Nomograms for Visualizing Support Vector Machines

Aleks Jakulin
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana,
Slovenia
jakulin@acm.org

Martin Možina, Janez Demšar, Ivan Bratko and Blaž Zupan
Faculty of Computer and Information Science, University of Ljubljana
Tržaška cesta 25, SI-1001 Ljubljana,
Slovenia
martin.mozina,janez.demsar,ivan.bratko,blaz.zupan@fri.uni-lj.si

ABSTRACT

We propose a simple yet potentially very effective way of visualizing trained support vector machines. Nomograms are an established model visualization technique that can graphically encode the complete model on a single page. The dimensionality of the visualization does not depend on the number of attributes, but merely on the properties of the kernel. To represent the effect of each predictive feature on the log odds ratio scale as required for the nomograms, we employ logistic regression to convert the distance from the separating hyperplane into a probability. Case studies on selected data sets show that for a technique thought to be a black-box, nomograms can clearly expose its internal structure. By providing an easy-to-interpret visualization the analysts can gain insight and study the effects of predictive factors.

Categories and Subject Descriptors

G.6 [**Probability and Statistics**]: [Multivariate Statistics]; H.5.2 [**Information interfaces and presentation (e.g., HCI)**]: User Interfaces—*Theory and methods*

General Terms

Theory, Human Factors

Keywords

nomogram, visualization, support vector machines, machine learning

1. INTRODUCTION

Within predictive data mining, methods that build classification models have received much attention. 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 (SVM) [23, 22] is a 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, as 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. It is possible to show the SVM classifier directly in the attribute space, but this is only appropriate when the attribute space does not have more than two or three dimensions. When the SVM model is a hyperplane, we can also present it with the hyperplane's normal vector, but this technique is of limited utility with multi-valued or continuous attributes.

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 for all common types of attributes and even for non-linear SVM kernels. 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. In the beginning of 2005 a search for 'nomogram*' on PubMed/MEDLINE, a database of biomedical article citations, yielded over 2400 papers (a search for 'support vector machine*' yielded fewer than 400). A search for nomograms on Google resulted in 77000 web pages. Nomograms are not an uncertain novelty, but a milestone in the history of visualization [6].

To visualize a logistic regression model, the use of nomograms was first proposed by Lubsen and coauthors [17]. With an excellent implementation of logistic regression nomograms in S-Plus and R statistical packages by Harrell [7], 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., [15]). A naïve Bayesian classifier can too be visualized in the form of a nomogram [19].

The nomograms for support vector machines that we in-

SIGKDD '05 Chicago, Illinois USA

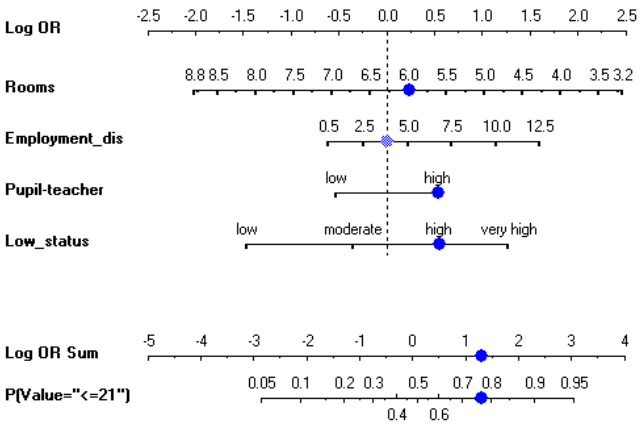

Figure 1: A nomogram of the SVM model that predicts the probability of costly housing in a given Boston area. The dots illustrate the classification of a specific instance.

troduce 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 [8]). The Housing data set consists of 506 different instances (areas of Boston); about 50% of the areas have the 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*, discretized to two nominal values), and proportion of lower status population (*Low_status*, discretized to four nominal values). There are two classes of areas, the expensive with the median values above $21000, and the cheap. To make a prediction using a nomogram, the contributions of attributes on the scale of the log odds ratios [11] (topmost axis of the nomogram) are summed up, and used to determine the probability whether the price is less than $21000 (bottommost axis of the nomogram). For instance summing the effects of 6 rooms per average dwelling, unknown distance from employment centers, a high pupil-teacher ratio and a high rate of lower status population results in the log odds ratio $0.21 + 0.0 + 0.49 + 0.5 = 1.20$ on the 'Log OR Sum' axis. This sum is then projected to the bottommost 'P(<= 21)' probability axis, yielding the final probability of the target class of approximately 0.76. On the other hand, if the area 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, and 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 the 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. The nomogram also exposes how different attribute values affect the outcome; for instance, the value of housing goes up when the employment centers are nearby. 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 near 0.0 on the Log OR axis. They do not affect the probability of the outcome. If a particular attribute value is not given for the test instance, it is these neutral values that 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. Assume an instance $[\boldsymbol{x}, y]$, where the range of the label is assumed to be $\Re_y = \{-1, 1\}$, and $\boldsymbol{x} \in \Re_{\mathcal{X}}$ is described by a set of attributes $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$. The nomogram can visualize a probability function of the type

$$\hat{P}(y = 1|\boldsymbol{x}) := F\Big(\beta_0 + \sum_{j=1}^{k} f_j(A_j(\boldsymbol{x}))\Big) \qquad (1)$$

where β_0 is the *intercept*, 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 for the instance $\boldsymbol{x}$ into a point score, and F is the inverse link function that maps the *response* of an instance into the outcome probability. The nomogram in Fig. 1 is based upon one effect function for each attribute. Each line in the nomogram corresponds to a single attribute, and a single effect function. Because the effect function $f_{Low\_status}(\text{high}) = 0.05$, the tick corresponding to the value 'high' for the attribute *Low_status* is aligned with 0.05 on the 'Log OR' axis.

The class of models of the above type are the generalized additive models (GAM, [9]). When each effect function is linear, we speak of a generalized linear model (GLM). For a GLM, the response or the systematic component is written as $\beta_0 + \sum_j [\boldsymbol{\beta}]_j [\boldsymbol{x}]_j$, where $[\boldsymbol{x}]_j$ is the j-th coordinate of the vector $\boldsymbol{x}$. Using the dot product $\langle \cdot, \cdot \rangle$ we may express it more simply as $\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$. We may refer to the vector $\boldsymbol{\beta}$ as the *effect vector*.

