TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Diplomarbeit

One Class Classification of Longitudinal Data

Ausgeführt am Institut für
Stochastik und Wirtschaftsmathematik
der Technischen Universität Wien

unter der Anleitung von

Ao.Univ.Prof.Dr. Peter Filzmoser

durch

Sebastian Hödlmoser
Habsburgerstraße 82/1/4, 2500 Baden

Wien, 1. Dezember 2015

Abstract

In clinical treatment, a variety of data is collected to assess a patient's condition and determine necessary interventions. The common practice is to periodically observe a selection of features and compare these measurements to defined thresholds. The question arose, if there is a benefit in considering not only the newest updates of features, but taking prior observations into account as well. To examine possible advantages, a dataset at the Lorenz Böhler intensive care unit was collected, where the progression over time of biomarker levels of trauma patients were documented. Motivated by a patient monitoring, this work deals with the problem of classification for this *longitudinal data*. The dataset demands a classification based on knowledge of a positive class only, hence it was approached by a *one class classification*. Further, the classifier has to deal with both, the unbalanced data as well as its updating nature. Based on a *linear mixed effects regression model*, characteristics of the class of survival patients are estimated. Deviations of observation from this estimations are penalized and evoke a negative classification. The mixed model approach allows not only for estimating class characteristics, but further features, e.g. it can expand the classification by a visual support. The gold standard method of brain injury assessment for trauma patients is used to benchmark the proposed longitudinal classifiers. Different views on evaluation show slight improvements to the benchmark, which however are in conflict with additional effort.

Zusammenfassung

Bei der klinischen Überwachung wird eine Vielzahl an Daten erhoben, anhand derer der Zustand eines Patienten und notwendige Behandlungen bestimmt werden. Ausgewählte Biosignale werden in periodischen Abständen gemessen und mit kritischen Werten verglichen. Es stellt sich die Frage, ob eine Berücksichtigung vorangegangener Messungen einen Informationsgewinn gegenüber einer Bewertung, welche ausschließlich auf den neusten Beobachtungen basiert, darstellt. Zu diesem Zweck wurde ein Datensatz in der Lorenz Böhler Intensivstation erfasst, welcher die zeitliche Entwicklung von Biomarker-Werten von Trauma-Patienten enthält. Die vorliegende Arbeit beschäftigt sich mit der Klassifikation dieser *Longitudinaldaten*. Der Datensatz erfordert eine Klassifikation, die auf Kenntnis von nur einer der auftretenden Klassen basiert, einer sogenannten *One Class Classification*. Diese muss sowohl mit den unterschiedlichen Beobachtungszeiträumen der Patienten, als auch mit der periodischen Neuerhebung der Merkmale umgehen können. Anhand eines gemischten linearen Regressionsmodells werden die Charakteristika der überlebenden Patienten geschätzt. Abweichung von diesen zu gemessenen Werten können - in Abhängigkeit ihres Ausmaßes - zu einer negativen Klassifikation führen. Neben der Schätzung der Klassen-Charakteristika dient das gemischte Modell weiteren Vorteilen, wie z.B. einer visuellen Hilfestellung zur Bewertung eines Patienten. Zur Auswertung der vorgeschlagenen Klassifikation dient die derzeitige Standardmethode zur Bewertung von Schädel-Hirn-Traumata, welche auf der jeweils aktuellsten Messung eines bestimmten Biomarkes basiert. Die longitudinale Datenauswertung konnte die Ergebnisse der (sehr einfachen) Standardmethode teilweise übertreffen, jedoch zum Preis eines erheblich größeren Aufwands.

Contents

1 Introduction

Monitoring of patients in clinical treatment involves a variety of different kinds of data. Features such as sex, weight, blood pressure, heart rate, oxygen saturation etc. are widely understood and contribute to the physician's assessment of a patient's health status. An educated and experienced physician takes all this information as well as their influences on each other into account and bases his/her further actions on them. This applies to every stage of treatment spanning from first aid over emergency treatment to intensive care. In the latter anyhow, a patient's monitoring contains a much broader sampling of body signs besides the already mentioned. Additional informations such as continuous EKG observations, blood analyses, injury scales etc. are collected. The hope behind this broad collection of data is to sharpen the physician's knowledge about the patient's health status and deliver fast and reliable alarming systems for necessary interventions. Considering the raw data in form of numbers, charts and tables tends to be an unpleasant task and implies the dangers of missing non-obvious phenomena and correlations between different features. This is why it is important to develop methods to analyze and summarize all available data to support physicians in their decisions.

The following thesis deals with the task to classify intensive care patients during clinical monitoring, based on the analysis of a combination of categorical and periodically measured data. To do this, a data set at the Lorenz Böhler intensive care unit was collected, where 103 patients under intensive care were observed. The data consist of clinical information as well as levels of biomarkers, which were observed every 24 hours since the beginning of treatment and until release of intensive care or decease. This mixture of a cross sectional sample (i.e. the patients) with a time series for each individual (i.e. each patient's biomarker levels over time) is referred to as *longitudinal data*. The goal is to assess the health status trough

- a combination of the most useful features
- the exploitation of an individual's 'history', meaning the use not only of the newest, updated observations of biomarker levels but also of the prior ones

The gold standard in clinical practice to screen trauma patients for *traumatic brain injuries* (TBI) is to measure the *s100* level (a biomarker sensitive to brain injuries) and compare the observations to a critical threshold. This neglects the prior observations as well as other features sensitive to brain injuries (e.g. other biomarkers). Further it uses one threshold for every patient, independent of age, sex, weight, length of treatment etc. Considering the longitudinal data trends of features enables us to analyse the changes over time in a patient's condition. Incorporating individual specific data allows us to tailor the thresholds/cutoffs to the current situation rather than declaring a universal reference value. With a constant monitoring of a combination of thoroughly selected features it is possible to deliver an automatized decision support system that can be extended by graphical visualizations of a patient's condition.

To get there, we will at first take a closer look at the dataset (chapter 2). We will highlight its important features and explain the crucial observations with respect to the classification process. In chapter 3 we will explain all the important parts of the classification in detail. We aim for an assessment of a patient's health status by assigning him/her to either a positive or a negative class, i.e. the survivors/non-survivors. A common practice to do this is to estimate the distributions of occurring classes and check each patient for resemblance. But as we will see, our situation requires a slightly different approach called *one class classification* (OCC). Our dataset makes it impossible (or at least inaccurate) to estimate the negative class' distribution, so the strategy will be to estimate the characteristics of the positive class only and check every patient for deviations.
The longitudinal nature of the data has to be taken into account when estimating class characteristics as well as in the interpretation and evaluation of the models and results. Chapter 4 explains the choice of parameters, the selection of the optimal features and ways to evaluate the predictions. The consecutive observations of biomarker levels lead to consecutive classifications and since we imagine a clinical monitoring during intensive care treatment, early predictions should be as reliable as possible. Further we are dealing with asymmetric misclassification costs, meaning a positively classified negative case is more severe than a negatively classified positive. Chapter 5 illuminates different views on the results of various models and interprets them. In chapter 6 we will conclude the results.

All graphs and numerical data analyses were computed with R 3.1.1. The implemented codes as well as a list of used R packages can be found in the appendix.

2 The Dataset

In the following chapter we will give a detailed view at the dataset collected at the Lorenz Böhler intensive care unit (ICU). It consists of 103 patients which have been in treatment due to multiple traumata and/or traumatic brain injuries. Six biomarkers where recorded subsequently every 24 hours after the first measurement:

- *s100* (μ/l), *gfap* (μg/l), *nse* are TBI specific biomarkers; *s100* is currently the gold standard to assess brain damage of emergency patients
- *il* (pg/ml), *crea* (mmol/l) are specific to inflammatory symptoms of an organism
- *pct* (ng/ml) is a biomarker sensitive to infections of bacterial origin.

So the data consists of injury and inflammatory/infection specific biomarkers. An ideal classifier should incorporate both symptoms in its decision.
For roughly a quarter of patients *nse* measurements are absent, for a few patients single observations of different markers are missing at random. For 73 patients the data contains a positive or negative diagnose of a septic shock. Further every patient is assigned to a TBI injury scale reaching from 1 to 3 with decreasing severity.
The outcome of every patient's treatment was recorded as well, we separate them into two classes. The *positive class*, the survivors, denotes the patients which were released of intensive care when considered fit. These amount to approximately three quarters of all recorded patients. The remaining quarter passed away during treatment and form the *negative class* or the non-survivors.

What makes this dataset particularly interesting are the consecutive observations of biomarker levels. In contrary to a sample of a feature at **one point in time** on a number of individuals (i.e. cross sectional data) or observations of **one individual** over some time (i.e. time series data) the data consists of the mixture of theses two types, called *longitudinal data*. Further the data is *unbalanced*, i.e. the observation times differ from individual to individual.
As [Diggle et al., 1994] point out in their introduction, the advantage of longitudinal data lies in the ability to analyse changes over time. If there are correlations of changes of features over time and the change of a patient's condition, longitudinal data lets us examine theses changes - which then can serve as indicators for complications.

2.1 Graphic Exploration

Every patient was observed at least 3 times, the longest time span is 22 days. Figure 2.1 shows boxplots of days under observation separated for the two classes. As we would expect, the mean length of ICU stay of a deceased patient is significantly less than a survivor's, but still the boxes overlap to a great extent. This is important, since a dataset were a patient's outcome strongly depends on the length of stay alone would bias our

results. Also we conclude that it will be of great interest to predict a patient's condition as early as possible - while keeping the predictions reliable.

Figure 2.2 depicts the progression of the six biomarkers against days post trauma (dpt). Every trajectory represents one patient, the curves are separated into the two classes survival and non-survival. Here one can already examine the trajectories for noticeable trends and promising biomarkers. The first three, *pct*, *s100* and *gfap* for instance show a clear distinctive behaviour in the different classes. The survival curves are decreasing in spite of high first measurements, while some of the non-survival curves increase after a while. This is especially striking for *s100* and *gfap*. For survivors a strong reciprocal trend similar to an exponential decay can be observed. Some non-survivors show quadratic and other non-linear time-trends. Not every marker appears to be as promising as *s100* or *gfap*, such as *il*, where differences between the positive and negative class are less obvious.

A subdivision not only into positive and negative class with respect to survival, but a further separation into subclasses with positive and negative diagnose on septic shock, reveals another interesting and important property of the data. Patients without sepsis diagnose are neglected for these plots. Figure 2.3 shows the additional subdivision. The graphs for *s100* and *gfap* are of special interest. Here we can obtain idiosyncrasies in the non-survival subclasses 'septic' and 'non-septic'. *s100* levels of non-survivors without a septic shock appear to be very similar to the *s100* levels of those who survived. But for septic non-survivors one can observe a strong increase. A similar phenomenon can be seen in *gfap* concentrations. As mentioned above, these markers are brain-injury specific. We obtain deceased patients with high *s100*/*gfap* levels (bottom right in the corresponding graphs), those are the ones who probably died of TBI. The top right trajectories of *s100*/*gfap* - the septic non-survivors - are inconspicuous, those patients most likely died of a septic shock.
This leads to two insights. First, a classifier that should determine the health status of a patient has to incorporate markers which are specific to different symptoms. It is desirable to not only detect brain injuries but other potential complications as well. Second, while we can expect a normal behaviour of different biomarkers for a healthy human, reasons for complications or decease can be diverse. This diversity transfers to the characteristics of biomarker curves as it can be observed in *s100*/*gfap* non-survival curves. This is strong evidence that the non-survivors are to be subdivided in two or more subclasses. Estimating the characteristics of these subclasses would require a much broader sample and is likely to be rather inaccurate. This is a key factor for a classification in terms of a *one class classification*, as discussed in section 3.2.

Figure 2.4 helps to examine the value of the TBI injury scale, which is recorded for each patient. With group 1 - 3 we denote patients with injury scales of 1 - 3, correspondingly. The graphs show a separation of survivor curves with respect to the TBI injury scale. We are looking for differences in the subclasses that have to be incorporated in the classifier later on. In order to get a clearer view, the limits of the y-axes are chosen manually, large values which skew the picture lie outside of the frames. For *s100*, *gfap*, *il* and *crea*, no significant differences can be observed. However, *pct* levels of group 1 and group 2 patients seem to be slightly increased compared to group 3. In group 2, *il*

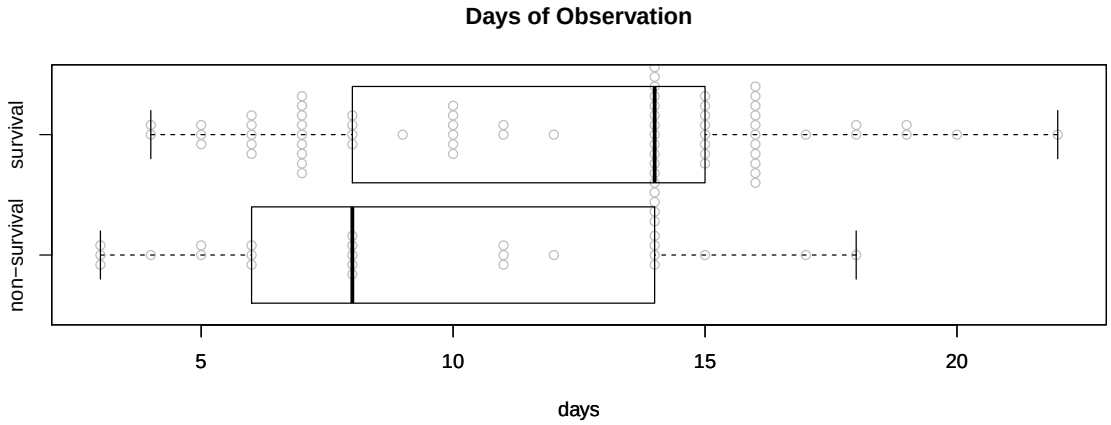


Figure 2.1: Boxplots of days of observation for positive and negative class.

levels tend to be higher than in the other two groups. Of course one has to be careful with observations like that. Roughly 50% of the recorded patients belong to group 1, 40% to group 2 and the remaining 10% form group 3. So the sample size of group 3 is significantly smaller that the others, which makes it problematic to determine typical phenomena. Furthermore, often one is tempted to detect patterns where there aren't any.

As mentioned, it is clinical practice to consider only the current observations of biomarker levels. Figure 2.5 depicts boxplots of the measurements over the first 15 days post trauma for the two classes and should show the discriminatory power of current observations. To get a clearer picture, the marker values were transformed via $\log(marker + 1)$. The graphs confirm some of the prior observations. *gfap* and *s100*, especially at the beginning of observation, are significantly higher in the negative class. Also *pct* and *nse* levels show this trend, but in a lesser extent. What is remarkable is that *il* levels - which didn't show striking differences in the prior considerations - develop a rising distinctive behaviour after about 5 days. Again we will see later, if this helps to improve classifications. The less promising marker in this view is *crea*, where nearly all boxes overlap to a great extent.

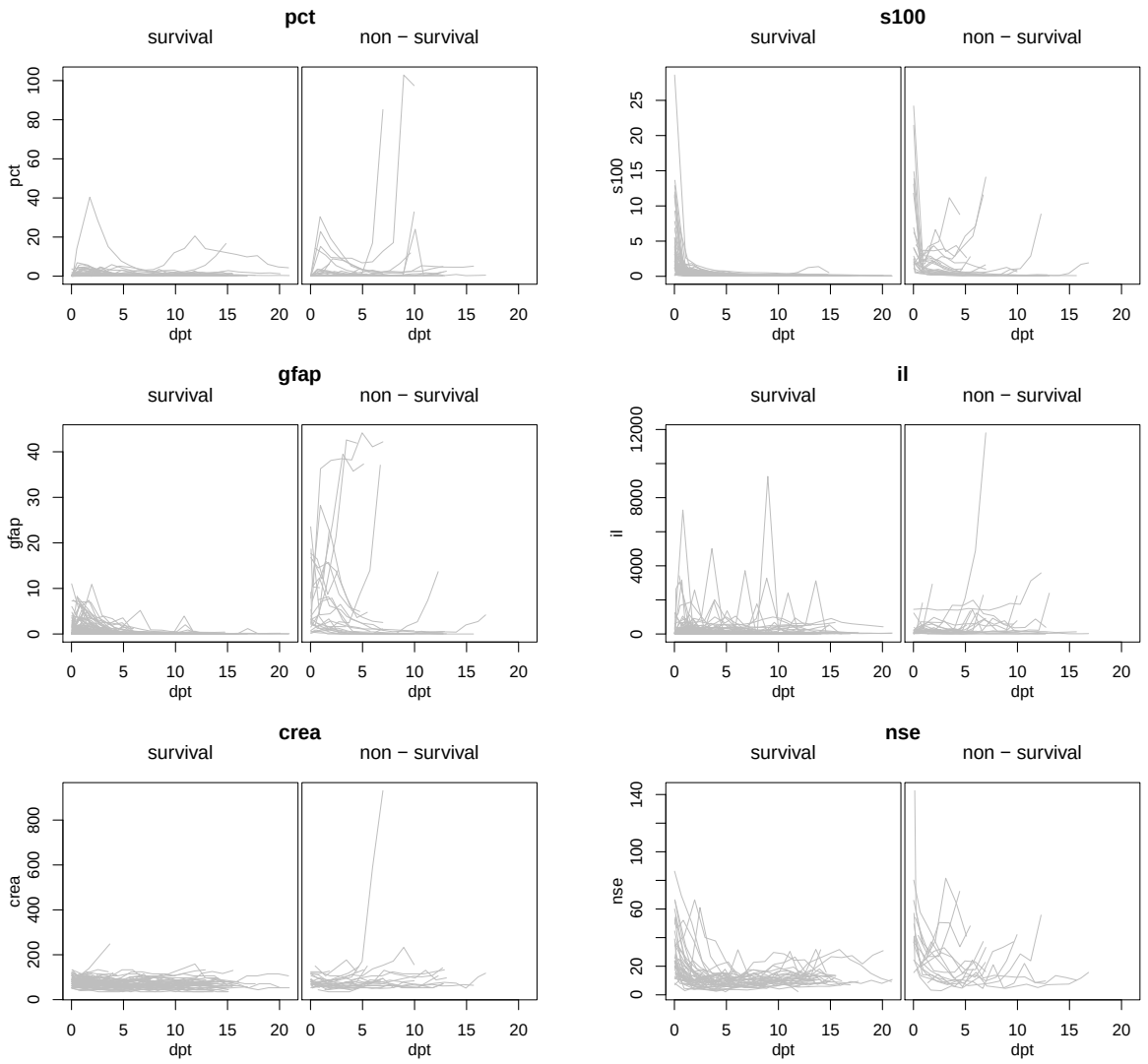


Figure 2.2: Biomarker over time, separated into positive (survivors) and negative (non-survivors) class.

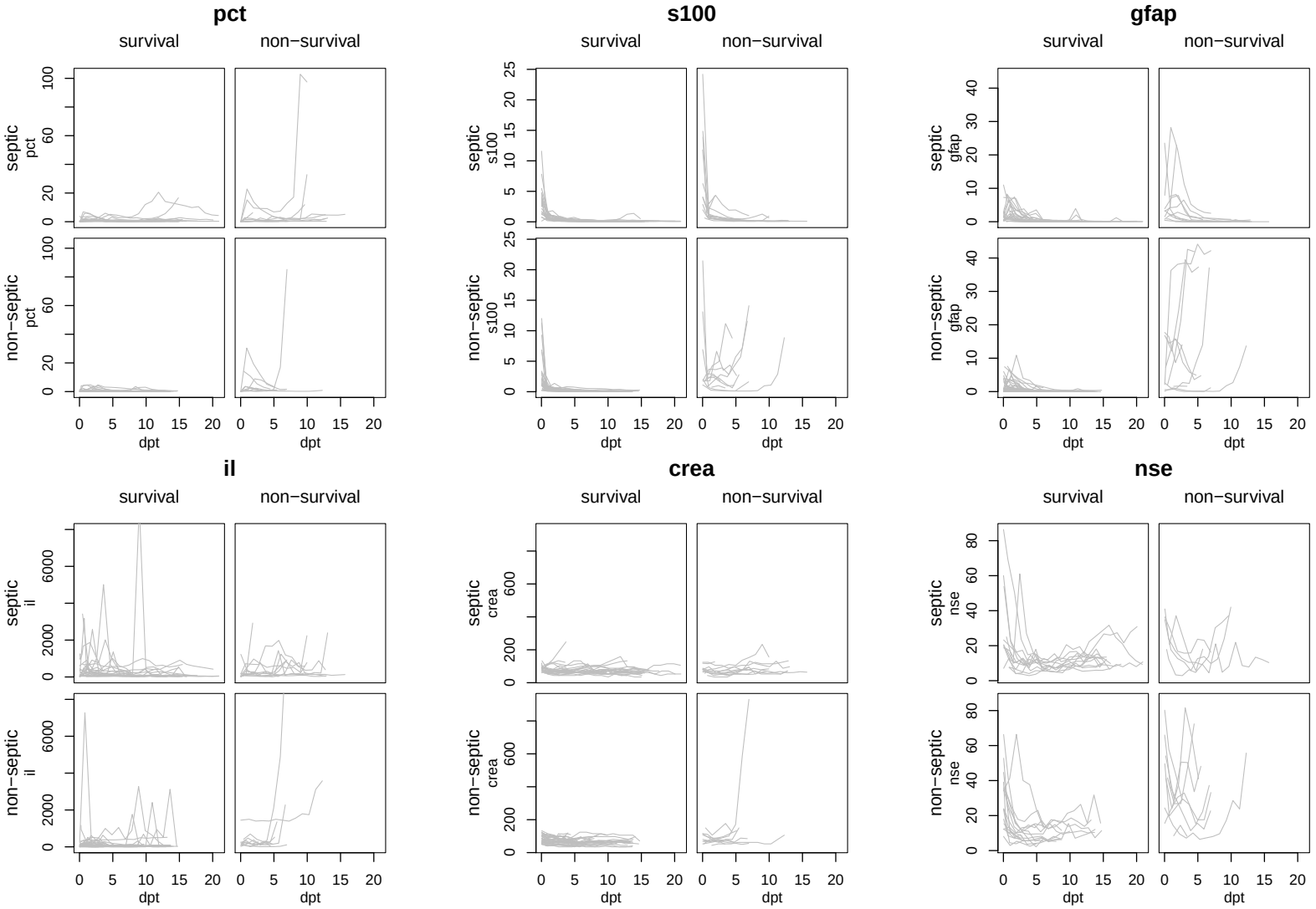


Figure 2.3: Patients divided into subclasses with pos/neg survival and pos/neg sepsis.

