Structured multicategory support vector machine with analysis of variance decomposition

By YOONKYUNG LEE

Department of Statistics, The Ohio State University, Columbus, Ohio 43210, U.S.A.

yklee@stat.ohio-state.edu

YUWON KIM

Statistical Research Center for Complex Systems, Seoul National University, Seoul 156-747, Korea

gary@stats.snu.ac.kr

SANGJUN LEE

Department of Statistics, Seoul National University, Seoul 156-747, Korea

seaphant@stats.snu.ac.kr

AND JA-YONG KOO

Department of Statistics, Korea University, Seoul 136-701, Korea

jykoo@korea.ac.kr

SUMMARY

The support vector machine has been a popular choice of classification method for many applications in machine learning. While it often outperforms other methods in terms of classification accuracy, the implicit nature of its solution renders the support vector machine less attractive in providing insights into the relationship between covariates and classes. Using structured kernels can remedy the drawback. Borrowing the flexible model building idea of functional analysis of variance decomposition, we consider multicategory support vector machines with analysis of variance kernels in this paper. An additional penalty is imposed on the sum of weights of functional subspaces, which encourages a sparse representation of the solution. Incorporation of the additional penalty enhances the interpretability of a resulting classifier with often improved accuracy. The proposed method is demonstrated through simulation studies and an application to real data.

Some key words: Classification; Feature selection; Linear programming; 1-norm penalty; Quadratic programming; Regularisation method; Reproducing kernel Hilbert space.

1. INTRODUCTION

A classification rule that maps the attributes of an individual to a class label is learned or estimated from a training dataset, a set of pairs of attributes and their known class memberships of individuals. The foremost goal of classification is to learn the prediction rule that attains the minimum error rate over novel cases. The support vector machine is a classification method which has been widely used recently in machine learning; see Vapnik (1998), Cristianini & Shawe-Taylor (2000), Schölkopf & Smola (2002) and references therein. The popularity of the support vector machine is in part ascribed to its versatility and competitive classification accuracy as demonstrated in many applications. The main aspect of flexibility when estimating possibly nonlinear classification boundaries is the embedding of attributes or variables into a high-dimensional feature space, where hyperplanes may well separate instances from different classes. However, such embeddings seldom need to be explicit in support vector machine applications. Rather, the support vector machine solution is expressed as a linear combination of the data representers determined by a chosen kernel function. Hence, for a nonlinear kernel function, the resulting classifier is given as a black box function, which does not allow for a clear interpretation of the importance of each variable to the final classifier. Identification of the variables that are predictive of the response is often crucial. It would be valuable to be able to achieve comparable classification accuracy, and at the same time to choose relevant features, i.e. variables or their transformations selectively.

To enhance the interpretability of the support vector machine, we propose structured learning through functional analysis of variance decomposition. For a general treatment of classification problems, we consider the multicategory support vector machine, an extension of the binary support vector machine proposed by Lee et al. (2004). It is important to note the distinction between hard and soft classification in order to illuminate the connection between this work and the existing literature. Soft classification refers to classification through estimation of the probability of each class, given attributes. For example, logistic regression is a method of soft classification. In contrast, the support vector machine provides hard classification, in which the probability estimation is not of primary interest. For more discussion about soft and hard classification, see Wahba (2002). In parallel to the work of Wahba et al. (1994), which addresses structured learning for soft classification via smoothing-spline analysis of variance, this paper presents structured learning for hard classification. As studied and suggested by Kohavi & John (1997), filtering informative attributes marginally may not be as efficient as wrapping the selection operation simultaneously in learning. In this paper, for feature selection we incorporate an additional penalty of ℓ_1 nature on the sum of weights of functional components with the multicategory support vector machine. Gunn & Kandola (2002), and in a technical report from the University of Wisconsin at Madison, Y. Lin and H. Zhang used the idea of selecting features by this component penalty for regression, generalising the LASSO to nonlinear function space generated by the spline analysis of variance kernel. The extra penalty makes the solution in the form of an expansion of functional components more compact and lucid. Just as in the LASSO of Tibshirani (1996) and the basis pursuit method of Chen et al. (1999) for regression, the ℓ_1 penalty effects selection by shrinking the weights of less predictive or redundant components to zero.

