural network architecture. The network has two input nodes and therefore supports two input features. Through the first (and only) hidden layer the network is able to solve non-linear problems. The arrows between the nodes demonstrate the different weights the output of the nodes have. The graphic was taken from wikipedia. [33]*

node, which is equivalent to the regression coefficients, and b the bias. The term inside the function ϕ is identical to a in equation (2) used in logistic regression.

The original perceptron by Rosenblatt [28] used a Heaviside step function as activation function. Today they mostly use a logistic sigmoid function as do most artificial neural networks. Now with ϕ the same function as in equation (1) for the logistic regression there is no difference left between the logistic regression method and the perceptron.

Artificial Neural Networks

Taking the next step in the evolution away from the perception we get a feed forward neural network with a sigmoid activation function which was used during the experiments described in this thesis. It consists of a series of perceptrons that are linked and is sometimes called the multilayer perceptron. Unlike a biological neural network the information in feedforeward networks is always directed to the output layer. It is possible to design networks that reuse their output as input, or cross-link nodes between layers in other ways.

As seen in figure 3 the structure of a neural network consist of different units Input units catch signals from the outer world, hidden units process the input data within the network and output units give signals back to the outer world. The units

are connected with each other, depending on the net-topology. The connections are labeled with weights representing the strength of the influence the signal has. A positive weight leads to excitatory reactions, negative weight inhibit the following node and a weight of zero means that the connection has no influence at all.

Therefore the activity a of a node $N_{i,j}$ (the jth node in layer i) is dependent on all input nodes $N_{i-1,1}, \ldots, N_{i-1,n}$ into this node and the corresponding weights between them. This again is the concept as seen in the logistic regression and the perceptron as well and the following equations describing the activity of a node in a neural network is identical to the equations (2) in the logistic regression part.

The only difference is the structure of the network (the hidden layers) and function of the activity of each node $\phi(a)$. This activation function can be linear, linear and using a threshold or a binary function also using a threshold. Most commonly used are sigmoid activation functions. [7] (This type of function was used during our experiments as well.)

$$z = \sigma(a) = \frac{1}{1 + e^{(-a)}} \tag{5}$$

The level of activity here is $z \in [0, 1]$ and making sure that the activity of the net does not overflow. a again is the sum of the activity the node is confronted with and the term is equal to the activation function for logistic regression (equation (1))[27]

Training the neural network

Our feedforward network is trained with supervised learning. Supervised learning is output-oriented which means, that the network tires to minimize the error between its own output and a given target vector containing the preset results.

Learning in the case of multi-layered neural networks means, that the weights between the nodes are iteratively changed using the backpropagation algorithm until either the error between the output vector of the network and the target vector (of the training data) is small or the weights are no longer updated. [27, p. 152][7]

The first step of backpropagation is presenting the input data is to the network and the error between the networks output o and the target $t$vector both of length

n is calculated using the an error function. During our experiments we used the cross-entropy error function[6], coupled with a 2-norm penalty on w:

$$E(w) = \sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]) + \lambda \parallel w \parallel^2 \tag{6}$$

Training the network implies updating the weights :

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \tag{7}$$

To do this, we use gradient descent that says that $\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}}$, where γ is a constant called learning rate defining the length of the steps along the error curve[6].

For networks without hidden layer, E is a direct function of the weights so we can easily compute $\frac{\partial E}{\partial w_{ij}}$ and update the network. For networks with hidden layers however, there are no target values for the hidden nodes, therefore we cannot compute $\frac{\partial E}{\partial w_{ij}}$ directly. To solve this problem, D. E. Rumelhart *et al.* rediscovered the backpropagation algorithm for neural networks in 1986 [29]), that computes the error of the complete network and propagates it backwards. So by using the backpropagation we connect ∂w_{ij} to the output nodes, whose errors we can calculate.[7]

Using the chain rule we now separate the gradient $\frac{\partial E}{\partial w_{ij}}$ (which is proportional to Δw_{ij})

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} \tag{8}$$

where the first term is the error of node j and is described as δ_j. If we modify equation (2) we can describe the the activity in a (hidden) layer as

$$a_j = \sum_i w_{ij} z_i$$

The second term in equation 8 is the error for w_{ij}'s target node which is the activation function z_j and is known to us. (See equation (5)).

$$z_j = \phi(a_j)$$

Now to calculate the gradient Δw_{ij} we still need to find δ_j the error in node j. Again we can use the chain-rule and get

$$\delta_j = \frac{\partial E}{\partial a_j} = \sum_k \frac{\partial E}{\partial a_k}\frac{\partial a_k}{\partial a_j} = \frac{\partial z_j}{\partial a_j}\sum_k \delta_k \frac{\partial a_k}{\partial z_j} \tag{9}$$

The factor $\frac{\partial E}{\partial a_k}$ in equation (9) is the error in node k described as δ_k. The factor $\frac{\partial a_k}{\partial z_j}$ is w_{jk} which is the weight between layer k and layer j. The factor $\frac{\partial z_j}{\partial a_j}$ finally is the derivate z'_j of the activation function $\phi'(a_j)$.

So we get

$$\delta_j = \phi(a_j)\sum_k w_{jk}\delta_k \tag{10}$$

The sum $\sum_k w_{jk}\delta_k$ is the sum over the weights in between layer j and k multiplied with the activation function δk. This sum is necessary because now there are multiple weights and multiple target nodes possible.

In case of networks with one hidden layer we're done then. Now all the factors known and the equation is solvable. We propagated the error backwards until we reached the output layer where we already knew the errors. We now can use the result of δ_j to solve equation (8). [22][6]

Support Vector Machines

Support Vector Machines started out as linear classifiers as well. They try to find the best hyperplane dividing datasets that maximizes the distance between itself and the nearest data-point. As non-linear classifiers they were introduced by Schlkopf and Smola in 2002 [30] after being worked on by Vapnik in 1995 [12]. The equation is the same as the one we used to describe logistic regression. The only thing that is different is the activation function.

$$y = \text{sign}(a(x)) = \text{sign}(\mathbf{w}^T\mathbf{x} + b) \tag{11}$$

Because the vector w leaves in the same space as the data, it can be expressed

as a linear combination of the training data : [7]

$$w = \sum (a_n x_n) \tag{12}$$

After replacing w in (11), we obtain

$$y = sign(\sum (a_n(\mathbf{x}_n^T \mathbf{x}) + b) \tag{13}$$

To be able to classify non-linear data as well, a kernel function is introduced that induces a new feature space in which it is easier to linearly separate the data. Usually the number of features becomes really high, which allows the construction of such a hyperplane. Boser *et al.* [8]describe 1992 the first kernel methods. The sparse kernel function was introduced by Vapnik in 1995[32]. The quantity $k(x_n, x)$ replaces the factor ($\mathbf{w}^T \mathbf{x}$ and can be seen as a similarity measure between x_n and x.