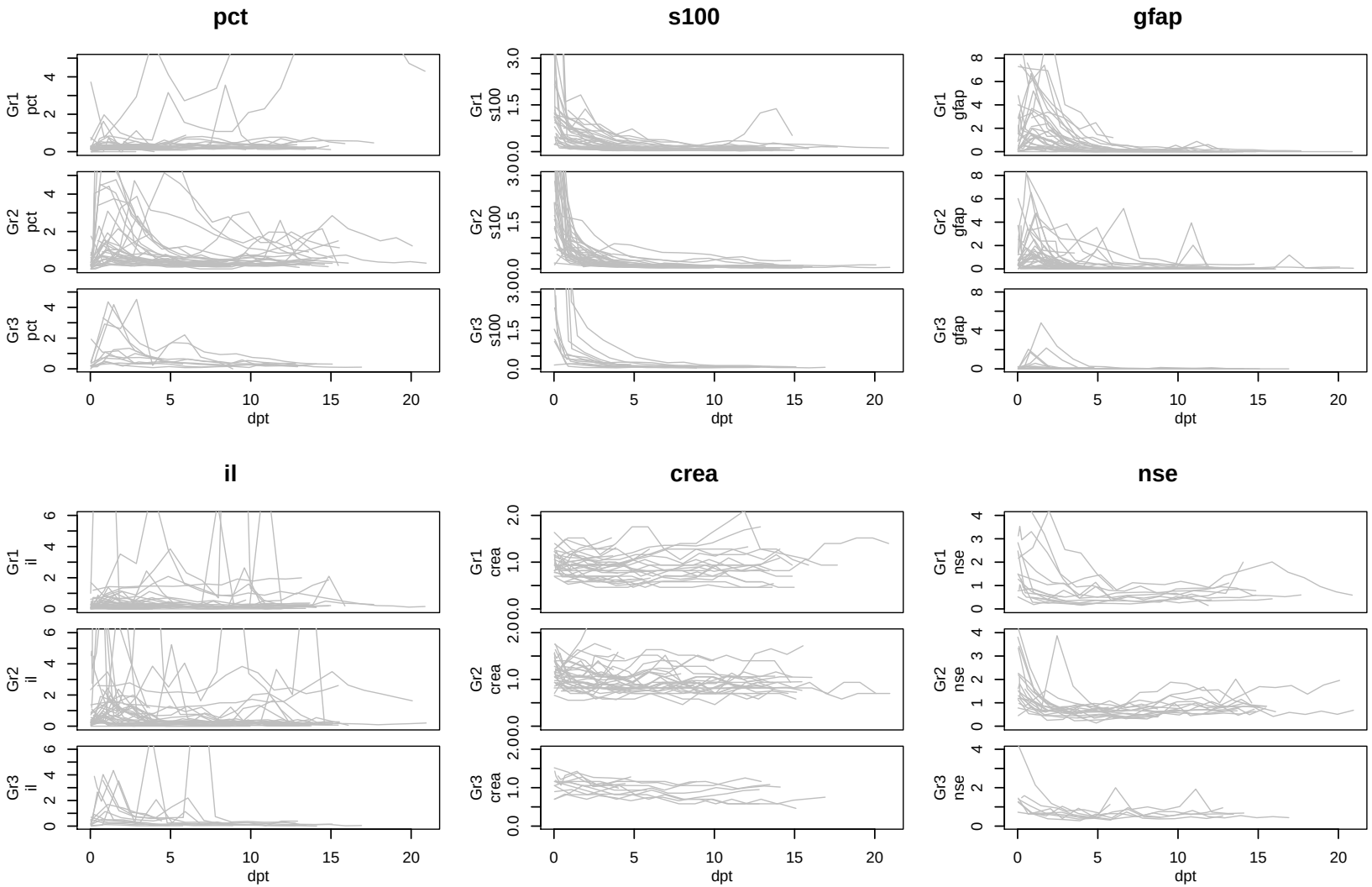


Figure 2.4: Positive class divided into the three TBI groups.

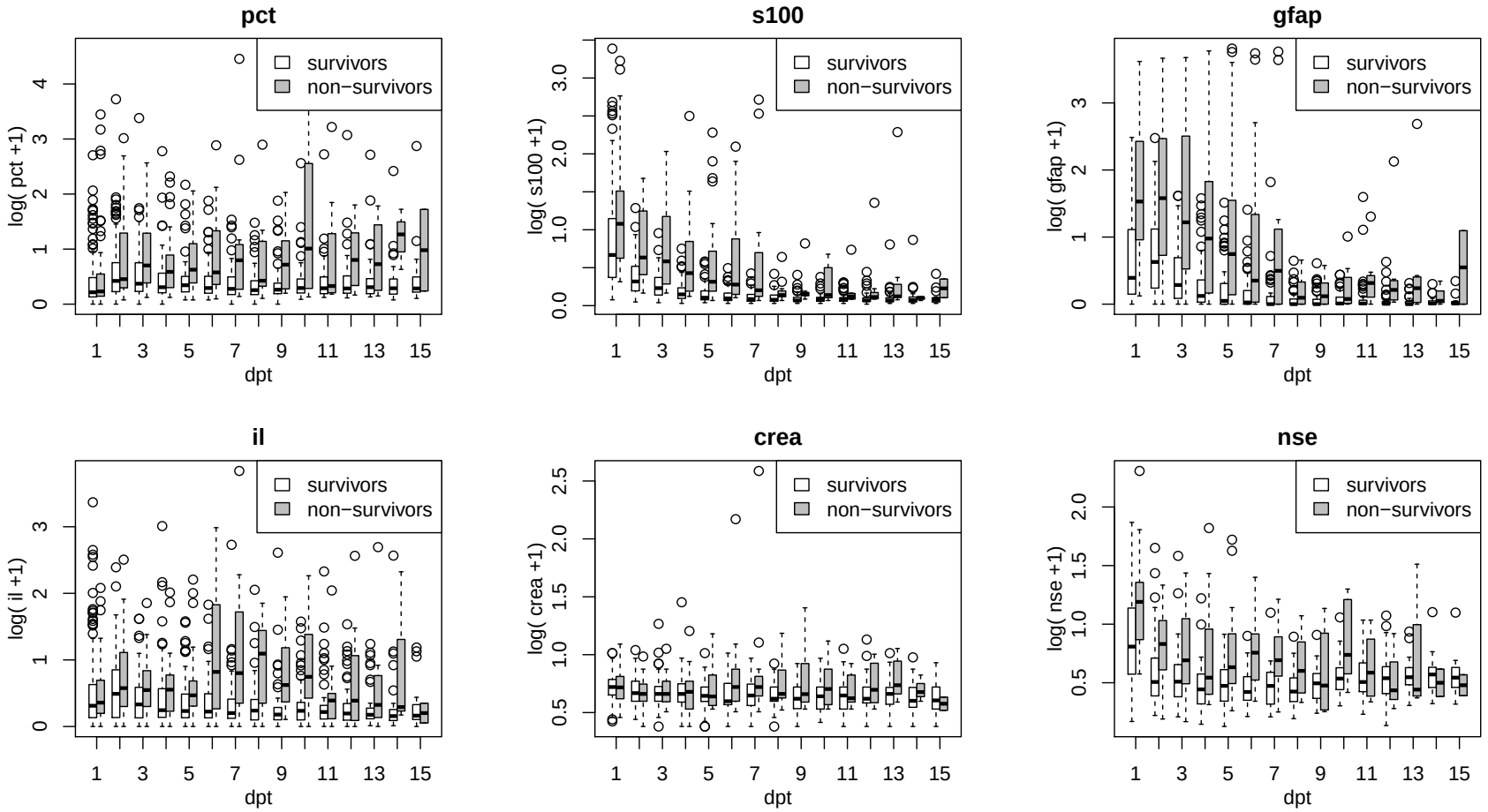


Figure 2.5: Boxplots of biomarker concentrations over time, separated for survivors and non-survivors.

3 Methods

The classification process and its evaluation consists of various elements. In the following section these elements and their theoretical background will be explained. We will give a brief overview about the procedure and then discuss all the parts in detail.

3.1 Classification Process

When a trauma patient is delivered to intensive care, various body signs are collected and analysed to check the patient's health status. The goal is to improve this analysis and provide a reliable decision support for the physicians. In the following, we imagine an optimal selection of features such as patient specific data, e.g. sex, age, weight, injury scales etc., and periodical observations of various biomarkers.
Unfortunately, our dataset consists besides of 'continuous' biomarker levels only of a traumatic brain injury (TBI) scale and an inaccurate sepsis diagnosis. So due to the lack of better patient specific data, we focus on the biomarker concentrations and the TBI scales. Any other information can easily be included in the model, compare section 3.3.
The classification process can be summarized as follows:

- Every patient's biomarker levels are surveyed in given intervals and the data is updated periodically.
- The measured progressions of marker levels over time are compared to expected biomarker curves, which are computed for every patient.
- Deviations from measured to expected curves are penalized with respect to their intensities; thus not only current observations are compared to a critical value, but also a patient's marker history is taken into account; also there doesn't have to be a 'global' critical value but an expected curve for each patient given he/she belongs to the positive class.
- With every update a probability of membership for the positive class is assigned to the patient. A cutoff determines if the patient is to be classified as a positive or negative case.
- This classification serves as a monitoring of the patient's health status and supports the physician's further actions.
- Observed marker levels and expected survival curves can serve for additional visual support.

The *expected survival curve*, *mean trajectory* or the *population mean*, meaning the trajectory of biomarker levels to be expected, given the patient belongs to the positive class, is crucial. Deviations from this mean curve determine a patient's classification. It is estimated through a *mixed effects regression model* (section 3.3). The response variable

is the biomarker level, explanatory variables can be functional expressions of time and patient specific data. The regression model then delivers the expected survival curve and variance in terms of a multivariate random variable. This allows to calculate a weighted distance between observed marker levels and expectation, which can be linked to a probability of the observation under the assumption of normal distribution with the calculated mean vector and covariance matrix (section 3.5).

As explained in chapter 2, we are dealing with two classes - the negative and the positive class. We can estimate the positive class' characteristics, but with the negative class we have to deal in another way. A better knowledge about the non-survivor class would allow an estimation of an expected non-survival curve as in the case of survivors. However this would require a much broader sample of negative cases, which we do not have. This is why we define a negative case as every case that is not positive and apply a *one class classification*, see section 3.2.

The same data set is used for both fitting the mixed model and testing the classifier's performance. It is not desirable to classify a patient based on a fitting where the patients data was included. To avoid this *overfitting*, we divide the dataset into ten folds of approximately same size with a similar survivor/non-survivor ratio in each fold. To classify one fold we use the remaining nine as data to fit the model. This process is repeated for each fold. This so called *cross validation* (section 3.6) avoids an advantage for the classifier which it would not have in reality, hence would bias the results.

Some data points are missing at random, for a bunch of patients the data lacks of *nse* observations. The mixed model approach to estimate the survival mean delivers a convenient way to impute these missing biomarker levels. With the mixed model fit we can declare the most probable values for missing biomarker levels (section 3.7)

To assess the different classifiers' performances we use the *receiver operator characteristic* (ROC) curve and its *area under curve* (AUC) as described in section 3.8. An AUC of 0.5 denotes a classification at random, while a value of 1 would describe the perfect classifier. This calculation can be done with every update to check the benefits of a longitudinal analysis. Further considerations on evaluation are discussed in chapter 4.

A visual support can be delivered through confidence intervals, which are not trivial for longitudinal data. Through conditional expectation it is possible to draw confidence intervals in longitudinal graphs, but one has to be careful with their interpretation (section 3.9).

3.2 One Class Classification

We will briefly point out the difference of a multi class and a one class classification as discussed in detail in [Pimentel et al., 2014]. Classification describes the problem of assigning data to pre-defined groups. Given a new object, a classifier has to decide to which group (or class) it belongs. This decision is based on the characteristics of the

occurring classes, where new data is assigned to the class with the most resemblance. The probabilistic approach to determine these characteristics is to assume an underlying probability distribution for each class and estimate these distributions with a *training set*, i.e. sampled data with known class labels. In a multi class classification, new data is then assigned to the class with the highest probability with respect to the class distributions. A problem occurs when (1) one class is sampled well but others are under-sampled or (2) there is evidence that a class consists of subclasses with discriminative distributions. A popular example of the first problem is the survey of a machine, where faults should be detected. While it is rather easy to observe the normal operating conditions, reasons for machine fault are diverse. To get representative data, one would have to destroy the machine in every possible way, an impossible or at least infeasible task (a scenario which can easily be adopted for a human body). The second problem we observed in chapter 2, figure 2.3, where we found idiosyncrasies in the non-survival patients. Estimates which rely on underrepresented or mixed training data will be inaccurate or biased.
In our dataset we oppose both of the mentioned problems, an under-sampling and idiosyncratic subclasses of the negative class. So we can't rely on estimates of the negative class based on our training data, which is why we approach the task of classification via a *one class classification* (OCC). In spite of better knowledge, all non-survivors are grouped into one negative class and every negative case is defined by being non-positive. Now instead of both characteristics, only the positive class' distribution is estimated. The classifier checks the deviation of this distribution for every new data and a threshold determines which samples are considered as outliers, i.e. negative cases.

3.3 Linear Mixed Effects Models

The estimation of class distributions can be achieved through a mixed effects regression model. For a good understanding of the linear *mixed effects model* (MEM), we will start with an easy example of a linear regression model and build the MEM out of it. The following definitions and notations are based on [Verbeke and Molenberghs, 2000], chapter 3, the incorporation of multiple response variables in a MEM follows the approach of [Morrell et al., 2012].

In ordinary linear regression, an entity is assumed to be dependent on one or more quantities. The entity is called *response variable*, or just response, and emerges trough some relation with the so called *explanatory variable(s)*. Another terminology for response and explanatory would be dependent and independent variable, as known from real functions. They both describe the dependence of one variable upon others. A simple example of a linear regression model would be

$$y = \beta_0 + \beta_1 x + \epsilon \tag{3.1}$$

where the *regression coefficients* β_0 and β_1 are real numbers and ϵ is a normally distributed random variable with mean 0 and variance σ^2, $\epsilon \sim N(0, \sigma^2)$. With $\hat{y} = \beta_0 + \beta_1 x$ we denote the estimated value for the data x. Thus the residual unfolds to $\epsilon = y - \hat{y}$.
Given a training data x_i and y_i, $i = 1 \dots N$, one can calculate the optimal values for β_0 and β_1 under certain conditions, e.g. such that the sum of squared residuals $\sum_{i=1}^{N} \epsilon_i^2$ with

$\epsilon_i = y_i - \hat{y}_i$ reaches a minimum.
In our situation y could for example be a patient's s100 level upon arrival and x the elapsed days since a trauma occurred. β_0, the intercept, could then be interpreted as a baseline for s100, β_1 would describe the increase or decrease of s100 level per day post trauma. Given a training data we could calculate the optimal β_i and simulate s100 levels with the model.

This linear regression can be expanded to a multiple regression in terms of multiple explanatory variables. Considering the shapes of the marker curves in figure 2.2 this seems necessary, since a model of the form (3.1) can only explain linear phenomena. So it will be inevitable to include non-linear expressions of time into the model. E.g., the model

$$y = \beta_0 + \beta_1 t + \beta_2 e^{-t} + \epsilon \tag{3.2}$$

defines a linear model that is able to account for a baseline, i.e. the intercept β_0, linear effects described by β_1 and an exponential decay determined by β_2, dependent on the time t. So the model (3.2) is far more flexible than model (3.1).

If $y = (y_1, y_2, \dots y_n)^T$ is a vector of observed response variables at times $t_1, t_2, \dots t_n$, model (3.2) can be written in matrix notation

$$y = X\beta + e \tag{3.3}$$

where $\beta = (\beta_0, \beta_1, \beta_2)^T$ is a 3 - dimensional vector of the 3 regression coefficients. X is a $n \times 3$ matrix containing the explanatory variables

$$X = \begin{bmatrix} 1 & t_1 & e^{-t_1} \\ 1 & t_2 & e^{-t_2} \\ \vdots & \vdots & \vdots \\ 1 & t_n & e^{-t_n} \end{bmatrix} \tag{3.4}$$

and e is an n - dimensional vector of residuals. Further on, we will consider the matrix form of the regression model and allow for an arbitrary number of explanatory variables.

Every data contains unobserved phenomena such as individual specific effects, measurement uncertainties, laboratory specific effects etc. To account for these unobservable effects we can extend the model (3.3) to a *mixed effects model*. We assume dependencies besides the ones explained by the explanatory variables in X, so we add another matrix of explanatory variables Z and multiply it with a random vector $b = (b_1, b_2, \dots, b_q)^T$. The univariate mixed model is then written as

$$y = X\beta + Zb + e \tag{3.5}$$

In this context the columns of X are called the *fixed effects* explanatory variables while the columns of Z are called *random effects* explanatory variables, hence the name mixed effects model. The regression parameters β are named *fixed effects parameters*, b are the *random effects parameters* or simply fixed / random effects.
Z is a $n \times q$ matrix which can be a submatrix of X, equal to X or contain additional

terms not included in X. The random effects are treated as a multivariate normal random variable $b \sim MVN(0, \tilde{D})$ with mean 0 and an unstructured covariance matrix $\tilde{D}$. They allow for varying effects for each individual. e is also multivariate normal with mean 0 and covariance matrix Σ, $e \sim MVN(0, \Sigma)$.

The mixed effects model (3.5) is univariate with respect to its response variable. In our context it only allows us to estimate a single biomarker per model. Since we want to incorporate multiple markers into the model, we have to transfer it into a multivariate setting.
Consider a patient i with n_i observations, so for each biomarker we have n_i measurements. Let $Y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]$ denote the response matrix for a patient i, where each column y_{ik} contains the measurements for the biomarker k, $k = 1, \ldots, m$ (when we consider m markers). Correspondingly let $E_i = [e_{i1}, e_{i2}, \ldots, e_{im}]$ be the error matrix defined in the same manner. In the following, y_i and e_i are the stacked $n_i \cdot m$ vectors of the columns of Y_i and E_i, respectively.
For every marker k there are two matrices $\tilde{X}_{ik} \in \mathbb{R}^{n_i \times p^*}$ and $\tilde{Z}_{ik} \in \mathbb{R}^{n_i \times q^*}$ with explanatory variables for fixed and random effects, given there are p^* fixed and q^* random effects for this marker. Let $X_i = diag(\tilde{X}_{i1}, \ldots, \tilde{X}_{im})$ and $Z_i = diag(\tilde{Z}_{i1}, \ldots, \tilde{Z}_{im})$ denote block diagonal matrices for each patient i. Then we can formulate a 'multivariate' mixed model as

$$y_i = X_i\beta + Z_i b_i + e_i \qquad i = 1, \ldots, N, \tag{3.6}$$

where N is the number of recorded patients. The real vector β contains the fixed effects for all markers, the random vector b contains the random effects correspondingly.
For a better understanding we will take a brief look at a small model in detail. Consider a patient i with n_i observations of the two biomarkers *s100* and *gfap* ($m = 2$). Then model (3.6) can be written as

$$\begin{bmatrix} s100_{it_1} \\ \vdots \\ s100_{it_{n_i}} \\ gfap_{it_1} \\ \vdots \\ gfap_{it_{n_i}} \end{bmatrix} = \begin{bmatrix} \tilde{X}_{i1} & 0 \\ & \\ 0 & \tilde{X}_{i2} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\frac{p}{2}} \\ \beta_{\frac{p}{2}+1} \\ \vdots \\ \beta_p \end{bmatrix} + \tag{3.7}$$

$$\begin{bmatrix} \tilde{Z}_{i1} & 0 \\ & \\ 0 & \tilde{Z}_{i2} \end{bmatrix} \begin{bmatrix} b_{i1} \\ \vdots \\ b_{i\frac{q}{2}} \\ b_{i\frac{q}{2}+1} \\ \vdots \\ b_{iq} \end{bmatrix} + \begin{bmatrix} e_{i1} \\ \vdots \\ e_{in_i} \\ e_{in_i+1} \\ \vdots \\ e_{in_i \cdot 2} \end{bmatrix} \tag{3.8}$$

when we assume the same number of fixed and random effects for *s100* and *gfap*. The parameters p and q denote the overall number of fixed/random effects.
This representation shows how the model works. Through the block structure of the model matrices X and Z, the corresponding β and b_i entries only act on the submatrices

responsible for the different markers. We still deal with a univariate model in the sense that the response is a vector as in (3.5), but with the block diagonal structure of the model matrices we incorporated the multivariate nature fo the data. The residual vector divides into residual subvectors for each marker.