For the binary linear support vector machine, Bradley & Mangasarian (1998) demonstrated the utility of the ℓ_1 penalty for feature selection, and Weston et al. (2003) further introduced the ℓ_0 penalty. We also note that there are other feature selection methods for the binary support vector machine, whose formulations are not based on the ℓ_1 penalty; the recursive feature elimination method for the linear support vector machine on the basis of sensitivity analysis by Guyon et al. (2002) is an example. To handle nonlinear support vector machines, Weston et al. (2001) and Chapelle et al. (2002) suggested an alternative approach of introducing a scale factor for each variable, keeping the embedding implicit through a choice of kernel function. They then treated the scale factors as further tuning parameters, and chose them by minimising generalisation error bounds as functions of a subset of variables via gradient descent. Grandvalet & Canu (2003) further integrated variable rescaling into the support vector machine formulation with some sparsity constraints. The optimisation problems involving scaling factors are computationally more challenging than the original problem. In contrast, our formulation with analysis of variance decomposition does not entail additional complexity except that a linear programming problem needs to be solved at intermediate steps. Above all, our proposed method is a structured approach for identifying a parsimonious subset of features that are relevant to classification without compromising the classification accuracy. It handles both linear and nonlinear features in a principled way for general multiclass problems. As a result, it can be applied to a wide range of problems, and can provide insights into the relationship between attributes and the response for a particular classification problem.

2. FUNCTIONAL ANALYSIS OF VARIANCE DECOMPOSITION

2·1. *Functional analysis of variance*

Here we review the functional analysis of variance decomposition as a structured representation of a multivariate function for describing a relationship f between p covariates $x = (x_1, \ldots, x_p)$ and the response y, where $x \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$ with $x_\alpha \in \mathcal{X}_\alpha$.

As generalisation of the classical analysis of variance decomposition of a function defined on a discrete domain, the analysis of variance decomposition of a function f is given by

$$f(x) = b + \sum_{\alpha=1}^{p} f_\alpha(x_\alpha) + \sum_{\alpha<\beta} f_{\alpha\beta}(x_\alpha, x_\beta) + \cdots , \tag{1}$$

where b is a constant, and the functional components f_S for $S \subseteq \{1, \ldots, p\}$ satisfy side conditions for identifiability. The component f_α can be interpreted as the main effect of x_α, $f_{\alpha\beta}$ as the two factor interaction of x_α and x_β, and so on. For simplicity and elucidation of f, the analysis of variance decomposition is truncated after lower-order interaction terms in practice because, as the order of interaction terms increases, accurate estimation of them becomes increasingly more difficult because of the curse of dimensionality (Bellman, 1961). For example, by restricting the effects of x to be additive, we have additive models of the form $f(x) = b + f_1(x_1) + \cdots + f_p(x_p)$, as introduced by Hastie & Tibshirani (1986). The smoothing-spline analysis of variance models (Wahba, 1990; Gu, 2002) are another family of multivariate function estimation methods based on functional analysis of variance decomposition. Wahba et al. (1994) and X. Lin, in a 1998 University of Wisconsin, Madison Ph. D. thesis discussed the smoothing-spline analysis of variance approach to logistic regression for soft classification in the dichotomous case and the polytomous case, respectively. The method is a regularisation approach with roughness penalty imposed on functions in a reproducing kernel Hilbert space. It is well known that the support vector machine can be cast as a regularisation method in a reproducing kernel Hilbert space (Wahba, 1998; Evgeniou et al., 2000). This connection to the reproducing kernel Hilbert space method makes structured support vector machine learning through the smoothing-spline analysis of variance a natural extension for hard classification.

2·2. *Reproducing kernel Hilbert space and component selection*

We briefly describe a smooth function space that facilitates the analysis of variance decomposition in (1). The function f is assumed to be in $\mathcal{H}$, a reproducing kernel Hilbert space of functions defined on $\mathcal{X}$. Details about reproducing kernel Hilbert spaces and their general properties can be found in Aronszajn (1950). The space $\mathcal{H}$ is constructed as a tensor product of functional subspace $\mathcal{H}_\alpha$, a reproducing kernel Hilbert space of functions on $\mathcal{X}_\alpha$ for $\alpha = 1, \ldots, p$. It is further decomposed as $\{1\} \oplus \bar{\mathcal{H}}_\alpha$, where $\bar{\mathcal{H}}_\alpha$ is the subspace of $\mathcal{H}_\alpha$ orthogonal to $\{1\}$. The space $\mathcal{H}$ is given by

$$\mathcal{H} = \otimes_{\alpha=1}^{p}(\{1\} \oplus \bar{\mathcal{H}}_\alpha) = \{1\} \oplus \sum_{\alpha=1}^{p} \bar{\mathcal{H}}_\alpha \oplus \sum_{\alpha<\beta}(\bar{\mathcal{H}}_\alpha \otimes \bar{\mathcal{H}}_\beta) \oplus \cdots . \tag{2}$$

