Nomograms for Visualizing Linear Support Vector Machines

Aleks Jakulin Martin Možina Janez Demšar Ivan Bratko
Blaž Zupan
Faculty of Computer and Information Science, University of Ljubljana
Tržaška cesta 25, SI-1001 Ljubljana, Slovenia
jakulin@acm.org

Abstract

Support vector machines are often considered to be black box learning algorithms. We show that for linear kernels it is possible to open this box and visually depict the content of the SVM classifier in high-dimensional space in the interactive format of a nomogram. We provide a cross-calibration method for obtaining probabilistic predictions from any SVM classifier, which control for the generalization error. If we employ logistic regression for calibration, the effect of each attribute can be represented on the log odds ratio scale. We also describe an approach to capturing nonlinear effects of continuous attributes with an ordinary linear kernel, and adapt the nomogram so that these nonlinear effects can be graphically rendered.

1. Introduction

Within predictive data mining, methods that build classification models have received much attention recently. These methods consider a set of class-labelled data instances and induce classification models that should both predict well and, preferably and through the model inspection, can uncover interesting relations and patterns. The latter is particularly important when predictive data mining is used for knowledge discovery, where presentation of the classification model should help the user to answer questions such as "Which are the most important factors that determine the class of the instance?", and "What is the magnitude of the effect of these?", and "How do various factors interact?", and alike.

A support vector machine [18] (SVM) is a recently very popular and much applied supervised machine learning method. It is known for good predictive performance, but may be at a disadvantage in terms of intuitive presentation of the classifier, particularly when compared to some other supervised learning techniques like classification trees and rules. While an SVM model can be presented as a weighted list of support vectors, e.g., a subset of learning instances that defines the decision boundary, this only reduces the number of instances to consider in the interpretation but does not answer any of the questions posed above directly.

In the paper, we propose a new approach for visualization of SVM models. The main advantage of our approach is that it captures a complete classification model in a single, easy-to-interpret graph and provides means to easily study the effects of predictive factors. The approach is limited to SVM models with linear kernel functions, which, despite the simplicity when compared to those that use more complex kernels, have shown useful in a number of practical applications.

The particular model visualization we use is called a *nomogram*. Nomograms were invented by French mathematician Maurice d'Ocagne in 1891 to graphically represent a class of mathematical functions. To visualize a logistic regression model, the use of nomograms was first proposed by Lubsen and coauthors [14]. With an excellent implementation of logistic regression nomograms in S-Plus and R statistical packages by Harrell [5], the idea has recently been picked up and the nomograms have been used much to present probabilistic classification models in, for instance, clinical medicine and oncology (e.g., [12]; see also http://www.baylorcme.org/nomogram/modules.cfm).

The nomograms for support vector machines that we introduce in the paper use a similar presentation as those of Harrell for logistic regression. To illustrate the general idea, consider the nomogram in Figure 1 which represents a linear SVM model induced from the Boston Housing data set (StatLib, http://lib.stat.cmu.edu/datasets/, also see [6]). The Housing data set consists of 506 different instances (areas), where about 50% of them have median value of housing price lower than $21000. For convenience of this presentation we use only four representative attributes: the average number of rooms per dwelling (*Rooms*), weighted distances to five Boston employment centers (*Employment_dis*), pupil-teacher ratio by town (*Pupil-teacher*, dis-

cretized to two nominal values), and proportion of lower status population (*Low_status*, discretized to four nominal values). We induced an SVM model with a linear kernel for classifying the median value into two groups: the expensive areas with the median values above $21000, and the cheap areas. Furthermore, we employed cross-calibration (described in Section 2.2) based on univariate logistic regression to obtain probabilistic estimates of the classes.

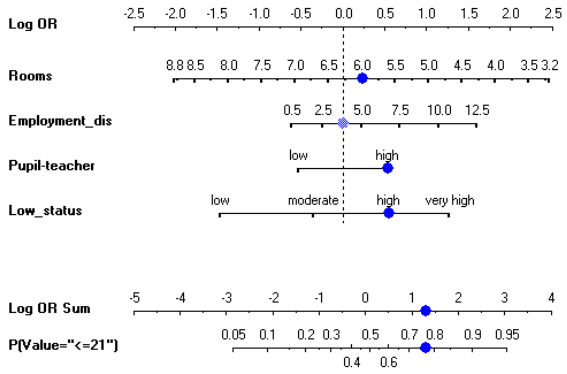

Figure 1. A nomogram of the SVM model that predicts the probability of costly housing in a given Boston area. The probability estimate for a specific instance is indicated by dots.