The model matrices X and Z contain the explanatory variables. A submatrix $\tilde{X}_{ik}$ consists of n_i rows corresponding to the n_i measurements of marker k. The columns declare the explanatory variables. These can be continuous values such as days post trauma *dpt* and functional expressions of *dpt*. An intercept as in model (3.1) is achieved by adding a column of ones to $\tilde{X}_{ik}$ ((3.4) is an example for such a submatrix). Any other explanatory variables could be added as well, such es sex, age, injury scales etc. Metric or ordinal data can be implemented directly. Group memberships are realized through dummy variables, where memberships are coded with zero or one. E.g. if we would have information about the patients' gender, every submatrix of X_i would include two columns representing 'male' and 'female'. For male patients the 'male' column would consist of ones and the 'female' column of zeros, for women vice versa. The same holds for the matrices $\tilde{Z}_{ik}$.

The fixed effects β are the same for every patient and account for effects which are shared by the whole population. As we will see they describe the population mean of the response. In contrary, the random effects b_i are patient specific and carry unobservable phenomena. That is why we have to treat them mathematically different, which is why they are defined as a random vector. In summary we deal with the random vectors

$$b_i \sim MVN(0, D) \qquad \text{and} \qquad e_i \sim MVN(0, \Sigma_i). \tag{3.9}$$

D is an unstructured covariance matrix that allows for correlations between the random effects. For simplicity we assume the residual covariance $\Sigma_i = \sigma^2 I_{n_i \cdot m}$ to be a diagonal matrix with a single parameter. This implies that Σ_i depends on the patient i just by its dimension. Under this assumption, errors are uncorrelated for every observation.

If we assume a mixed model (3.6), we infer a distribution of the response variable y_i. From the linearity of the expected value and $E(X_i\beta) = X_i\beta$, $E(Z_i b_i) = Z_i E(b_i) = 0$ and $E(e_i) = 0$ it follows that

$$\mu_i := E(y_i) = E(X_i\beta + Z_i b_i + e_i) = X_i\beta. \tag{3.10}$$

Similarly, for the variance of y_i it holds that

$$V_i := var(y_i) = var(X_i\beta + Z_i b_i + e_i) = Z_i^T D Z_i + \Sigma_i \tag{3.11}$$

since $var(X_i\beta) = 0$, $var(Z_i b_i) = Z_i^T D Z_i$ and $var(E_i) = \Sigma_i$. Thus, y_i can be interpreted as a multivariate normal vector

$$y_i \sim MVN(X_i\beta, Z_i^T D Z_i + \Sigma) =: MVN(\mu_i, V_i). \tag{3.12}$$

This is the so called marginal distribution of the model. Note that some parts contain an index i to emphasise the dependence on the patient's data. X_i and Z_i are the matrices of explanatory variables, i.e. observed objects such as days post trauma and group memberships. D is the covariance matrix of the random effects and allows for correlations

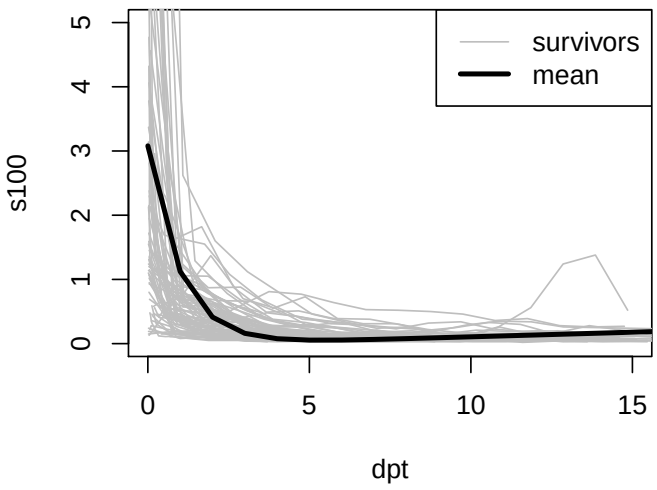


Figure 3.1: *s100* trajectories of survivors (grey) and estimated population mean (black).

between the markers and points in time. With (3.12) it is possible to calculate the expected survival curve and its variance for every patient at any time, given the assumption that he/she belongs to the positive class. If (3.6) is fitted with a training data, we can estimate the expected survival curves through the population mean (3.10), hence $X_i\beta$. An example is given in figure 3.1. It shows the *s100* trajectories of the survival patients in grey, the black thick line depicts the estimated mean survival curve gained form a model of the form (3.6) with model submatrices as in (3.4). Here one can obtain how the fixed effects β contribute to the population mean.

Since, as explained in the prior chapter, we deal with an under represented negative class, it is not feasible to estimate the negative class's marginal distribution. That is why our training data will only consist of the positive cases. What we gain is an estimation of biomarker curves under the assumption that the patient belongs to the positive class, thus is a survivor. We can then compare measured observations with expected biomarker levels on the basis of the assumed distribution (3.12). We need to define a degree of deviation of expectation and set a critical value for this degree as explained in section 3.5. Based on this threshold we classify a patient as survivor or non-survivor.

3.4 Parameter Estimation

The estimation of the marginal model (3.12) is based on fitting observed data to a linear mixed model of the form (3.6). To do this, we use the R - routine `lmer` contained in the `lme4` package. The (very technical) documentation of this routine can be obtained in [Bates et al., 2014], this section will focus on the essential ideas as discussed in [Verbeke and Molenberghs, 2000], chapter 5.

Recall that the marginal mean and covariance of y_i are $\mu_i = X_i\beta$ and $V_i = Z_i^T D Z_i + \Sigma_i$. Let α denote the covariance parameters and $\theta = (\beta, \alpha)$ the vector of all parameters to be estimated. Estimation of θ is based on the maximum likelihood (ML) approach

$$L_{ML}(\theta) = \prod_{i=1}^{N} \left[(2\pi)^{-\frac{n_i}{2}} |V_i(\alpha)|^{-\frac{1}{2}} \times exp\left(-\frac{1}{2}(y_i - X_i\beta)^T V_i^{-1}(\alpha)(y_i - X_i\beta) \right) \right] \quad (3.13)$$

Observed data is plugged into this function, which is then maximized with respect to the parameter vector θ to gain the maximum likelihood estimator, denoted by $\hat{\theta}$.
Note the dependency of V_i on α. If α is assumed to be known, the maximum likelihood estimator $\hat{\beta}$ of β conditional on α is gained by maximizing (3.13) and unfolds to

$$\hat{\beta}(\alpha) = \left(\sum_{i=1}^{N} X_i^T V_i^{-1}(\alpha) X_i \right)^{-1} \sum_{i=1}^{N} X_i^T V_i^{-1}(\alpha) y_{obs} \quad (3.14)$$

Plugging (3.14) into (3.13) and maximizing with respect to θ yields the ML estimate $\hat{\theta}$ of the parameter vector.

As Verbeke and Molenberghs (2000) point out, an estimation of the covariance based on an estimation of the mean (in this case the estimation of β) is biased. This bias can be avoided as follows: Let

$$y = X\beta + Zb + \epsilon \quad (3.15)$$

be the model gained when stacking all vectors y_i, b_i, ϵ_i as well as the matrices X_i. Z then is the block diagonal matrix of all Z_i. The marginal distribution of y can be obtained as in (3.12), with mean $X\beta$ and a covariance $V(\alpha)$. Now let A be an arbitrary full-rank $(n \times (n-p))$ matrix (p is the total number of fixed effects) where the columns of A are orthogonal to the columns of X. Then the transformation $u = A^T y$ follows a normal distribution with mean zero and covariance $A^T V(\alpha) A$. By the transformation, the impact of β on the estimation of $V(\alpha)$ is eliminated. It can be shown, that the likelihood function of u is equal to

$$L(\alpha) = C \left| \sum_{i=1}^{N} X_i^T V_i^{-1}(\alpha) X_i \right|^{-\frac{1}{2}} L_{ML}(\hat{\beta}(\alpha), \alpha) \quad (3.16)$$

where L_{ML} on the right hand side corresponds to the likelihood function (3.13). The constant C does not depend on α, further note how the first term in (3.16) is independent of β. These results are used to define the *restricted maximum likelihood* (REML) function

$$L_{REML}(\theta) := \left| \sum_{i=1}^{N} X_i^T V_i^{-1}(\alpha) X_i \right|^{-\frac{1}{2}} L_{ML}(\theta) \quad (3.17)$$

which is maximized to estimate the parameter vector $\theta = (\beta, \alpha)$. The parameter vector $\hat{\theta}_{REML}$, which maximizes this function, is called the *restricted maximum likelihood estimator*.

3.5 Mahalanobis Distance

The key idea of section 3.3 was to find a way to estimate the mean survival curve. Now we need a method to compare measured data with that estimation, which we will accomplish with the *Mahalanobis distance* (MD). The Mahalanobis distance is a weighted distance which can be used for an assessment, if measured data occurs under a given normal distribution. It is defined through

$$MD^2(y) = (y-\mu)^T \Sigma^{-1} (y-\mu) \tag{3.18}$$

with given mean μ and covariance Σ.

To show how this distance works it is useful to consider a bivariate normal distributed example. Let the two-dimensional random vector y be distributed as

$$y \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \right) \tag{3.19}$$

In figure 3.2, this distribution is illustrated by drawing contour lines of its density in the $y_1 - y_2$ plane. The mean $(0,0)^T$ obviously has a MD of zero. Moving away from the center, the MD ascends, where every ellipse represents a set of points with equal Mahalanobis distances. Although the point A is closer to the center than B in terms of Euclidean distance, its MD is larger, which we know since B lies within and A outside the 95% tolerance region. This definition of distance takes the elliptical shape of the distribution into account. When the covariance matrix is the identity I_2, the Mahalanobis distance degrades to the common Euclidean distance.
An important feature of the Mahalanobis distance is its distribution. When an n - dimensional random vector is normally distributed, then MD^2 follows a χ^2_n distribution with n degrees of freedom,

$$y \sim MVN(\mu, \Sigma) \Rightarrow MD^2(y) \sim \chi^2_n \tag{3.20}$$

(a proof of this result can be obtained in [Bilodeau and Brenner, 1999], chapter 4). Examining a sample under the assumption of an n-dimensional normal distribution, we can thus define a quantile q of χ^2_n as cutoff and consider samples with a squared MD greater than q as outliers. Going back to our example, the left side of figure 3.3 shows a random sample under (3.19) with a 90% tolerance ellipse. Points with $MD^2 > \chi^2_{2;0.9}$,where $\chi^2_{2;0.9}$ denotes the 90% percentile of the χ^2_2 distribution, are depicted by '$\circ$' and defined as outliers. The right hand side shows the samples' squared Mahalanobis distances compared with the χ^2_2 distribution. As we can observe, samples with $MD^2 > q_{0.9}$ coincide with samples outside of the ellipse.

Recalling the aim of this section, given an expected survival curve, deviations of measured biomarker data should be assessed and penalized, depending on their intensities. The Mahalanobis distance delivers exactly that. Assuming a normal distribution of the biomarker curves - with known (estimated) mean and variance - , measured data that lies far away from expectation corresponds to a high MD^2. But we oppose some difficulties. An MD is a distance measure in an n-dimensional space. In our case this can be

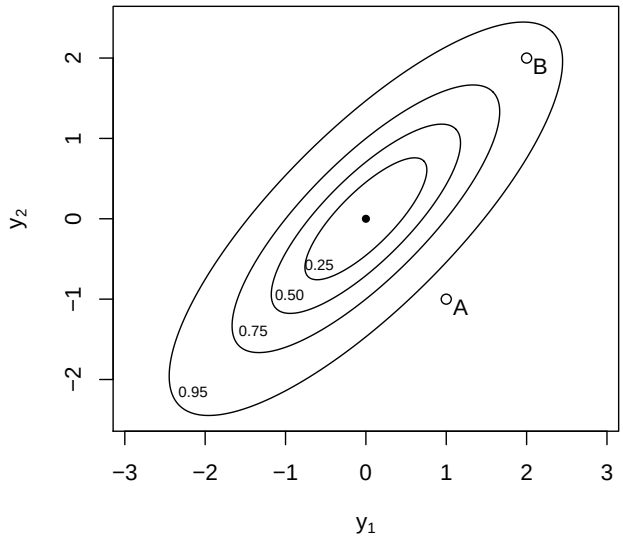


Figure 3.2: Contour lines of density of (3.19) of a two-dimensional normal distribution; points A and B with different Mahalanobis distances where $MD(A) > MD(B)$.

interpreted as follows: given n measurements, the MD declares a 'skew' distance in the n-dimensional space of all possible biomarker concentrations, where higher distances correspond to lower probabilities that the observations occur under the assumption (3.12). But there is a problem. Since we deal with an unbalanced dataset with respect to observation times per patient, we are confronted with different lengths of marker vectors reaching from 3 to 22. Additionally, we want to update a patient's data with every 'new' observation, meaning for n observations we deal with vectors of lengths $1, 2, \ldots, n$. Further the length of the marker vectors multiply with the number of markers used for the model (consider the left hand side of (3.7)). Up to six marker vector with lengths form 1 to 22 leaves us with a span of 1 to $22 \times 6 = 132$ different dimensions. Note that the varying dimensions - which origin from varying observation times - are reflected in the distribution (3.20) in the degrees of freedom. What we need is a reasonable way to compare Mahalanobis distances that originate from spaces of different dimensions.
Consider a patient i with observed marker levels y_i for a marker k at the times $t_1, t_2, \ldots, t_{n_i}$. Let $X(t_1, t_2, \ldots, t_{n_i})$ be the model matrix of (3.6) with the patient's data at the time points plugged in. Further let β be the fixed effects regression parameters and V_i the covariance matrix as in (3.11). To classify this patient we calculate the expected marker level through $\mu_i = X(t_1, t_2, \ldots, t_{n_i})\beta$. We then determine the consecutive squared Mahalanobis distances

$$MD^2_{t_1} := (y_{t_1} - \mu_1)^T V_{11}^{-1} (y_{t_1} - \mu_1) \tag{3.21}$$

$$MD^2_{t_2} := \left(\begin{bmatrix} y_{t_1} \\ y_{t_2} \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \right)^T \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} y_{t_1} \\ y_{t_2} \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \right) \tag{3.22}$$

$$\vdots$$

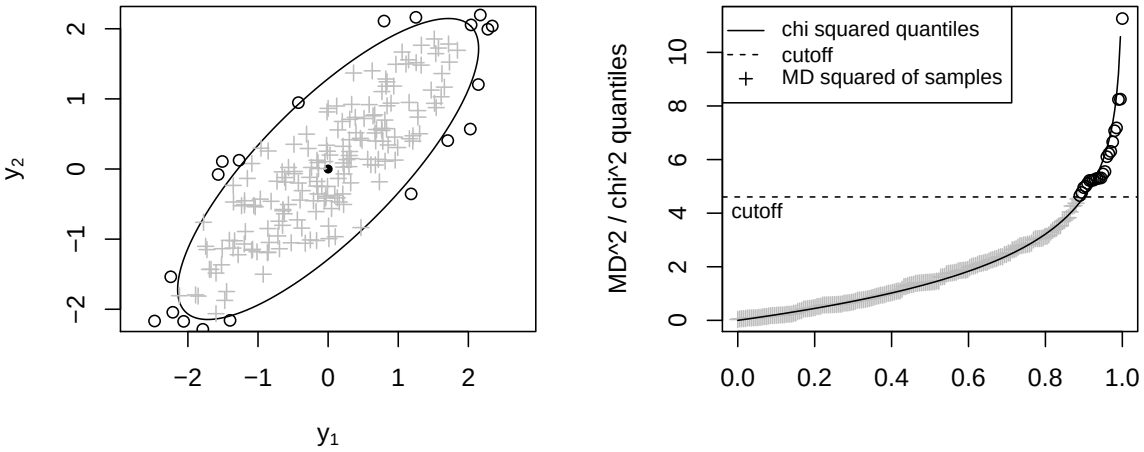


Figure 3.3: Left: random samples with 0.9 confidence ellipse; right: MD^2 of samples compared to quantiles of χ^2_2; samples with $MD^2 > \chi^2_{2;0.9}$ ('$\circ$') are considered outliers.

and so on. The theoretical distributions of the squared Mahalanobis distances are $MD^2_{t_n} \sim \chi^2_n$, where n denotes the length of the observation vector. Let f_n be the probability density function of the χ^2 distribution with n degrees of freedom. Then for every observation vector $y_{t_1}, y_{t_2}, \ldots, y_{t_n}$ the probabilities of the Mahalanobis distances $MD^2_{t_1}$, $MD^2_{t_2}, \ldots, MD^2_{t_n}$ can be calculated through $f_1(MD^2_{t_1}), f_2(MD^2_{t_2}), \ldots, f_n(MD^2_{t_n})$, in short

$$y_{t_n} \Rightarrow MD^2_{t_n} \Rightarrow f_n(MD^2_{t_n}) \tag{3.23}$$

These probabilities can be compared regardless of the dimension of their corresponding biomarker vectors. Mahalanobis distances with a probability below some cutoff are assumed to origin from outliers. Theses outliers are considered as data that doesn't belong to the underlying distribution but rather to an unknown distribution denoting the negative class.

Before we move on, we want to give a brief summary of the process so far. Based on a training set of positive cases - a sample of survival curves - a distribution (3.12) for expected biomarker levels is estimated. Individual specific and cohort effects can be incorporated in this distribution via the model matrices of the mixed model (3.6). Observations of new patients are compared to the estimated population mean by computing their MD^2 and corresponding probabilities under χ^2. They are assumed to behave similar to the training data, given the patient belongs to the positive class. High MDs correspond to low probabilities that the patient's data occurs under these assumptions, hence the patient is to be classified as a negative case.

3.6 Cross Validation

Since we have limited amount of data, we would like to use a maximum of patients to contribute to the training data. We want to estimate the parameters of the positive class only, so a nearby choice are all positive cases - the survivors - as training data. To classify the negative cases this is the legitimate approach. To classify survivors anyhow we have to be careful. Using a patient to estimate a population mean and then checking the patient's deviation from this mean gives the classifier an unfair advantage which would bias the results. To avoid this overfitting we need to *cross validate* our results. In a *k-fold cross validation* the data is divided into k folds. The model is then fitted with $k-1$ folds as training data and the remaining fold - the so called test data - is classified based on this fitting. This procedure avoids overfitting, so the results are more reliable. A typical choice for k is 10. Detailed considerations of cross validation can be obtained in [Hastie et al., 2009].

In summary, for every classifier there are $k+1$ mixed model fits. One for every fold to classify the survivors and one to classify the negative cases. The choice of $k = 10$ is common, however other values are possible. The approach where a maximum of data is used is a $leave-one-out$ cross validation where there are as many folds as individuals. For 75 survivors this would mean there are 76 model fittings to be calculated. Since computation costs can be quite intense this is no feasible solution during model selection. This favours a k-fold cross validation.

3.7 Imputation

Some observations of the data are missing at random, for a quarter of patients there are no *nse* observations. The few randomly missing samples are less severe, but if the absent *nse* levels are not dealt with, all models which incorporate this marker lack of a big part of training data. If these patients are neglected, also all their other marker curves are neglected, thus a lot of information is thrown away. The mixed model can deal with missing data with the use of the estimated distribution (3.12). If a model is fitted using all available data - survivors as well as non-survivors - as training set, absent marker levels can be predicted by $X\beta$, were X contains the available information (*dpt*, injury scales or other patient specific data).

The missing values of the dataset at hand were imputed in such a manner. The explanatory matrix of the imputation model is the same as for model 5 described in chapter 4, section 4.1.

3.8 ROC and AUC

A summary of a classifier's performance is given with the *receiver operator characteristic* (ROC) curve and its *area under curve* (AUC). To explain these in detail we need the terms *sensitivity* and *specificity* for a predictor.

Consider a set of patients to be classified as either survivors or non-survivors. Let TP (*true positives*) denote the true positive cases, meaning patients who survived and are classified as such. TN (*true negatives*) are the true predicted negative cases, i.e. non-survivors with a negative prediction. FP (*false positives*) are non-survivors predicted as survivors and FN (*false negatives*) vice versa. Then the sensitivity of a predictor - also called the true positive rate - is defined as $\frac{TP}{TP+FN}$. It can be interpreted as the probability that a positive case is recognized as such. The specificity - the true negative rate - calculates as $\frac{TN}{TN+FP}$ and is the percentage of true predicted non-survivors. A perfect test has 100% sensitivity and 100% specificity, in practice anyhow an increase of one will lead to decrease of the other.

Sensitivity and specificity of a test depend on the classifier's cutoff. If we set a high threshold for a positive prediction we enhance the chances to correctly predict negative cases, but we risk to assign a high number of positives to the negative class. Thus we get high specificity and low sensitivity. If we lower the cutoff, chances to catch survivors rise, while more non-survivors will be classified as positives as well, leading to lower specificity and higher sensitivity. Typically there is a trade-off between these two parameters which can be visualised as follows. Let the false positive rate (fpr) be 1 minus false negative rate, so 1− specificity. Since we classify in terms of probabilities of memberships to the positive class, every number between 0 and 1 represents a cutoff. For every cutoff the sensitivity and the fpr = 1−specificity can be calculated. A cutoff of 1 (all patients are classified as negatives) leads to sensitivity 0 and specificity 1, thus fpr is 0. A cutoff of 0 (only positive classifications) implies sensitivity 1 and specificity 0, fpr then is 1. If we plot sensitivity against the false positive rates, the resulting graph is a path leading form $\underline{0} = (0, 0)$ to $\underline{1} = (1, 1)$ with every point corresponding to a cutoff. The rise of fpr (thus the descend of specificity) causes a rise of sensitivity.
Such a plot is called the *receiver operator characteristic* (ROC) and can be used to asses the quality of classifiers. A straight line from $\underline{0}$ to $\underline{1}$ determines a classification at random, the closer the path is to the upper left corner of the unit square, the better is the classifier's performance.
To compare different classifiers we generate their ROCs and calculate the areas under the curves, the AUCs. Since a straight line from the lower left to the upper right corner denotes a random prediction, the AUC will be 0.5 while the perfect classifier has an AUC of 1. A classifier with an AUC close to 0.5 is considered to be near to random, while higher AUCs correspond to classifiers of higher prediction success. Figure 3.4 shows an example of a ROC. The AUC, the area under the black curve, is 0.902. The grey, straight line denotes a random classifier.