Truncation of subspaces for higher-order interactions yields the corresponding simplification of $f \in \mathcal{H}$. Relabel the remaining subspaces as $\mathcal{F}_\nu$, $\nu = 1, \ldots, d$, after truncation and let the resulting reproducing kernel Hilbert space be $\mathcal{F} = \{1\} \oplus \bar{\mathcal{F}}$, where $\bar{\mathcal{F}} = \oplus_{\nu=1}^{d} \mathcal{F}_\nu$. Suppose that $f \in \mathcal{F}$. Then f is represented as a sum of functional components, each of which is an element of the corresponding subspace of $\mathcal{F}$. Using $\mathcal{F}$, we sketch the regularisation approach in general terms. Let $\mathcal{L}$ denote a loss function and let $\mathcal{T} = \{(x_i, y_i), i = 1, \ldots, n\}$ be a training dataset of n observations. A regularisation method finds $\hat{f} \in \mathcal{F}$ so as to minimise

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\{y_i, f(x_i)\} + \lambda \sum_{\nu} \theta_\nu^{-1} \|P^\nu f\|^2,$$

where $\|\cdot\|$ is the norm defined on the reproducing kernel Hilbert space $\mathcal{F}$, P^ν is the orthogonal projection operator on to $\mathcal{F}_\nu$, and $\theta_\nu \geq 0$. If $\theta_\nu = 0$, the minimiser is taken to satisfy $\|P^\nu f\|^2 = 0$. The scalar λ is a tunable parameter which balances the empirical risk and the penalty associated with f. The penalty functional $J(f) = \sum_\nu \theta_\nu^{-1}\|P^\nu f\|^2$ with rescaling parameters θ_ν's entails the following reproducing kernel for $\bar{\mathcal{F}}$ (Wahba 1990, Ch. 10):

$$K(s,t) = \sum_{\nu=1}^{d} \theta_\nu K_\nu(s,t) \text{ for } s,t \in \mathcal{X}, \tag{3}$$

where K_ν is the reproducing kernel of $\mathcal{F}_\nu$. Tuning the θ_ν's amounts to rescaling of the component spaces $\mathcal{F}_\nu$, and the model complexity is controlled through a set of θ_ν values as well as λ.

The expression in (3) as a sum of component reproducing kernels weighted by θ_ν values allows a systematic way of selecting the most relevant components to y. By imposing an additional penalty on the sum of these weights, we can further force those components or features with negligible weights to be zero. Motivated by the LASSO method in linear models that produces sparse solutions, in their technical report Y. Lin and H. Zhang proposed the following COmponent Selection and Smoothing Operator, COSSO, in smoothing spline regression: find $\hat{f} \in \mathcal{F}$ to minimise

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\{y_i - f(x_i)\}^2 + \lambda \sum_{\nu} \theta_\nu^{-1} \|P^\nu f\|^2 + \lambda_\theta \sum_{\nu} \theta_\nu, \\ &\text{subject to } \theta_\nu \geq 0, \text{ for } \nu = 1, \ldots, d. \end{aligned} \tag{4}$$

Here the θ_ν's are obtained as part of the minimiser of (4). They showed that finding $\hat{f}$ to minimise (4) is equivalent to finding the minimiser of

$$\frac{1}{n}\sum_{i=1}^{n}\{y_i - f(x_i)\}^2 + \tau \sum_{\nu} \|P^\nu f\| \tag{5}$$

with $\tau = 2\sqrt{\lambda\lambda_\theta}$, and the method reduces to the LASSO in linear models. For a general set-up, the squared error can be replaced by other proper loss functions $\mathcal{L}$.