To make a prediction using a nomogram, the contributions of attributes on the scale of the log odds ratios [10] (topmost axis of the nomogram), are summed up and used to determine the probability whether price is less than $21000 (bottommost axis of the nomogram). For instance, an area with 6 rooms per average dwelling, where the distance from employment centers is unknown, with a high pupil-teacher ratio and a high rate of lower status population, the overall sum of contributions is $0.21 + 0.00 + 0.49 + 0.5 = 1.20$. This sum is then projected from the 'Log OR Sum' axis to the bottommost 'P($<= 21$)' probability axis, where the final probability of the target class is approximately 0.76. On the other hand, if the town was known to be far away from employment centers (12.5), *Employment_dis* contribution to final sum would be around 1.5 instead of 0. Accounting for this change in the overall sum, the final probability would be higher than 0.93.

Besides prediction, nomograms provide a clear and comprehensive presentation of the underlying model. Our SVM nomogram from Fig. 1, for instance, clearly exposes that housing values in Boston from a particular data set are most associated with the average number of rooms. The corresponding line in the nomogram is the longest, and trying to predict housing values for a certain area simply with the information that the average number of rooms is small (3.2), the probability for price under $21000 jumps from a priori 0.5 to over 0.9 a posteriori. The other three attributes carry less importance, especially the pupil-teacher ratio. Our nomogram also exposes how different attribute values affect the outcome; for instance, the value of housing goes up when the employment centers are close. Note that we can include continuous as well as discrete attributes in the nomogram. The nomogram also clearly exposes the "neutral" values of the attributes, e.g. values which do not affect the probability of the outcome from the prior. If a particular attribute value is not given for the test instance, those neutral values will be effectively imputed.

Nomograms – like the one from our example – are used to assess the probability of the observed outcome, where the effects of the attributes are independent given the class and are added up to form the final prediction. For some instance $\boldsymbol{i}$, described by a set of attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, and a label y, the nomogram can visualize a probability function of the type

$$\hat{P}(y|\boldsymbol{i}) = F\Big(b + \sum_j f_j(a_j(\boldsymbol{i}))\Big) \qquad (1)$$

where b is a constant delineating the prior probability in the absence of any attributes, f_j is an *effect* function that maps the value of an attribute a_j into a point score, and F is a function that maps the *response* to the instance into the outcome probability. Notice that the class of models based on the above model type are the generalized additive models (GAM, [7, 8]).

We start our paper by showing how certain support vector machines can be decomposed into the above additive model. To enable the use of the nomograms for support vector machines, we need them to predict outcome probabilities. The basic SVM alone does not attempt to model the probability, just the distance of an instance to a separating hyperplane in the instance space, each side of which represents a different class. Therefore, the effect functions need to be calibrated and thus placed on the log odds ratio scale, which is used by the nomogram representation. The benefits of this visualization technique and its application on several data sets are given in the discussion. Beyond the study of SVM models, we also show that nomograms are suitable when these are compared to other generalized additive models, such as the naïve Bayesian classifier. In the discussion, we also experimentally assess our underlying assumption that distance to the separating hyperplane can provide a good estimate of outcome probabilities.

2. Methodology

2.1. Obtaining the Effect Functions from an SVM

Not every support vector machine is appropriate for visualization using a nomogram. For that purpose, we will describe a restricted formulation. The first restriction is based

on the ability to additively separate the individual contribution of each attribute towards the response as in (1). This is made feasible by using a linear kernel, based on the dot product. The second restriction is linked to how we represent the lack of information: ideally, zero value of the transformed attribute should indicate the lack of information about an attribute.

For support vector machines, as with logistic regression, all the attributes need to be transformed into real-valued variables before a model can be trained. Support vector machines are not disturbed neither by attribute spaces of high dimensionality, nor by collinear or coplanar placement of instances. We standardize continuous attributes so that zero implies the mean, and ± 1 implies one standard deviation distance from the mean. A K-valued discrete attribute a is transformed into a set of k variables $x_1, x_2, \ldots, x_K$, so that given the value of $a = v$, $x_v = 1$ and $x_i = 0, v \neq i$. This way, the transformation function can represent each attribute value with its own dimension and also provides ground to handle the missing values by setting all corresponding x_i to zero. Altogether, the transformation will be captured by the *transformation* function $t(\boldsymbol{i})$. The range of the label is usually $\Re_y = \{-1, 1\}$.