We start by showing how support vector machines based on appropriate kernels 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, but attempts to achieve the separation of instances in the feature space with a separating hyperplane, 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. In the experimental section we examine the performance of linear SVM in comparison to other methods that can be visualized with nomograms. We also compare linear SVM to the SVM with the RBF kernel, which cannot be visualized with a low-dimensional nomogram, observing that the losses are not very large. We also show that nomograms are suitable for graphically comparing support vector machines to other generalized additive models, such as the naïve Bayesian classifier and logistic regression.

2. METHODOLOGY

2.1 Overview

Not every support vector machine is appropriate for visualization using a nomogram. The first requirement is the ability to additively separate the contribution towards the response of an individual attribute or of a small group of attributes as in (1). We achieve this goal by using a kernel of a particular family. The second requirement is related to how we represent the lack of information: ideally, zero value of the transformed attribute should indicate missing attribute values.

Our approach to visualization will take the following steps, which will be addressed in detail in the subsequent sections.

1. Transform each instance $\boldsymbol{x} \in \Re_{\mathcal{X}}$ into the feature space using a decomposable kernel map $\Phi : \Re_{\mathcal{X}} \to \mathcal{H}$.

2. Train the support vector machine using the dot product kernel, and obtain the hyperplane's normal $\boldsymbol{w}$ and bias b.

3. Employ univariate logistic regression to obtain two parameters Δ and Υ, so that

$$\text{logit}(\hat{P}(y = 1|\boldsymbol{x})) = \Upsilon + \Delta(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b)$$

   is well-calibrated. Obtain the effect vector and the intercept on the log-odds scale by elementary algebraic manipulation of the above. This step is only necessary to assure that the log-odds scale can be used in the nomogram.

4. Collect the terms of the effect vector that belong to a specific attribute. This results in a possibly nonlinear effect function $f(A(\boldsymbol{x}))$ for each attribute A. It is also conceivable to visualize effect functions of multiple attributes simultaneously. For example, the joint effect of attributes A,B and C would take the form $f(A(\boldsymbol{x}), B(\boldsymbol{x}), C(\boldsymbol{x}))$. Such joint effect functions are useful when the attributes in the group interact.

5. Visualize the effect functions and the intercept in a nomogram.

A Simple Example. As an example, we have taken the Fisher's iris data, selected the I. versicolor and I. virginica species, and the petal length and the petal width attributes. We have trained the SVM model with a linear kernel with both attributes standardized, and have used the cross-calibration to obtain probabilistic outputs from SVM. The top-down visualization of the data and the model to the left of Fig. 2 should be familiar. The true SVM's separating hyperplane would lie at the contour corresponding to the probability of approximately 0.49. The hyperplane is specified with the equation $1.82\mathit{length} + 3.64\mathit{width} = 15.21$. The vector $[1.82, 3.64]^T$ can be understood as a weight vector, but we should note that it depends on the scaling of the attributes: it should not seem that the width is more than twice as important as the length. On the other hand, the nomogram shown to the right of Fig. 2 clearly shows the effect of individual attribute values on the outcome. Unlike the top-down view which is restricted to two attributes and two dimensions, we can include a larger number of attributes in the nomogram without increasing the dimensionality.

2.2 The Kernel Map

Support vector machines can be applied purely with a kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ and the resulting Gram matrix. However, nomogram visualization requires us to concern ourselves with the kernel map from the *instance space* $\Re_{\mathcal{X}}$ into the *feature space* $\mathcal{H}$ using the *reproducing kernel map* $\Phi : \Re_{\mathcal{X}} \to \mathcal{H}$. First we will discuss various kernel maps. Later we will describe the notion of a decomposable kernel, which is limited by the number of dimensions required for visualizing the resulting classifier in the form of a nomogram.

2.2.1 Non-Linear Univariate Kernel Maps

All attributes need to be transformed into real-valued variables before a model can be trained. We standardize continuous attributes so that zero implies the mean, and ± 1 implies one standard deviation distance from the mean. Some m-th coordinate of the linearly transformed instance $[\Phi(\boldsymbol{x})]_m$ equals the standardized value of a continuous attribute $(A(\boldsymbol{x}) - \mu_A)/\sigma_A$. SVM based on a purely linear kernel and logistic regression both have linear effect functions, and can therefore be seen as generalized linear models.

However, attributes may have non-linear effects on the outcome. For example, both very high and very low body temperatures indicate risks when predicting the health status, and this pattern cannot be captured by a single real-valued variable. We can allow for the nonlinear effect functions by employing non-linear kernel maps. This way, a single attribute is internally represented with more than one dimension, and the actual support vector machine can be trained using the dot product on the feature space $\mathcal{H}$, but not on the attribute space $\Re_{\mathcal{X}}$.

The simplest example of a non-linear map relates to handling multi-valued nominal attributes. A discrete attribute B with V values is transformed into a set of V features $[\boldsymbol{x}]_m, [\boldsymbol{x}]_{m+1}, \ldots, [\boldsymbol{x}]_{m+V-1}$, so that given the value of $B = b_{v+1}$, $[\boldsymbol{x}]_{m+v} = 1$ and $\forall j = 0, \ldots, v-1, v+1, \ldots, V : [\boldsymbol{x}]_{m+j} = 0$. Thus, the kernel map assigns its own dimension to each attribute value, and also provides ground to interpret setting all corresponding $[\boldsymbol{x}]_{m+j}$ to zero as a missing value.

The same concept can be applied to continuous attributes. Using *discretization*, we convert a continuous attribute x into a V-valued discrete one, each value of which corresponds to an interval of the range of x. This is an extremely simple method for handling nonlinear effects. For example, we could discretize the body temperature into a 3-valued nominal attribute with the range $\{<36.6, 36.6\text{-}37.4, >37.4\}$. The corresponding effect vector $[b_1, b_2, b_3]^T$ is obtained from SVM, and the effect function $f(x)$ then takes the following form:

$$f(x) = \begin{cases} b_1 & ; \; x < 36.6 \\ b_2 & ; \; 36.6 \leq x \leq 37.4 \\ b_3 & ; \; x > 37.4 \end{cases}$$

Discretization essentially involves modelling the effect of an attribute with a piecewise-constant function.

