Machine Learning Methods for Mortality Prediction in Patients with ST Elevation Myocardial Infarction*

Jiří Vomlel
Institute of Information Theory and Automation of the AS CR
Pod vodárenskou věží 4, 182 08, Prague, Czech Republic
and
Faculty of Management, University of Economics
Jarošovská 1117/II, 377 01, Jindřichův Hradec
e-mail: vomlel@utia.cas.cz

Hynek Kružík and **Petr Tůma**
Gnomon, s.r.o.
Faltysova 1500/18, 156 00, Prague – Zbraslav, Czech Republic
kruzik@gnomon.cz, tuma@gnomon.cz

Jan Přeček and **Martin Hutyra**
1st Department of Internal Medicine, University Hospital Olomouc
I. P. Pavlova 6, 779 00 Olomouc, Czech Republic
jan.precek@seznam.cz, martinhutyra@seznam.cz

Abstract

ST Elevation Myocardial Infarction (STEMI) is the leading cause of death in developed countries. The objective of our research is to design and verify a predictive model of hospital mortality in STEMI based on clinical data about patients that could serve as a benchmark for evaluation of healthcare providers. In this paper we present results of an experimental evaluation of different machine learning methods on a real data about 603 patients from University Hospital in Olomouc.

1 Introduction

In developed countries ST Elevation Myocardial Infarction (STEMI) is responsible for more than a half of deaths. Its treatment has a significant socio-economic

*Our research was supported by the Czech Science Foundation through grant nr. 201/08/0539.

impact. The main objective of our research is to design and verify a predictive model of hospital mortality in STEMI based on clinical data about patients available at the beginning of their hospitalization. This model can be used not only as a decision support tool that supports medical decisions about patients' treatments but also as a benchmark for evaluation of healthcare providers, which is our main motivation for the research reported in this paper.

The motivation for this type of benchmarking is that mere mortality does not reflect severity of the illness at the hospital admission. There are hospitals that more often treat complicated cases and mere mortality would not be fair to them. Therefore the mortality should be risk adjusted. For this purpose a good model describing influence of risk factors on the mortality is needed.

In this paper we will present the results of our experimental evaluation of different machine learning methods on a real data from University Hospital in Olomouc.

2 Dataset of patients with STEMI

Our dataset contains data of 603 patients admitted to University Hospital in Olomouc for STEMI. The average age was 65 years. There were 431 men (71%) and 172 women (29%) in the dataset. Our goal is to classify patients into two classes according to whether they survive 30 days or not. This criteria is called *30-days mortality* [8]. The value 0 will correspond to survival while the value 1 to non-survival. Since the intended use of a constructed classifier is the evaluation of healthcare quality we use only information about patients' health state at the time of their hospital admission. In data we have 23 attributes of different types and value range. They were selected by cardiologists since they may influence STEMI mortality. The attributes are listed in Table 1. In the first group there are basic demographic characteristics and body measurements. The attributes of the second group describe the location and the mortality risk of STEMI. The third group consists of laboratory tests.

Some attribute values are missing for some patients. In total 3.2% of values are missing. As it can be seen from Table 1 the attributes are of different types by their nature. Some classification methods can handle certain types of attributes only and thus require a transformation of attributes' values.

2.1 Ordinal attributes

Ordinal attributes are attributes whose values have an ordering of values that is natural for the quantification of their impact on the class. This is satisfied by all attributes that can take only two values – even if they are nominal, e.g. by Gender[1]. In our data it seems it can be assumed for most real-valued attributes, but note that there might exist laboratory tests whose values deviating from a normal range in both directions (i.e. both lower and higher values) may increase

[1]For this purpose we encode Gender using two numbers: 0 for male and 1 for female.

Table 1: Attributes

Attribute	Code	type	value range in data
Gender	SEX	nominal	{male, female}
Age	AGE	real	$[23, 94]$
Height	HT	real	$[145, 205]$
Weight	WT	real	$[35, 150]$
Body Mass Index	BMI	real	$[16.65, 48.98]$
STEMI Location	STEMI	nominal	{inferior, anterior, lateral}
Killip classification at admission	KILLIP	integer	$\{1, 2, 3, 4\}$
Kalium	K	real	$[2.25, 7.07]$
Urea	UR	real	$[1.6, 46.5]$
Kreatinin	KREA	real	$[17, 525]$
Uric acid	KM	real	$[109, 935]$
Albumin	ALB	real	$[23, 53.5]$
HDL Cholesterol	HDLC	real	$[0.38, 2.21]$
Cholesterol	CH	real	$[1.8, 9.59]$
Triacylglycerol	TAG	real	$[0.31, 8.13]$
LDL Cholesterol	LDLC	real	$[0.63, 7.79]$
Glucose	GLU	real	$[4.2, 25.7]$
C-reactive protein	CRP	real	$[0.3, 359]$
Cystatin C	CYSC	real	$[0.38, 5.22]$
N-terminal prohormone of brain natriuretic peptide	NTBNP	real	$[22.2, 35000]$
Troponin	TRPT	real	$[0, 25]$
Glomerular filtration rate (based on MDRD)	GFMD	real	$[0.13, 7.31]$
Glomerular filtration rate (based on Cystatin C)	GFCD	real	$[0.09, 7.17]$