3.9 Conditional Confidence Interval

Our objective is to support physicians in their decisions in clinical practice. A classification as explained above can monitor a patient's health status and induce necessary interventions. But besides a strict classification, the model also enables us to deliver a visual monitoring for a patient's condition. The knowledge of mean survival curves and

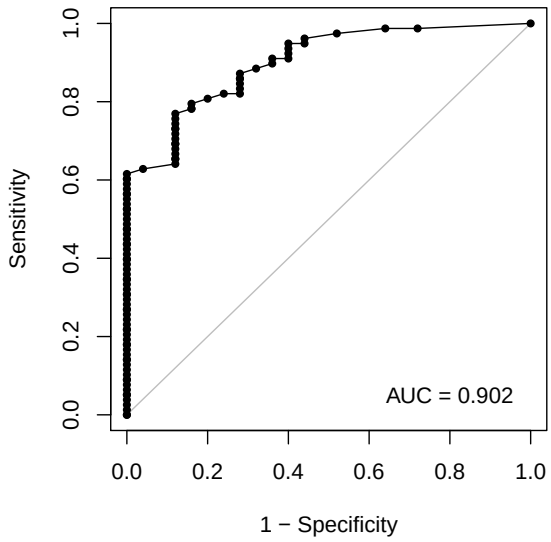


Figure 3.4: ROC curve of a classifier with an AUC of 0.902; the grey line denotes a classification at random, ever mark of the ROC corresponds to a cutoff.

variances for the biomarkers allows for a graphical display of desired marker concentrations and anomalies. This comes down to drawing confidence regions for the biomarker levels. Confidence regions for longitudinal data are somewhat troublesome. Note that the marker vectors correspond to multivariate normal random vectors of different dimensions. Also the updating nature of the problem - the constant patient monitoring - calls for a special treatment of confidence bounds.

A possibility to depict confidence regions is what we call a conditional confidence interval (CI). That is a CI that takes the updating nature of the monitoring into account, meaning that we aim for a clinical monitoring system where the data is updated periodically. The conditional CI uses the prior observations to deliver a prognosis of the marker levels for the upcoming time period.

Consider the estimated distribution of a biomarker vector y_i (which can include multiple biomarkers)

$$y_i \sim MVN(X_i\beta, Z_i^T D Z_i + \Sigma) = MVN(\mu_i, V_i). \tag{3.24}$$

We remember that the matrices X_i and Z_i are block diagonal matrices with submatrices $\tilde{X}_{ik}$ and $\tilde{Z}_{ik}$, $k = 1, \ldots, m$, where m denotes the number of biomarkers used in the model. Every row of $\tilde{X}_{ik}$ corresponds to an observation time t_j, $j = 1, \ldots, n_i$, the columns denote the fixed effects explanatory variables. For every subset $j_1, j_2, \ldots, j_h$ of $1, 2, \ldots, n_i$, let $X(t_{j_1}, t_{j_2}, \ldots, t_{j_h})$ be the rows of the matrix X_i corresponding to the observation times $t_{j_1}, t_{j_2}, \ldots, t_{j_h}$. The same notation applies to Z_i.

In general, if the normally distributed random vector y is partitioned as $y = (y_1, y_2)^T$ and

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \sim MVN(\mu, V) \qquad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \tag{3.25}$$

then the distribution of y_1 conditional on y_2 is also a multivariate normal distribution

$$y_1|y_2 \sim MVN(\mu_1 + V_{12}V_{22}^{-1}(y_2 - \mu_2), V_{11} - V_{12}V_{22}^{-1}V_{21}) \tag{3.26}$$

([Bilodeau and Brenner, 1999], chapter 5). Assume we have n measurements of different biomarkers stacked in the vector $y = y(t_1, \ldots, t_n)$. We know the distribution of the survival curve (3.24), not only for n dimensions but for an arbitrary number of observations. Thus we can calculate the mean and variance

$$\mu = X_i(t_{n+1}, t_1, \ldots, t_n)\beta, \qquad V_i = Z_i(t_{n+1}, t_1, \ldots, t_n)^T D Z_i(t_{n+1}, t_1, \ldots, t_n) + \Sigma \tag{3.27}$$

for the marker levels until t_{n+1}. Through (3.26) we know the conditional distribution $y(t_{n+1})|y(t_1, \ldots, t_n)$ of marker levels at time t_{n+1} given the observations $y(t_1, \ldots, t_n)$.

$$\underbrace{y(t_{n+1})}_{yet\ unobserved} \quad |(\underbrace{y(t_1, \ldots, t_n)}_{observed}) \sim MVN(\mu_{cond}, V_{cond}) \tag{3.28}$$

The conditional mean μ_{cond} of this distribution represents the expected marker levels at t_{n+1} based on the prior n observations, the conditional covariance matrix V_{cond} contains their variances as diagonal elements.

Based on these results it is possible to declare conditional confidence regions for the $(n+1)$th observation of each marker. Considering the first n observations of all markers, a 95% conditional confidence region for a specific marker at t_{n+1} is given as

$$(\mu_{cond} - 1.96\sigma_{cond}, \mu_{cond} - 1.96\sigma_{cond}) \tag{3.29}$$

where σ_{cond} is the square root of V_{cond} (note that V_{cond} is not a matrix, but a scalar; for each of the markers an individual conditional mean and variance (3.28) is calculated, so these are univariate distributions). Such a conditional confidence interval can be obtained in figure 3.5. The grey error bars represent the conditional mean and the confidence bounds (3.29). Again, these are confidence regions in one dimension, one for each marker, based on all the prior observations.

It has to be mentioned that one has to be careful with the interpretation of these confidence bounds because of two reasons. First, since we assume a normal distribution, the CIs are symmetric around the mean, which leads to negative lower confidence bounds of marker levels. In figure 3.5, the lower bound is set to zero if it is indeed negative. Second, these confidence regions determine the area where we would expect the curves to be, given the prior observations and the assumption that the patient belongs to the positive class. The left graph shows a desirable behaviour. Single measurements are outside of the confidence region, the curves then descend to 'normality'. The right hand side shows odd confidence bounds for the *gfap* trajectory. High observations at the beginning lead to high conditional expectation. At the second measurement, the observed curve is below the lower confidence bound, but that of course is not a bad sign, since it converges to the desired survival mean. The upper confidence bounds are of more interest and relevance than the lower ones.

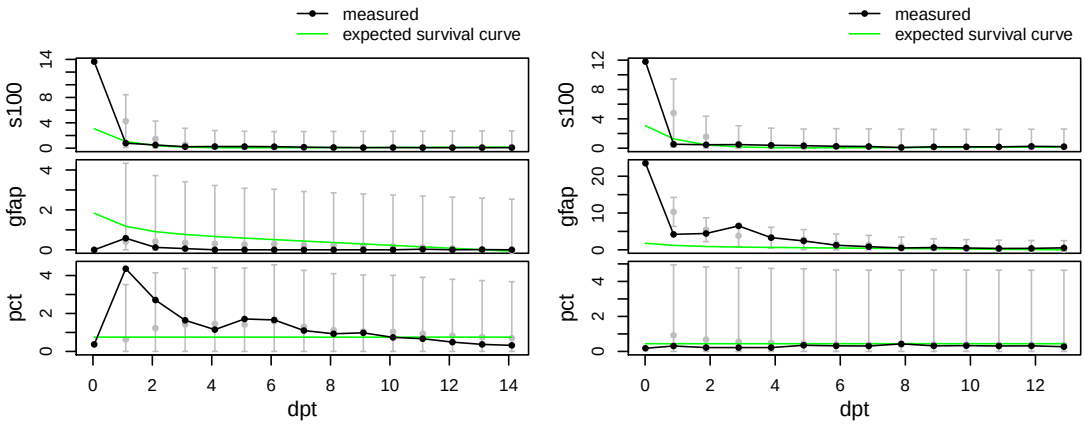


Figure 3.5: Patient trajectories (black), expected survival curve (green), conditional confidence intervals (grey); error bars represent conditional means and 95% confidence bounds.

4 Model Selection

So far we introduced the process of longitudinal classification with updating data. In the following, we will show how specific models are chosen and compared. A variety of possible classifiers based on the upper approach emerge out of following considerations:

- **best subset of features**
  The biomarker trajectories endorse the hope to distinguish potential non-survivors form survival patients. Although some markers are more promising than others, it is unclear which combinations of features lead to the best results.

- **explanatory variables**
  The purpose of the mixed model is to find a functional explanation of the survival curves for each biomarker. The power of the model to explain the observed curves depends on the explanatory variables of the model matrices. These will be mainly functional expressions of time, depending on which trends can be observed in the trajectories.

- **the classifier's memory**
  The classifiers are designed to look 'into the past', meaning to exploit prior observations as well as current. But should the classifier be able to 'forget' after a while? As we will see, classification quality improves when the classifier only considers a few of the old observations instead of all available data.

These are model parameters which have different impacts on the classification results. Further, when evaluating and comparing different models, we have to consider different aspects:

- **evaluation**
  One way to assess the performance of a classifier is to use the ROC/AUC as explained earlier. The longitudinal nature of the biomarker levels in clinical monitoring leads to updated classifications with every new measurement, thus it is important to consider the development of AUCs with progressing observation time. Moreover, this can only be considered as one aspect of classification quality. Some important information can't be read out of these figures. How are the misclassification rates? Are there more false negative or false positive predicted cases? We deal with asymmetric misclassification costs, meaning false positives should be considered worse than false negatives.
  How early does the classifier react? Predictions on the last day of clinical treatment are of less use than earlier assessments.
  How helpful are the classifications in forecasting health conditions? Can they serve as a reliable warning system?

- **cutoff**
  AUCs are calculated without the choice of a specific cutoff. Other aspects need to

define a threshold which influences the upper criteria. The choice of a cutoff should incorporate the asymmetric misclassification costs.

Considerations on feature selection are discussed in the upcoming section, different angles on evaluation are shown in section 4.2.

4.1 Feature Selection

One of the main 'screws to adjust' a classifier are the model matrices of the mixed model. They are block-diagonal-matrices (recall section 3.3, especially example (3.7)), where the number of submatrices corresponds to the number of markers used. Every submatrix contains the explanatory variables of the respective marker. In the following we will use the terms 'classifier' and 'model' interchangeably.
The selection of the best models starts with the determination of the explanatory variables. For each marker, these should capture the behaviour of the trajectories of figures 2.2 to 2.4. A possible approach would be to try various functional expressions of the days post trauma (dpt), such as a linear, a quadratic, a logarithmic etc. and determine the best models. If this is done, the number of models to be calculated increases very fast. Computation times grow exponentially with a growing number of explanatory variables, especially for the mixed effects model fitting, computation can be quite intense. This is why we have to try to keep the number of explanatory variables to a minimum. Trails showed that the impact of complex submatrices is negligible, thus the selection of explanatory variables was done by examining the marker trajectories and picking reasonable components:

- *s100* and *gfap* both show a behaviour similar to an exponential decay, their explanatory variables will be set to an intercept, a linear time trend and an exponential decay (e^{-t})
- For *pct* and *il*, no characteristic time trend can be observed, but a separation of TBI groups could be advantageous; hence the model submatrices of these markers will contain only intercepts, one for each TBI group.
- The explanatory variables of *crea* are an intercept and a linear time trend.
- *nse* will be modelled by an intercept and an exponential decay (it showed that a linear time trend doesn't enhance prediction results).

The submatrices for the markers will thus be:

$$s100, gfap: \begin{bmatrix} 1 & t_1 & e^{-t_1} \\ 1 & t_2 & e^{-t_2} \\ \vdots & \vdots & \vdots \\ 1 & t_n & e^{-t_n} \end{bmatrix} \quad pct, il: \begin{bmatrix} 1_{i\in GR1} & 1_{i\in GR2} & 1_{i\in GR3} \\ 1_{i\in GR1} & 1_{i\in GR2} & 1_{i\in GR3} \\ \vdots & \vdots & \vdots \\ 1_{i\in GR1} & 1_{i\in GR2} & 1_{i\in GR3} \end{bmatrix} \quad nse: \begin{bmatrix} 1 & e^{-t_1} \\ 1 & e^{-t_2} \\ \vdots & \vdots \\ 1 & e^{-t_n} \end{bmatrix} \tag{4.1}$$

$1_{i\in GRj}$ denotes a one if the patient i belongs to TBI group GRj, $j \in 1, 2, 3$ and zero otherwise. The fixed effects of these explanatory variables represent baselines of the markers *pct* and *il* for each TBI group. Depending on the markers used for the classifier, the

upper matrices are combined to a block-diagonal-matrix forming the model matrix X of the fixed effects. The random effects matrix Z will be identical to X, although other choices are possible.

High first measurements influence a classification of a patient, even if the marker levels descend to a desired level very early. It should be considered to clear a classifier's 'memory' after a while. By that we mean, the classifier should only take a certain time span into account and forget prior observations to that span. This 'short term memory' is realized through the Mahalanobis distance. If all available data is used, then the consecutive MD^2 are calculated by adding every updated measurement to the already observed biomarker vector. By 'cutting' the vector to the last e.g. 5 observations, the Mahalanobis distance neglects all values prior to the newest 5. In the following, the parameter *span* will denote the length of a classifier's short term memory. Since the longest ICU treatment in the data is 22 days, this span can take on values between 1 and 22, where $span = 22$ corresponds to the case where all available data is used.

For each classifier, a subset of the six available markers has to be chosen. The total number of combinations is 63. Further, for every choice there are 22 different classifiers corresponding to 22 different time spans. Thus the total amount of models to be compared and evaluated would sum up to $63 \times 22 = 1386$. To compare all these model is infeasible, especially since evaluation consists of various factors. WE reduced the number of considered models by (1) choosing only reasonable and promising feature subsets (e.g. a model which incorporates only *crea* and *il* will be of limited power) and (2) limiting the spans to a reasonable length (for *span* greater than ~ 15 the results mostly don't show significant differences). The best models - under different point of views - will be selected and discussed in detail in chapter 5.

4.2 Evaluation Aspects

The ROC and its AUC as discussed in section 3.8 deliver a convenient way to compare different classifiers. The ROC'/AUC is a good way to narrow down the number of possible models, for a detailed comparison however, more considerations - as mentioned briefly above - should be taken into account. To introduce different aspects of evaluation, we consider the model with all available features and varying spans to point out crucial observations.

The top of figure 4.1 depicts ROC curves with AUCs of two models. These refer to the last predictions of each patient. For both models, all features are used, but the spans differ. Top left depicts the model with $span = 2$ while top right shows results with $span = 5$. We obtain that AUCs of the left classifier are larger, hence one could conclude, that the smaller span is superior to the larger one. But since these ROCs refer to predictions on the last day of ICU treatment, they assess the classification quality at a time where it is too late for physicians to react. Due to the longitudinal nature of the data, we have to consider the progression of this assessment over time. In a clinical

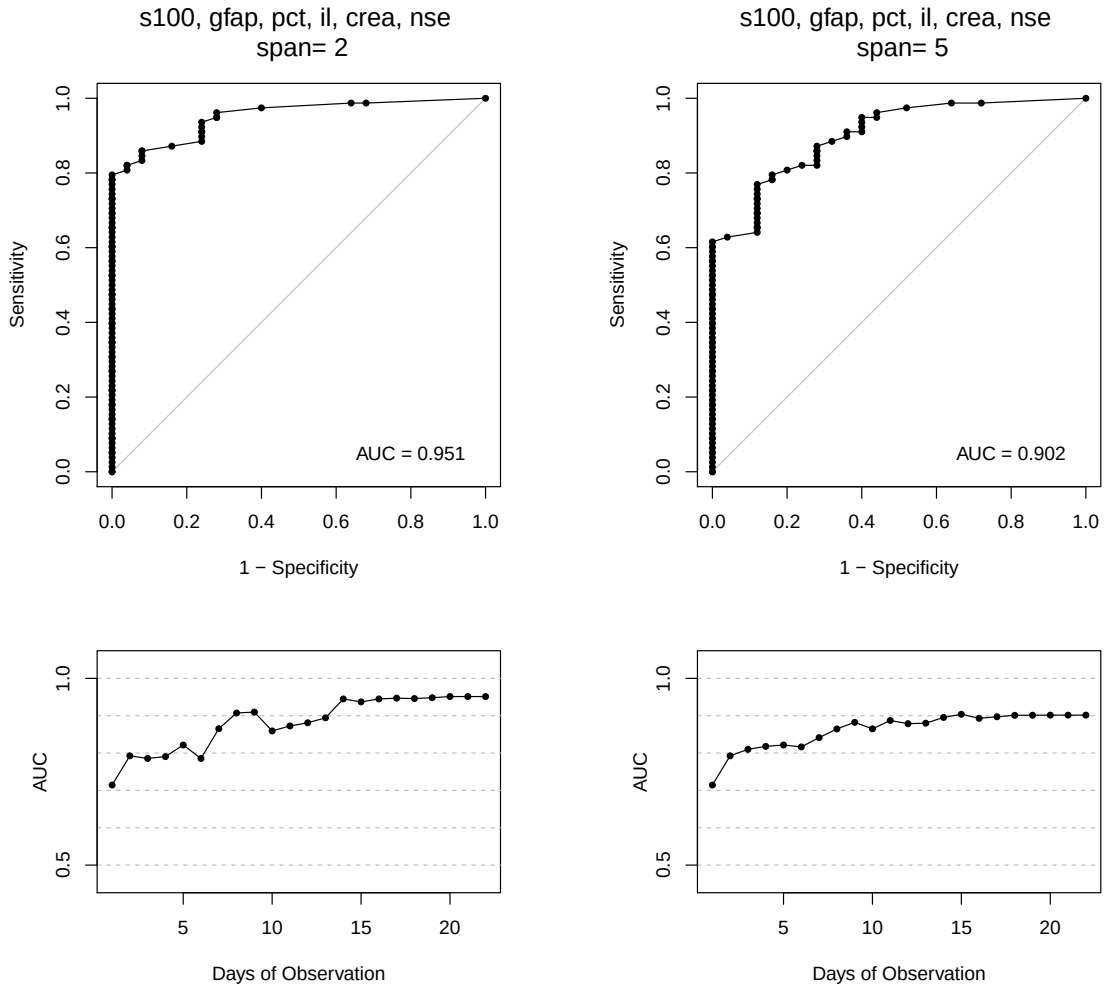


Figure 4.1: ROCs of two models (upper) and corresponding development of AUC over time (lower)

monitoring setting, every patient is classified with every new observation, which yields a progression of classifications over time, hence also a progression of AUCs. This progression can be obtained in the bottom of figure 4.1. We obtain that while the curves corresponding to the smaller span reach higher peaks, the other AUCs appear more stable.

4.2.1 Progression of AUC

Such progressing AUC curves demands further explanation. For a fixed day post trauma d_0, a way to depict classification performance on that day is to calculate the AUC using all predictions of patients with an observation time greater or equal to d_0. The question is, how to deal with patients with less observations? If these are ignored, results at later sample times will be rather meaningless, since AUCs that rely on a small number of samples do not bear useful insights. That is why we want to incorporate all patients to the AUC evaluation. This is achieved by setting predictions at higher times than there

are samples to the last available prediction. E.g. for a patient with a treatment length of 10 days, predictions after the 10th day are equal to the one on day 10.
While this enables to calculate AUCs incorporating all patients at any time, one has to be careful in interpretation. Late AUCs (with respect to days post trauma) correspond not only to late classifications but to prior classifications of patients with shorter treatment lengths. Good AUCs in a late point in time thus not necessarily mean good classification in late observation phase but a good mixture in early and late prediction.
Considering figure 2.1, we obtain that only 9 patients were observed longer than 16 days, 7 of those 9 survived. The last non-survivor deceased on day 18, after that the data consists only of three survivors. For completeness, we depict AUC curves over 22 days post trauma, but the focus should be on the results of the first $\sim$ 16 days. After that, the sparsity of data doesn't allow for meaningful interpretation.

4.2.2 Progression of Confusion

AUC evaluation bears to disadvantage of not being rather inaccessible. It is hard to tell how actual classifications differ, when the AUCs of two classifiers differ about a few decimals. That is why we look for a more descriptive way to depict model differences.
A typical way to summarize a classifier's results is a confusion table, where the number of true positives/negatives and false positives/negatives are summarized in a matrix:

TP	FN
FP	TN

This is a summary for one point in time only. For a longitudinal data classification, a time dependent pendant of the confusion table is needed. But before that, another factor has to be mentioned. Evaluation with the ROC curves doesn't rely on a specific cutoff, since all possible thresholds are considered. In the current and upcoming section, this changes. A confusion table depends on the cutoff since it determines the number of TP/TN/FP/FN. It has to be chosen in a manner which maximizes TP and TN while keeping FP and FN to an acceptable minimum. Further, false positives aren't equally bad to false negatives. The first denote deceased patients, which were classified as positives, the latter vice versa. These asymmetric misclassification costs have to be taken into account. The choice of an optimal cutoff is described in detail in section 4.2.4, for now we consider it as known.

A time dependent confusion development can be obtained in figure 4.2. The x-axis depicts the days of observation, the y-axis corresponds to the number of patients. At every point in time, absolute numbers of TP/FN/TN/FP are stacked and connected over time. Different observation lengths were treated in the same way as for progressing AUC curves. The gained areas deliver an impression for the change of confusion over time, the false positive area is coloured in red to emphasize the asymmetry in misclassification costs.