Given N training instances $\boldsymbol{x}_j$, $j = 1, 2, \ldots, N$, the resulting support vector model can be described with a weight vector $\boldsymbol{\alpha}$ and the bias b. We will not describe the actual learning procedure and its criteria, which are better described elsewhere, e.g., [18]. The response $\delta(\boldsymbol{i})$ for an instance, given a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be described as:

$$\delta(\boldsymbol{i}) = b + \sum_{j=1}^{N} y(\boldsymbol{x}_j)\boldsymbol{\alpha}_i K(t(\boldsymbol{i}), \boldsymbol{x}_j) \tag{2}$$

Here, $\hat{P}(y|\boldsymbol{i}) = F(\delta(\boldsymbol{i}))$. While b in (2) and (1) correspond exactly, the effect functions need to be calculated. If the kernel function K is a dot product, we can decompose this equation into the form of (1). Assume each continuous attribute a_l and each *value* of a discrete attribute a_l correspond to the k-th component of the transformed instance $x = t(\boldsymbol{i})_k$. Thereby, for a continuous attribute a_l, x will indicate the standardized value of a_l. For a discrete attribute a_l, x_i will take the value of 0 or 1, depending on whether $a_l = i$. The effect function in all cases is simply linear $f_k(x) = \beta_k x$, and the factor is calculated as:

$$\beta_k = \sum_{j=1}^{N} y(\boldsymbol{x}_j)\boldsymbol{\alpha}_j \boldsymbol{x}_{j_k} \tag{3}$$

From this, it is easy to see that $\delta(\boldsymbol{i}) = b + \sum_k \beta_k t(\boldsymbol{i})_k$.

2.2. Cross-Calibration

In general, the response $\delta(\boldsymbol{i})$ to an instance in SVM is the (signed) distance of instance to the bounding hyperplane. Obtaining the function that maps the label posterior probability into a response is a requirement for using the nomogram-based visualization of a model, as the scale of the visualized effect functions is based on probability. While the link function is simply the logit transform for logistic regression, there is no direct mapping for SVM, but there are methods for performing this transformation. Instead of special-purpose algorithms such as [17], we can interpret the task of obtaining the probability of a particular instance's label given the instance's response simply as a *calibration* learning problem considering the response as the single continuous attribute in this problem. In that sense, all generalized additive models can be seen as constructive induction methods that yield a single continuous attribute useful for predicting the label. In the work we report on here, we have applied the univariate logistic regression.

It is also possible to formulate the learning problem so that the error arising from generalization is accounted for. Namely, a classifier might achieve perfect separation on the training set, but not on a separate test set. One can moderate the predictions by using Bayesian priors or regularization, but a particularly simple and powerful approach is based on an analogy with the wrapper approach [13]. Therefore, if we calibrate on data that was not used for training the response, we capture the uncertainty associated with generalization to unseen data. We can perform this procedure for all learning algorithms, but care must be taken to prevent the arbitrariness of the response function range. For that reason, we use the training data to find a scalar τ so that $\max_{\boldsymbol{i}} \tau\delta(\boldsymbol{i}) - \min_{\boldsymbol{i}} \tau\delta(\boldsymbol{i}) = 1$.

There are two parameters to such a calibration procedure. The first parameter is the data hiding protocol used for separating training from test data. For example, for 10-fold cross-calibration, 90% of the data is used for training and 10% remains hidden for calibration. The more data we hide, the more conservative are our predictions. The second parameter is the number of replications. A single cross-calibration depends on a particular shuffling of instances. To remove this dependence, the cross-calibration procedure should be replicated as many times as it is practical.

If we use univariate logistic regression as the calibration learning algorithm C, the end result can be represented as $\hat{P}(y(\boldsymbol{i}) = 1|\delta(\boldsymbol{i})) = (1 + \exp(b' + \beta'\tau\delta(\boldsymbol{i})))^{-1}$, where b', β' and τ represent the maximum likelihood logistic regression model learned by calibration. The inverse link function is here defined as $F(\delta') = (1 + \exp(\delta'))^{-1}$, and the calibrated response function on the log odds ratio scale is $\delta'(\boldsymbol{i}) = b' + \beta'\tau\delta(\boldsymbol{i})$. It is then simple to transform the effect functions and the bias in (3), so that they match the logistic regression coefficients precisely. This way we obtain the zero threshold $\hat{b}$ which marks both the outcome probability of 0.5 and the log odds ratio of 0.0, and the $\hat{\beta}_k$ which indicates the effect of a particular nominal attribute value, or

$\mathcal{R} \leftarrow \emptyset$ {Calibration training set.}
for all $r : 1 \leq r \leq R$ **do** {for each replication}
  $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_X \leftarrow \mathcal{T}$ {Generate folds.}
  for all $x : 1 \leq x \leq X$ **do** {for each fold}
    $\hat{\delta} \leftarrow L\left(\bigcup_{i \neq x} \mathcal{F}_i\right)$ {Train.}
    $\hat{\tau} \leftarrow \left(\max_{\boldsymbol{i},\boldsymbol{j} \in \mathcal{T}} (\hat{\delta}(\boldsymbol{i}) - \hat{\delta}(\boldsymbol{j}))\right)^{-1}$ {Range.}
    for all $\boldsymbol{i} \in \mathcal{F}_x$ **do** {for each test instance}
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{\langle \hat{\tau}\hat{\delta}(\boldsymbol{i}), y(\boldsymbol{i})\rangle\}$ {Record the response.}
    end for
  end for
end for
$\delta \leftarrow L(\mathcal{T})$ {Response function.}
$\tau \leftarrow (\max_{\boldsymbol{i},\boldsymbol{j} \in \mathcal{T}} (\delta(\boldsymbol{i}) - \delta(\boldsymbol{j})))^{-1}$ {Range.}
$\hat{P}(y(\boldsymbol{i}) = 1 | \tau\delta(\boldsymbol{i})) \leftarrow C(\mathcal{R})$ {Probability function.}