For example a popular kernel is the Gaussian kernel :

$$k(x_n|\widehat{x}) = \exp\,(-\frac{\|x-y\|^2}{\sigma^2})$$

Random forests

A decision tree is a classifier that takes the form of a binary tree, where nodes contain logical tests, and leafs distributions over labels. At each node, a test is computed on a data point and depending on the answer to that test (true or false), the data-point goes down one branch or the other until it reaches the next node containing the next test. When the data-point reaches a leaf, classification can be made according to the distribution of that leaf.

A test usually takes the following form: $x_i < \delta$, where x_i denotes the i-th variable in point x and delta is a threshold. Of course, the tests and the distributions need to be learnt during the creation of the tree.

Decision-tree algorithm

The data set consist of K classes, with labels $1 \rightarrow k$. At first the root node is created as leaf, with all training data sitting there. Now while there are still leafs that contain a mixture of different classes the following is done.

For each leaf in the current node, a sample T of random tests is created. Those test than are applied on the training data D_i sitting in the current leaf l_i. The result of each test t_j is a different split of the data, described as $\{D_{t1}, D_{t0}\}$. For each test t_j a measure of the corresponding data D_{t1} and D_{t0} is computed. A typical measure used is the entropy. This is done to identify the test that separates class 1 and class 2 in the best way possible.

If we do have the test t^* that maximizes the separation the leaf is transformed into a node, t^* is attached to it and two now branches now lead away from it (One for true, one for false) terminating with a leaf. On the branches 1 and 2 now the data D_{t1} and D_{t0} is seated. The last step of the loop is to compute the distribution over the classes for each new leaf based on D_{t1} and D_{t0}.

Decision trees are very intuitive but they tend to overfit dramatically. A good answer to this problem is to build several such trees, on different subsets of the dataset. This approach is called random forest.[17]

3.2.4 Cross validation

Model selection and error estimation

Usually when training a classifier, you need a data-set to train the classifer and a independent data-set to test the performance. The data used for training cannot be used for testing and vice versa. Next to training and testing we need to choose the right parameters as well. The number of nodes in a neural network for example, effects directly the learning possibility of the network. Too few nodes in the hidden layer lead to underfitting, during which the network is simply too small to generalize the complexity of the problem. Both test and training error are high during underfitting. Overfitting results from too many nodes in the hidden layer. Now the network topology is too big and instead of generalizing the data the network memorizes the training data. The training error here is very small where the test error is high. [14]

To avoid this, we need a validation set that allows us to asses the performance of the classifier when using a specific set of parameters. Because the parameters will be chosen to maximize this performance, we still need the test set to estimate the real performance of the classifier.

The lack of data is a problem for training classifiers.The ratio of the available patients and the number of features for classifiers to learn properly should be as low as possible. In fact the best situation would be a infinite count of patients. Otherwise inaccurate estimates in classifiers appear because the dataset is too small to represent each features distribution correctly. So we need every possible data-point during the training process.

If we separate the data into training set, a validation set to choose the model parameters and a test set to estimate the performance of the chosen model, then we loose data-points we need for training. Unfortunately, our dataset is very limited so we cannot afford splitting the data and therefore cutting off training-data. So we need a method to perform model selection and error estimation reliably while using all the data for training. With cross-validation we can do that.

Cross-validation algorithm

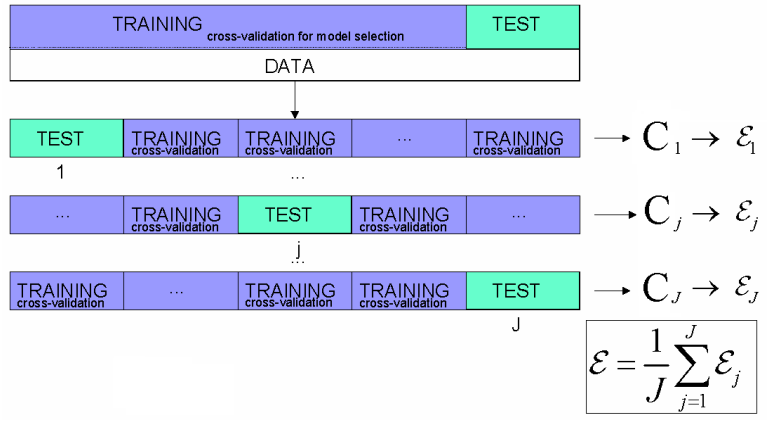


Figure 4: *Figure of the cross-validation algorithm. The whole data is split into training and test data. The position of the test data changes with the iteration over j. For each test set the performance ϵ is calculated.*

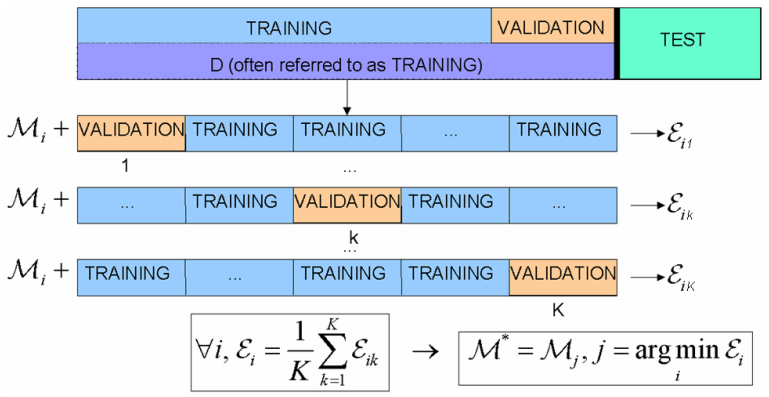


Figure 5: *The nested cross-validation algorithm uses two loops to divide the data into training, validation and test set.*

The cross-validation algorithm consists of one loop that separates the data into J pieces (Figure 4). One subset of the data is used as test data whereas the rest of the data is used for training. For each j the test set changes position until each data point was used once for testing. In each iteration j the test error ϵ_j is calculated. The final test error ϵ is the averaged error over all ϵ_j.

Nested cross-validation

The nested cross-validation introduces a second loop. (Figure 5). It basically is a cross-validation within a cross-validation algorithm. The outer loop iterates over j as described before (**??**). The new inner loop iterates over k and separates the training subset into K pieces where one subset is called validation set. The error ϵ_i for each set of the outer loop is the error averaged over all validation errors ϵ_{jk}.