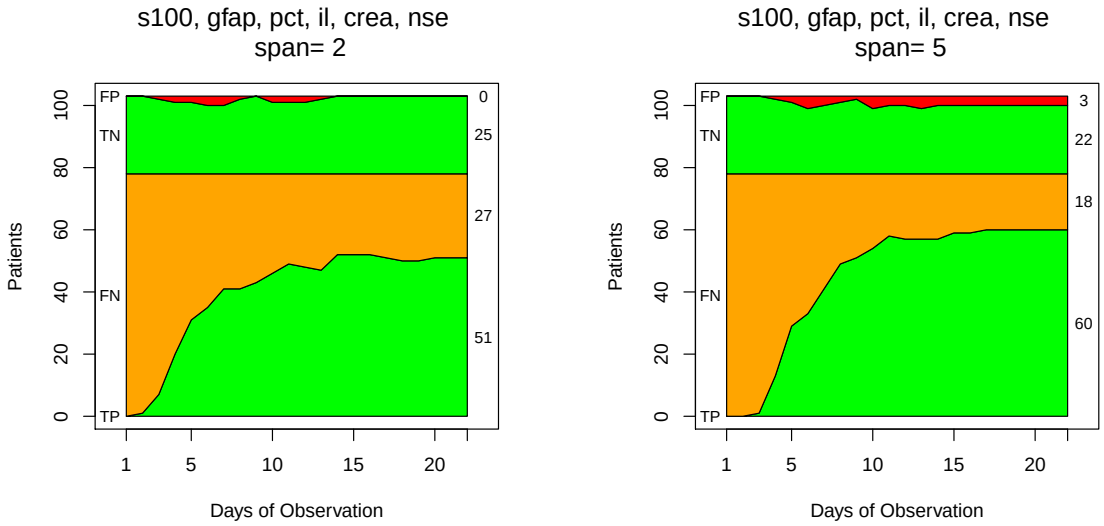


Figure 4.2: Time dependent confusion of model with *s100* and *gfap* (left) and full model (right), both with $span = 7$.

4.2.3 Forecasting - An Early Warning System

Besides a good overall classification performance and acceptable confusion results, we want to add another criterion to the model evaluation. Predictions in clinical monitoring need to be reliable, but if they should contribute to a patient's treatment, they have to be available early enough. A model which assigns a patient to the negative class on the very same day as he/she dies, will get good results for this point in time, but can't actually support physicians in their interventions. It is important, how early a model truly detects patients in the negative/positive class. By that we mean, how many days prior to the endpoint of a patient (i.e. decease/release) the model detects the true class without changing its decision meanwhile. For every observation vector, the classifier produces a vector of squared Mahalanobis distances and further a vector of probabilities. These are linked to prediction vectors which contain the class memberships. The count of 'tails' of these prediction vectors, where the prediction was correct, tells us how early the model recognized the patients' class. If the model changes the assignment from one to the other class, but gets it right at the end, we are interested in how long before the endpoint the model was right.
Figure 4.3 shows true prediction rates in dependence of days before last observations, which from now on I will call the 'forecasting rates'. Note that in contrary to the former plots, the time axis is reversed. Early values correspond to late observations. $t = 1$ denotes the day before the last observation for each patient, $t = 2$ two days before end of treatment and so on. The y-axis shows the percentage of truly detected patients. Circled marks, '$\circ$', correspond to the overall positive predicted rate, dotted marks '$\cdot$' denote to percentage of recognized survivors. The line marked with '$\times$' is the one of special interest, since it shows the model's ability to correctly detect non-survivors early enough.

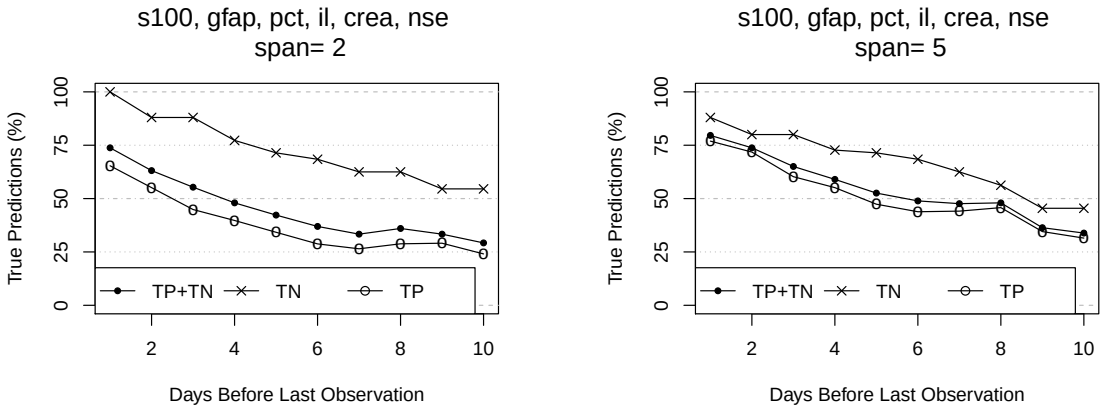


Figure 4.3: Forecasting rates for true positives (TP), true negatives (TN) and the sum TP+TN before last observation.

4.2.4 Choice of Cutoff

AUCs enable us to compare classifiers without the need to specify a fixed cutoff. Anyhow they neglect important and interesting aspects in assessing a model's quality. For a detailed consideration, it is necessary to define a (reasonable) cutoff. As explained in section 3.8, high cutoffs will lead to high specificity and low sensitivity, thus a large number of true negatives but also a large number of false negatives. A low cutoff on the other hand delivers low specificity and high sensitivity, i.e. more true positives but also increased false positives. There is a tradeoff between specificity and sensitivity which basically comes down to a trade-off between false positives and false negatives. An optimal cutoff should lead to a minimum of both FP and FN, FP however is the worse misclassification. Assigning an unhealthy patient who might die to the class of survivors is the worst case, thus it should be treated different to a negative prediction of a survivor. We deal with asymmetric misclassification-costs.
A way to find an optimal cutoff that incorporates these asymmetric costs is to define a cost function and look out for its minimum. That is a function which counts the number of FN and FP for each cutoff but weights the number of FP by a factor c:

$$Costs_{cutoff} := \sum FN + c \cdot \sum FP \tag{4.2}$$

The factor c can be interpreted as follows: every falsely predicted non-survivor weights c times more than every falsely predicts survivor. A screening for non-survivors would demand a high c, $c = 1$ corresponds to weighting FN and FP equally. The optimal cutoff is the one where $Costs_{cutoff}$ reaches a minimum. If there are multiple cutoffs which minimize the cost function, the maximum is chosen. Other options such as the minimum or mean are possible, in this very case anyhow the maximum makes sense since in the case of doubt higher levels will decrease the number of false positives.
Figure 4.4 demonstrates the effect of the costs parameter c on a classifier (the notation m4/s2 is explained in chapter 5). c ascends from 5 to 9 and as we can observe, lower costs correspond to higher true positive and higher false positive rates.

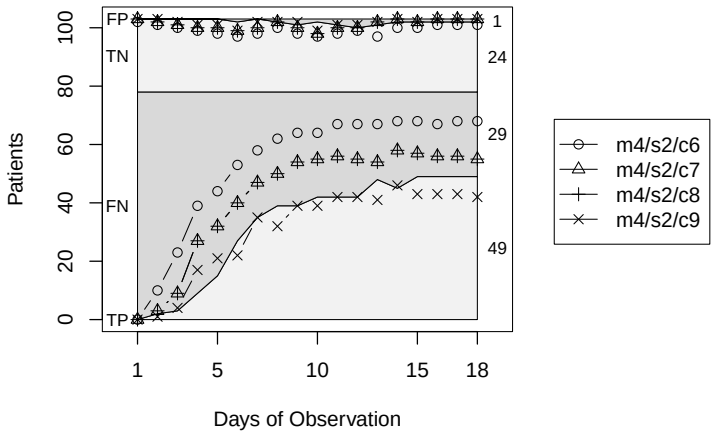


Figure 4.4: Confusion progress with varying costs parameter c.

Including the dataset to determine the optimal cutoff bears the same problem as in fitting the mixed model. To avoid a bias, cutoffs for patients shouldn't be based on their own data, the cutoffs have to be cross validated. Considering patient i, a threshold which minimizes the costs has to be calculated with a dataset excluding the patient's data. Thus in fact we gain a different cutoff for each patient.
For every of the following models as well as for the plots above, cutoffs where determined by a *leave-out-one* cross validation. We gain an individual cutoff for every patient, based on the misclassification costs of all other patients. It turned out that the variance of the cutoffs is very small, which is why for the results we used the maximum of all cross validated cutoffs. This leads to pessimistic values, but then again, a higher cutoff leads to lesser false positive cases, which is the mistake that should be avoided as much as possible.

5 Results

Section 4.2 introduced different points of view in assessing a classifier's performance. The examples were chosen in a way to depict the different aspects. In the following chapter, we will discuss various classification results in detail. We are mainly interested in three things: (1) how do the models change when different feature sets are used, (2) how does the *span* parameter influence the models and (3) how good is the longitudinal classification compared to the current clinical practice?

Since, as mentioned, it is infeasible to examine all possible marker combinations, we will focus on the most promising. Figures 2.2 to 2.5 help to decide which models - i.e. which feature sets - should be considered. The most striking differences can be observed in *s100* and *gfap* trajectories, we conclude that every model should at least incorporate these markers. Also *pct* curves seem promising, further *pct* is an infection specific marker, while the other two are TBI specific, so we can hope to benefit from *pct* because we take other symptoms into account as well. Further we want to incorporate *nse*, *il* and *crea* to see if the classifiers benefit from a broader feature set. All in all we will take a look at five models, which we denote with model 1 to model 5, with an increasing number of markers as can be obtained in table 5.1.

classifier	**features**
model 1	s100, gfap
model 2	s100, gfap, pct
model 3	s100, gfap, pct, il
model 4	s100, gfap, pct, il, crea
model 5	s100, gfap, pct, il, crea, nse

Table 5.1: model 1 to 5 with incorporated markers

The current clinical practice to assess TBI for trauma patients is to consider *s100* levels, where a concentration of 0.1 $\mu g/l$ and above is considered critical. We chose the current *s100* level as a benchmark for the longitudinal classification, although it has to be mentioned that the purposes of these two classifications differ. As TBI-specific marker, *s100* is used to assess brain damage in trauma patients, especially upon arrival in intensive care. The classifiers introduced in this work examine the patient's data with interest in longitudinal observations - especially regarding the possibility of patient monitoring. Despite that, current *s100* levels still can be used as an appropriate benchmark for the model, see section 5.4.

A classification summary of the benchmark can be obtained in figure 5.1. The upper left figure depicts the progression of AUCs over observation time. As we will see, the increase in classification quality over time is typical for our problem. The number of correct classifications will improve with time, since a large part of patients stabilize after

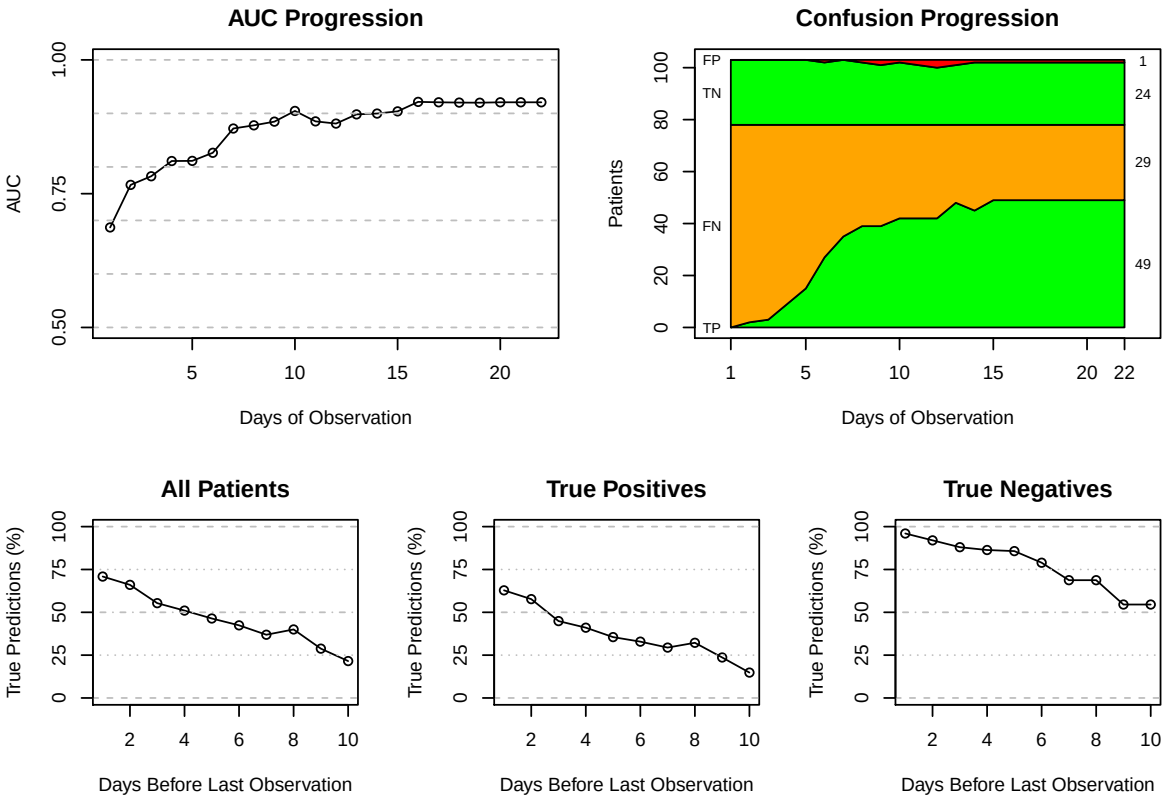


Figure 5.1: Classification summary of benchmark with a cutoff of 0.1 $\mu g/l$; AUC progression (upper left); confusion progress (upper right); true rates before last observations (lower).

a while and are thus correctly classified as positive cases.
Bad classification rates at the beginning do not necessarily mean that the classifiers work bad in early prediction. The data lacks of information about the patients' treatments and change of health condition. It is possible (and likely) for patients to belong to the negative class at the beginning of observation - i.e. their condition is critical - but after proper treatment they switch to the positive class since they are considered fit again. That means we deal with labelled training data, but the labels refer to the endpoint of observation time. It is unknown if and how these labels changed over time. This further complicates interpretations of confusion progressions and forecasting rates.
The confusion progression (upper right) and the forecasting rates rely on a cutoff, which was set to 0.1 in accordance to the clinical threshold. This cutoff is designed to be very specific, but rather insensitive. That's why in the first five days, nearly all patients are classified as negatives (and most of them probably are indeed at that time negative cases), but even at the end some 40% of survivors are assigned to the negatives.
The forecasting rates show a similar behaviour. The classifier's capability of recognizing negatives is very high - with the price of low true positive and overall rates. This of course is a sensitivity/specificity trade-off. A higher cutoff would lead to worse true negative rates but increase true positives.

Below, different models with different spans will be compared. For confusion progressions and forecast rates, cutoffs have to be defined. In all following figures, we will denote the model x with a span y and a cutoff based on the costs parameter c=z with mx/sy/cz. In order to compare the results with the benchmarks, the benchmark curves can be seen in the plots as a grey shade in the background.

5.1 AUC Over Time

The AUC over time can be considered as an 'overall' measure of the goodness of classification. This section will examine the differences of AUC progressions with respect to both, different feature sets and different span lengths.

Let us consider the five models of table 5.1. First we examine the change in classification performance when different time spans are used. Figure 5.2 shows the comparison of AUC progressions. Each sub-figure contains AUC curves of model 1 to 5 with coinciding spans. To get an overall impression, we consider spans of 2,3,5,7,10 and 15.
The first interesting observation is the decline in differences of model performances with increasing spans. For small spans of 2 or 3 (upper graphs), variations in AUC curves are stronger than for large spans of e.g. 10 and 15 (lower graphs).
The best values are reached by models with small spans, but using larger ones stabilizes the performances. In early prediction, AUCs of models with spans ≥ 5 show the best results, thus we conclude, that in the first ~ 5 days, all available information should be taken into account. Overall we observe an improvement in classification when smaller spans are used.
Models 4 and 5 are the largest in terms of incorporated features. Except for spans 10 and 15, they are superior to the others. These models benefit from their 'short term memory', especially for final AUCs for $dpt \geq 15$. On the first day of observation, model 5 receives the best predictions.
Models 1, 2 and 3 perform equal or below the others, we conclude that there is a benefit in using larger feature sets.

With this first impression in mind, we take a closer look at each of the models with different spans, as is done in figure 5.3. Here every graph contains AUC curves of one model with varying spans, which slightly differ for each model to depict the best results. For a prediction at a certain point in time, any model can only use as much information as is available at that time. This means that for each model, results with larger spans are equal to smaller ones at the beginning. Considering two spans a, b where $a > b$, the results coincide for predictions on observation times $dpt < b$.
The focus on the plots should lie on the question, how high spans (marked with '+' and '$\times$') perform compared to low spans ('$\circ$' and '$\triangle$').
AUCs of model 1 - the one with the smallest feature set - improve with increasing span, but this represents the exception. All other models show an interesting phenomenon: for short observation times, larger time spans are superior; but for the final AUCs at times $\gtrsim 12$, shorter memory delivers better performances.

Models 1, 2 and appear mediocre, the top values are reached by models 4 and 5. Although peak AUCs are achieved by m4/s2 and m5/s2, longer spans ~ 5 deliver more stable results.

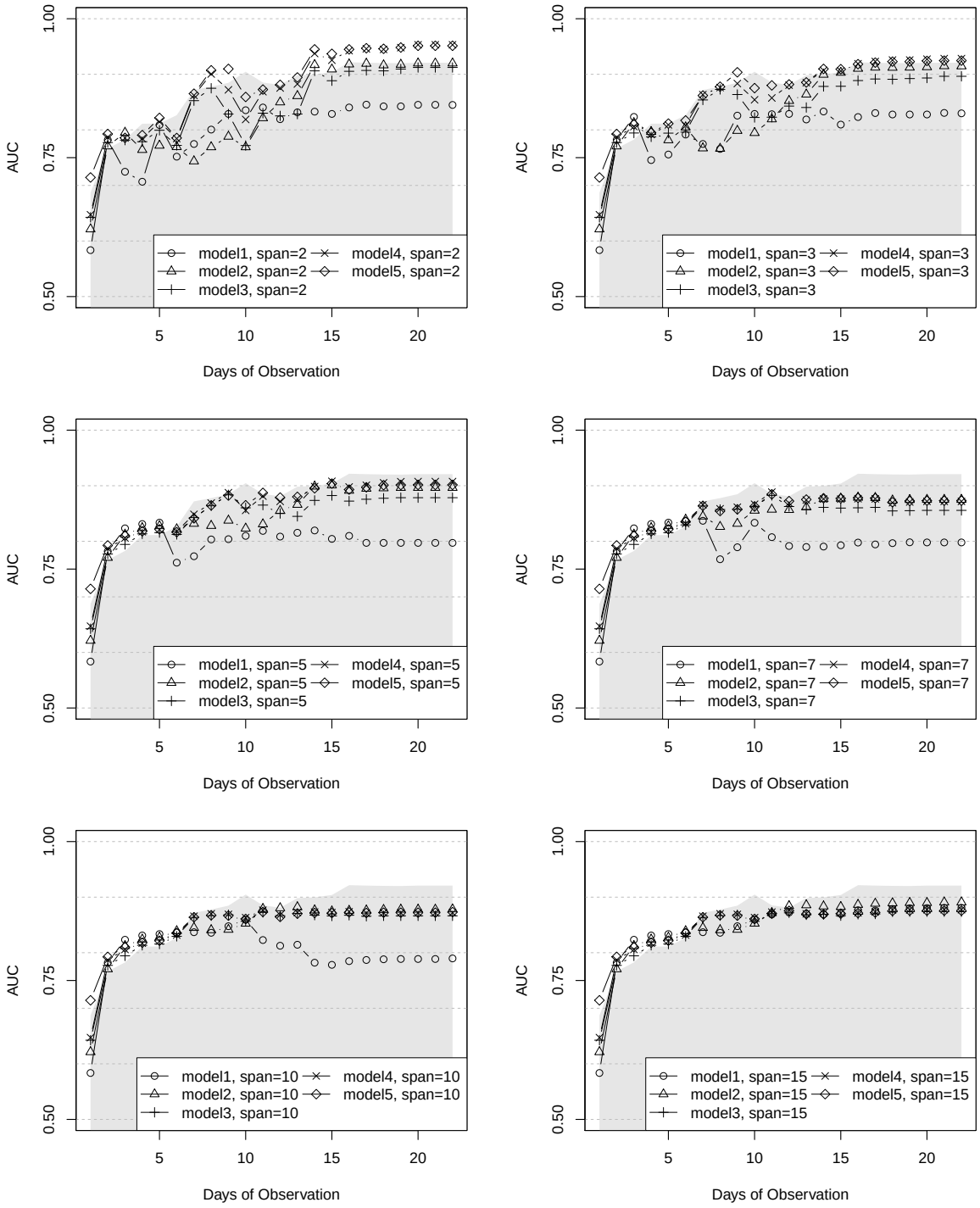


Figure 5.2: AUC progressions of model 1 to model 5 with different spans; sub-figures with varying models and equal spans.