Using *polynomialization* we transform a continuous attribute x into a vector of features $[x, x^2, \ldots, x^d]^T$. The corresponding effect vector $[a_1, a_2, \ldots, a_d]^T$ results in a polynomial effect function for x: $f(x) = a_1 x + a_2 x^2 + \ldots + a_d x^d$. Other forms of transforming continuous attributes may be employed while maintaining the dot product kernel. Such functions can be easily rendered inside the nomogram, so

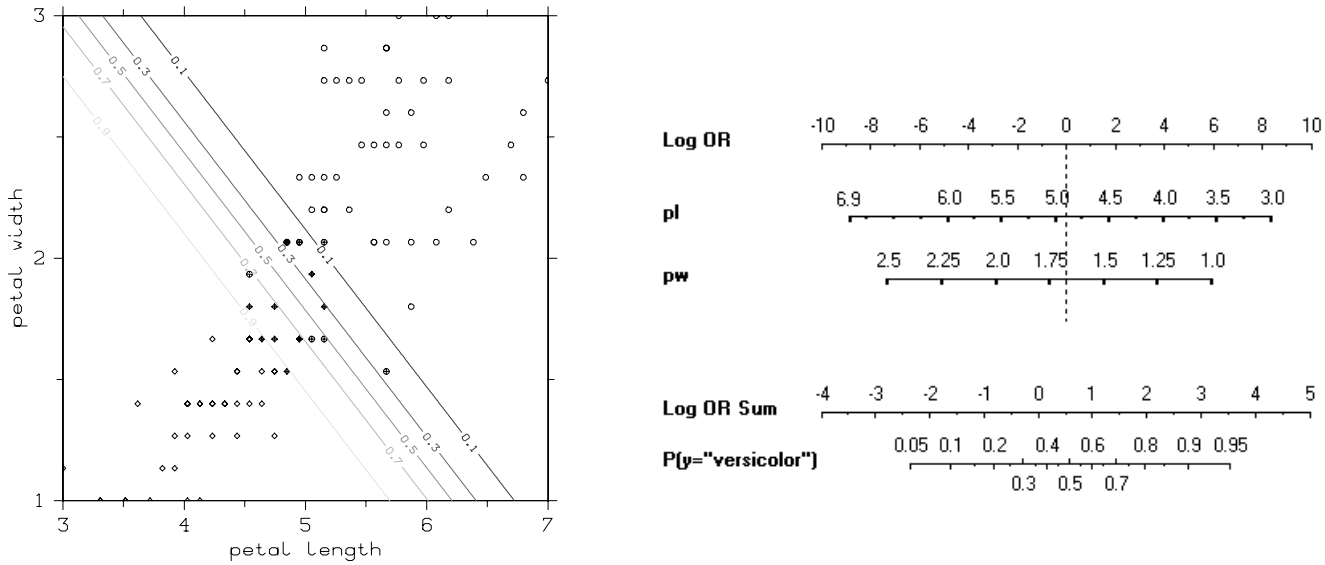

Figure 2: A top-down projection of SVM (Left) as applied to Fisher's iris data, separating the I. virginica (top-right) from I. versicolor (bottom-left). The probability contours resulted from cross-calibration using a logit link function. The support vectors are marked. The above model can be summarized in the form of a nomogram (Right).

that $f(x)$ is shown on the horizontal axis, and x on the vertical axis or using a label.

In addition to discretization and polynomialization, other univariate expansions can be employed, such as piecewise-linear functions, splines, sigmoids, or even univariate radial-basis functions.

Visualizing Non-Linear Effects. The effects of continuous attributes can be shown on a single line as in Figs. 1 and 2. However, non-linear effect functions, especially non-monotonic ones, could be confusing if presented in such a way. An alternative approach is illustrated in Fig. 3, where the effect of an attribute is presented in the form of a two-dimensional graph. The vertical dimension is used to list different values and the horizontal dimension shows the effect of the value on the outcome. The 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 also suitable for ordered discrete attributes.

We have used the 'Horse Colic' data set from the UCI repository [10]. Among the attributes, we have chosen the respiratory rate and the body temperature, as they are continuous attributes with potentially non-linear effects. It is clear that there is a particular range of normal body temperatures centered near 38° with low risk. The deviations in any direction (fever, hypothermia) carry increased risk. The pulse appears more monotonic, but the effect of the pulse is distinctly non-linear with respect to the pulse scale.

The intervals for the piecewise constant effect function were set manually. The RBF effects are defined through 4 radial bases, covering separate intervals of an attribute's range.

2.2.2 Decomposable Kernels

Discretization and polynomialization correspond to non-linear kernels, but the non-linearity is always restricted to within a single attribute. We will now employ an example to show why general non-linear kernels introduce problems for nomogram visualization. Let us focus on the quadratic kernel $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^2$. Specifically, for an instance $\boldsymbol{x} = [x_1, x_2]^T$ in a two-dimensional continuous attribute space, we can introduce the following kernel map $\Phi(\boldsymbol{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]^T$. Then the quadratic kernel can be linearized [22]:

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^2 = x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2 x_2'^2$$

We can see that in addition to polynomializing each attribute to the degree of 2, the quadratic kernel introduces interactions involving each pair of attributes and the label, corresponding to the coordinates x_1 and x_2. The effect function would take the form of $f(x_1, x_2)$ so that one attribute's effects only appear in a single place. Otherwise, the effect of x_1 would appear under $[\boldsymbol{x}]_2$ corresponding to x_1x_2, under $[\boldsymbol{x}]_1$ corresponding to x_1^2, and under the intercept term. Such effect functions are more difficult to be effectively visualized in two dimensions: the effect of x_1 depends on the value of x_2. Such visualization would involve simulating the third dimension either with color or with shape on a two-dimensional computer monitor.