the probability of death[2]. However, there is no natural ordering of the values of the nominal attribute STEMI since its values are locations. Fortunately, this problem can be simply overcame by creating one binary attribute for each state of STEMI indicating whether STEMI takes this state or not. We denote new binary attributes as STEMI_inferior, STEMI_anterior, and STEMI_lateral. We will refer to data in this form as D.ORD.

[2]In order to allow modeling this type of influence we will transform such attributes into two attributes. We will discuss this in the next subsection.

2.2 Discrete attributes

Some classification methods require a finite number of values of each attribute – i.e., discrete attributes. In order to get statistically reliable estimation the number of values should be as low as possible (and sensible). The transformation of a real-valued attribute into an attribute with finitely many values is called discretization. We performed discretization of all real-valued attributes. We used different number of values depending on the nature of each attribute. Generally, it is difficult to find the optimal number and the values of split points in discretization. Fortunately, there exists the Czech National Code Book that classifies numeric laboratory results, with respect to age and gender, into nine groups $1, 2, \ldots, 9$. Group 5 corresponds to standard values in the standard population. The groups < 5 to decreased values and groups > 5 to increased values. We discretized all laboratory tests X so that for each test we created two new attributes:

- One attribute for a decreased value of the test – denoted X_low – with state 0 if the value is within the normal range. Values $1, 2, 3, 4$ became values of this attribute.
- Another attribute for the increased value of the test – denoted X_high – again with state 0 if the value is within the normal range. Values $6, 7, 8, 9$ became values of this attribute.

The attributes Age, Height, and Weight were discretized into more than two groups (10, 4, and 4, respectively). We will refer to data in this form as D.DISCR.

2.3 Binary attributes

However, as we will see in Section 4 the performance of tested methods using discretization described in Section 2.2 was inferior to discretization to only binary attributes, where all laboratory tests are encoded using two binary attributes. The first attribute takes value 0 for the standard values of the test and value 1 if the values are decreased. The second attribute takes value 0 for the standard values of the test and value 1 if the values are increased. The attribute Killip classification was transformed by replacing value 1 by 0 and by joining the values $2, 3, 4$ into one value 1. The attributes Age, Height, and Weight were removed since they appeared not to be relevant for mortality. From the demographic group of attributes only Gender and the Body Mass Index (BMI) were kept with BMI being encoded using two binary attributes BMI_high and BMI_low. We will refer to data in this form as D.BIN.

2.4 Attribute selection

When learning classifiers from datasets we used every dataset in two different ways:

- all attributes were included or
- only attributes selected by the attribute selection method CfsSubsetEval from Weka [6] were included.

CfsSubsetsEval method [5] selects a subsets of attributes that are highly correlated with the class while having low intercorrelation. We searched the space of all subsets by a greedy best first search with backtracking. Data D after the application of this attribute selection method will be suffixed as D.AS.

3 Tested classifiers

For tests we used a large subset of classifiers implemented in Weka [6]. Classifiers that performed best in the preliminary tests qualified for the final tests. In the final tests we compared following classifiers:

- Logistic regression (two versions):

  LOG.REG – logistic regression model with a ridge estimator [10].

  LOG.BOOST – LogitBoost with simple regression functions as base learners used for fitting the logistic models [9].

- Decision tree C4.5 – pruned C4.5 decision tree [11].
- Naive Bayes classifier (two versions):

  NB.SIMPL – Naive Bayes classifier which estimates Gaussian distribution when learned from real-valued (numeric) attributes [3].

  NB – Naive Bayes classifier which also uses estimator classes. Numeric estimator precision values are chosen based on analysis of the training data [7].

- NN – Artificial Neural Network Multilayer Perceptron. The nodes in this network model sigmoid functions [2].
- Bayesian network classifier (two versions):

  BN.K2 – Bayesian Network classifier learned by K2 algorithm [1] (with unrestricted number of parents).

  BN.TAN – Tree Augmented Naive Bayes classifier [4].