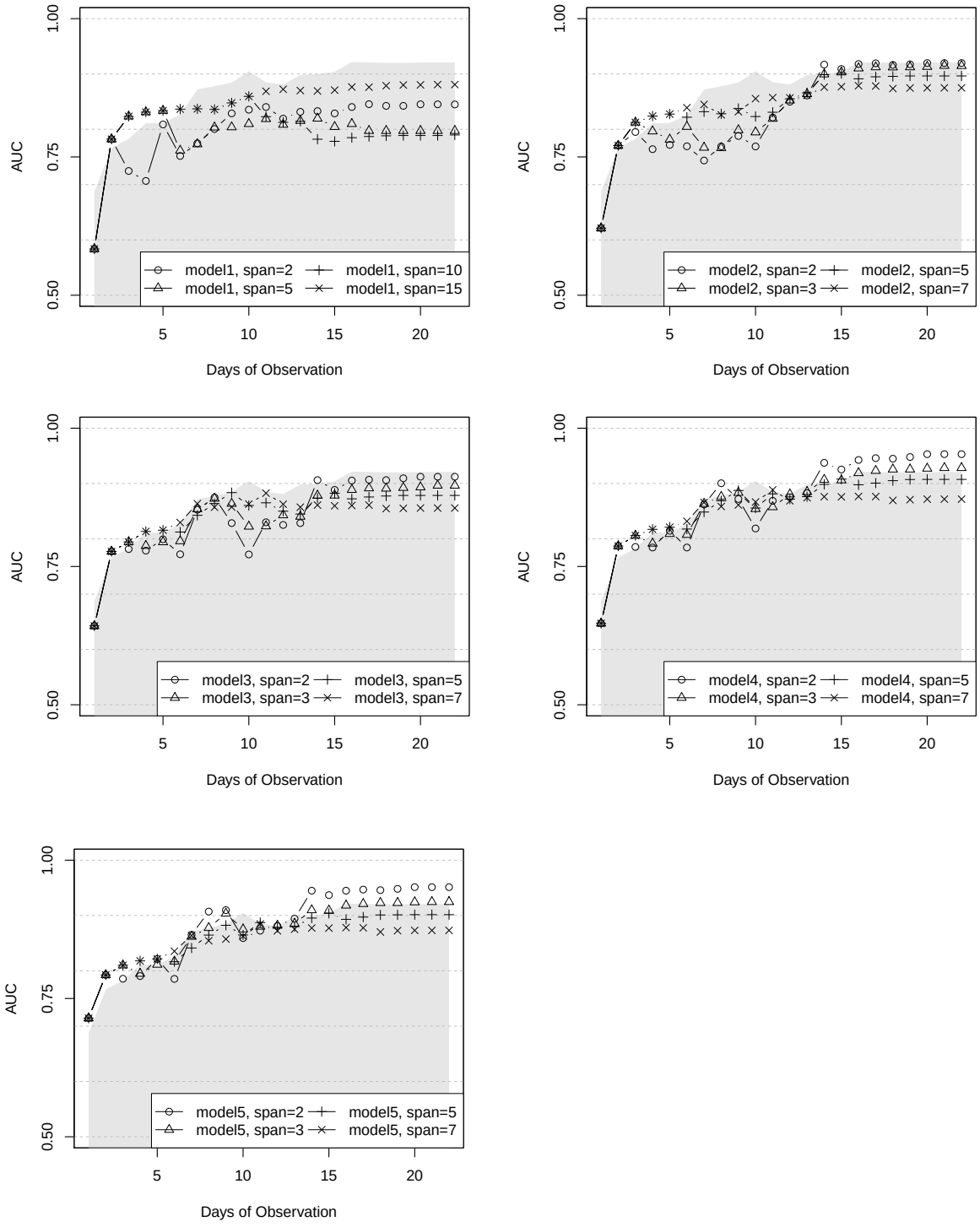


Figure 5.3: AUC progressions of model 1 to model 5 with different spans; sub-figures with equal models and varying spans.

5.2 Confusion Progression

A clearer impression of actual prediction success yields the confusion over time as introduced in section 4.2. Such a progression of confusion contributes to a more differentiated evaluation. Instead of interpreting a single figure, we can obtain the number of correctly and incorrectly classified patients directly and examine the change of prediction success over time.
To compare all patients at all times, the various sample lengths have to be treated in the same manner as for AUCs. If a patient's sample length is smaller then the considered day post trauma, his/her latest available prediction is used. Thus again, confusion results at late points in time not only correspond to late predictions but also to predictions of patients with earlier endpoints.
Confusion rates don't change significantly after the 18th day post trauma, which is why in the following we only consider the progress up to that time.

The drawback of confusion tables/progressions is the need for a specific cutoff. Earlier we described the possibility to define a cutoff based on misclassification rates via a cost-function. In the following, for each model and span, such a cutoff was calculated, where the costs parameter in (4.2) was set to $c = 6$. This value was chosen to meet the rather high misclassification cost of the benchmark classification, while still being able to spot differences in the upcoming plots.

Each trajectory in the graphs of figure 5.4 represents a model/marker combination. Since we already know that the best AUCs are obtained when a small number of past observations is used, we focus on the spans 2, 3, 5 and 7. The benchmark curves should serve as reference points, though one has to be careful with a direct comparison to it. Misclassification is a trade-off between true positives and true negatives. As can be obtained, models with less false positive predictions are poorer in correctly identifying true positives. If we either set c higher than 6 or increase the benchmark cutoff, confusion progressions that now seem different can then be the same (recall figure 4.4). A detailed comparison to the benchmark follows in section 5.1, for now we are interested in the differences of the models with varying spans.
The graphs show that longer spans evoke a smaller number of false positive cases with the costs of decreasing the number of true positives. Imagine patients with high first measurements that decrease early, but worsen after a while. Models with short spans treat these as positives rather soon and take a while to assign the patient to the negatives again. Longer spans on the other hand cause the classifier to remember the high measurements upon arrival for a longer time, thus patients who's condition deteriorates are classified as negatives earlier. A monitoring system should probably favour the latter, in the case of doubt it should be pessimistic.
For spans ≥ 3, models 3 to 5 deliver better classification success than 1 and 2, they keep the number of false positives rather low while having acceptable (with respect to the benchmark) true positive rates. m5/s5 in the lower left graph delivers approximately the same number of false positives as the benchmark, while it identifies a larger number of true positives.

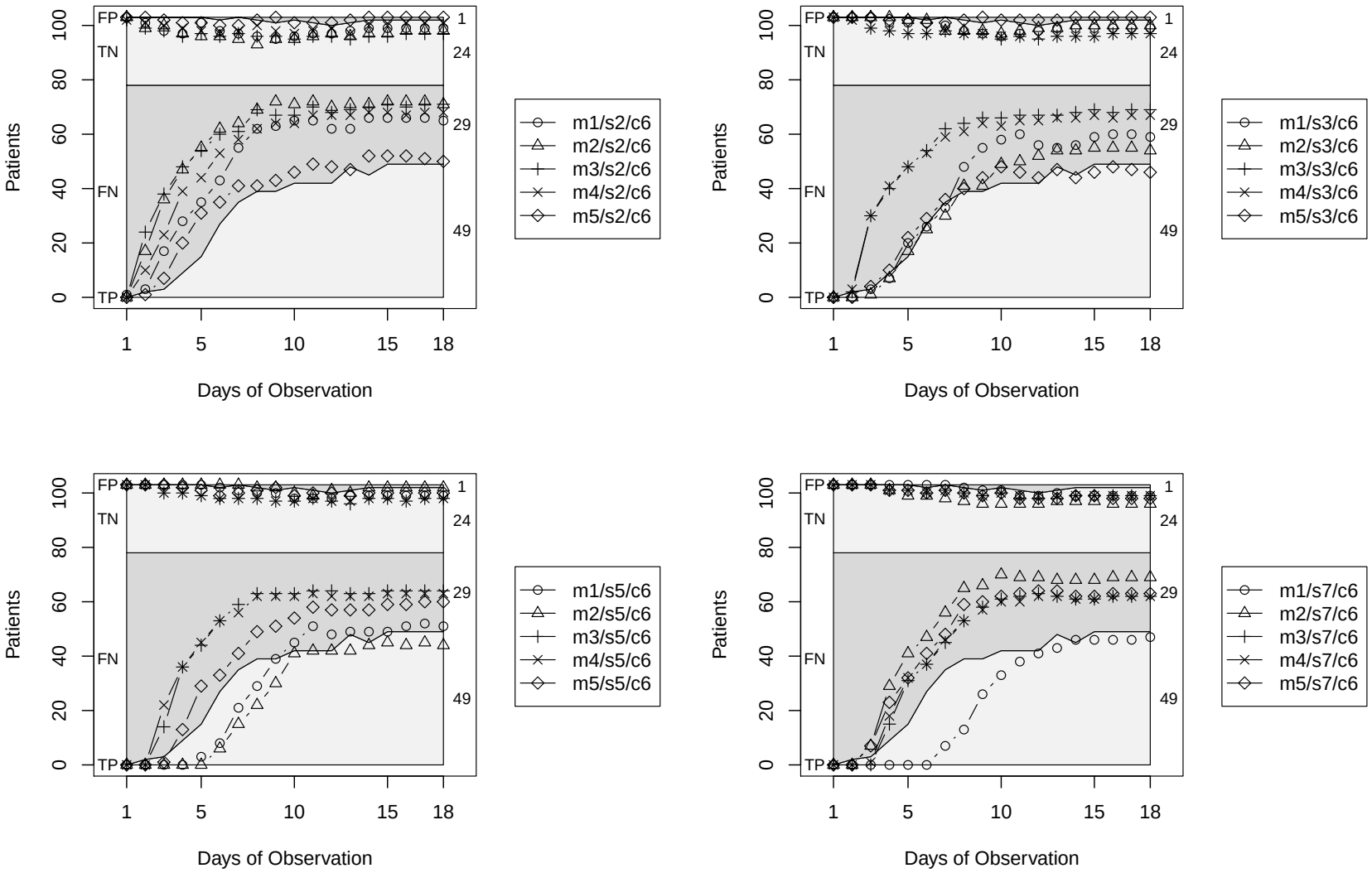


Figure 5.4: Confusion progressions of model 1 to model 5 with different spans; sub-figures with varying models and equal spans.

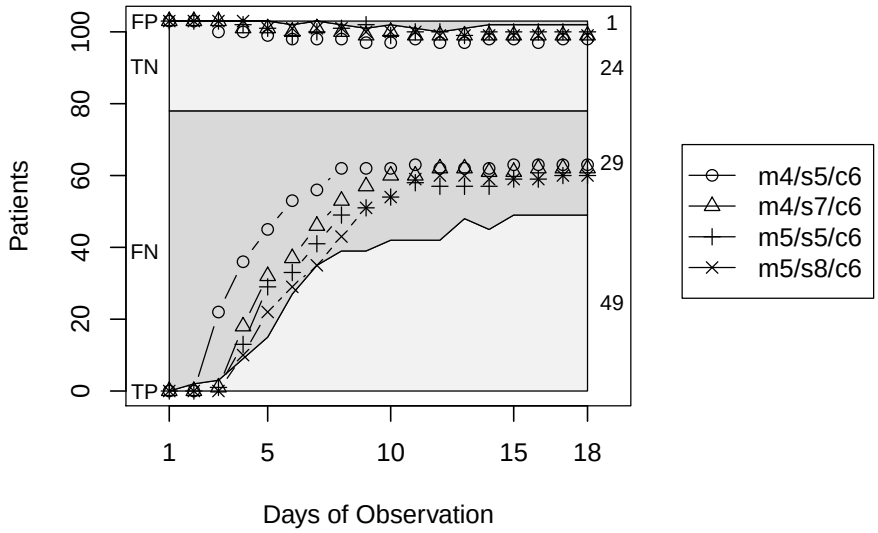


Figure 5.5: Confusion progressions of model 4 and model 5 with different spans.

Figure 5.5 shows the comparison of confusion progressions of four models which are chosen with respect to findings in section 5.3. Note that the combination model4/span7 probably yields the best balance between a low number of false positives and a reduction of false negatives.

5.3 Forecasting Rates

The results so far showed, that the best performances of the longitudinal classification are accomplished by the more sophisticated models 4 and 5, which is why in the following we will focus on these two. While the AUC curves summarize the overall performance and the confusion progressions helped to sharpen those impressions, we now take a look at the classifiers' reliabilities in forecasting a patient's condition. This is especially interesting when we imagine an ICU monitoring system which should alert physicians.
Instead of examining classifications in dependence of days under observation, we compare how early the classifiers are able to determine a patient's condition correctly. The following plots depict, how many days before a patient's endpoint - decease or release - the model correctly classified the patient without changing its decision meanwhile.

The forecasting rates of models 4 and 5 with varying spans can be obtained in figure 5.6. The left column depicts the rates of successful classified patients, the columns in the middle and right separate for truly predicted positives and negatives, respectively. The time axis denotes the remaining days before the last observation of each patient. we choose a maximum of 10 days, although it should be expected, that predictions 10 days before a patients endpoint are rather inaccurate. The rates of correct classifications x days before the last observation refer only to patients which are observed at least for x days. As for confusion progression, the benchmark curves should act as reference, but a direct comparison requires an adjustment of the cutoffs, which is done in the next section.

The graphs confirm that forecast rates decline with days before last observation, some more than others. With respect to a monitoring system, we want the trajectories of true negatives (right column) to be as high as possible while gaining reasonable true positive and overall forecasting rates. The top row of figure 5.6 shows forecast rates of model 4 with spans 2, 3, 5 and 7. Recall that AUCs for a span=7 were unremarkable while for spans 2 and 5 they were among the top levels. We obtain something interesting: forecast rates of true negative cases are superior with span=7 to the other three models, while true positive rates are about the same. The second row shows that model 5 delivers the most promising results with spans 5, 7 and 8.
As for AUC curves, the results converge for greater time spans, which can be observed in the last row, where the models with spans of 10 and 15 are compared. The different curves are almost identical.

To summarize, the best forecast rates for non-survivors with an acceptable amount of misclassified survivors are reached through the model/marker combinations m4/s7, m5/s5 and m5/s8. Other models were considered as well but couldn't perform that good, hence they are left out of this evaluation to keep the amount of figures feasible. Note that those models didn't show remarkable AUC or confusion results. We see that an assessment based on AUCs can be misleading, depending on the purpose of the classifier.

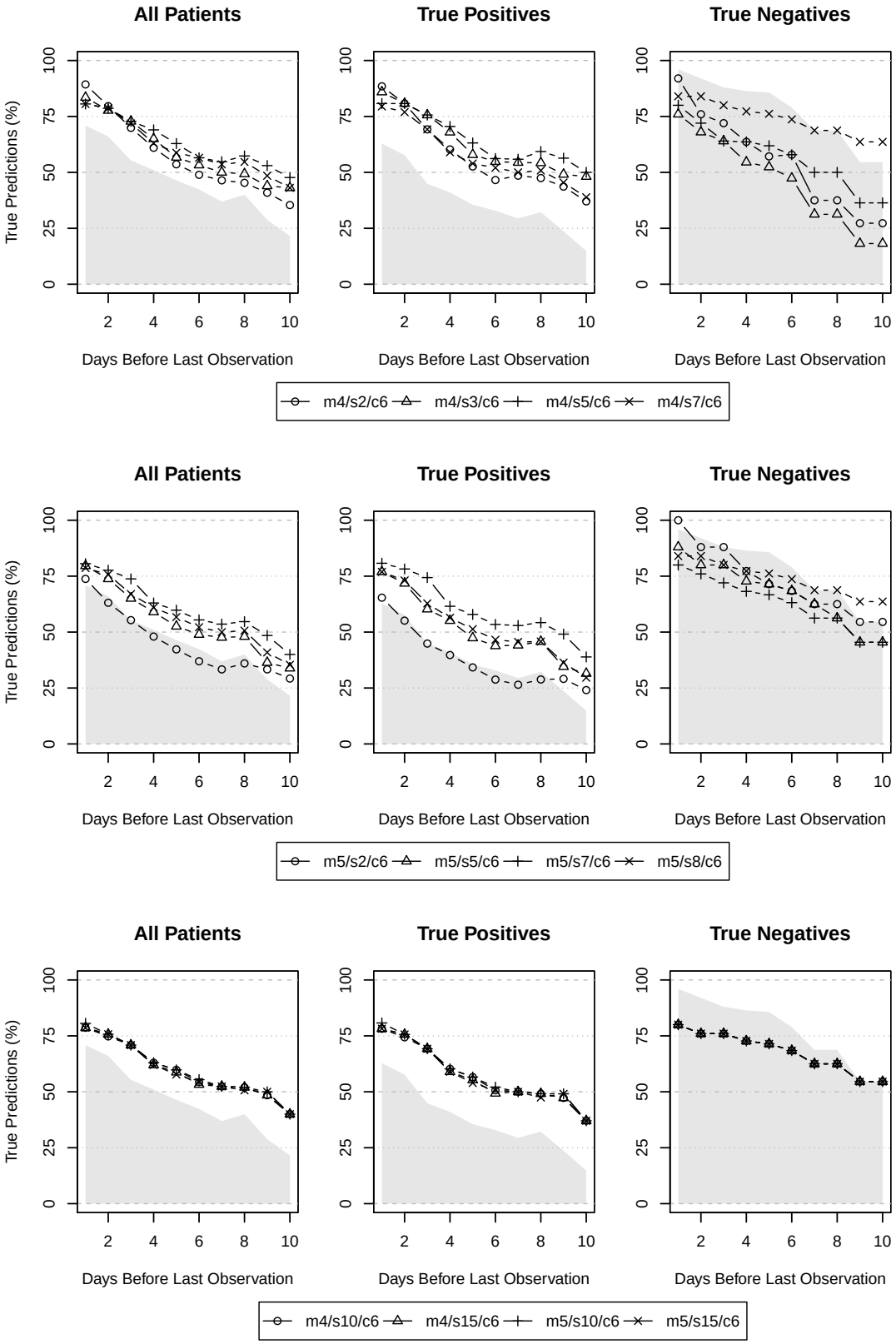


Figure 5.6: Forecasting rates for all patients (left), true positives (center), true negative (right) for models 4 and 5 with varying spans.

5.4 Benchmark Comparison

We now want to examine, if the longitudinal classification bears advantages compared to the rather easy classification via current *s100* levels. Considering the AUC progressions in figures 5.2 and 5.3, we can only obtain slight improvements compared to the benchmark. Model5/span3, model5/span5 and model4/span3 are among the most promising model/span combinations. These models trump the benchmark either at the beginning or in late AUCs, or deliver an overall better classification.
As we saw, confusion progressions are a more accessible way to depict classification performances, but they rely on specific cutoffs. The cutoffs of the longitudinal classifiers are determined by the parameter c in the cost function (4.2). For a comparison with the benchmark, the costs were set to a level where the numbers of false positives in the confusion progression approximately match the confusion of the benchmark. If the model's true positive predictions are superior to the benchmark, we conclude a benefit in the longitudinal classification.

The top of figure 5.7 compares the confusion progression of the models m4/s3, m5/s3 and m5/s5. The costs were set to 6.5, 6 and 7 respectively. Then all models are able to identify almost all non-survivors correctly. But while the positive predictions of m5/s5 are below the benchmark results, the other two, m4/s3 and m5/s3, are above. The best predictions in the positive class are achieved by m4/s3.
The lower graph in figure 5.7 contains confusion curves for the models with good forecasting rates, m4/s7 and m5/s8, compare section 5.3. Here no improvement (with respect to confusion progression) can be observed, recall that also AUCs of that models were unremarkable.

When we consider forecasting rates anyhow, the situation changes. Figure 5.8 shows forecasting results of the upper models. Since it was harder to match true negative rates of the models to the benchmark, I did the opposite and adjusted the costs such that true positive rates are similar. Again the upper graph contains the models with the best AUC curves. The one with the most disappointing confusion curve, m5/s5, scores best when forecasting performance is considered. Starting from 100% of correctly classified non-survivors, rates stay at over some 80% even 10 days before the last observation. The true positive and overall rates are slightly below the benchmark results though. The other two represent no improvement to the benchmark.
The bottom of figure 5.8 depicts forecast results of m4/s7 and m5/s8. Theses are the models with the strongest forecasting rates. While delivering approximately the same or slightly better overall and true positive forecasts, the true negative rates are more stable.

In forecasting, the considered models with a longer span are superior. Also the best forecasting models are not the ones with the best AUC/confusion results. The reason for that was discussed earlier in a slightly different context. Patients with bad first measurements and a fast decrease of to high marker levels are classified as positives by models with short spans rather soon. If conditions worsen, the classifier takes longer to shift the patient to the negative class. Longer spans cause more negative predictions for such cases. On the other hand, the idea of short term memory originated in the

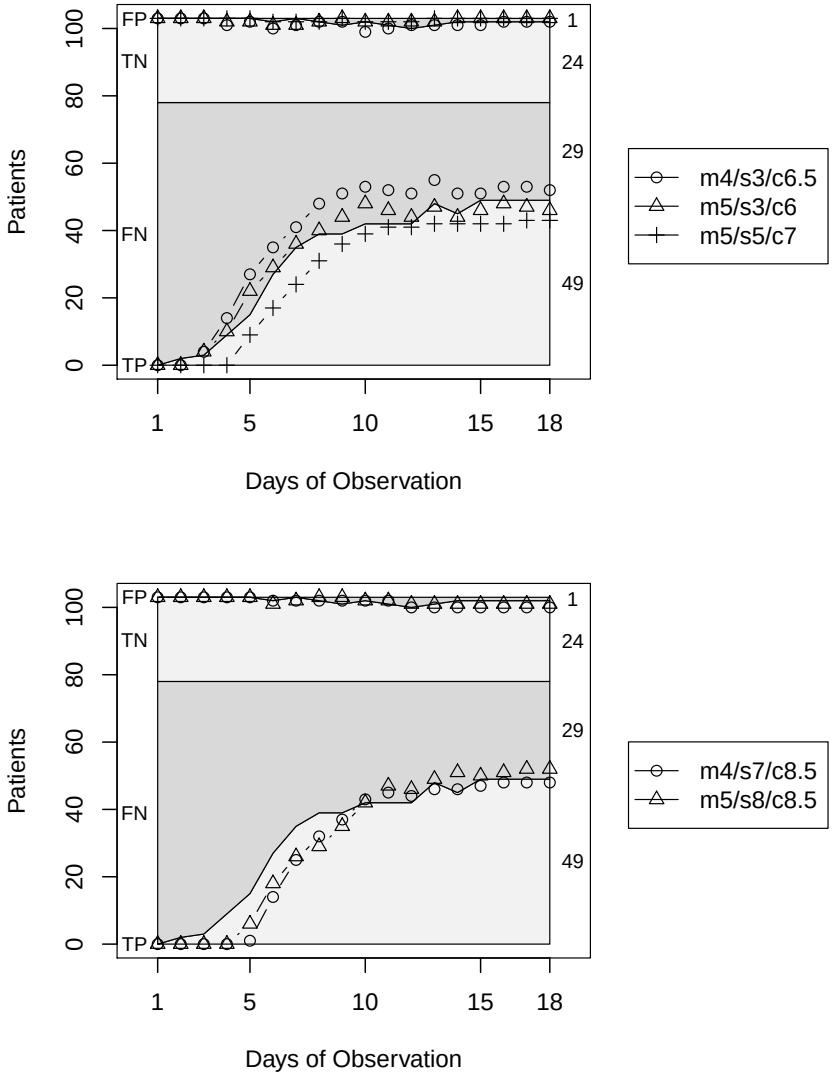


Figure 5.7: Confusion progression of the most promising models with varying costs.

observation, that classifiers should have the property to forget high first measurements. Once more we obtain a trade-off, short memory will decrease true negative forecast rates, long memory will be too pessimistic and evoke a large number of false negatives.