Of course, non-linear kernels can be used for nomogram visualization as well, under some restrictions. We will now describe a general form of a kernel suitable for visualization. Assume a partitioning of the set of attributes $\mathcal{X}$ into m disjoint subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$, so that $\bigcup_{i=1}^m \mathcal{S}_i = \mathcal{X}$ and $\sum_{i=1}^m |\mathcal{S}_i| = |\mathcal{X}|$. The underlying assumption is that all the interactions between attributes happen within each subset $\mathcal{S}_i$, but not across the subsets. We can then visualize any SVM based on such a kernel K that is expressible in terms of such a sum:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{m} \dot{K}_{\mathcal{S}_i}([\boldsymbol{x}]_{\mathcal{S}_i}, [\boldsymbol{x}']_{\mathcal{S}_i}) \qquad (2)$$

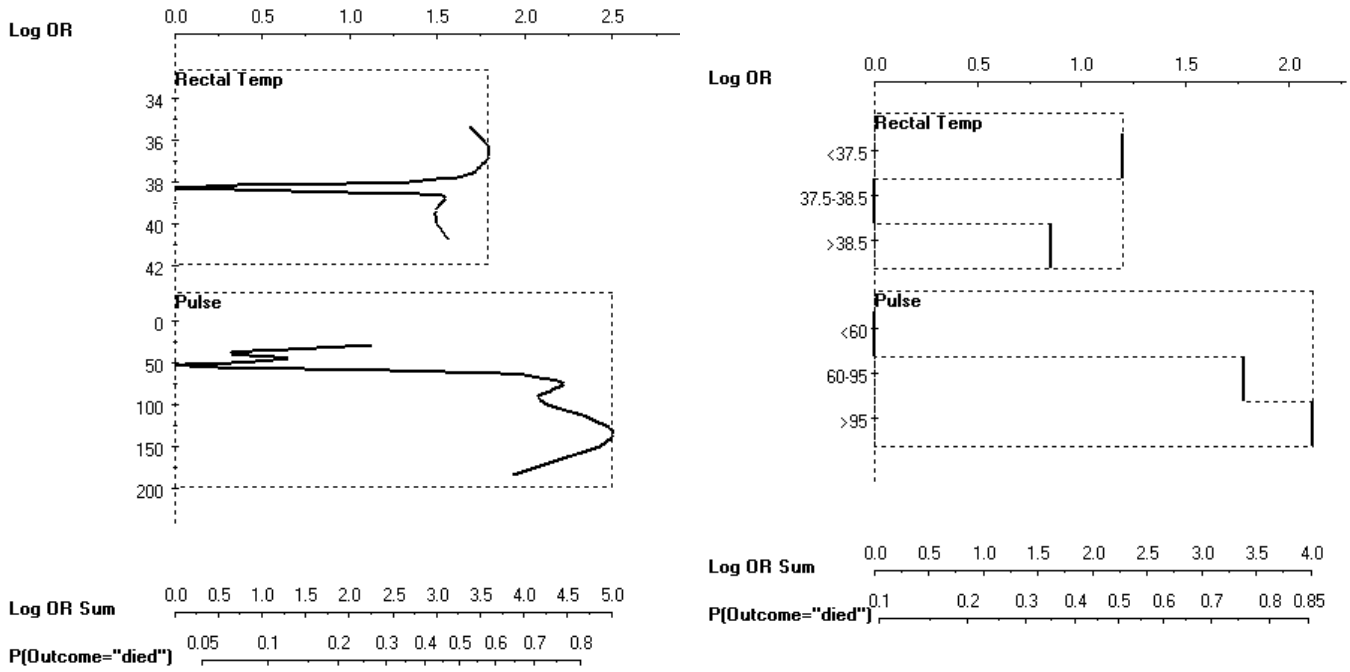

Figure 3: A nomogram with the univariate RBF effects (left) and the piecewise constant univariate effects (right) in the 'Horse Colic' data set. We can observe that increased pulse and increased or decreased body temperature all indicate an increased risk of death.

A kernel that can be expressed in such a way will be referred to as an (additively) *decomposable kernel*. Here, $\dot{K}_{\mathcal{S}_i}$ is an arbitrary nonlinear positive semidefinite *sub-kernel* that acts upon $[\boldsymbol{x}]_{\mathcal{S}_i}$, the subset of the coordinates of $\boldsymbol{x}$ that correspond to the attributes in $\mathcal{S}_i$. The full reproducing kernel Hilbert space is then a concatenation of the reproducing kernel Hilbert spaces for each $\dot{K}_{\mathcal{S}_i}$. It is then possible to retrieve the the effect functions $f(\mathcal{S}_i)$, localized for each subset of interacting attributes. The dimensionality of the resulting nomogram visualization is $\max_i |\mathcal{S}_i| + 1$. The kernel (2) is a special case of the kernels proposed by [5, 1, 16]. In particular, [5] motivated the choice of these kernels through the ability to effectively visualize them.

Visualizing Interaction Effects. Fig 4 shows the comparison between models that use interactions in the learning phase and models that do not. If no interaction is assumed, we employ the linear kernel. If an interaction between attributes A and B is assumed, we employ the following subkernel:

$$\dot{K}_{\{A,B\}}\left([a,b]^T,[a',b']^T\right) := \begin{cases} 1 & ; \; a = a' \wedge b = b' \\ 0 & ; \; \text{otherwise} \end{cases}$$

We have used German credit risk data set that contains 20 attributes and 1000 past applicants. Each applicant was classified according to the risk (high vs. low). The risk is low if the applicant is very likely to return the money, and high otherwise. Due to space restrictions, we have used only 6 attributes: the status of applicant's account in the bank, duration of the credit in months, purpose of the credit, the amount of credit asked for, the duration of the applicant's present employment, and applicant's duration of residence. The joint effect can be illustrated in the nomogram by picking one attribute as the 'control', a condition. The other attribute's influence is then interpreted in the context of the control. We examined the effect of *employment duration*, controlling for *residence duration*. This pair of attributes in 'German-credit' has been identified as significantly interacting, using the methodology of interaction analysis [14].

Both nomograms are very similar when comparing the first four attributes. Among those four, the most influential attribute is the *purpose* of the credit: buying used cars incurs low risk, and education incurs high risks. The difference between two model occurs in the effect of unemployment on the risk: without the interaction, the model regards unemployment as almost unimportant, while the second regards it as highly important for determining low risk (if residence duration is higher than 2.5 years) and for determining high risk (if residence duration is less than 2.5 years). It is probable that people that live at this place for more than 2.5 years are unemployed because they do not need or can not work, i.e. are retired, while people that are residents for less than 2.5 years are unable to find a job. This comparison stresses the importance of interactions and shows that they can be effectively visualized with nomograms.