Two papers closely related to this formulation, namely Micchelli & Pontil (2005) and

Argyriou et al. (2005), as well as other references therein, were brought to our attention by a referee. Although the main motivation of the regularisation problem (4) is simultaneous function fitting and feature selection, it can be cast as a variational problem of learning an optimal kernel configuration in the convex hull of prescribed kernels as studied in the papers. Their general treatment allows for a broader perspective on this sparse kernel approach to feature selection with the analysis of variance kernels, and they provide rigorous characterisation of the optimal kernel as a solution to a saddlepoint problem.

3. SUPPORT VECTOR MACHINES WITH ANALYSIS OF VARIANCE DECOMPOSITION

3·1. *The binary case*

In the classification problem, the response y_i denotes the class that the ith instance falls into, where $y_i \in \{1, \ldots, k\}$, for prescribed k. Using the training sample $\mathcal{T}$, we want to construct a classification rule $\phi : \mathcal{X} \to \{1, \ldots, k\}$ that can generalise the relationship between x_i and the class label y_i to novel instances.

In the binary case with $k = 2$, the class labels y_i are coded either 1 or -1. Instead of finding a category-valued mapping ϕ directly, the support vector machine looks for a real valued function f which will induce a classification rule via $\phi(x) = \text{sign}\{f(x)\}$.

For structured representation of f, we consider the analysis of variance decomposition corresponding to functional subspaces in (2). As illustrated in §2, a truncated sum of subspaces is prespecified for f. Suppose that $f = b + h \in \{1\} \oplus \bar{\mathcal{F}}$. Then h can be expressed as $\sum_{\nu=1}^{d} h_\nu$ with $h_\nu \in \mathcal{F}_\nu$. In addition to the decomposition, we prefer f to be only in terms of informative covariates. This can be achieved by introducing the rescaling parameter θ_ν for $\mathcal{F}_\nu$ and imposing the ℓ_1 component penalty on the sum of the parameters as in (4). The component penalty encourages elimination of irrelevant features, and hence provides a sparse and succinct description. Modifying the standard support vector machine with the component penalty, we seek to find $\hat{f}$ that minimises

$$\frac{1}{n}\sum_{i=1}^{n}\{1-y_i f(x_i)\}_{+}+\lambda\sum_{\nu=1}^{d}\theta_\nu^{-1}\|P^\nu h\|^2+\lambda_\theta\sum_{\nu=1}^{d}\theta_\nu, \tag{6}$$
$$\text{subject to } \theta_\nu \geq 0, \text{ for } \nu=1,\ldots,d,$$

where $(z)_+ = \max(z,0)$ for $z \in \mathbb{R}$. The dependence of $\hat{f}$ on $(\lambda, \lambda_\theta)$ is suppressed for conciseness. Note that the tuning parameters $(\lambda, \lambda_\theta)$ and the rescaling parameters $\theta = (\theta_1, \ldots \theta_d)^\top$ are not uniquely defined. Any λ, λ_θ and θ with the same values of λ/θ_ν and $\lambda_\theta\theta_\nu$ for $\nu = 1, \ldots, d$ are equivalent. Thus, the θ_ν's can be scaled arbitrarily. Without loss of generality, we will assume that they are bounded above by 1 when implementing the proposed method. $\theta = (1, \ldots, 1)^\top$ corresponds to no change from the original set of features. The effect of feature selection and shrinkage will be gauged in comparison to this benchmark. It was noted by a referee that the above formulation can be simplified by using the constraint $\sum_\nu \theta_\nu = 1$, which would result in only one tuning parameter λ. This amounts to renormalisation of the θ_ν's by their sum, so that θ determines a convex combination of kernel functions just as in Micchelli & Pontil (2005). Then, λ simultaneously regularises both the coefficients and the rescaling parameters. However, we will keep the above formulation to make use of λ_θ for a more direct handle on the magnitudes of the rescaling parameters after the overall complexity of $\hat{f}$ is controlled by λ.