Algorithm 1: A general scheme of a cross-calibration procedure, based on X folds, R replications, the response learning algorithm L, the calibration learning algorithm C, and the training data $\mathcal{T}$.

the change in the effect of a particular continuous attribute when it increases by 1:

$$\hat{b} = b' + \beta' \tau b \tag{4}$$

$$\hat{\beta}_k = \beta' \tau \beta_k \tag{5}$$

Both $\hat{b}$ and $\hat{\beta}_k$ are now on log odds ratio scale, and it is obvious how they can be visually presented in a nomogram.

2.3. Nonlinear Effects for Continuous Attributes

The above learning algorithms are primarily appropriate for nominal attributes. Both logistic regression and linear SVM have linear effect functions, and are therefore generalized linear models for continuous attributes. Unfortunately, in real-life data many attributes have non-linear effects on the outcome. For example, trying to predict health status from body temperature, both high and low temperatures indicate trouble, and this pattern cannot be captured by a single real-valued variable. Non-linear kernels in SVM can capture these effects, but we cannot use nomograms for visualizing them. Instead, we can allow for the nonlinear effect function of a single attribute, but using ordinary linear SVM. A general solution that applies to all these methods is *discretization*. If a continuous attribute is converted into a nominal attribute based on intervals, we obtain an extremely simple method for handling nonlinear effects. The knowledge that a particular nominal attribute is based on a continuous one can be profitably employed in the presentation of the results.

3. Experiments

In this section, we address two questions. The first one is on performance of support vector machines with linear kernels and with probability estimation and calibration as proposed in this paper. To address this, we present an experimental analysis and compare the nomogram-based probability estimations with those obtained from SVM with RBF kernel (Did we lose anything assuming the linearity?) and two popular methods for probabilistic classification, namely logistic regression and naïve Bayesian classifier (What is the overall performance in class probability prediction?). The second question addresses the utility of visualization, and we show that besides revealing the structure of the SVM classifier, nomograms may well be used as a data mining tool to depict different properties of problem domains. Also, as applicable to other generalized linear models, nomograms may be used to study the differences between various modelling methods. For the later, we present a nomogram-based comparison of a linear-kernelled SVM and the naïve Bayesian classifier model.

3.1. Classification Performance

All experiments were performed within the Orange toolkit [4]. We employed LIBSVM [2] with default settings for training the SVM classifiers, and iteratively re-weighted least squares fitting [15] of the logistic regression model, as implemented in the Orange extensions package [11]. We experimented on 16 well-known UCI [9] data sets with a binary outcome. For data sets with more than 1000 examples ('mushroom' and 'spam base') we have selected a stratified random subset of 1000 examples which were used throughout the experiments.

We evaluated each method on three criteria: classification accuracy, outcome probability estimation (as measured by Brier score, the mean square error of predicted class probabilities given the true class probabilities for each instance [1]), and instance ranking with respect to the outcome (as measured by the area under the receiver operating characteristic). Table 1 compares the naïve Bayesian classifier (NB), logistic regression (LR), support vector machines with RBF kernels (SVM), and support vector machines with a linear kernel (dot and dot') on each of these three criteria. The first six data sets (the upper part of the table) include no continuous attributes. Elsewhere, the continuous attributes were discretized for NB and dot' into 10 intervals with approximately equal number of examples for each value, as to provide the capacity for handling nonlinear effects. In computation of the Brier score, the predicted probabilities were calibrated for all methods, except for logistic regression (which is considered not to require calibration). Note that Brier score measures the loss, so lower values are better