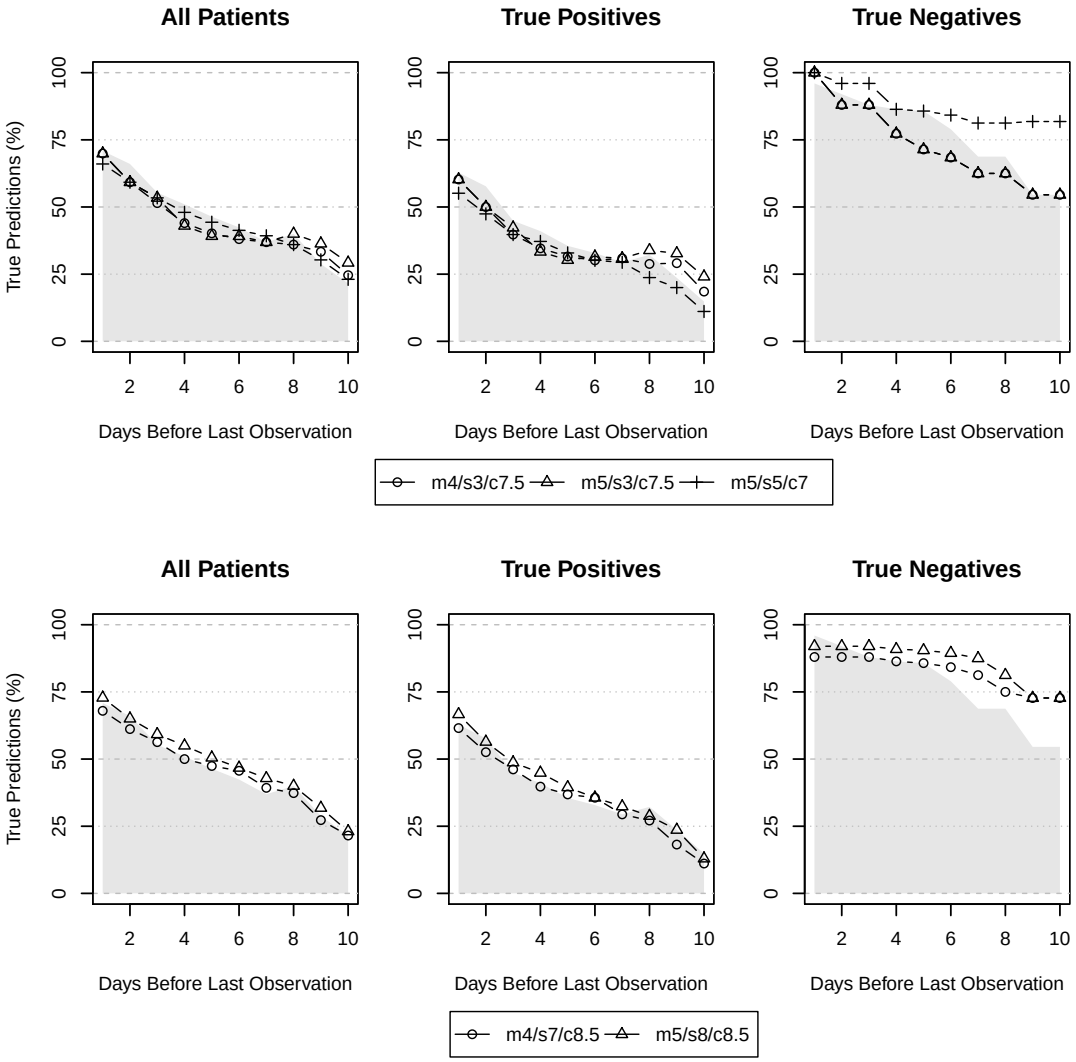


Figure 5.8: Forecasting rates of different model/marker combinations.

5.5 Visual Support

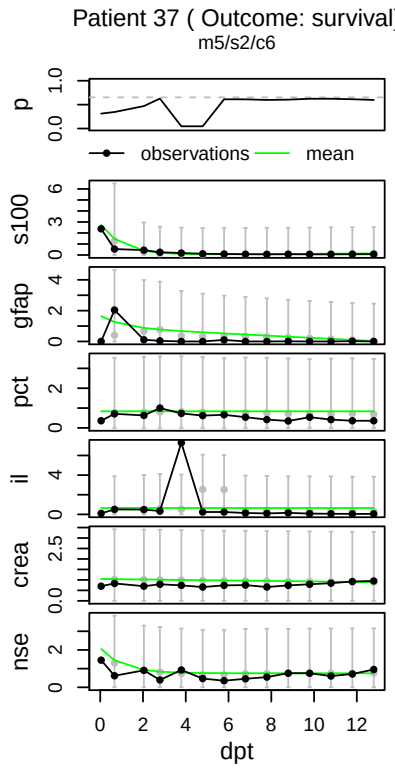
An individual's condition is not a matter of black and white. A 'non-survival' classification doesn't mean, that a patient's fate is decided, but rather that his biosignals show deviations from behaviours of 'typical survivors' and that he/she should receive special attention. Further, it is of interest what caused the classifier's decisions, especially the negative assignments. A feedback, which observation lead to a negative prediction, can help to find appropriate treatment measures.
Thus besides a binary assessment of health condition, physicians can benefit from a visual support. To achieve this, we described a conditional confidence interval (section 3.5), a confidence bound for each biomarker conditional on the prior observations.

Such conditional CIs can be obtained in figure 5.9. Each subfigure contains observed marker levels of a patient, together with expected survival curves and conditional confidence bounds. Above the marker curves, one can observe the classification progression of the patient. The higher the p-value, the better the resemblance of the data to expectation. The dashed line denotes the cutoff based on costs $c = 6$, a p-value above this line corresponds to a positive classification. For subfigures (a) and (b), model 5 with span 2 was used, (c) and(d) depict model 5 with span 5.

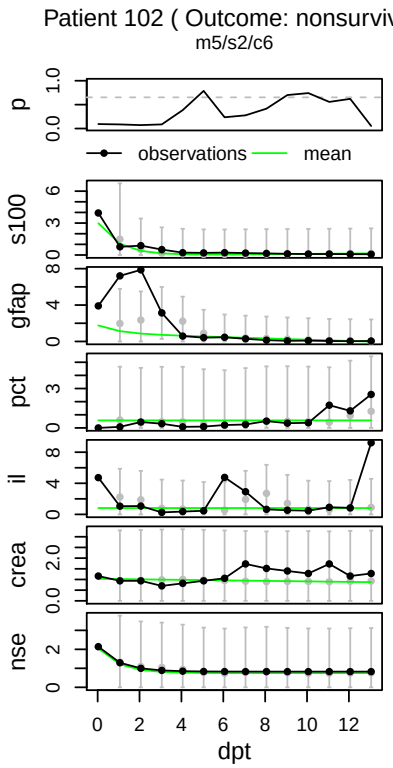
Figure 5.9 (a) and (b) show the fast reaction of low spans. In (a), at day 4 *il* increases and causes a fall of the health assessment. The next day, the *il* concentration drop to normal and after another day, p is back to its prior level. In figure 5.9 (b), the model makes a positive prediction after marker curves settle, changes its decision when *il* increases and then goes back to the positive prediction. In the end, an even larger increase in *il*-concentration causes a (correct) negative classification, but at a rather late point in time.
Subfigures (c) and (d) show slower assessment changes because of the longer memory of the models. The patient in subfigure (c) early shows desirable marker levels, still it takes half of the observation time for the model, to assign him/her to the positive class. In (d) we observe a negative classification after six days, 9 days before the actual outcome of the patient.

This visual support mainly serves two purposes. First, it gives physicians a feedback, which marker levels (or other biosignals) are out of bound and cause a bad prediction of health condition. Second, the assessment can be observed continuously rather than binary. Instead of considering only a positive/negative classification, one can obtain its severity and furthermore its progression over time.

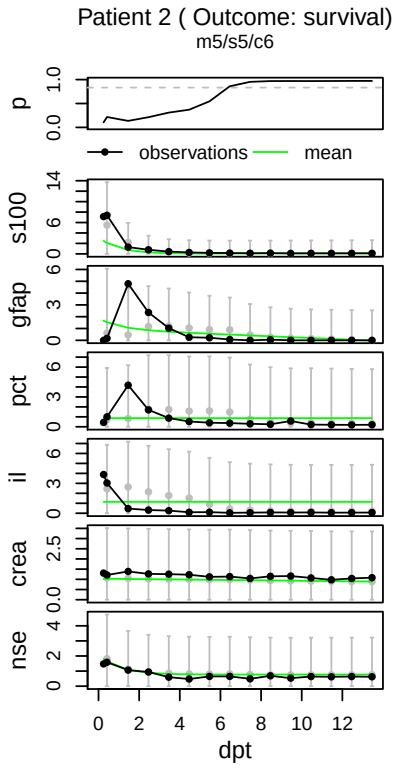
(a)



(b)



(c)



(d)

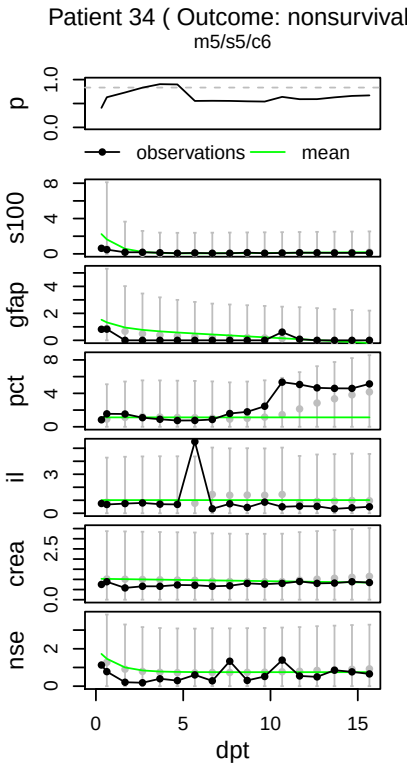


Figure 5.9: Conditional confidence intervals for four patient trajectories, based on model5/span2 (upper) and model5/span5 (lower).

6 Conclusion

This work discussed an approach of longitudinal data classification with updating data samples. The motivation of that approach originated from the idea of a patient monitoring in intensive care treatment. It is clinical practice, to consider only the newest observations of biosignals, disregarding a possible benefit in taking a patient's history into account. In order to explore this, a dataset of trauma patients containing longitudinal biomarker levels was considered.
For classification, characteristics of the occurring classes have to be known or estimated. The graphical exploration revealed, that the data at hand makes it impossible or at least very inaccurate, to estimate the negative class's traits. That is why a one class classification was chosen, a classification based only on knowledge about one class - the survivors. The mixed model approach to estimate the survivors' distribution delivers a lot of advantages. The fitting of the mixed model represent one time costs only, once the regression coefficients are estimated, classification is rather simple. The multiple regression allows for imputation of missing data as well as the handling of the unbalanced dataset, further it enabled an incorporation of both, multiple features and multivariate response variables. By assessing deviations of expectation using the squared Mahalanobis distance, the longitudinal classifiers are flexible when a different number of prior observations are considered, as well as when different misclassification costs are examined.
Further, the described approach allowed for a visual support which can improve a physicians impression on a patient's condition. In addition, the visual support explains how the classification works and thus contributes to the transparency of the process.
Evaluation represented a challenge. Treatment outcomes of the patients are labelled, but those labels can't be considered as fixed upon arrival in intensive care. Besides an assessment of prediction performance using ROC curves and their AUCs, progressing confusion results and forecasting rates were introduced. By these we gained deeper insights of various models and their differences.
The results showed that longitudinal classification benefits from the use of a broader feature set. Also short term memories showed significant improvements. To benchmark the results, current levels of one marker only were used, as it is done to assess brain injury for trauma patients. Most classifiers presented in this work were at least able to match the benchmark results. Depending on the evaluation aspect, some models were able to trump them, but not to a great extent.
The most significant improvements were obtained in forecasting the patients' condition. Here longitudinal classifiers were able to correctly predict the same amount of survivors as the benchmark, while stabilizing non-survivor predictions. With regard to a patient monitoring, this represents an important improvement.
Overall, when considering the additional effort, longitudinal classification was not able to convincingly improve the easier assessment by current marker levels. Nevertheless, it does bear advantages, maybe larger sample sizes as well as a broader patient specific data could further improve and justify the more sophisticated longitudinal approach.

7 Appendix: R Codes

The following R libraries were are used in the R scripts and functions below:

- `lme4`: fitting the linear mixed model
- `mvtnorm`: handling multivariate normal random variables
- `reshape2`: handling data to gain the block-diagonal form necessary for the MEM
- `parallel, foreach, doParallel`: enables parallel computing to reduce computation time

7.1 Data and Model Structure

```
structure_data <- function(data){
#structure data to diagonal form and generate necessary matrix
   for the MEM

  marker <- c("s100", "gfap", "pct", "il", "crea", "nse")
  data <- melt(data, id.vars = c("fold","id","dpt","group","
     sepsis","survival"), measure.vars = marker)
  names(data)<-c("fold","id","dpt","group","sepsis","survival","
     biomarker","value")
  data <- data[with(data, order(id)),]

  data <-   cbind(data,
      "interc_s100" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="s100"),
      "dpt_s100" = data$dpt * as.numeric(data$biomarker=="s100"),
      "dptinv_s100" = exp(-data$dpt) * as.numeric(data$biomarker
         =="s100"),
      "interc_gfap" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="gfap"),
      "dpt_gfap" = data$dpt * as.numeric(data$biomarker=="gfap"),
      "dptinv_gfap" = exp(-data$dpt) * as.numeric(data$biomarker
         =="gfap"),
      "interc_pct_gr1" = rep(1,length(data$id)) * as.numeric(data
         $biomarker=="pct") * as.numeric(data$group==1),
      "interc_pct_gr2" = rep(1,length(data$id)) * as.numeric(data
         $biomarker=="pct") * as.numeric(data$group==2),
      "interc_pct_gr3" = rep(1,length(data$id)) * as.numeric(data
         $biomarker=="pct") * as.numeric(data$group==3),
      "interc_il_gr1" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="il") * as.numeric(data$group==1),
```

```
      "interc_il_gr2" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="il") * as.numeric(data$group==2),
      "interc_il_gr3" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="il") * as.numeric(data$group==3),
      "interc_crea" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="crea"),
      "dpt_crea" = data$dpt * as.numeric(data$biomarker=="crea"),
      "interc_nse" = rep(1,length(data$id)) * as.numeric(data$
         biomarker=="nse"),
      "dpt_nse" = data$dpt * as.numeric(data$biomarker=="nse"),
      "dptinv_nse" = exp(-data$dpt) * as.numeric(data$biomarker==
         "nse")
  )
  return(data)
}
```

```
impute <- function(data, covariates.fix, covariates.ran){

  form.imp <- as.formula( paste("value ~ ", paste(covariates.fix,
      collapse="+"), "- 1 + (", paste(covariates.ran, collapse="+
     "), "-1 | id)") )
  na.sel <- which(is.na(data$value))
  if( length(na.sel)>0 ){
    fm.impute <- lmer(form.imp, data = data[-na.sel,])
    data$value[na.sel]<-predict(fm.impute,newdata=data[na.sel,],
       allow.new.levels=T,re.form=NA)
  }
  return(data)
}
```

7.2 Model Fit

```
model_fit <- function(data, covariates.fix, covariates.ran){

# cross-validated mixed model fit with given fixed and random
   effects to data; fits are computed parallel
# input:  data ... allready structured dataframe in 'long fomat'
   (block-matrices) & with assigned fold to each patient
#   covariates.fix, covariates.ran ... character vectors with
   corresponding random/fixed effects, matching with colums of '
   data'
# output: list containing:  - "id": patient id
#         - "fold": fold of patient (to connect patient with
   corresponding fit)
#         - "true": true outcome (1=survival, 0=nonsurvival)
#         - "cv.fits": 11 fits (lmer-outputs), for survivors one
   for each fold, one for the non-survivors
```

```
10
# SCAN FEATURES IN USE & GENERATE FORMULA FOR MODEL FIT

15  features <- unique(data$biomarker)[unlist(lapply(unique(data$
      biomarker), function(x) TRUE %in% grepl(x, covariates.fix)))
      ]
    form <- as.formula( paste("value ~ ", paste(covariates.fix,
      collapse="+"), "- 1 + (", paste(covariates.ran, collapse="+"
      ), "-1 | id)") )

# PARALLEL CLUSTER

20  cl <- makeCluster(detectCores() -1)
    registerDoParallel(cl, cores = detectCores()-1)

# CV: FIT FOR POSITIVE CLASS; IN EACH FIT ONE FOLD IS EXCLUDED

25  cv.fits <- foreach(fold = unique(data$fold), .packages=c("lme4"
      , "nlme")) %dopar% {

      fold_sel <- (data$fold == fold)
      fm1 <- lmer(form, data=data[!fold_sel & data$survival == 1 &
        data$biomarker %in% features,], REML = TRUE)
      return(fm1)
30  }

    stopCluster(cl)

# FIT WITH ALL POSITIVE UNITS TO CLASSIFY NEGATIVE CASES
35
    fm1.allpositive <- lmer(form, data=data[data$survival == 1 &
      data$biomarker %in% features,], REML = TRUE)

# PREPARE OUTPUT: CONTAINS ID, FOLD, TRUE OUTCOME & FITS

40  true.outcome <- tapply(data$survival, data$id, function(x) as.
      numeric(1 %in% x) )
    id.fold <- tapply(data$fold, data$id, function(x) unique(x) )

    return(list("id" = unique(data$id), "fold" = id.fold, "true" =
      true.outcome, "cv.fit" = list(cv.fits, fm1.allpositive)))
}
```

7.3 Mahalanobis Distances and Probabilities

```
cmd <- function(data, mem.fits, span){

# mahalanobis distance for data with fits gained from mem.fits;
   only a maximum of 'span' observations are used
# input:  data ... same as in model_fit
#   mem.fits ... output object of model_fit
#   span
# output: CMD ... list of consecutive mahalanobis distances for
   each patient

# READ PARAMETERS

  covariates.fix <- intersect(strsplit(as.character(formula(mem.
     fits$cv.fit[[2]])), split= " " )[[3]], names(data))
  covariates.ran <- covariates.fix
  features <- unique(data$biomarker)[unlist(lapply(unique(data$
     biomarker), function(x) TRUE %in% grepl(x, covariates.fix)))
     ]
  nfeat <- length(features)

# CONSECUTIVE MAHALANOBIS DISTANCES FOR EACH PATIENT; MEAN AND
   VARIANCE ARE COMPUTED RELYING ON FITS

  CMD <- list()
  for(id in unique(data$id)){
    data.pat <- data[(data$id==id) & (data$biomarker %in%
       features),]
    if( unique(data.pat$survival) == 1 ) { fit <- mem.fits$cv.fit
       [[1]][[mem.fits$fold[id]]] } else { fit <- mem.fits$cv.fit
       [[2]] }
    n <- length(data.pat$id)
    Z <- as.matrix(data.pat[covariates.ran])
    X <- as.matrix(data.pat[covariates.fix])
    mu1 <- X %*% fixef(fit)
    Y <- as.numeric(as.matrix(data.pat$value))
    distance.progress <- vector(length=n/nfeat)
    for(step in 1:(n/nfeat)){
      sel <- c(sapply(1:nfeat,function(k) max(1, step - span +1):
         step+(n/nfeat*(k-1))))
      if(length(sel) == 1){
        sigma1 <- sum((Z[sel,] %*% VarCorr(fit)$id) *  t(Z[sel,])
           ) + sigma(fit)^2
      } else {
        sigma1 <- (Z[sel,] %*% VarCorr(fit)$id) %*% t(Z[sel,]) +
           sigma(fit)^2*diag(dim(Z[sel,])[1])
      }
```

```
      distance.progress[step] <- sqrt( t(Y[sel] - mu1[sel]) %*%
         solve(sigma1) %*% (Y[sel] - mu1[sel]) )
    }
    CMD <- c(CMD, tapply(distance.progress, factor(data.pat$id)
       [1:length(distance.progress)], function(x) x ))
  }

  CMD <- CMD[unique(data$id)]
  return(CMD)
}
```

```
assign_pval <- function(CMD, span){

# assign probability to the vector CMD of Mahalanobis Distances
# input:  CMD ... output object of 'cmd'
#   span
# output: pval ... p values for patients at different points in
   time

  pval <- lapply(CMD, function(x) pchisq(x,c(1:min(span, length(x
     )) ,rep(span,max(0,length(x)-span))),lower.tail=F))
  max.length <- max(unlist(lapply(CMD, length)))
  pval.matrix <- matrix(NA, ncol=length(pval), nrow=max.length)
  colnames(pval.matrix) <- names(pval)
  for(id in names(CMD)){ pval.matrix[1:length(pval[[id]]),id] <-
     pval[[id]]
            pval.matrix[-c(1:length(pval[[id]])),id] <- tail(
               pval[[id]],1) }

  return( list("list"=pval, "matrix"=pval.matrix) )
}
```