Through this more complex example we show that apart from revealing the structure of the SVM classifier, nomograms may be used as a data mining tool to depict different properties of problem domains. Gunn and Kandola [5] present examples of how interactions of real-valued variables can be visualized using 3D plots, but not in the context of the nomograms.

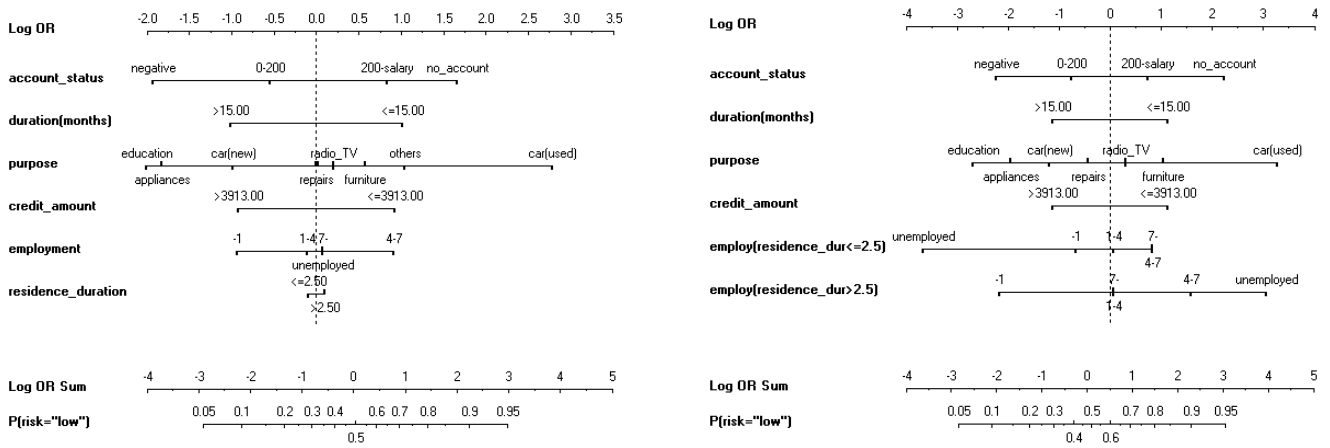

Figure 4: A nomogram with the linear kernel (left) and the non-linear kernel that assumes a joint effect function for the *employment duration* and *residence duration* attributes (right) in the 'German-credit' data set. We can see that controlling for the residence duration amplifies and differentiates the effects of employment duration on the assessment of credit risk.

2.3 Distance from the Separating Hyperplane

Given N training instances $[\boldsymbol{x}^{(j)}, y^{(j)}]$, $j = 1, 2, \ldots, N$, the resulting support vector model can be described with the vector $\boldsymbol{\alpha}$ and the *bias* b. The (signed) separating *hyperplane distance* of an instance $[\boldsymbol{x}, y]$ is denoted as $\delta(\boldsymbol{x})$. Given a dot product kernel $\langle \boldsymbol{x}, \boldsymbol{x}^{(j)} \rangle$, the distance be described as:

$$\delta(\boldsymbol{x}) := \tau \left(b' + \sum_{j=1}^{N} y^{(j)} [\boldsymbol{\alpha}]_j \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}^{(j)}) \rangle \right) \tag{3}$$

Here, b is the bias, while the scaling constant τ assures that the distance is Euclidean. The sign of the hyperplane distance indicates the predicted value of the label.

Because of the kernel map Φ, we work in the feature space with the dot product kernel. We can now remove the reference to support vectors $\boldsymbol{x}^{(\cdot)}$, and represent the distance with the bias b and the vector $\boldsymbol{w}$. We now define the weight vector $\boldsymbol{w}'$. It has the same dimensionality as the feature space, and is defined as:

$$\boldsymbol{w}' := \sum_{j=1}^{N} y^{(j)} [\boldsymbol{\alpha}]_j \Phi(\boldsymbol{x}^{(j)}) \tag{4}$$

The length of the weight vector $\boldsymbol{w}'$ thus obtained is not 1. For that reason we needed the normalizing constant τ in (3), where $\tau = 1/\|\boldsymbol{w}'\|$. The signed hyperplane distance of an instance $\boldsymbol{x}$ can thus be expressed as

$$\delta(\boldsymbol{x}) = \frac{b' + \langle \boldsymbol{w}', \Phi(\boldsymbol{x}) \rangle}{\|\boldsymbol{w}'\|} \tag{5}$$

To simplify further notation, $b = b'/\|\boldsymbol{w}'\|$, $\boldsymbol{w} = \boldsymbol{w}'/\|\boldsymbol{w}'\|$. The $\boldsymbol{w}$ is the separating hyperplane normal.

If we are working with a non-linear kernel of the form (2), it is easy to see that a particular coordinate $[\boldsymbol{w}']_k$ corresponds to one or more attributes, but only within a single group $\mathcal{S}_i$. Of course, multiple coordinates can correspond to a single attribute if nonlinear univariate sub-kernels are used, and one coordinate may correspond to multiple attributes if nonlinear multivariate sub-kernels are used. This way, (5) can be written as:

$$\delta(\boldsymbol{x}) = \frac{b' + \sum_{i=1}^{m} \langle \boldsymbol{w}'_{\mathcal{S}_i}, \dot{\Phi}_{\mathcal{S}_i}([\boldsymbol{x}]_{\mathcal{S}_i}) \rangle}{\|\boldsymbol{w}'\|} \tag{6}$$

Therefore, each sub-kernel $\dot{K}_{\mathcal{S}_i}$ is independently linearized by $\dot{\Phi}_{\mathcal{S}_i}$. The approach works even if $\dot{K}$ is an RBF kernel involving a potentially infinite-dimensional reproducing kernel Hilbert space: the dimensionality of $\dot{\Phi}_{\mathcal{S}_i}([\boldsymbol{x}]_{\mathcal{S}_i})$ will depend on the size of the data set, but will be finite for a finite data set.

2.4 Cross-Calibration

The horizontal scale in nomogram-based visualizations is based on the probability of the label. However, the signed hyperplane distance $\delta(\boldsymbol{x})$ of an instance $\boldsymbol{x}$ has no probabilistic meaning. This is the role of the link function. The link function connects probability (the random component) with the response (the systematic component). The link function in classification maps a probability p into a response d. The inverse link function F instead maps a response d into a probability. The most frequently used link functions are the identity$(p) = p$, probit (the inverse of the cumulative Gaussian distribution) and logit$(p) = \log(p/(1-p))$. The inverse logit link function is $F(d) = 1/(1 + \exp d)$, and it has been used in the past [21].