We used this method to select the model for a classifier (i.e. the topology and weight decay of a neural network) and to estimate the error of the selected model. In the inner loop we averaged over all possible model parameter combination and used the validation set to estimate each models error ϵ_{jk}. The model with the lowest error ϵ_{jk} averaged over all errors ϵ_{jk} is chosen. In the outer loop this model then is trained on the training data (D in figure 5) and the models final error is calculated on the test set.

experimental settings

The experimental settings in cross-validation are preset variables that set the range of possible model parameters and the number of loops used for the nested cross-validation algorithm.

- The *nested cross-validation* loops in our experiments are both set $i = k = 5$, splitting the whole data in the outer loop and the training data in the inner-loop, both into five equal data-sets.
- In *neural networks* we used a range of $1 \rightarrow 20$ possible nodes in the hidden layer and a the weight decay λ out of $[2^{-10}, 2^{10}]$.
- In *SVM*s model parameters were σ^2 and c which both were out of $[2^{-10}, 2^{10}]$, as well.
- Both *logistic regression* and *random forests* models are fixed. In logistic regression there are no parameters, so we don't need the inner loop. Random forests don't need an outer loop at all.

3.2.5 feature-selection

Motivation

The first results showed that the best AUCs were around 0.65, which is a better than random but not great. We hoped for values around 0.75 or 0.8.

Considering the small number of patients and the large number of features, overfitting is clearly a risk.To minimize this risk, we decided to perform feature-selection which reduces the number of features and allows to determine which ones contribute. So by removing the redundant features we try to simplify the problem. An alternative way would have been to add more patients, which we couldn't.

feature-selection was performed on all datasets using neural networks, random forests and logistic regression. The data and features used were the same as during CV.

The simple feature-selection algorithm

We have a set of available features F of the length m. And we have a set of already selected features S. The length of S in the first step is 0.

For each F_i in m we create a model/network and train it on the features of S combined with F_i. The classifier is trained using the nested CV algorithm 3.2.4. The feature that produced the best AUC is removed from F and added to S. As m decreases in each step the length of S increases. This is done as long as there are still features to consider, so while $m > 0$.

A result of the feature-selection should be a of the AUC and the # of features that increases (while features are added that contain information) for a few points and then decreases (because features are added that don't support better information than already supported). The intercept is the AUC of the first feature selected (which is the models best feature). The features until the maximum of the curve are features that contribute valid information to the model. The rest is noise or data that does not cooperate with the other features.

3.3 Experiments

Unfortunately, gathering the data took much longer than expected. Additionally running the feature-selection with the nested CV took really long (even on the cluster of the MPI institute it took up to a week). So we were limited in the number of experiments we could carry out. Nevertheless, we could perform two tests using different classifiers (training them using nested CV, training them using nested CV and feature-selection), were able to choose one filling method of filling the missing values and did perform a ranking of the most important features.

In each following figure showing multiple curves comparing classifiers the red curve represents neural networks, the blue curve logistic regression, the green curve random forests and the cyan one SVMs. If the figures compare different methods of filling in the data, the red curve always represents sample-filled data, the blue one zero-filled data and the green one predicted data.

3.3.1 How to fill in the missing values

- We want to asses three methods we used to fill in the missing values. For each of these methods we ran all the classifiers on all the features and we used the nested cross-validation (Section 3.2.4) where appropriate, i.e. everywhere except for logistic regression (simple cross-validation) and the random forest (no CV at all).

- The whole data with 696 patients and 44 features was used for training. Of the features 28 are discrete. As described in section 3.2.2 we chose three ways of dealing with missing values. Filling in the missing values by setting them to zero, by choosing sample values or by predicting the values using linear and logistic regression. For testing the 10 percent data (80 patients) were used which we put aside at the very beginning.

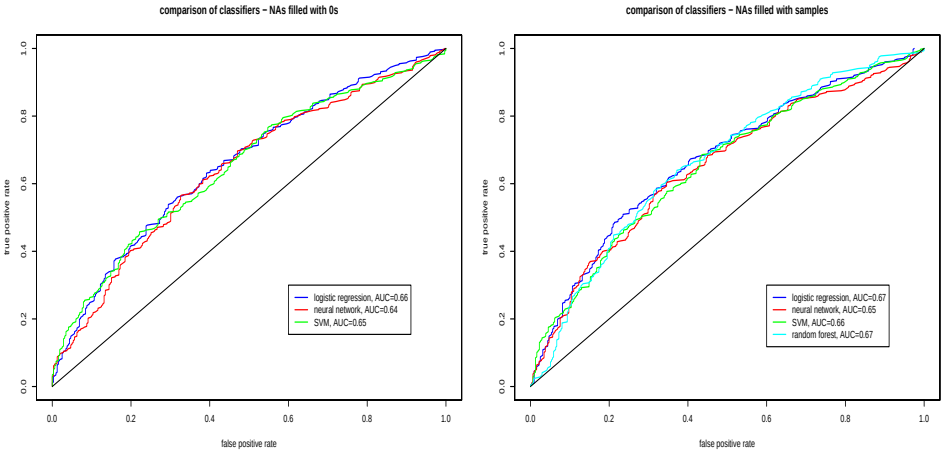
- Figures 6 show the ROC curve for each method, one classifier a time. (Except random forests, which can't handle the zero-filled data-set.) We can see that, for any classifier, the three methods perform very similarly. This is probably due to the fact that we discharged the problematic variables and kept very few missing values.

- The question now is, if there is a difference between choosing the method of filling when using all features, and when performing feature-selection at first. We expected the results to be different, probably better, after performing feature-selection. Figure 7 shows the result of the same experiment, but this time, with feature-selection performed. Unfortunately the results for the SVMs cannot be shown due to the fact, that while finishing this thesis the algorithm was still running.

3.3.2 Comparison of the classifiers when using all the features

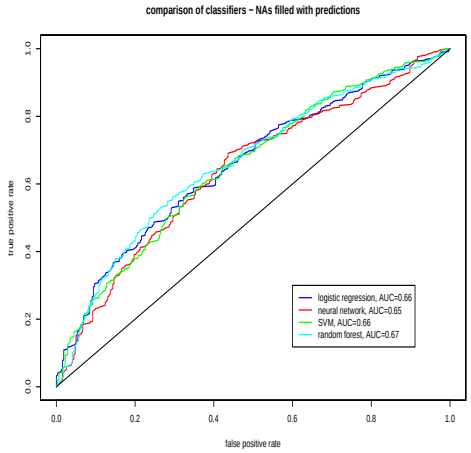
- Now we want to know which classifier used performs best on the data. The three classifiers ran again on all data with all the features and we again used the nested cross-validation (Section 3.2.4) where appropriate.

- The test set-up is equal to the one in (section 3.3.1). But as mentioned before we will only discuss the results for the third method of filling in the data, as the other results are nearly the same.