4 Results of experiments

We compared the classifiers using Weka [6]. We used the 10-fold cross-validation methods. The results are summarized in Table 2 using the following two measures of prediction quality:

- Accuracy (ACC), which is the number of true positive and true negative classification divided by total number of classifications. It is reported using percentage scale (i.e. multiplied by 100).
- Area under the ROC curve (AOC). The ROC curve depicts the dependence of True Positive Rate (vertical axis) on False Positive Rate (horizontal axis) both as functions of the threshold.

In Table 2 we can observe several interesting things:

First, if we compare results of a single classifier on different versions of data, we can see that the best results are mostly achieved with D.BIN.AS, i.e. with discretized data, where each attribute is binary. This observation confirms the general recommendation that if the number of data records is not large then the discretization should not be fine-grained. We were able to improve the classifiers' performance due to a good discretization of original ordinal data based on expert knowledge of the domains of attributes.

Secondly, attribute selection methods also helped to improved performance. Originally, we did't have large number of attributes since we started with only 23 attributes. But the performance of most classifiers improved if only few of the most relevant attributes were included. This also confirms the general recommendation that in order to avoid overfitting of training data the models should be as simple as possible.

Finally, when comparing different classifiers we can see that there is not big difference between their accuracy. Actually, the high accuracy could be achieved by a primitive classifier that would assign all instance to class 0, i.e. all patients would survive 30 days. Its accuracy would be 94.03%, which is the relative number of patients that survive STEMI in our data. However, its AUC would be very low, only 0.465. Therefore we prefer classifiers that maximize both criteria at the same time. From this point of view the classifiers C4.5 and NN seem inferior to LOG, NB, and BN families. There are not huge differences between later three families, but if we should choose two best performing classifiers it would be LOG.BOOST and BN.TAN that have the best AUC and ACC from all classifiers, respectively.

Next we will present our choice of the best performing classifiers in more detail. In Figure 1 we compare LOG.BOOST for two versions of data – original ordinal and binarized data. Both formulas are for logit of probability of Mortality=1. Although there are some similarities between these two classifiers they are not exactly the same. Note that splitting laboratory tests ALB and CYSC into two attributes ALB_low, ALB_high and CYSC_low and CYSC_high helps to make explicit the impact of low values of ALB and high values of CYSC on the mortality. Also note that while in the first formula KILLIP takes values $1, 2, 3, 4$ in the second one it is only 0 (corresponding to the original 1) and 1 corresponding to the original $2, 3, 4$. Albeit the second model is simpler it has substantially higher value of AUC. Actually, according to AUC it is the best performing classifier.

The AUC values of C4.5 classifiers were quite low. However, it is interesting to see that the C4.5 for binarized data despite its extreme simplicity has quite

Table 2: Results of experiments

Classifier	Criteria	D.ORD	D.ORD.AS	D.DISCR	D.DISCR.AS	D.BIN	D.BIN.AS
LOG.BOOST	ACC	94.03	94.20	93.86	88.23	94.03	93.86
	AUC	0.618	0.646	0.722	0.640	0.802	**0.832**
LOG.REG	ACC	92.54	93.86	90.05	87.56	92.87	93.70
	AUC	0.792	0.821	0.646	0.607	0.743	0.798
C4.5	ACC	93.86	94.69	94.20	88.72	93.53	94.53
	AUC	0.618	0.569	0.600	0.544	0.547	0.610
NB	ACC	89.22	91.04	86.90	87.73	86.90	94.20
	AUC	0.820	0.813	0.806	0.649	0.811	0.809
NB.SIMPL	ACC	89.72	90.88	86.90	87.73	86.90	94.20
	AUC	0.828	0.769	0.806	0.649	0.811	0.809
NN	ACC	91.38	93.86	93.20	87.40	92.04	93.53
	AUC	0.763	0.746	0.737	0.550	0.767	0.759
BN.K2	ACC	NA	NA	92.04	94.53	94.03	94.36
	AUC	NA	NA	0.769	0.783	0.769	0.821
BN.TAN	ACC	NA	NA	92.04	88.89	94.20	**94.86**
	AUC	NA	NA	0.787	0.590	0.811	0.818

```
0.87   + STEMI_lateral * -0.41   + ALB * -0.08
       + HDLC * 0.21   + CYSC * 0.24 + KILLIP * 0.31

-1.64  + ALB_low * 0.76  + CYSC_high * 0.62  + KILLIP * 0.68
```