Note that the support vector machine is an example of regularisation methods in a reproducing kernel Hilbert space with the so-called hinge loss function $\mathcal{L}\{y, f(x)\} = \{1 - yf(x)\}_+$. Thus, by the representer theorem (Wahba, 1990), its solution admits a finite-dimensional representation. For fixed θ, substituting the rescaled reproducing kernel (3) into the finite-dimensional representation of $\hat{f}$ gives

$$\hat{f}(x) = b + \sum_{i=1}^{n} c_i \sum_{\nu=1}^{d} \theta_\nu K_\nu(x_i, x), \tag{7}$$

which yields $\hat{h}_\nu(x) = \theta_\nu \sum_{i=1}^{n} c_i K_\nu(x_i, x)$ as the νth functional component. In connection with (5), the same argument in the report by Y. Lin and H. Zhang applies to verify that the structured support vector machine in (6) is equivalent to the following regularisation problem with an intermediate functional norm: find the minimiser of

$$\frac{1}{n}\sum_{i=1}^{n}\{1 - y_i f(x_i)\}_+ + \tau \sum_{\nu=1}^{d} \|P^\nu h\|. \tag{8}$$

Suppose that $\mathcal{X} = [0,1]^p$, $\mathcal{F}$ is the direct sum of $\mathcal{F}_\nu = \{x_\nu - 1/2\}, \nu = 1, \ldots, p$, the subspaces of additive linear models, and $\mathcal{F}_\nu$ is equipped with the inner product $(f, g) = \int fg$. Let $f(x) = \beta_0 + \sum_{\nu=1}^{p} \beta_\nu x_\nu$. Then $f \in \mathcal{F}$ and the penalty functional $J(f) = \sum_{\nu=1}^{p} \|P^\nu h\|$ is proportional to $\sum_{\nu=1}^{p} |\beta_\nu|$. Thus, the above general formulation (8) subsumes the ℓ_1-norm-based feature selection for the linear support vector machine proposed by Bradley & Mangasarian (1998).

3·2. *Computation of the structured support vector machine*

Let $\mathcal{K}_\nu$ stand for n by n matrix with the (l, m)th entry $K_\nu(x_l, x_m)$ and set $c = (c_1, \ldots, c_n)^\top$. By the reproducing property and (7), $\sum_{\nu=1}^{d} \theta_\nu^{-1} \|P^\nu \hat{h}\|^2 = c^\top \left(\sum_{\nu=1}^{d} \theta_\nu \mathcal{K}_\nu\right) c$. Given θ, let $\mathcal{K}_\theta = \sum_{\nu=1}^{d} \theta_\nu \mathcal{K}_\nu$. Then the structured support vector machine in (6) can be rewritten as a finite-dimensional problem of finding θ and (b, c) that minimise

$$\begin{aligned} \Phi(\theta, b, c) &= \frac{1}{n} e^\top \Big\{ e - Y(be + \mathcal{K}_\theta c) \Big\}_+ + \lambda c^\top \mathcal{K}_\theta c + \lambda_\theta \sum_{\nu=1}^{d} \theta_\nu, \\ &\text{subject to } \theta_\nu \geq 0, \text{ for } \nu = 1, \ldots, d, \end{aligned} \tag{9}$$

where e is the vector of n ones, $Y = \mathrm{diag}(y_1, \ldots, y_n)$, and $(z)_+$ is the vector with ith coordinate $(z_i)_+$ for $z = (z_1, \ldots, z_n)^\top$. This finite-dimensional problem associated with the optimisation criterion (6) involves (b, c) and θ jointly. However, its inherent structure renders it natural to carry out alternating minimisation as proposed in Y. Lin and H. Zhang's report for regression. The alternating approach gives rise to two well-defined convex optimisation problems referred to below as the c-step and the θ-step. The more general problem of learning the optimal kernel in Micchelli & Pontil (2005) is formulated by adding another layer of optimisation over θ, which amounts to the kernel configuration, to the c-step of regularisation based on a given kernel. Consider the following iterative scheme for finding $\hat{f}$. After initialising $\theta^{(0)} = (1, \ldots, 1)^\top$ and $(b^{(0)}, c^{(0)}) = \mathrm{argmin}\, \Phi(\theta^{(0)}, b, c)$, we alternate evaluation of θ and (b, c) given λ and λ_θ, as follows. At the mth stage ($m = 1, 2, \ldots$), carry out the following double step:

in the θ-step, find $\theta^{(m)}$ to minimise $\Phi(\theta, b^{(m-1)}, c^{(m-1)})$ with $(b^{(m-1)}, c^{(m-1)})$ fixed;

in the c-step, find $(b^{(m)}, c^{(m)})$ to minimise $\Phi(\theta^{(m)}, b, c)$ with $\theta^{(m)}$ fixed.