	Classification accuracy					Brier score					Area under ROC				
	NB	LR	RBF	dot	dot'	NB	LR	RBF	dot	dot'	NB	LR	RBF	dot	dot'
breast (lju)	0.73	0.70	0.73	0.69		0.40	0.42	0.38	0.39		0.70	0.67	0.56	0.58	
breast (wsc)	0.97	0.93	0.98	0.96		0.05	0.15	0.05	0.06		0.98	0.91	0.98	0.96	
mushroom	1.00	1.00	0.99	1.00		0.02	0.01	0.03	0.01		1.00	0.99	0.99	1.00	
shuttle	0.93	0.99	0.93	0.96		0.08	0.02	0.11	0.10		1.00	0.99	0.94	0.96	
titanic	0.78	0.78	0.79	0.78		0.33	0.33	0.32	0.35		0.71	0.76	0.68	0.70	
voting	0.90	0.96	0.95	0.95		0.13	0.06	0.07	0.07		0.97	0.99	0.96	0.95	
australian	0.86	0.85	0.86	0.85	0.84	0.21	0.30	0.22	0.24	0.24	0.92	0.85	0.86	0.86	0.85
german	0.77	0.76	0.73	0.76	0.76	0.33	0.33	0.36	0.34	0.34	0.79	0.79	0.59	0.68	0.69
hepatitis	0.86	0.83	0.84	0.85	0.83	0.21	0.26	0.23	0.24	0.27	0.86	0.85	0.70	0.76	0.71
horse-colic	0.79	0.82	0.82	0.82	0.79	0.32	0.29	0.26	0.28	0.30	0.81	0.86	0.80	0.80	0.77
housing	0.81	0.86	0.87	0.86	0.83	0.27	0.19	0.19	0.22	0.25	0.89	0.94	0.87	0.86	0.83
ionosphere	0.91	0.83	0.94	0.81	0.90	0.15	0.26	0.13	0.31	0.17	0.92	0.84	0.92	0.77	0.89
liver	0.65	0.69	0.71	0.68	0.73	0.43	0.42	0.41	0.44	0.41	0.69	0.72	0.68	0.66	0.70
pima	0.75	0.78	0.76	0.78	0.75	0.32	0.31	0.34	0.33	0.35	0.83	0.83	0.70	0.73	0.72
post-op	0.66	0.68	0.73	0.70	0.69	0.40	0.49	0.39	0.40	0.39	0.41	0.36	0.50	0.48	0.48
spam base	0.91	0.91	0.91	0.92	0.92	0.16	0.19	0.14	0.20	0.12	0.94	0.89	0.90	0.91	0.91
avg rank (cont)	3.3	3.5	2.2	2.6	3.4	2.8	3.1	2.3	3.7	3.1	1.9	2.6	3.4	3.4	3.7
	$F = 1.24, p = 0.31$					$F = 1.04, p = 0.40$					$F = 2.51, p = 0.06$				
avg rank (all)	2.7	2.7	2.0	2.6		2.5	2.5	2.0	3.0		1.6	2.3	3.1	3.1	
	$F = 0.95, p = 0.42$					$F = 1.67, p = 0.19$					$F = 6.19, p = 0.00$				

Table 1. Comparison of the naïve Bayesian classifier (NB), logistic regression (LR), SVM with the RBF kernel (RBF), SVM with the linear kernel (dot) and linear SVM with discretization (dot') on several UCI data sets.

than higher.

The observed methods perform similarly, with few exceptions. For instance, linear SVM performs poorly on 'ionosphere' unless the attributes are discretized, apparently indicating non-linear attribute effects (which will be studied further in Sect. 3.2). The SVM using the RBF kernel captures this nonlinearity better than any method based on discretization. An unexpectedly good performer is the naïve Bayesian classifier, which achieved very good probability estimation results, probably helped by calibration, and the best ranking results.

Since our paper shows how to visualize SVM with linear kernels, it is of interest how much performance needs to be given up by not using the more powerful RBF kernels. As expected, SVM with RBF kernels generally performs best of all methods in classification and probability estimation, but not in ranking. Nonetheless, for classification alone, SVM without discretization is quite good. The difference in classification accuracy between RBF and dot kernels is only a few percent (except in the already mentioned 'ionosphere'), with RBF kernels being better on 5 and linear on 6 data sets. Results with respect to the other two criteria are similar, with 6 (RBF) vs 5 (linear) wins on Brier score, and 3 vs 3 on area under ROC. By Wilcoxon signed ranks tests none of these results contradicts the null hypothesis of equivalence between the methods.

We also assumed that discretization could be used to alleviate the linear restrictions of the model. Experimental results (dot vs dot') do not confirm that. With exception of 'ionosphere' data set, discretization does not seem to have a large and consistent effect on linear SVM, and Wilcoxon test again fails to reject the null hypothesis. This question should, however, be further investigated. It is possible that our data sets do not contain larger non-linear relations, and that this is the reason that discretization had little beneficial effect on the results. Furthermore, a more sophisticated discretization algorithm might affect the results: our discretization is quite granular.