Figure 1: LOG.BOOST for D.ORD.AS (up) and D.BIN.AS (down).

```
CYSC <= 1.64: 0 (553.0)
CYSC > 1.64
|   HDLC <= 0.56: 1 (5.0)
|   HDLC > 0.56
|   |   KILLIP <= 1
|   |   |   ALB <= 25.2: 1 (2.21)
|   |   |   ALB > 25.2: 0 (29.79)
|   |   KILLIP > 1
|   |   |   UR <= 15.8: 1 (6.0)
|   |   |   UR > 15.8: 0 (7.0)


CYSC_high = 0: 0 (526.0)
CYSC_high = 1
|   ALB_low = 0: 0 (63.29)
|   ALB_low = 1: 1 (13.71)
```

Figure 2: C4.5 for D.ORD.AS (up) and for D.BIN.AS (down).

good accuracy ACC and performs actually better than more complex C4.5 build from ordinal data. See Figure 2. In each leaf the first number after colon is the classification. The number in parenthesis is the total number of instances reaching that leaf (since our data has missing attribute values we got decimal numbers).

Finally, we add a comment on two Bayesian network classifiers. In Figure 3 we compare Tree Augmented Naive Naive Bayes classifier (up) and Bayesian Network classifier learned by K2. Despite the BN learned by K2 was allowed to have more parents of each attributes than TAN[3] it finally contains less edges (only four edges between attributes) and its performance is comparable with BN.TAN.

5 Conclusions

In this paper we compare different machine learning methods using a real medical data from a hospital. The best performance was achieved on discretized data

[3]In TAN each but one attribute has exactly two parents, the class and one other attribute.

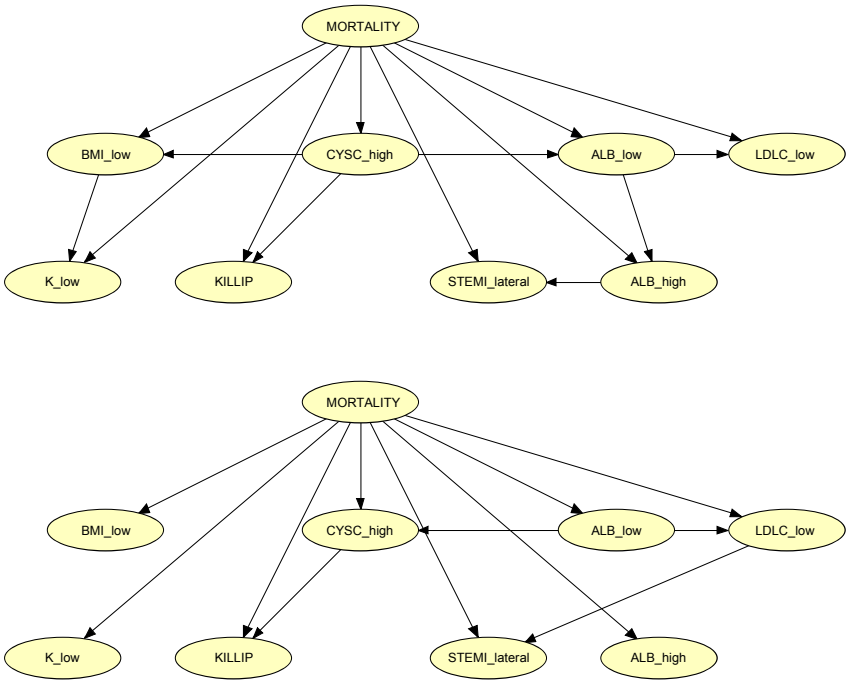

Figure 3: Tree Augmented Naive Bayes classifier (up) and Bayesian Network classifier learned by K2 (down). Both were learned from D.BIN.AS.

where the discretization was based on the expert knowledge about the attributes (mostly on standard scale of results of laboratory tests) and the attributes had only two values. The best performing classifiers were based on logistic regression and on simple Bayesian networks. In our future research we would like to extend the set of attributes with other clinical data and get datasets with a larger number of patients.

Acknowledgments

This work would not be so easy without the Weka system. We are grateful to the authors of Weka for making their tool freely available.

References

[1] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[2] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[3] Richard Duda and Peter Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[4] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3):131–163, 1997.

[5] Mark Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1999.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[7] George H. John and Pat Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[8] Harlan M. Krumholz, Sharon-Lise T. Normand, Deron H. Galusha, Jennifer A. Mattera, Amy S. Rich, Yongfei Wang, and Yun Wang. Risk-adjustment models for AMI and HF 30-day mortality, methodology. Technical report, Harvard Medical School, Department of Health Care Policy, 2007.

[9] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59(1–2):161–205, 2005.

[10] S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[11] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.