When the rescaling parameters θ are fixed, the nonnegativity constraint on θ is irrelevant, as is the last term in (9). Thus, the c-step problem reduces to the ordinary support vector machine with the reproducing kernel rescaled by θ. The θ-step of shrinkage and selection is reminiscent of the nonnegative garrote in a parametric setting by Breiman (1995), and is essential for feature selection. For the optimisation problem defined by the θ-step, we introduce nonnegative slack variables denoted by $\xi = (\xi_1, \ldots, \xi_n)^\top$ for the truncation function $(\cdot)_+$ in (9). In terms of the slack variables, the θ-step optimisation is to choose θ for fixed b

and c, to minimise

$$\begin{aligned}
\Phi_{b,c}(\theta,\xi) &= \frac{1}{n}e^\top\xi + \sum_{\nu=1}^{d}\theta_\nu\left(\lambda c^\top\mathcal{K}_\nu c + \lambda_\theta\right), \\
\text{subject to} \qquad & e - Y(be + \sum_{\nu=1}^{d}\theta_\nu\mathcal{K}_\nu c) \le \xi, \\
& \xi \ge 0, \text{ and } \theta_\nu \ge 0, \text{ for } \nu = 1,\ldots,d.
\end{aligned}$$

This is a linear programming problem in θ and ξ. Denoting the minimiser at the mth step by $\hat{f}^{(m)}$, we observe that $\hat{f}^{(0)}$ is the ordinary support vector machine solution with $\theta^{(0)}$. We now mention some properties of $\hat{f}^{(m)}$ as generated by the alternating algorithm.

Theorem 1. *Given λ and λ_θ, the algorithm yields a sequence of $\hat{f}^{(m)}$ with feasible $(\theta^{(m)}, b^{(m)}, c^{(m)})$ and nonincreasing $\Phi(\theta^{(m)}, b^{(m)}, c^{(m)})$. That is, $\Phi(\theta^{(m+1)}, b^{(m+1)}, c^{(m+1)}) \le \Phi(\theta^{(m)}, b^{(m)}, c^{(m)})$. For strictly positive definite $\mathcal{K}_\nu$, $\nu = 1,\ldots,d$, and nonzero $\theta^{(m+1)}$, the equality holds only if $c^{(m)} = c^{(m+1)}$.*

Corollary 1. *Given λ and λ_θ, the sequence of $\Phi(\theta^{(m)}, b^{(m)}, c^{(m)})$ generated by the algorithm converges as $m \to \infty$.*

Proofs are straightforward and can be found in an Ohio State University technical report by the authors. A formal proof of convergence of the arguments $(\theta^{(m)}, b^{(m)}, c^{(m)})$ is not pursued here. This would require strict conditions on $\mathcal{K}_\theta$ and the uniqueness of solutions to the θ-step linear programming problem and determination of b. Numerical studies showed that the iterative algorithm typically gives convergent solutions in a few steps, and often taking a one-step update was in practice sufficient for reaching the final approximate solution. A similar observation was made by Y. Lin and H. Zhang in their report.

3·3. *The multicategory case*

For the multicategory case, we adopt the extension in the paper by Lee et al. (2004), which retains good theoretical properties of the binary support vector machine. For a k-category problem, the multicategory support vector machine attempts to find a k-tuple of separating functions $f = (f^1, \ldots, f^k)$ with the zero-sum constraint, $\sum_{j=1}^{k} f^j(x) = 0$ for any $x \in \mathcal{X}$, which induces a classifier $\phi(x) = \text{argmax}_{j=1,\ldots,k} f^j(x)$. Throughout this paper, superscripts are used to indicate coordinates. A vector-valued class code is introduced in place of the nominal class label, and when appropriate, $y_i = (y_i^1, \ldots, y_i^k)$ denotes a vector with $y_i^j = 1$ and $-1/(k-1)$ elsewhere if the ith observation falls into class j. Just as the class code vector y_i contrasts the coordinate of 1 to the rest that are $-1/(k-1)$, f with the zero-sum constraint is designed to contrast $\max_j f^j$ to the rest of the f^j's. Also, $L(y_i) = (L_{y_i}^1, \ldots, L_{y_i}^k)$ is a k-dimensional misclassification cost vector, where $L_j^{j'}$ is the cost of misclassifying j as j'. The extended hinge loss function $\mathcal{L}\{y_i, f(x_i)\} = L(y_i)\{f(x_i) - y_i\}_+$ can be written explicitly as $\mathcal{L}\{y_i, f(x_i)\} = \sum_{j=1}^{k} L_{y_i}^j \{f^j(x_i) - y_i^j\}_+$. When the misclassification costs are equal, that is, $L_j^{j'} = I(j \neq j')$, it is simplified to $\mathcal{L}\{y_i, f(x_i)\} = \sum_{j \neq y_i} \{f^j(x_i) + 1/(k-1)\}_+$.