To test whether the differences between the tested algorithms are statistically significant, we used non-parametric statistical tests. We computed average ranks of the methods on all and, separately, on data sets that included continuous attributes. Neither Friedman test nor the stronger Iman-Davenport variation of it [3] detected any significant differences in any of the three performance criteria, except in the area under the ROC. The Iman-Davenport statistic is illustrated as F in the Table 1, and the associated p-value as p. Two average ranks over all the domains with continuous attributes in the AUC group are significantly different ($p = 0.10$) if their difference is greater than 1.3. Two av-

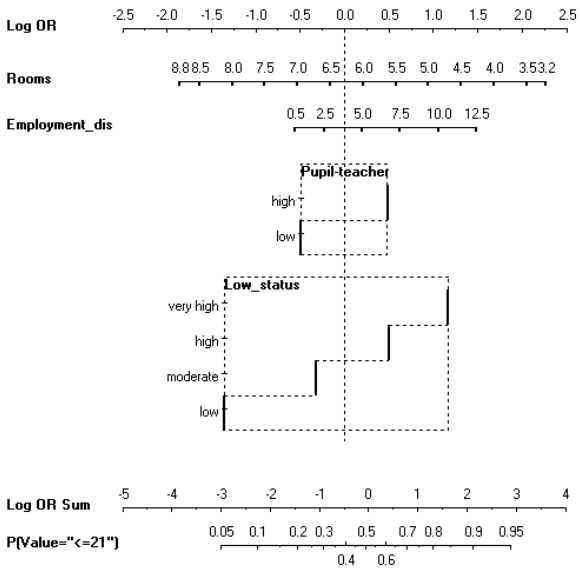

Figure 2. An SVM nomogram for the 'Housing' data set with a 2-dimensional presentation of ordered variables.

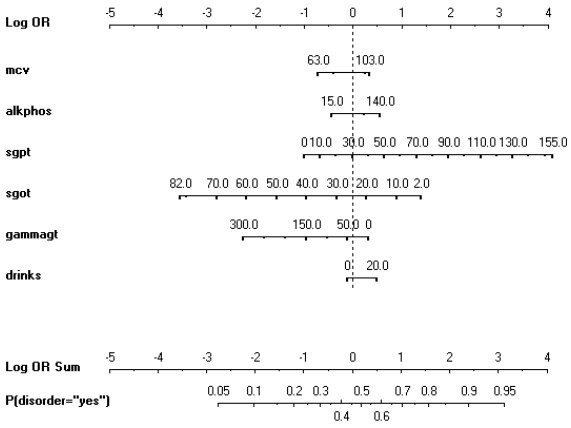

Figure 3. An SVM Nomogram induced from the 'BUPA' data set.

erage ranks over all the data sets are significantly different ($p = 0.05$) if their difference is greater than 0.9. Thereby, the AUC of NBC is significantly different from that of all SVM classifiers.

3.2. Examples of Nomograms

We here try to justify the value of nomograms by providing some practical examples where nomograms depict interesting features from the data that would otherwise remain hidden. We start by showing a two-dimensional graphical depiction of effect functions. Next, we show the difference between linear and discretized attributes, where the nomogram exhibits some of the problems that are known for linear SVM's assumption of effect linearity. In the end of this section we compare the SVM and the naïve Bayesian classifier on the 'Titanic' data set using nomograms. We will show how and why these two methods differ and explain why they have comparable classification accuracy, but different when observing Brier score.

3.2.1 Two-Dimensional Depiction of Effect Functions

The effects of a discretized continuous attribute are not very clear from the nomogram in Fig. 1, with labels appearing in the same line. An alternative approach is illustrated in Fig. 2, where the effect of a discrete attribute is presented in the form of a two-dimensional graph. The vertical dimension is used to list different discrete values and the horizontal dimension shows the effect of the value on the outcome. This graph reveals how the attribute's impact on the outcome probability gradually changes as its value changes from the lowest to the highest interval. This kind of presentation is suitable for ordered discrete attributes, as vertically ordering unordered attribute values would imply structure that does not exist in the data. It is also easy to see that continuous attributes can also be rendered in such a way, but SVM's linear effect functions are not particularly illuminating.

3.2.2 Linear vs. Non-Linear SVM

In the experimental comparison of SVM to other machine learning techniques, we showed that the linear SVM is as good as non-linear SVM using the RBF kernel on most data sets. Linear SVM, however, has difficulties when dealing with nonlinearities in attributes. The problems may be solved without non-linear kernels simply by discretizing the problematic attribute before employing linear SVM. In this section we will compare both approaches in the 'BUPA liver disorder' data set [9]. The first five attributes are the blood tests which are thought to detect liver disorders. On the other hand, liver disorders themselves might arise from excessive alcohol consumption, and the *drinks* attribute corresponds to number of half-pint equivalents of alcoholic beverages drunk per day.