(a) zero filled

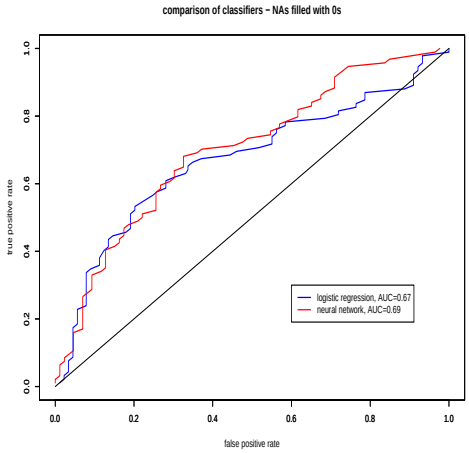
(b) sample filled



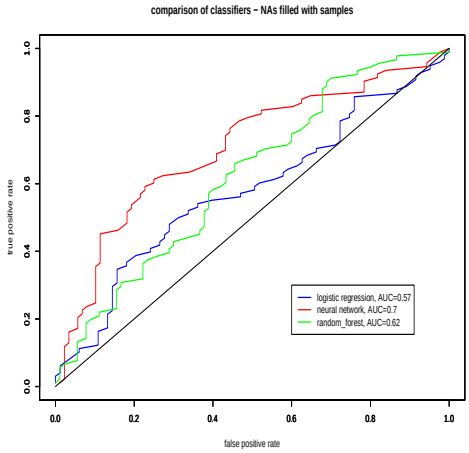
(c) predicted

Figure 6: *The three methods used to fill in the missing values in comparison.*

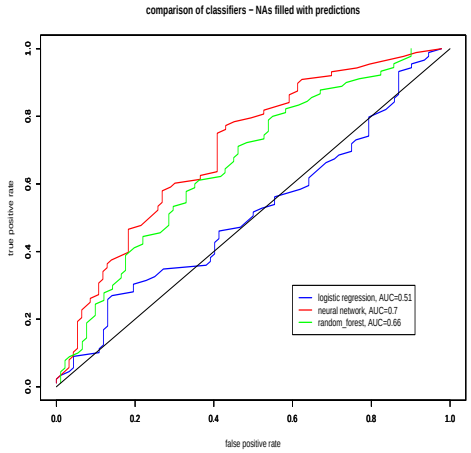
- In figure 8 you can see the ROC curve for each classifier on the three different datasets. The classifiers perform nearly the same and the AUCs are very similar. We expected to see a difference between logistic regression and neural networks due to the better learning abilities of the neural network. But as you can see the classifier show no significant difference.



(a) zero filled



(b) sample filled

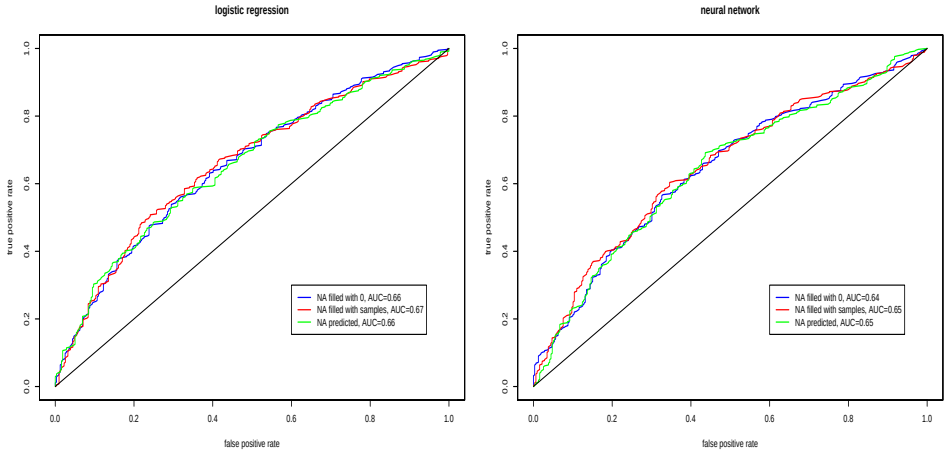


(c) predicted

Figure 7: *The three methods used to fill in the missing values in comparison (performed after feature-selection)*

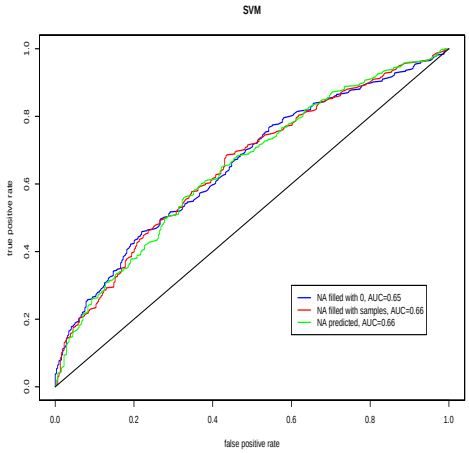
3.3.3 Comparison of the classifiers when using feature-selection

- Using the feature-selection algorithm as described in the methods part (section 3.2.5) we are now eliminating redundant features, that is features that are not containing information needed for predicting the outcome.