While the logistic regression too employs a generalized linear model with the logit link, the effect vector $\boldsymbol{\beta}$ is optimized directly in order to minimize the probabilistic loss (deviance) of the resulting model. The hyperplane distance δ does not attempt to optimize the calibration performance using the logit link, merely achieve the separation. For that reason, Platt linearly transforms the SVM output with two additional parameters, Υ and Δ, using a procedure that resembles univariate logistic regression with the hyperplane distance acting as the independent variable, and the label as the dependent variable. The two parameters ensure that $F(d)$ based on $d = \Upsilon\delta(\boldsymbol{x}) + \Delta$ is a well-calibrated probabilistic classifier using the logit link.

It often happens that the separation of the support vector machine on the training set is perfect. In such a case,

the inverse of the logistic link will tend towards a step function. However, on a separate test set, the same performance is rarely as good. For that reason Platt [21] proposed performing internal cross-validation where the training set is partitioned into two sets of instances, one is used for SVM training, and the other for learning the parameters Υ and Δ. The error arising from generalization is thus accounted for: the two parameters capture the uncertainty associated with generalization to unseen data.

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.

With the logit link the end result can be represented as

$$\hat{P}(y=1|\Delta,\Upsilon,\delta(\boldsymbol{x})) = \frac{1}{1+\exp\{\Delta+\Upsilon\delta(\boldsymbol{x})\}} \qquad (7)$$

If we apply logistic regression to the problem of associating the hyperplane distance with the label, we find such values of Δ and Υ that maximize the thus defined conditional log-likelihood of y given $\boldsymbol{x}$ in the above model across the N training instances $[\boldsymbol{x}^{(j)}, y^{(j)}]$:

$$\Delta,\Upsilon = \arg\max_{\Delta,\Upsilon} \prod_{j=1}^{N} \hat{P}(y=y^{(j)}|\Delta,\Upsilon,\delta(\boldsymbol{x}^{(j)})) \qquad (8)$$

The calibrated response function on the log odds ratio scale is $d(\boldsymbol{x}) = \Delta + \Upsilon\delta(\boldsymbol{x})$. We can now map these symbols $\Delta, \Upsilon, \boldsymbol{w}$ and b so that they will correspond to the notation of a (linearized) generalized additive model (1) based on the intercept β_0 and the effect vector $\boldsymbol{\beta}$. The intercept β_0 marks both the outcome probability of 0.5 and the log odds ratio of 0.0, so it can be seen as probabilistically calibrated bias b. The k-th coordinate $[\boldsymbol{\beta}]_k$ of the effect vector corresponds to the probabilistically calibrated coordinate of the normal $[\boldsymbol{w}]_k$. The mapping is as follows:

$$\beta_0 = \Delta + \Upsilon b, \qquad [\boldsymbol{\beta}]_k = \Upsilon[\boldsymbol{w}]_k \qquad (9)$$

The linear effect function for the set of attributes $\mathcal{S}_i$ is simply $f_{\mathcal{S}_i}([\boldsymbol{x}]_{\mathcal{S}_i}) = \langle [\boldsymbol{\beta}]_{\mathcal{S}_i}, [\Phi(\boldsymbol{x})]_{\mathcal{S}_i} \rangle$. Both f_k and β_0 are on the log odds ratio scale, and they can thus be directly presented in a nomogram. It is important to distinguish the weight vector $\boldsymbol{w}'$, the hyperplane normal $\boldsymbol{w} = \boldsymbol{w}'/\|\boldsymbol{w}'\|$, the GLM effect vector $\hat{\boldsymbol{\beta}}$, and the Lagrange multipliers $\boldsymbol{\alpha}$: all are different.

3. MODEL COMPARISON

In this section, we examine the performance of support vector machines in comparison to other methods that can be visualized with the nomograms. To address this, we compare the nomogram-based probability estimations with those obtained from SVM with RBF kernel (Did we lose anything assuming the decomposability into effect functions?) and two popular methods for probabilistic classification, namely logistic regression and the naïve Bayesian classifier (What is the overall performance in class probability prediction?). Nomograms may be used to study the differences between various modelling methods from the family of generalized additive models. We present a nomogram-based comparison of SVM and the naïve Bayesian classifier model.

Figure 5: A general scheme of a cross-calibration procedure, based on N folds, R replications, the response learning algorithm L, the calibration learning algorithm C, and the training data $\mathcal{T}$.

$\mathcal{R} \leftarrow \emptyset$ {Calibration training set.}
for all $r : 1 \le r \le R$ **do** {for each replication}
  $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_N \leftarrow \mathcal{T}$ {Generate folds.}
  for all $n : 1 \le n \le N$ **do** {for each fold}
    $(\hat{\delta} : \Re_{\boldsymbol{x}} \to \mathbb{R}) \leftarrow L\left(\bigcup_{i \neq n} \mathcal{F}_i\right)$ {Train.}
    for all $\boldsymbol{x}^{(i)} \in \mathcal{F}_n$ **do** {for each test instance}
      $\mathcal{R} \leftarrow \mathcal{R} \cup \left\{ \left[\hat{\delta}(\boldsymbol{x}^{(i)}), y^{(j)}\right] \right\}$ {Record the distance.}
    end for
  end for
end for
$(\delta : \Re_{\boldsymbol{x}} \to \mathbb{R}) \leftarrow L(\mathcal{T})$ {Hyperplane distance.}
$(\hat{P}\,(y=1|\boldsymbol{x}) : \Re_{\boldsymbol{x}} \to [0,1]) \leftarrow C(\mathcal{R}, \delta)$ {Calibrated prob.}

3.1 Accuracy

As for earlier nomograms, all experiments were performed within the Orange toolkit [4]. We employed LIBSVM [3] with default settings for training the SVM classifiers, and iteratively re-weighted least squares fitting [18] of the logistic regression model, as implemented in the Orange extensions package [12]. We experimented on 16 well-known UCI [10] 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 [2]), and 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 discretized 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 than higher.