Fig. 3 shows an SVM nomogram with continuous attributes in the original form. The nomogram is quite transparent and shows that a high value of *sgpt* indicates almost a certain presence of liver disorder, while low *sgot* implies that the person has very probably good liver health. We have discretized the *drinks* and *gammagt* attributes and Fig. 4 shows the resulting SVM nomogram of these attributes alone. The thickness of each bar represents the

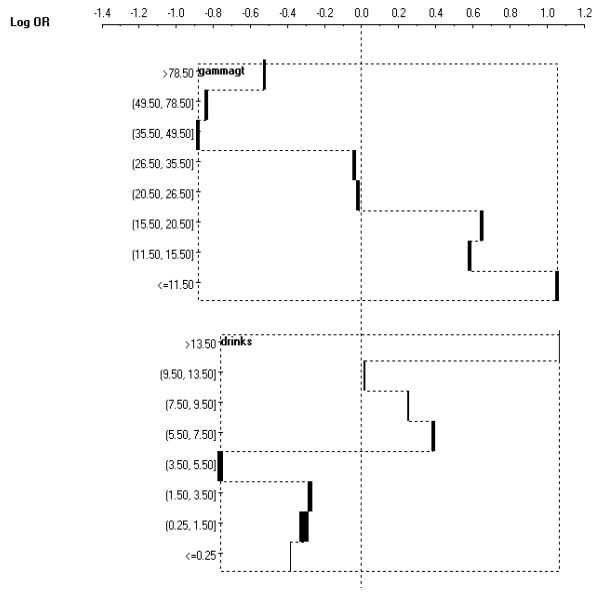

Figure 4. An SVM nomogram of two discretized attributes the 'BUPA' data set, with a two-dimensional graph presentation of the effect functions.

number of instances in the corresponding interval. The interesting point of comparison between the two nomograms in Fig. 3 and Fig. 4 is the attribute *drinks*. Whilst the first model assumed the linearity of influence of this attribute, it is clear from the second model that drinking less than certain amount of alcohol a day (5.5) has a low effect on liver health. However, when drinking more than this amount, the effect of alcohol may have drastic consequences.

The attribute *gammagt* as depicted in Fig. 4 manifests another possible problem that we might encounter when dealing with linear models. Notice that the attribute *gammagt* in Fig. 3 has a strong positive impact on liver health prognosis, but a comparison to the same attribute in the first nomogram (Fig. 4) shows that the impact of *gammagt* reaches a plateau at the value of around 35, while the linear model continues to extrapolate the growth of the impact in spite of the data. This is a very nice example of over-emphasizing effect when using linear SVM and other linear models.

3.2.3 Support Vector Machines vs. the Naïve Bayesian Classifier

Judging from the experimental comparison of SVM to other machine learning techniques, SVM sometimes achieves worse results on Brier score while having comparable classification accuracy at the same time. 'Shuttle' and 'Titanic' are examples of such data sets. The reason for the problem can be easily explained with a nomogram. We will compare the naïve Bayesian classifier (NBC) and SVM to predict the probability for passenger's survival of the HMS Titanic disaster. The NBC nomogram [16] on Fig. 5 (the data set was obtained at http://hesweb1.med.virginia.edu-/biostat/s/data/), includes three attributes: the passenger *status* (first, second, and third class, or a crew member), the *age* (adult or child), and the *sex* of the passenger. For NBC, the attribute with the biggest potential influence on the probability of survival is gender of the passenger: being female increases the chances of survival most (log odds of 1.7), while being male decreases the odds (log odds of about -0.6). Of the three attributes, the age is apparently the least influential, although children had a higher probability of survival. Most lucky were the passengers of the first class for which – considering the status only – the prob-

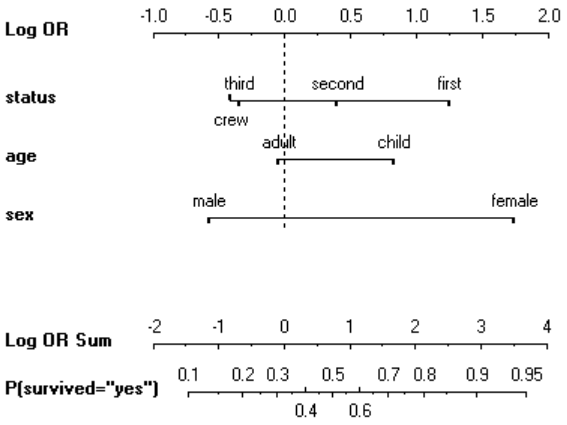

Figure 5. A naïve Bayesian nomogram for the 'Titanic' data set.