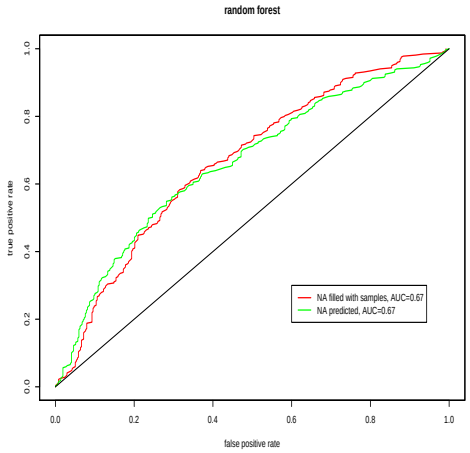


(a) Logistic regression

(b) Neural network



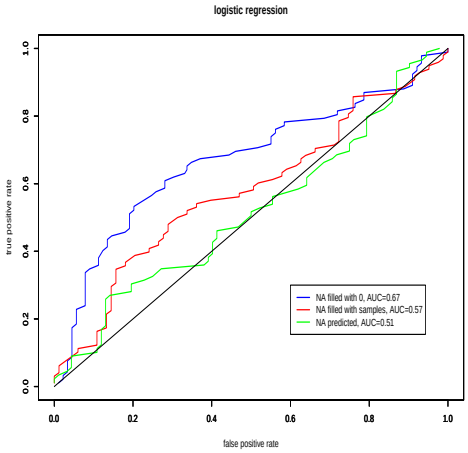
(c) SVMs



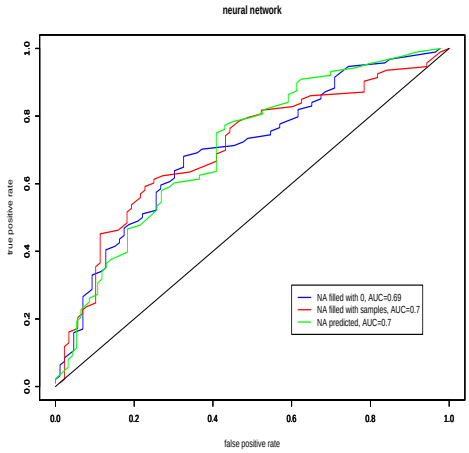
(d) Random forests

Figure 8: *The four classifiers used in comparison.*

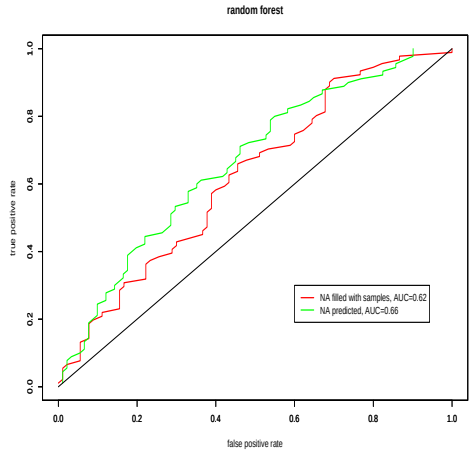
- The test set-up again is equal to the one in section 3.3.1).
- In figure 9 you can see the ROC curve for the classifiers logistic regression, neural networks and random forests on the three different datasets. After feature-selection the ROC curves on the different data-sets are not as similar as in figure 8 where only the nested CV was performed.



(a) Logistic regression



(b) Neural network



(c) Random forests

Figure 9: *The three classifiers trained using feature-selection.*

– The features chosen during the feature-selection are known to us and are the following:

– Logistic regression chose seven features during the feature-selection. The selected features are *the donor age, the number of previous transplants (of the recipient), the donors CMV status, the recipients weight, the donors HBV status, the donors creatinine value and the HLA B*

broad mismatch.

- Neural networks chose in feature-selection seven features, which are *the donor age, the recipients weight, the donors HCV status, the recipients number of previous transplants, the donors death by respirational causes, the donors CMV status and the donors creatinine value.*
- During feature-selection the random forests chose the following fourteen features: *the donor age, the recipients number of previous transplants, the donors glucose level, the donors creatinine value, the recipient age, the donors HCV status, the donors sodium value, the donors HBV status, the HLA broad mismatch, the recipients CMV status, the HLA DR broad mismatch, the donors sex, the recipients height, and the donors blood type.*

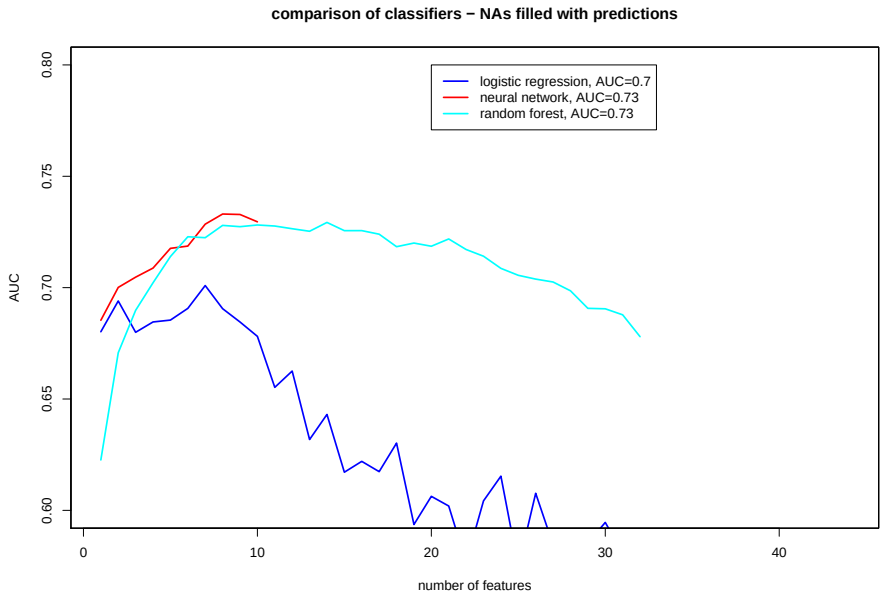


Figure 10: *The classifiers AUC on the predicted data in respective to the number of features used*

- When feature-selection is used to train an classifier, for each feature added to the model the performance in form of the AUC is calculated. In figure 10 we show the AUCs performed for neural networks, logistic regression and random forests in relation to the number of features used.

3.3.4 Importance ranking of the features

- After being able to see which features were used during feature selection we want to create a reliable ranking of the features dependent on our data. Using random forests as classifier we were able to create such a ranking of the features used. It is quite probable that the other classifiers used the same or a similar ranking or weighting of the features.
- The ranking was performed on the test set as well. At first we included all features, than we used feature selection.

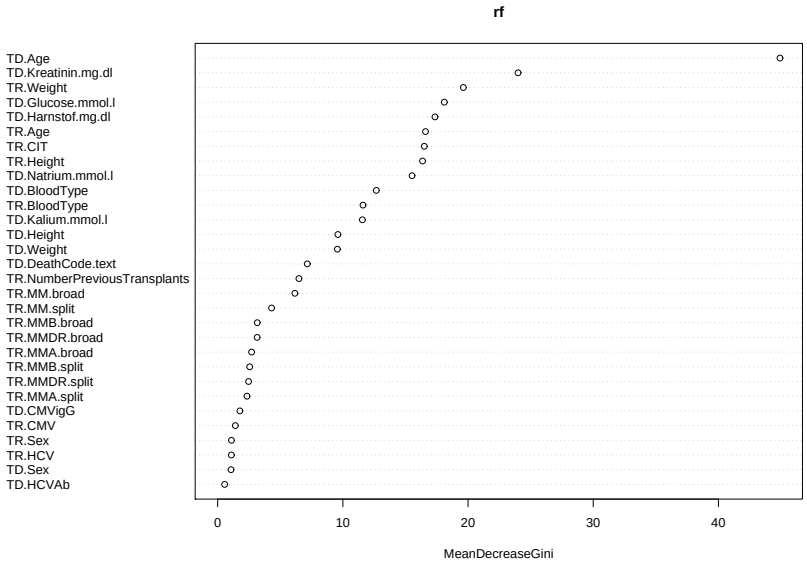


Figure 11: *The feature importance predicted by random forests using all features*