The observed methods perform similarly, with some exceptions. For instance, linear SVM performs poorly on 'ionosphere' unless the attributes are discretized. This indicates non-linear attribute effects in this data set, and we illustrate an example of them in Fig. 6. 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 good probability estimation results and reasonable ranking results.

	Classification accuracy					Brier score					Area under ROC				
	NB	LR	RBF	dot	dot'	NB	LR	RBF	dot	dot'	NB	LR	RBF	dot	dot'
breast (lju)	0.71	0.70	0.73	0.69		0.39	0.42	0.37	0.39		0.70	0.67	0.71	0.66	
breast (wsc)	0.97	0.93	0.97	0.95		0.05	0.15	0.05	0.08		0.99	0.91	0.99	0.98	
mushroom	0.99	1.00	0.99	1.00		0.02	0.01	0.01	0.01		0.99	1.00	1.00	1.00	
shuttle	0.96	0.99	0.97	0.97		0.09	0.02	0.05	0.08		0.96	0.99	1.00	1.00	
titanic	0.78	0.78	0.79	0.78		0.34	0.33	0.32	0.35		0.72	0.76	0.68	0.72	
voting	0.89	0.96	0.96	0.96		0.13	0.06	0.07	0.08		0.98	0.99	0.99	0.99	
australian	0.86	0.85	0.87	0.85	0.86	0.21	0.30	0.20	0.24	0.22	0.92	0.85	0.93	0.91	0.91
german	0.75	0.76	0.73	0.73	0.74	0.33	0.33	0.34	0.34	0.34	0.79	0.79	0.79	0.79	0.79
hepatitis	0.84	0.83	0.85	0.84	0.84	0.21	0.26	0.23	0.24	0.27	0.86	0.85	0.74	0.76	0.75
horse-colic	0.81	0.82	0.84	0.83	0.77	0.31	0.29	0.26	0.29	0.31	0.83	0.86	0.88	0.86	0.85
housing	0.81	0.87	0.88	0.86	0.84	0.27	0.19	0.18	0.19	0.23	0.90	0.94	0.95	0.94	0.92
ionosphere	0.91	0.84	0.94	0.84	0.89	0.15	0.26	0.11	0.28	0.19	0.93	0.84	0.98	0.83	0.94
liver	0.67	0.69	0.70	0.68	0.72	0.41	0.42	0.41	0.44	0.39	0.73	0.72	0.74	0.72	0.77
pima	0.75	0.78	0.77	0.78	0.75	0.32	0.30	0.34	0.33	0.35	0.83	0.83	0.71	0.73	0.72
post-op	0.66	0.68	0.73	0.71	0.70	0.40	0.49	0.39	0.40	0.39	0.40	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	0.22	0.13	0.94	0.89	0.90	0.92	0.91

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.

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. Nonetheless, the difference between SVM with RBF and dot kernels is only a few percent (except in the already mentioned 'ionosphere'). We expected that the discretization would alleviate the linear restrictions of the model, but experimental results (dot vs dot') do not confirm that. Still, dot' provided a considerable improvement in the 'ionosphere' and 'liver/BUPA' data sets. This indicates that the non-linearities appear only in certain data sets. We need to apply the more sophisticated models and visualizations only if the non-linearity is justified through a higher classification performance.

3.2 Comparing Models With Nomograms

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 (NB) and SVM to predict the probability for passenger's survival of the HMS Titanic disaster. The NB nomogram [19] in Fig. 7 (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 NB, 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 probability of survival was much higher than the prior. Comparing this nomogram to the SVM nomogram in Fig. 7 of 'Titanic', we observed a very interesting difference between them. SVM,

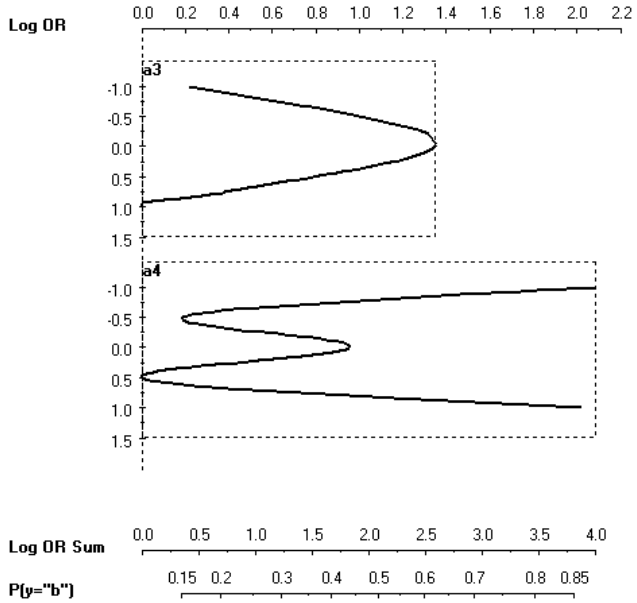

Figure 6: The nonlinearities in the 'ionosphere' data set. Two features were used for each attribute, representing an RBF basis.

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, NB and SVM, consider this attribute as very important, but unlike NB, 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.

3.3 Interpretation of Effect Functions

The role of the nomogram is to visualize the probabilistic predictions of a support vector machine without losing