In parallel to the binary case, structured representation of each f^j of f is considered by using the functional analysis of variance decomposition; that is, if $f^j = b^j + h^j \in \{1\} \oplus \bar{\mathcal{F}}$, then $h^j = \sum_{\nu=1}^{d} h_\nu^j$ with $h_\nu^j \in \mathcal{F}_\nu$. By the ℓ_1-type penalty on θ_ν to encourage a sparse representation of each f^j in terms of its components, the multicategory support vector machine is modified to find $\hat{f}$, with the zero-sum constraint to minimise

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i)\{f(x_i) - y_i\}_+ + \frac{\lambda}{2}\sum_{j=1}^{k}\left(\sum_{\nu=1}^{d}\theta_\nu^{-1}\|P^\nu h^j\|^2\right) + \lambda_\theta \sum_{\nu=1}^{d}\theta_\nu, \quad (10)$$
$$\text{subject to } \theta_\nu \geq 0, \text{ for } \nu = 1, \ldots, d.$$

The method will be referred to as the structured multicategory support vector machine hereafter. When the reproducing kernel is rescaled by θ, by the multicategory version of the representer theorem proved in Lee et al. (2004), each coordinate of $\hat{f}$ is given by

$$\hat{f}^j(x) = b^j + \sum_{i=1}^{n} c_i^j \sum_{\nu=1}^{d} \theta_\nu K_\nu(x_i, x), \tag{11}$$

with $\hat{h}_\nu^j(x) = \theta_\nu \sum_{i=1}^n c_i^j K_\nu(x_i, x)$ as its νth functional component. Note that the same rescaling parameters θ_ν are used for each h^j. As a result, an equivalence between (10) and its COSSO-type formulation given by

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i)\{f(x_i) - y_i\}_+ + \tau \sum_{j=1}^{k}\sum_{\nu=1}^{d} \|P^\nu h^j\| \tag{12}$$

may not be established in general, in contrast to the binary case. This can be explained as follows. Following the arguments in Lemma 2 of Lin and Zhang's report, we can verify that, for each ν,

$$\begin{aligned}(\lambda/2)\theta_\nu^{-1}\sum_{j=1}^{k}\|P^\nu h^j\|^2 + \lambda_\theta\theta_\nu \ &\geq\ (\lambda/2)\theta_\nu^{-1}\Big(\sum_{j=1}^{k}\|P^\nu h^j\|\Big)^2/k + \lambda_\theta\theta_\nu \\ &\geq\ 2\sqrt{\{(\lambda/2)(\lambda_\theta/k)\}}\sum_{j=1}^{k}\|P^\nu h^j\|.\end{aligned}$$

Equality in the first inequality holds if and only if $\|P^\nu h^1\| = \cdots = \|P^\nu h^k\|$, and therefore, the COSSO interpretation of the penalty terms in (10) is possible only in rare cases when these equality conditions are met for all ν. If each h^j is allowed to have different rescaling parameters θ_ν^j in place of θ_ν, say, and $\sum_{\nu=1}^d \theta_\nu$ in (10) is replaced with $\sum_{j=1}^k\sum_{\nu=1}^d \theta_\nu^j$, then the equivalence between the two formulations (10) and (12) can be shown analogously. However, such generality in the rescaling parameters would seldom be necessary. Thus, numerical studies in this paper are based on (10) using the same θ_ν's for each h^j.