- Figures **??** show the result of the importance ranking of the features. Using feature-selection, as you can see, the

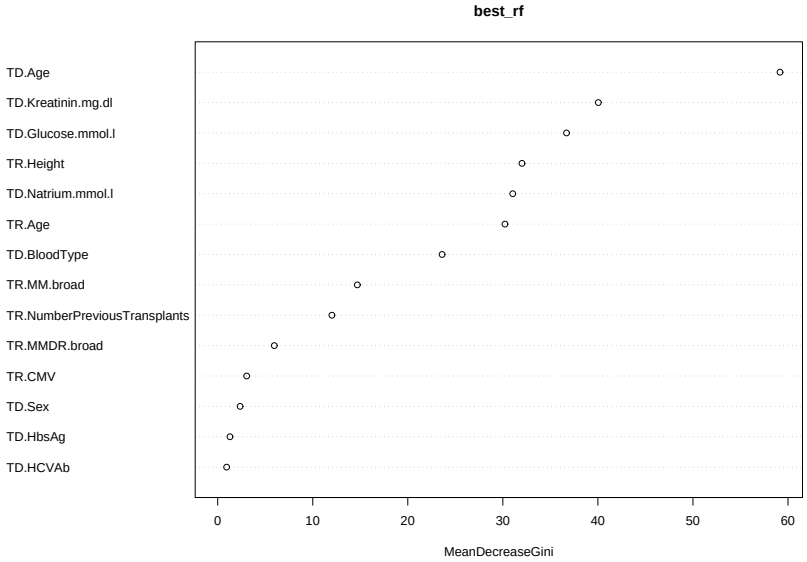


Figure 12: *The feature importance predicted by random forests using feature-selection*

4 Conclusion

4.1 The experiments results

4.1.1 How to fill in the missing values

Description of figures 6 (a)-(c): Figures 6 show the result of the nested CV algorithm performed for each method of filing in the missing values (section 3.2.2). Each figure shows all the used classifiers at a time. The random forests method is an excpetion, due to the fact that it was unable to process the zero-filled data, so figure 6 (a) shows only three classifiers.

We expected to see a difference within each figure between the classifiers and a difference between the different method of filling in the data (figures (a), (b), (c)). What we thought would happen was that in figure 6 (a) the average of the classifiers AUC is the lowest, with the neural network having the best performance, and the logistic regression having the worst, with a clear difference between the ROC curves.

The figure of classifiers being trained on the sample-filled data (Figure 6 (b)) should show similar ROC curves as the zero-filled data, but with a better averaged AUC of the classifiers - so the figure should be the same except the ROC curves should rise higher. The predicted data should show the best ROC curves (figure 6 (c)) for the classifiers, again with neural networks the best classifier and logistic regression the worst.

But in each figure the classifiers perform nearly the same, the ROC curves do overlap in each picture. Also the difference of the averaged AUCs for each figure in comparison with another figure is very small. The AUC of the classifiers is in every figure between 0.64 and 0.67,

Interpretation of figures 6 (a)-(c): To see clear differences between the filling methods there should be a significant amount of values that were filled in. In our final test set - the eighty patients that were never used for training - there were less than ten percent missing values in each feature. These are probably too few to influence the learning of the classifiers.

Another factor still is the size of the training data, which due to nested CV is used in the best possible way, but still might be too small to train the classifiers on. One way of dealing with this problem was to reduce the size of features using feature-selection discussed in section 3.2.5

Description of figures 7 (a)-(c): The figures 7 show the results of the same experiments after using feature-selection before training the classifiers.

As written before we expected to see a difference within each figure between the classifiers and a difference between the different method of filling in the data (figures (a), (b), (c)). Now we actually can see this. For zero-filled data (figure (a)) logistic regression reaches an AUC of 0.67 and the neural network 0.69. For sample-filled data (figure (b)) logistic regression reaches an AUC of 0.57, the neural network reaches 0.7 and the random forests 0.62. And for the predicted data the logistic regression preforms an AUC of 0.51, the neural network again have an AUC of 0.7 again and the random forests one of 0.66.

The ROC curves in each figure (a) - (c) clearly differ, there is no overlapping anymore and one can simply identify the best ROC curve.

Interpretation of figures 7 (a)-(c): The results are far better than without feature selection. After performing it we reduced the data-set and optimized it for each classifier. Feature-selection did not affect the data in a way of disturbing the method of filling in. I.e. by removing all the features that had missing values in it. Otherwise the ROC curves for one classifier would have been the same for different filling methods. But as you can see, they are not. So with this new experiment we actually are able to chose the best method of filling in missing data.

conclusion: By just looking at the results presented in figure 6 (a)-(c) we cannot decide which filling method is the best. We can't even decide which classifier performs best on which method. Not even the AUC is conclusive. So, we decided to combine this experiment with the feature-selection algorithm as well. Figure 7 (a)-(c) shows the results of this experiment. Now we see the results we expected before. Each figure (a) - (c) shows ROC curves that have significant different AUCs for each classifier and each filling method. In each figure neural networks perform the best AUC, random forests second best and logistic regression the worst. And all three classifiers show a difference between the filling-methods after feature selection was performed. So there is a clear difference between the filling methods, as well.

These experiments lead us to three conclusion. At first it is obviously better to use feature-selection in combination with the nested CV, when working with smaller data-sets. Then, when dealing with missing-data the best method to fill in the data is by predicting it (but only if the data is missing at random (section 3.2.2)) using a combined attempt of linear and logistic regression. At last this experiments leads to the conclusion that neural networks are the best way of predicting the outcome, no matter which form of filling-method was used.

4.1.2 Comparison of classifiers

Description of figures 8 (a)-(d): The figures 8 show separately how the classifiers performed on each dataset including all features. Here again you can see that the classifiers performance is nearly the same, no matter which filling method was used. Random forests (figure 8(d)) perform the best with a slightly better AUC of 0.67 for the sample-filled data and data filled using prediction. SVMs (figure 8 (c)) reached an AUC of 0.66 on each data-set, so no difference is seen there. Logistic regression (figure 8(a)) reached an AUC of 0.67 for sample-filled data and an AUC of 0.66 for zero-filled and predicted data. But still this performance is the second

best. The worst results were performed by the neural networks. They reached an AUC of 0.64 for zero-filled data and a AUC of 0.65 on the sample-filled and the predicted data.

Interpretation of figures 8 (a)-(d): All the classifiers do work nearly the same. The difference is so small that we consider it to be caused due to the noise of the data. It might just be a coincidence that in our graphics the random forests performed best. If we had more time we could perform more test to proof or falsify this thesis.