7.4 Benchmark Evaluation

```


ids <- data$id[data$biomarker=="s100"]
trues <- tapply(data$survival[data$biomarker=="s100"] , ids,
   unique)
Cutoff <- 0.1

#AUC PROGRESSION

  auc <- function(pval, true){

    cutoff <- sort(data$value[data$biomarker=="s100"])
    sens <- vector(length=length(cutoff))
```

```
    fpr <- vector(length=length(cutoff))

    for(i in 1:length(cutoff)){
      pred.class <- as.numeric(pval < cutoff[i])
      TP <- sum(true==1 & pred.class==1)
      TN <- sum(true==0 & pred.class==0)
      FN <- sum(true==1 & pred.class==0)
      FP <- sum(true==0 & pred.class==1)

      sens[i] <- TP/(TP+FN)
      fpr[i]  <- 1 - TN/(TN+FP)

    }
    return( c("AUC"=trapz((fpr),(sens))) )
  }

  s100.list <- tapply(data$value[data$biomarker=="s100"], ids,
     function(x) x)
  max.length <- max(unlist(lapply(s100.list, length)))
  s100.matrix <- matrix(NA, ncol=length(s100.list), nrow=max.
     length)
  colnames(s100.matrix) <- names(s100.list)
  for(id in names(s100.list)){ s100.matrix[1:length(s100.list[[id
     ]]),id] <- s100.list[[id]]
          s100.matrix[-c(1:length(s100.list[[id]])),id] <- tail
             (s100.list[[id]],1) }

  s100.auc <- apply(s100.matrix, 1, function(x) mahal_auc(x,trues
     ))[1,]

#CONFUSION PROGRESS

  TP=TN=FP=FN<- vector(length = dim(s100.matrix)[1])
  for(i in 1:dim(s100.matrix)[1]){
    pred.class <- as.numeric(s100.matrix[i,] < Cutoff)

    TP[i] <- sum(trues==1 & pred.class==1)
    TN[i] <- sum(trues==0 & pred.class==0)
    FN[i] <- sum(trues==1 & pred.class==0)
    FP[i] <- sum(trues==0 & pred.class==1)
  }

#FORECAST RATES

  tail_count <- function(x){
    x <- unlist(x)
    count <- 0
    index <- length(x)
```

```
  while( (tail(x,1) == x[index]) && (index > 0) ){ count <-
     count + 1; index <- index -1 }
  return(count)
}

detect <- lapply(s100.list, function(x) x < Cutoff)
final.predictions <- lapply(detect, function(x) tail(x,1))
sel <- names(final.predictions[unlist(final.predictions) ==
   unlist(trues)]) # TP & TN Patients

trues.before.end <- unlist(lapply(detect[sel], tail_count))
length.obs <- unlist(lapply(s100.list, function(x) length(x)))
max.length <- 10

# TRUE PREDICTION BEFORE ENDPOINT: ALL PATIENTS
  rates.all <- (sapply(1:max.length, function(x) length(which(
     trues.before.end>=x))) / unlist(lapply( 1:max.length,
     function(x) length(which(length.obs>=x))))  )*100

# TRUE NEGATIVE PREDICTIONS BEFORE ENDPOINT
  TN.sel <- names(which(trues == 0))
  detect <- lapply(s100.list[TN.sel], function(x) x < Cutoff)
  final.predictions <- lapply(detect, function(x) tail(x,1))
  sel <- names(final.predictions[unlist(final.predictions) ==
     unlist(trues[TN.sel])]) # TN Patients
  trues.before.end <- unlist(lapply(detect[sel], tail_count))
  length.obs <- unlist(lapply(s100.list[TN.sel], function(x)
     length(x)))
  rates.tn <-  ( sapply(1:max.length, function(x) length(which(
     trues.before.end>=x))) / unlist(lapply( 1:max.length,
     function(x) length(which(length.obs>=x))))  )*100

# TRUE POSITIVE PREDICTIONS BEFORE ENDPOINT
  TP.sel <- names(which(trues == 1))
  detect <- lapply(s100.list[TP.sel], function(x) x < Cutoff)
  final.predictions <- lapply(detect, function(x) tail(x,1))
  sel <- names(final.predictions[unlist(final.predictions) ==
     unlist(trues[TP.sel])]) # TP Patients
  trues.before.end <- unlist(lapply(detect[sel], tail_count))
  length.obs <- unlist(lapply(s100.list[TP.sel], function(x)
     length(x)))
  rates.tp <- ( sapply(1:max.length, function(x) length(which(
     trues.before.end>=x))) / unlist(lapply( 1:max.length,
     function(x) length(which(length.obs>=x))))  )*100

s100.rates.all <- rates.all
s100.rates.tp <- rates.tp
s100.rates.tn <- rates.tn
```

```
  save(file="s100.benchmarks.RData",s100.auc,s100.pauc,s100.rates
     .all, s100.rates.tp, s100.rates.tn, s100.sens, s100.spec,
     s100.matrix)
```

7.5 Cutoff

```
#CROSS VALIDATED SINGLE VALUE CUTOFF


  #-----------------------------------------------------
  opt_cutoff <- function(pval, true){

    cutoff <- c(0, unlist(lapply(seq(0,1,0.01), function(x)
       quantile(pval[true==1], x, na.rm=T))) , 1)
    loss <- vector(length=length(cutoff))
    for(i in 1:length(cutoff)){
      pred.class <- as.numeric(pval > cutoff[i])
      TP <- sum(true==1 & pred.class==1)
      TN <- sum(true==0 & pred.class==0)
      FN <- sum(true==1 & pred.class==0)
      FP <- sum(true==0 & pred.class==1)

      loss[i] <- FN + costs*FP
    }
    #plot(cutoff,loss)
    Cutoff <- max(cutoff[which(loss==min(loss))])  #optimal
       cutoff based on costs
    return(Cutoff)
  }
  #-----------------------------------------------------

  for(costs in seq(5,10,0.5)){
    cutoff.opt <- matrix(ncol=15, nrow=5)
    for(model.number in 1:5){
      load(paste0("model_fits/model",model.number,".RData"))

      for(span in 1:15){

        CMD <- cmd(data, mem.fits, span)
        pval <- unlist(lapply(CMD, function(x) pchisq(x,c(1:min(
           span, length(x)) ,rep(span,max(0,length(x)-span))),
           lower.tail=F)))
        trues <- unlist(data$survival[data$biomarker == "s100"])
        id <- data$id[data$biomarker == "s100"]

        cutoff.cv <- lapply(unique(id), function(x) opt_cutoff(
           pval[-which(id == x)], trues[-which(id == x)]))
```

```
        cutoff.opt[model.number, span] <- max(unlist(cutoff.cv))
      }
    }

    save(file = paste0("cutoffs/optimal.cutoff.costs",costs,".
       RData"), cutoff.opt)
  }
```

7.6 Conditional CI (with Plot)

```
# PATIENT PLOTS FOR CV MODEL

conditional_ci <- function(model.number, span, costs = 6, ids =
   NA){

  load(paste0("model_fits/model",model.number,".RData"))
  load(paste0("cutoffs/optimal.cutoff.costs",costs,".RData"))
  cutoff <- cutoff.opt[model.number,span]

  CMD <- cmd(data, mem.fits, span)
  pval <- assign_pval(CMD, span)
  true <- mem.fits$true

  covariates.fix <- intersect(strsplit(as.character(formula(mem.
     fits$cv.fit[[2]])), split= " " )[[3]], names(data))
  covariates.ran <- covariates.fix
  features <- unique(data$biomarker)[unlist(lapply(unique(data$
     biomarker), function(x) TRUE %in% grepl(x, covariates.fix)))
     ]
  nfeat <- length(features)

  add.error.bars <- function(X,upper,lower,width,col=col,lwd=1,
     lty=1){
    segments(X,max(0,lower),X,max(0,upper),col=col,lwd=lwd,lend
       =1,lty=1)
    segments(X-width/2,max(0,lower),X+width/2,max(0,lower),col=
       col,lwd=lwd,lend=1,lty=1)
    segments(X-width/2,max(0,upper),X+width/2,max(0,upper),col=
       col,lwd=lwd,lend=1,lty=1)
  }

  # PATIENT PLOTS

  if(is.na(ids)){ ids <- unique(data$id) }
  for(id in ids){

    # PATIENT DATA
```

```
data.pat <- data[which((data$id==id) \& (data$biomarker %in%
   features)),]

if(unique(data.pat$survival) == 0) { fit <- mem.fits$cv.fit
   [[2]]
} else { fit <- mem.fits$cv.fit[[1]][[unique(data.pat$fold)]]
    }

n <- length(data.pat$id)
Z <- as.matrix(data.pat[covariates.ran])
X <- as.matrix(data.pat[covariates.fix])
dpt <- data.pat$dpt[1:(n/nfeat)]
dpt <- dpt[dpt!=0]
beta <- fixef(fit)
mu.pat <- X %*% beta
Y <- as.numeric(as.matrix(data.pat$value))

# CONDITIONAL CONFIDENCE REGION

mu.marginal <- array(dim=c(nfeat,1,(n/nfeat-1)))
sigma.marginal <- array(dim=c(1*nfeat,1,(n/nfeat-1)))

for(step in 1:(n/nfeat-1)){

  sel2 <- c(sapply(1:nfeat,function(k) max(1, step - span +1)
     :step+(n/nfeat*(k-1))))
  sel1 <- c(sapply(1:nfeat,function(k) (step+1):(step+1)+(n/
     nfeat*(k-1))))

  for(j in 1:length(sel1)){
    mu.step <- mu.pat[c(sel1[j],sel2)]


    sigma1.step <- (Z[c(sel1[j],sel2),] %*% VarCorr(fit)$id)
       %*% t(Z[c(sel1[j],sel2),]) + sigma(fit)^2*diag(dim(Z[c
       (sel1[j],sel2),])[1])


    mu.marginal[j,,step] <-   mu.pat[sel1[j]] +
              sigma1.step[1:length(sel1[j]),(length(sel1[j])
                 +1):(length(sel1[j])+length(sel2))] %*%
              solve(sigma1.step[(length(sel1[j])+1):(length(
                 sel1[j])+length(sel2)),(length(sel1[j])+1):(
                 length(sel1[j])+length(sel2))]) %*%
              (Y[sel2] - mu.pat[sel2])

    sigma.marginal[j,,step] <-  sigma1.step[1:length(sel1[j])
       ,1:length(sel1[j])] +
```

```
                  sigma1.step[1:length(sel1[j]),(length(sel1[j])
                     +1):(length(sel1[j])+length(sel2))] %*%
                  solve(sigma1.step[(length(sel1[j])+1):(length(
                     sel1[j])+length(sel2)),(length(sel1[j])+1):(
                     length(sel1[j])+length(sel2))]) %*%
                  t(t(sigma1.step[1:length(sel1[j]),(length(sel1[
                     j])+1):(length(sel1[j])+length(sel2))]))
  }
}

# PLOT

mu.pat <- matrix(mu.pat, byrow=T, nrow=nfeat)
Y <- matrix(Y, byrow=T, nrow=nfeat)

nf <- layout(matrix(c(1,2,3:(length(features)+2)),ncol=1,
   byrow=F), widths = 5, heights = c(0.65,0.5,rep(1,length(
   features))), TRUE)
par(oma = c(4,1,3,1), mgp = c(2, 1, 0))

par(mar = c(0.1,3,0.1,3),xpd=F)
plot(dpt, pval$list[[id]], ylim = c(0, 1), type = "n", xlab =
    NA, xaxt = "n", ylab = NA, yaxt="n", xlim = c(0, tail(dpt
   ,1)))
mtext("p", side = 2, line= 2.5, cex = 0.8)
axis(2, at = c(0,0.5,1))
abline(h=cutoff, lty=2, col = "grey")
lines(dpt, pval$list[[id]], lty=1)

par(mar=c(0,4.5,0.8,3), xpd=T)
plot(dpt, dpt, type = "n", axes = F, xlab = NA, ylab = NA)
legend("bottom", lty=c(1,1), pch=c(20,NA), col=c("black","
   green"), legend=c("observations", "mean"), horiz=T, bty =
   "n")

for (i in 1:nfeat){
  par(mar = c(0.2,3,0.2,3),xpd=F)
  x.lab=NA
  x.axt <- "n"
  if(i == nfeat){ x.lab <- "dpt"; x.axt <- "l" }
  plot(dpt, dpt, xlab=x.lab, xaxt=x.axt, ylab = NA, type = "n
     ", xlim = c(0, tail(dpt,1)),
    ylim=c(0, max(  mu.pat[i,],
          Y[i,],
          max(sapply(1:dim(mu.marginal)[3], function(x) mu.
             marginal[i,,x]+1.96*sqrt(sigma.marginal[i,,x])))
              )))
```

```
      lines(dpt[2:length(dpt)], mu.marginal[i,,], pch=20, col = "
         grey", type = "p")
      sapply(1:dim(mu.marginal)[3], function(x) add.error.bars(
         dpt[x+1], upper=mu.marginal[i,,x]+1.96*sqrt(sigma.
         marginal[i,,x]), lower=mu.marginal[i,,x]-1.96*sqrt(sigma
         .marginal[i,,x]), width=0.2, col="grey", lty=1))

      lines(dpt, mu.pat[i,], col="green")
      lines(dpt, Y[i,], type="o", pch=20, col = "black")
      mtext( paste(features[i]), side = 2, line = 2.5, cex = 0.8)
    }

    mtext(paste0("Patient ", id, " ( Outcome: ", ifelse(unique(
       data.pat$survival)==1,"survival","nonsurvival"),")"),
       outer=T, line = 2, cex = 0.8)
    mtext(paste0("m", model.number, "/s", span, "/c6"), outer=T,
       line = 1, cex = 0.7)
    mtext("dpt",outer=T,side=1, line=2, cex = 0.8)
  }
}
```

7.7 Evaluation

```
roc_auc <- function(model.number, span, day = 22){

  if(title==T){par(mar=c(5,5,5,5))}

  load(paste0("model_fits/model",model.number,".RData"))
  CMD <- cmd(data, mem.fits, span)
  pval <- assign_pval(CMD, span)
  final.pval <- pval$matrix[day,]

  cutoff <- c(0,unlist(lapply(seq(0,1,0.01), function(x) quantile
     (final.pval[mem.fits$true==1], x))) , 1)
  sens=fpr <- vector(length=length(cutoff))
  for(i in 1:length(cutoff)){
    pred.class <- as.numeric(final.pval > cutoff[i])
    TP <- sum(mem.fits$true==1 & pred.class==1)
    TN <- sum(mem.fits$true==0 & pred.class==0)
    FN <- sum(mem.fits$true==1 & pred.class==0)
    FP <- sum(mem.fits$true==0 & pred.class==1)

    sens[i] <- TP/(TP+FN)
    fpr[i]  <- 1 - TN/(TN+FP)
  }

  return(trapz(rev(fpr), rev(sens)))
```

```
}
```

```
auc_progress <- function(model.number, span){

  load(paste0("model_fits/model",model.number,".RData"))
  CMD <- cmd(data, mem.fits, span)

  #-----------------------------------------------------
  auc_fun <- function(pval, true){

    cutoff <- c(0, unlist(lapply(seq(0,1,0.01), function(x)
       quantile(pval[true==1], x, na.rm=T))) , 1)
    sens=fpr <- vector(length=length(cutoff))
    for(i in 1:length(cutoff)){
      pred.class <- as.numeric(pval > cutoff[i])
      TP <- sum(true==1 & pred.class==1)
      TN <- sum(true==0 & pred.class==0)
      FN <- sum(true==1 & pred.class==0)
      FP <- sum(true==0 & pred.class==1)

      sens[i] <- TP/(TP+FN)
      fpr[i]  <- 1 - TN/(TN+FP)
    }

    return( c("AUC"=trapz(rev(fpr),rev(sens))) )
  }
  #-----------------------------------------------------

  pval <- assign_pval(CMD, span)

  auc.progression <- apply(pval$matrix, 1, function(x) auc_fun(x,
     mem.fits$true))
  return(auc.progression)
}
```

```
confusion_progress <- function(model.number, span, costs){

  load(paste0("model_fits/model",model.number,".RData"))
  load(paste0("cutoffs/optimal.cutoff.costs",costs,".RData"))
  cutoff <- cutoff.opt[model.number,span]
  CMD <- cmd(data, mem.fits, span)
  pval <- assign_pval(CMD, span)

  TP=TN=FP=FN <- vector(length = dim(pval$matrix)[1])
  for(i in 1:dim(pval$matrix)[1]){
    pred.class <- as.numeric(pval$matrix[i,] > cutoff)

    TP[i] <- sum(mem.fits$true==1 & pred.class==1)
    TN[i] <- sum(mem.fits$true==0 & pred.class==0)
```

```
    FN[i] <- sum(mem.fits$true==1 & pred.class==0)
    FP[i] <- sum(mem.fits$true==0 & pred.class==1)
  }

  return( rbind("TP"=TP,"TN"=TN,"FN"=FN,"FP"=FP)
}
```

```
forecast_rates <- function(model.number, span){

  load(paste0("model_fits/model",model.number,".RData"))
  load(paste0("cutoffs/optimal.cutoff.costs",costs,".RData"))
  cutoff <- cutoff.opt[model.number,span]
  CMD <- cmd(data, mem.fits, span)
  pval <- assign_pval(CMD, span)

  # counts length of elements of tial equal to last entry until
     first inequality
  #----------------------------------------
  tail_count <- function(x){
    x <- unlist(x)
    count <- 0
    index <- length(x)
    while( (tail(x,1) == x[index]) && (index > 0) ){ count <-
       count + 1; index <- index -1 }
    return(count)
  }
  #----------------------------------------

  max.length <- 10

  #OVERALL TRUE PREDICTIONS BEFORE ENDPOINT
  detect <- lapply(pval$list, function(x) x > cutoff)
  final.predictions <- lapply(detect, function(x) tail(x,1))
  sel <- names(final.predictions[unlist(final.predictions) ==
     unlist(mem.fits$true)]) # TP & TN Patients
  trues.before.end <- unlist(lapply(detect[sel], tail_count))
  length.obs <- unlist(lapply(pval$list, function(x) length(x)))
  overall.rate <- sapply(1:max.length, function(x) length(which(
     trues.before.end>=x))) / unlist(lapply( 1:max.length,
     function(x) length(which(length.obs>=x)))) )*100


  #TRUE NEGATIVE PREDICTIONS BEFORE ENDPOINT
  TN.sel <- names(which(mem.fits$true == 0))
  detect <- lapply(pval$list[TN.sel], function(x) x > cutoff)
  final.predictions <- lapply(detect, function(x) tail(x,1))
  sel <- names(final.predictions[unlist(final.predictions) ==
     unlist(mem.fits$true[TN.sel])]) # TN Patients
  trues.before.end <- unlist(lapply(detect[sel], tail_count))
```

```
  length.obs <- unlist(lapply(CMD[TN.sel], function(x) length(x))
     )
  tn.rate <- sapply(1:max.length, function(x) length(which(trues.
     before.end>=x))) / unlist(lapply( 1:max.length, function(x)
     length(which(length.obs>=x)))) )*100


  #TRUE POSITIVE PREDICTIONS BEFORE ENDPOINT
  TP.sel <- names(which(mem.fits$true == 1))
  detect <- lapply(pval$list[TP.sel], function(x) x > cutoff)
  final.predictions <- lapply(detect, function(x) tail(x,1))
  sel <- names(final.predictions[unlist(final.predictions) ==
     unlist(mem.fits$true[TP.sel])]) # TP Patients
  trues.before.end <- unlist(lapply(detect[sel], tail_count))
  length.obs <- unlist(lapply(CMD[TP.sel], function(x) length(x))
     )
  tp.rate <- sapply(1:max.length, function(x) length(which(trues.
     before.end>=x))) / unlist(lapply( 1:max.length, function(x)
     length(which(length.obs>=x)))) )*100, type = "o", pch = "o")

  axis(2, at = c(0,25,50,75,100))
  legend("bottomleft", legend=c("TP+TN ", "TN", "TP"), lty=c
     (1,1,1), pch=c(20,4,1), cex=1, bg = "white", horiz = T)

      return( rbind(overall.rate, tn.rate, tp.rate) )
}
```

Bibliography

Bates, D., Mächler, M., Bolker, B., and Walker, C.: *Fitting Linear Mixed-Effects Models using lme4*, https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf, 2014

Bilodeau, M., and Brenner, D.: *Theory of Multivariate Statistics*, Springer, New York, 1999

Diggle, P., Liang, K., and Zeger, S.: *Analysis of Longitudinal Data*, Oxford Science Publications, Oxford, 1994

Hastie, T., Tibshirani, R., and Friedman, J.: *The Elements of Statistical Learning*, Springer, New York, 2009

Morrell, C., Brant, L., Sheng, S., and Metter, E.: *Screening for Prostate Cancer Using Multivariate Mixed-Effects Models*, J Appl Stat., 2012 June 1; 39(6): 1151–1175.

Pimental, M., Clifton, D., Clifton, L., and Tarassenko, L.: *A Review of Novelty Detection*, Signal Processing 99, 2014: 215-249

Verbeke, G., and Molenberghs, G.: *Linear Mixed Models for Longitudinal Data*, Springer, New York, 2000