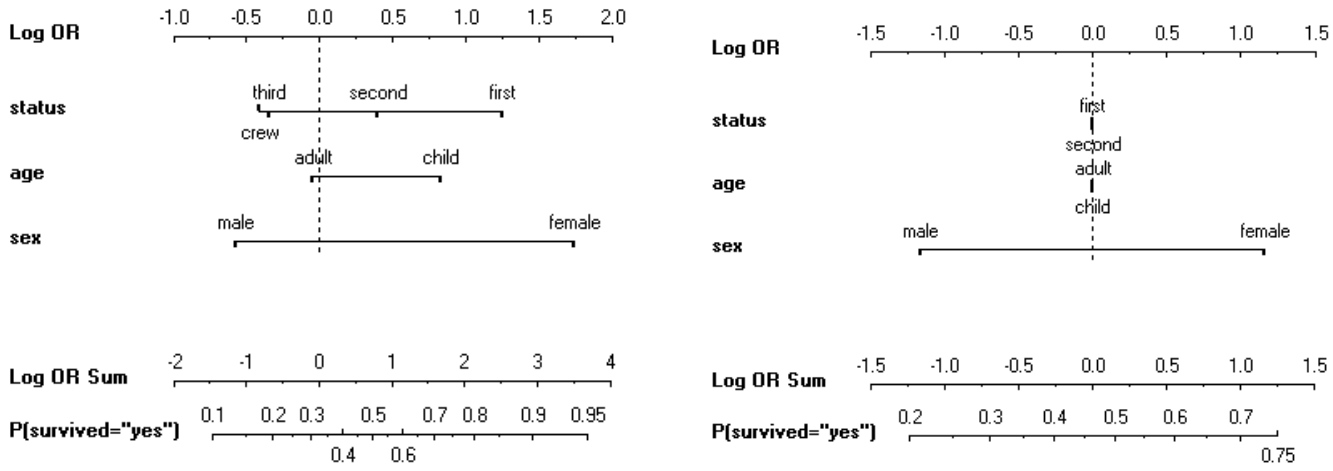

Figure 7: A naïve Bayesian nomogram (left) and the SVM nomogram for the 'Titanic' data set.

any information. The effect function is a full representation of the contribution of an individual attribute towards the probability of the outcome. The visualization is not a partial approximation to the SVM model: instead it captures the SVM model completely and exactly. Clearly, our intuitive conception of attribute 'importance' might not match that of the effect function. There are other ways of interpreting attribute importance that do not correspond to effect functions, perhaps the most popular of which is mutual information.

There are some pitfalls to interpreting the importance of attributes derived from the nomogram-based visualization of a support vector machine. We distinguish two distinct situations:

- If there are several highly correlated attributes, it is difficult to distinguish the individual effect of any attribute in particular. Instead, the effect functions will somewhat arbitrarily allocate the net effect among the attributes. If we hold the value of one attribute constant, the effects of other correlated attributes will decrease. This problem is referred to as a negative interaction or as *attribute redundancy* [13]. We illustrated this on an example in Fig. 7, where all attributes became irrelevant once the *sex* attribute was accounted for. It does not mean that the other attributes are irrelevant, just that *sex* takes their credit too.

- In some cases, an attribute A appears to have no effect. However, if we control the value of another attribute B, the effect of A will increase. The example of this phenomenon of a positive interaction or *attribute synergy* [13] are the familiar XOR and parity problems. We illustrated this on an example in Fig. 4, where the influence of the *employment* attribute increased if *residence_duration* was controlled for.

4. CONCLUSION

We have shown that support vector machines can be effectively visualized even in attribute spaces with many dimensions, using nomograms. Namely, individual attributes are stacked vertically in a nomogram, packing multiple dimensions into a single one. We have described the methodology for converting a support vector machine into the form of a generalized additive model. Furthermore, we have extended the form of a nomogram with two-dimensional graph representations of a nonlinear and non-monotonic effect function, as we have seen in Sect. 2.2. In addition to nonlinear univariate effects, we also show how interactions between attributes can be modelled and visualized.

We did not discuss the problem of determining what decomposable kernel to use in detail. There are three ways of addressing this. First, interaction analysis [14] is a heuristic that can aid the construction of kernels that capture the interactions. Secondly, we can see it as an issue of model selection. Finally, it is possible to express a preference for sparse and smooth kernels as a part of the optimization problem, combining the quest for decomposability and the actual learning [5, 20].

With the example 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.

KDD practitioners are often concerned with data sets that contain hundreds or thousands of attributes. Nomograms have no inherent problems with such situations: the dimensionality of the visualization depends on the structure of interactions, not on the number of attributes. To simplify the interpretation, the attributes should be arranged by importance, and the more important attributes would be examined first. Nomograms provide a measure of importance that is based on the length of the effect line: it indicates the range of the effects provided by the attribute. Although this measure should be weighted by the frequency of individual attribute values, it it nonetheless intuitive and useful.

An interesting question is also the stability of the model. The effect of a particular attribute can be thought as an uncertain quantity. To present the uncertainty, we can em-

ploy the notion of an error bar or a confidence interval. We obtain the error bars by training a separate SVM model for each bootstrap resample of the original data. Each separate model results in an effect function, and for each value of the attribute, we can obtain the lower and upper bound of the effect across the resamples. This yields the effect error bar.

Finally, all that was said about classification applies also to regression. The only difference is that the range of the dependent variable replaces the log-odds, and that the calibration is not required.

5. ACKNOWLEDGEMENTS

The authors are grateful to J. Brank for helpful advice. This work was supported by a grant from the Slovene Ministry of Education, Science and Sports and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

6. REFERENCES

[1] Y. Altun, A. Smola, and T. Hofmann. Exponential families for conditional random fields. In M. Chickering and J. Halpern, editors, *Proc. of 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 2–9, Banff, Alberta, Canada, July 2004.

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

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2005. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[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, Slovenia.

[5] S. R. Gunn and J. S. Kandola. Structural modelling with sparse kernels. *Machine Learning*, 48(1):137–163, 2002.

[6] T. L. Hankins. Blood, dirt, and nomograms: A particular history of graphs. *ISIS: Journal of the History of Science in Society*, 90:50–80, 1999.

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

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

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

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

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

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

[13] A. Jakulin and I. Bratko. Analyzing attribute dependencies. In N. Lavrač, D. Gamberger, H. Blockeel, and L. Todorovski, editors, *Proc. of Principles of Knowledge Discovery in Data (PKDD)*, volume 2838 of *LNAI*, pages 229–240. Springer-Verlag, September 2003.

[14] A. Jakulin and I. Bratko. Testing the significance of attribute interactions. In *Proc. of 21st International Conference on Machine Learning (ICML)*, pages 409–416, Banff, Alberta, Canada, 2004.

[15] 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.

[16] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In R. Greiner and D. Schuurmans, editors, *Proc. of 21st International Conference on Machine Learning (ICML)*, Banff, Alberta, Canada, 2004.

[17] 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.

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

[19] M. Možina, J. Demšar, M. W. Kattan, and B. Zupan. Nomograms for visualization of naive Bayesian classifier. In J.-F. Boulicaut, F. Esposito, and F. Giannotti, editors, *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 337–348, Pisa, Italy, 2004.

[20] K. Pelckmans, J.A.K. Suykens, and B. De Moor. Building sparse representations and structure determination on LS-SVM substrates. *Neurocomputing*, 64:137–159, March 2005.

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

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

[23] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 2nd edition, 1999.