This effect might be due to the lack of data in respect of the number of the features. As discussed in section 3.2.4 when talking about cross-validation, we need as many data-points as possible.

conclusion: It seems like the nested CV is not sufficient enough to train the classifiers efficiently on our data. So in the next experiment we use feature-selection to reduce the number of features.

4.1.3 Comparison of the classifiers when using feature-selection

Description of figures 9 (a)-(c): The logistic regression (figure 9 (a)) shows the biggest differences between the three data-sets. Paradoxically, the zero filled data (the blue curve) has the best AUC with 0.67. The ROC curve on the sample-filled data is represented by the red curve and has an AUC of 0.57, and the ROC curve on the predicted data has an AUC of 0.51. We expected the the results of this experiment in inverse order. The predicted data should do best.

The neural networks (figure 9 (b)) perform better, when looking on the AUCs. The blue ROC curve represents the zero-filled data and has a AUC of 0.69, the red ROC curve represents the sample filled data and has an AUC of 0.7, just like the green ROC curve, representing the predicted data. The curves still overlap but the result is better than the curves without feature-selection.

Random forest (figure 9 (c)) do also perform better with feature-selection. The red ROC curve represents the classifiers performance on the sample-filled data and reaches an AUC of 0.62, where the performance on the predicted data reach an AUC of 0.66, represented by the green ROC.

Interpretation of figures 9 (a)-(c): The paradox performance of logistic regression should be further investigated, but due to the lack of time, we can't say why the performs best on the zero-filled data and worst on the predicted. It is possible that the feature selection algorithm does not work on logistic regression

As presumed the neural networks perform worst on the zero-filled data and best on the predicted data. But there difference is quite small. Either the neural network is able to generalize well enough or the fact that the missing values in the test-data are so few, is responsible for the little difference, where we expected to see a bigger one. The random forests perform best on the predicted data, which is what we expected. There even is a difference of 0.04 in the AUC.

conclusion: There is a big improvement between the results using all features (figures 8 (a)-(c)) and using feature-selection (figures 9 (a)-(c)). Now it is possible to see the differences between the performances of the classifiers. So once again the feature selection was helpful in simplifying the model and therefore in producing better training results on each classifier.

As presumed the neural network again is the classifier with the most reliable performance on our data. The AUC of 0.7 on the test set is a good value that leads to the presumption that our neural network model and our training algorithm is able to predict the outcome on data from other transplant centers as well.

Description of figure 10: The curves for the neural network (red), the logistic regression (blue) and the random forests (cyan) are of different length, height, form and they all start at a different position on the y-axis. The length is number of features chosen, the height, the AUC performed according to the number of features chosen and the starting point at the y-axis is the AUC performed after choosing the first (which is the best) feature out of the dataset.

Interpretation of figure 10 & the features chosen during feature-selection: The figure shows the AUC during each step of the feature-selection iteration. For each feature added a new AUC is calculated. So we are able to reconstruct the role of each feature for each classifier. Surprisingly the feature selected at first is always the donor age. In our data this feature seems to play the most important role, or the classifiers would choose different features at first. The following features chosen are different even though a few features occur in the selection of other classifiers well, or even in all of them.

The features that occur in each classifiers selection are:

- the donor age
- the number of previous transplants
- the donors creatinine value

In two classifiers the following features appear:

- the recipients weight
- the donors HCV status
- the donors HBV status

The other features appear only once in each selection.

conclusion: In our data the features chosen by each of the three classifiers used, seem to be the most important features. So it seems like they also are the factors most important when choosing matching recipients and donors. But, surprisingly, non of the features used in medicine like the CIT or the HLA did appear more than once or did appear at all.

Further studies are needed to confirm or to negate these features. It is possible that due to the small amount of data, in our case it these features seem correct, but after adding data, or using different data, other features are chosen. Again we can't be sure, because we only have limited time and data available.

4.1.4 Importance ranking of the features

Description of figures 11 & 12: The figures show the estimated ranking of the features performed by random forests. In both figures the features *donor age* and *creatinine value* are on top. *Glucose* also appears in both figures within the top five features. In figure 11 the HLA is not even in the first half, and in figure 12 most of the HLA representing variables were already excluded during feature selection. Only the HLA DR broad still is in the dataset, which is considered crucial in combination with the -A and -B locus. [24]

Interpretation of figures 11 & 12: These graphics as well as the feature-selection for neural networks, logistic regression and random forests, put the donor age on top. This is quite strange, though considered a factor influencing the transplantation the donor age was never during our research considered more important than the HLA or the CIT. Putting the creatinine value this high is strange as well, but less confusing. The creatinine usually is used to estimate the current kidney function. So there is a direct connection between the creatinine, a healthy or marginal kidney and the transplantation outcome.

conclusion: Our data shows in two experiments using three different classifiers that the donor age is the most important feature during transplantation. This is as fascinating as confusing though the features commonly used to choose a recipient and donor match, are not high or not at all in the list of selected features. It's highly improbable that three different classifiers did this by mistake. We assume that this prediction is correct on our data. But as mentioned several times before, we have doubts about the data. Right now we cannot presume any ranking made reliable. There are things like the high ranking of creatinine that are understandable and go along with

4.2 Summarized conclusion

During our experiments we saw, that a good way of filling in missing values i.e. in clinical data is to use linear and logistic regression to create regression models on the discrete and continuous data. Our classifiers confirmed that this method is in fact better than filling the missing values with sampled values or by just setting them to zero.

Furthermore we saw that feature selection can be used to remove redundant features, so only the features that perform information are kept. The reduction of the features creates a better ratio of the number of features and the data-points and therefore helps classifiers to generalize the data and reduces the probability of overfitting the model.

We also saw, that using feature selection it is possible to create a ranking of the used features, and that this list of features partially accorded to the list of features selected by other classifiers, or using the ranking method of random forests.

In the ranking we found, that the HLA in our data did not have a high rank at all. The features mostly used were **the donor age, the number of previous**

transplants and the **creatinine value** in feature selection and **the donor age** and the **creatinine value** during the ranking of random forests.

Further studies are needed to show whether our results are limited to our small amount of data, or if the classifiers work the same way on bigger datasets.

4.3 Future work

We highly recommend for further work to include more patients. With a bigger number of transplants it is quite possible that the results for the same methods will improve. It might even be possible to include features we had to exclude due to missing values. (In our case we had to cut off half of the features.)

More complex information might be hidden in the data. There might be dependencies between features so subgroups can be identified. A subgroup might be a overweight, $>$65 year old male donor with a history of hypertension and diabetes. We expect this group to have a different outcome than a 25 year old donor who's BMI is normal. By clustering the data into subgroups like this we reduce the number of features and therefore reduce the problem complexity. A ad hoc attempt of clustering could be looking at the distribution of each feature and then, in dialog with a medical advisor, setting subgroups of each feature. In the second step than a variation of the feature selection algorithm could be used to identify the one subgroup of a feature that works best with the subgroups already identified. For the network to identify the subgroups dummy variables could be introduced.