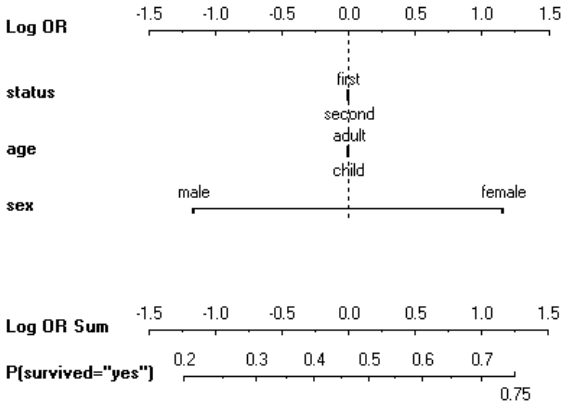

Figure 6. An SVM nomogram for the 'Titanic' data set.

ability of survival was much higher than the prior. Comparing this nomogram to the SVM nomogram in Fig. 6 of 'Titanic', we observed a very interesting difference between them. SVM, as it is known, aims to optimize the classification accuracy and considering this it induced a model that predicts survival of a passenger by considering only the *sex* attribute. Both methods, NBC and SVM, consider this attribute as very important, but unlike NBC, SVM disposes of age and status as completely irrelevant attributes. Using only the sex attribute, SVM achieves comparable classification accuracy, but the fidelity of the outcome probability estimates are slightly worse, as measured by Brier score.

4. Discussion

We have shown that support vector machines with linear kernels are not black box models even in spaces of high dimensionality, counter to the popular belief. We have provided the algorithm for converting such a support vector machine into a form of a generalized additive model. Furthermore, we have given a novel calibration algorithm based on logistic regression which captures the generalization error of any additive model. Finally, we have extended the form of a nomogram with two-dimensional graph representations of a nonlinear effect function. With the examples of Sect. 3.2, we pointed out that nomograms may be the right tool for experimental comparison of different models and modelling techniques, as it allows to easily spot the similarities and differences in the structure of the model. Furthermore, we can use nomograms to outline possible weaknesses of models, such as those of linear models by comparing them to the models obtained on discretized data.

Using linear kernels in SVM does reduce the performance slightly in comparison with more powerful kernels, but the differences were not statistically significant. Although RBF kernels are slightly better than linear ones, the differences in performance are small, and a minor improvement from the RBF kernel is rarely worth the resulting opaqueness of the model. Nonetheless, we have described how nonlinearity can be attained using a linear kernel through attribute discretization, remedying the performance in certain nonlinear data sets, such as 'ionosphere'. A further approach would be to seek and allow for interactions, which can be represented as joint effect functions of two attributes $f(a, b)$.

Our methodology can be improved. The normalization (which is a part of calibration) can be sensitive to outliers, and a more robust approach could be based on the percentiles of the response function range, rather than its minimum and maximum values. The nomogram could be generalized from using log odds ratio scale so that other approaches calibration could be supported; calibration with logistic regression is quite restrictive, and may be inappropriate for imbalanced or skewed data sets. Finally, our discretization method is a baseline one and was not tuned for performance which would improve with better algorithms.

References

[1] G. W. Brier. Verification of forecasts expressed in terms of probability. *Weather Rev*, 78:1–3, 1950.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] J. Demšar. Statistically correct comparisons of classifiers over multiple datasets, 2004. in preparation.

[4] J. Demšar and B. Zupan. Orange: From experimental machine learning to interactive data mining, 2004. White Paper (`http://www.ailab.si/orange`) Faculty of Computer and Information Science, University of Ljubljana.

[5] F. E. Harrell. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer, New York, 2001.

[6] D. Harrison and D. L. Rubinfeld. Hedonic prices and the demand for clean air. *J Environ Economics & Management*, 5:81–102, 1978.

[7] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, 1990.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: data mining, inference, and prediction*. Springer, 2001.

[9] S. Hettich and S. D. Bay. The UCI KDD archive. Irvine, CA: University of California, Department of Information and Computer Science, 1999.

[10] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, New York, 2000.

[11] A. Jakulin. Extensions to the Orange data mining framework, Jan. 2002. `http://www.ailab.si/aleks/orng/`.

[12] M. W. Kattan, J. A. Eastham, A. M. Stapleton, T. M. Wheeler, and P. T. Scardino. A preoperative nomogram for disease recurrence following radical prostatectomy for prostate cancer. *J Natl Cancer Inst*, 90(10):766–71, 1998.

[13] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[14] J. Lubsen, J. Pool, and E. van der Does. A practical device for the application of a diagnostic or prognostic function. *Methods of Information in Medicine*, 17:127–129, 1978.

[15] A. J. Miller. Algorithm AS 274: Least squares routines to supplement those of Gentleman. *Appl. Statist.*, 41(2):458–478, 1992.

[16] M. Možina, J. Demšar, M. W. Kattan, and B. Zupan. Nomograms for visualization of naive Bayesian classifier, 2004. submitted to PKDD-2004.

[17] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[18] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.