It could also help to introduce dummy variables to reduce the data. Obesity for example could be a discrete variable used, instead of the whole range of possible weights. Simplifying the dataset like this should have a similar effect as feature-selection had $\rightarrow$ redundant information is removed.

During our experiments we used the outcome variable as described in section 3.1.2. We already mentioned there that the setting of the outcome is both crucial, for training classifiers, and difficult, due to the fact that mostly the outcome is not obvious. Another way of setting the outcome would be using the 3-year GFR instead of the 1-year GFR. For three years after the transplantation the kidney function is more reliable. Another attempt would be not to use a boolean classifier at all, but to predict the numeric GFR, or multiple binary values representing intervals along the GFR.

Using a different programming language could improve the time needed to cal-

culate. We used "Gnu R" which is known for being really slow during loops. In the feature-selection algorithm we have to perform $(\text{features} \cdot \text{outerloop}^{(\text{innerloop} \cdot \#\text{nodes} \cdot \#\text{decay})})$ iterations, which for 44 features is quite a lot. Another language that could be used is matlab that just like R has packages for artificial neural networks, like the netlab toolbox[2]. As does python with its pybrain[3] package.

References

[1] "Eurotransplant international faundation annula report," pp. 38–40, 2008. [Online]. Available: http://www.eurotransplant.nl/files/annual_report/ar_2008.pdf

[2] M. Ahmad, E. Cole, C. Cardella, D. Cattran, J. Schiff, K. Tinckam, and S. Kim, "Impact of deceased donor diabetes mellitus on kidney transplant outcomes: a propensity score-matched study," *Transplantation*, 2009.

[3] K. Ahmed, N. Ahmad, M. Khan, G. Koffman, F. Calder, J. Taylor, and N. Mamode, "Influence of number of retransplants on renal graft outcome," *Transplant Proc.*, 2008.

[4] K. Armstrong, "Impact of obesity on renal transplant outcomes," *Nephrology (Carlton)*, 2005.

[5] M. Bartels, H. Otten, B. E. van Gelderen, and A. V. der Lelij, "Influence of hla-a, hla-b, and hla-dr matching on rejection of random corneal grafts using corneal tissue for retrospective dna hla typing," *Br J Ophthalmol*, 2001. [Online]. Available: http://bjo.bmj.com/content/85/11/1341.long

[6] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1995.

[7] ——, *Pattern Recognition and Machine Learning*. Oxford University Press, 2006.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers."

[9] M. E. Brier, P. C. Ray, and J. B. Klein, "Prediction of delayed renal allograft function using an artificial neural network," *Nephrol Dial Transplantation*, 2001. [Online]. Available: http://ndt.oxfordjournals.org/cgi/content/abstract/18/12/2655

[2] http://www.ncrg.aston.ac.uk/netlab/index.php

[3] http://www.pybrain.org/

[10] Y. Cho, J. Cecka, D. Gjertson, and P. Terasaki, "Prolonged hypertension (> 10 years) is a significant risk factor in older cadaver donor renal transplants," *Transplant Proc*, 1999.

[11] B. Cohen, J. Smits, B. Haase, G. Persijn, Y. Vanrenterghem, and U. Frei, "Expanding the donor pool to increase renal transplantation," *Nephrol Dial Transplantation*, 2005. [Online]. Available: httphttp://ndt.oxfordjournals.org/cgi/content/full/20/1/34

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, 1995.

[13] S. Fox, *Human Physiology, Tenth Edition*. McGraw-Hill Science/Engineering/Math, 2004.

[14] S. Geman, E. Bienenstock, and R. Doursat, *Neural Computation*, 1992. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/neco.1992.4.1.1

[15] Guyton and J. E. Hall, *Textbook of Medical Physiology*. McGraw-Hill Science/Engineering/Math, 2005.

[16] C. Hall, J. Sansom, M. Obeid, P. Dawson-Edwards, B. Robinson, A. Barnes, and J. Blainey, "Agonal phase, ischaemic times, and renal vascular abnormalities and outcome of cadaver kidney transplants." *Br Med J*, 1975. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pubmed&pubmedid=1100186

[17] T. K. Ho, *Random decision forests*, 1995. [Online]. Available: http://www2.computer.org/portal/web/csdl/doi/10.1109/ICDAR.1995.598994

[18] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, New York, 1987.

[19] S. V. Mahadevan and G. M. Garmel, *An Introduction to Clinical Emergency Medicine: Guide for Practitioners in the Emergency Department*. Cambridge University Press, 2005.

[20] B. M. Marlin, "Missing data problems in machine learning," 2008. [Online]. Available: http://www.cs.toronto.edu/~marlin/research/phd_thesis/marlin-phd-thesis.pdf

[21] G. Mayer and G. G. Persijn, "Eurotransplant kidney allocation system (etkas): rationale and implementation," *Bephrology Dialysis Transplantations*,

November 2005. [Online]. Available: http://ndt.oxfordjournals.org/cgi/content/full/21/1/2

[22] G. Orr, "Cs-449: Neural networks," willamette University. [Online]. Available: http://www.willamette.edu/~gorr/classes/cs449/backprop.html

[23] N. Petrovsky, S. K. Tam, V. Brusic, G. Russ, L. Socha, and V. B. Bajic, "Use of artificial neural networks in improving renal transplantation outcomes," *Graft*, 2002. [Online]. Available: http://graft.edina.clockss.org/cgi/reprint/5/1/6.pdf

[24] C. Ponticelli, *Medical Complications of Kidney Transplantation*. Informa HealthCare, 2007.

[25] L. Resende, "Reconsideration of the limits on donor age acceptability."

[26] L. Resende, J. Guerra, A. Santana, C. Mil-Homens, F. Abreu, and A. da Costa, "Impact of donor age on renal allograft function and survival," *Transplant Proc*, 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19376354

[27] R. Rojas, *Neural Networks - A Systematic Introduction*, chapter 7. The back-propagation algorithm.

[28] F. Rosenblatt, "Principles of neurodynamics: Perceptrons and the theory of brain mechanisms," *Washington, DC: Spartan)*, 1962.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *MIT Press*, 1986.

[30] B. Scholkopf and A. Smola, "Learning with kernels," 2002.

[31] C. K. Stone and R. Humphries, *Diagnosis and Treatment Emergency Medicine*. McGraw-Hill Medical, 2007.

[32] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 1995.

[33] Wikipedia, "Neural network." [Online]. Available: http://en.wikipedia.org/wiki/File:Neural_network_example.svg