To describe the necessary computation, let L^j denote the jth coordinates of the n misclassification cost vectors, $(L^j_{y_1}, \ldots, L^j_{y_n})^\top$ and define $y^j = (y^j_1, \ldots, y^j_n)^\top$. Let the coefficient vector $c^j = (c^j_1, \ldots, c^j_n)^\top$, for $j = 1, \ldots, k$, let $b = (b^1, \ldots, b^k)^\top$, and let $C = (c^1, \ldots, c^k)$. By the same argument as in the binary case, we can rewrite the structured multicategory support vector machine in (10) as a finite-dimensional problem of finding θ and (b, C) that minimise

$$\Phi(\theta, b, C) = \frac{1}{n}\sum_{j=1}^{k}(L^j)^\top\left(b^j e + \mathcal{K}_\theta c^j - y^j\right)_+ + \frac{\lambda}{2}\sum_{j=1}^{k}(c^j)^\top\mathcal{K}_\theta c^j + \lambda_\theta\sum_{\nu=1}^{d}\theta_\nu, \tag{13}$$

$$\text{subject to } \; \textstyle\sum_{j=1}^{k}\left(b^j e + \mathcal{K}_\theta c^j\right) = 0 \text{ and} \tag{14}$$
$$\theta_\nu \geq 0, \text{ for } \nu = 1, \ldots, d.$$

Analogously, the solution is obtained by alternating the c-step and θ-step of the redefined objective function $\Phi(\theta, b, C)$. Given θ, the c-step is simply that for the ordinary multicategory support vector machine with the rescaled kernel $\mathcal{K}_\theta$, whose dual formulation leads to a quadratic programming problem with $n(k-1)$ unknowns. To derive the θ-step optimisation problem, let $\xi^j = (\xi^j_1, \ldots, \xi^j_n)^\top$, for $j = 1, \ldots, k$, denote nonnegative slack variables and $\xi = (\xi^1, \ldots, \xi^k)$. Then the θ-step is given by a linear programming problem of finding θ, for fixed b and C, to minimise

$$\Phi_{b,C}(\theta, \xi) = \frac{1}{n}\sum_{j=1}^{k}(L^j)^\top\xi^j + \sum_{\nu=1}^{d}\theta_\nu\left\{\frac{\lambda}{2}\sum_{j=1}^{k}(c^j)^\top\mathcal{K}_\nu c^j + \lambda_\theta\right\}, \tag{15}$$

$$
\text{subject to} \qquad b^j e + \sum_{\nu=1}^{d} \theta_\nu \mathcal{K}_\nu c^j - y^j \le \xi^j, \ \text{for } j = 1, \ldots, k,
$$
$$
\xi^j \ge 0, \ \text{for } j = 1, \ldots, k, \ \text{and } \theta_\nu \ge 0, \ \text{for } \nu = 1, \ldots, d.
$$

Note that the zero-sum constraint (14) is satisfied by any θ once the c-step is carried out, so it becomes irrelevant at the θ-step. This additional linear programming problem for the θ-step involves $n(k-1)+d$ unknowns. Our current implementation of the θ-step is based on a simplex method, whose computational complexity in practice is approximately polynomial in the number of unknowns. The θ-step computing time relative to that of c-step depends on the ratio of the number of features to the sample size in general. Theorem 1 and Corollary 1 hold true also for the structured multicategory support vector machine with b and c replaced with their multicategory counterparts. The re-expression of the c-step and θ-step optimisation problems in the standard form of quadratic and linear programmes, respectively, can be found in our technical report.

3·4. *Reproducing kernels and choice of tuning parameters*

The choice of a reproducing kernel determines basis functions in which the solution is expanded as in (11). In principle, any positive definite function K can be chosen as a reproducing kernel. However, we consider only flexible and structured kernels that facilitate the analysis of variance decomposition. Since reproducing kernels are closed under tensor summation and multiplication, it is often sufficient to define a reproducing kernel for a univariate as a building block of the basis functions. For example, the spline kernel on the unit interval $[0,1]$ is $K(s,t) = k_1(s)k_1(t) + k_2(s)k_2(t) - k_4(|s-t|)$ for s and $t \in [0,1]$, where $k_1(t) = t - 1/2$, $k_2(t) = \{k_1^2(t) - 1/12\}/2$, and $k_4(t) = \{k_1^4(t) - k_1^2(t)/2 + 7/240\}/24$; see Wahba (1990) for more details of the spline kernel. Whenever necessary, one can transform a covariate so that it lies in $[0,1]$ via $x'_\alpha = \{x_\alpha - \min(x_\alpha)\}/\{\max(x_\alpha) - \min(x_\alpha)\}$, where $\min(x_\alpha)$ and $\max(x_\alpha)$

are the minimum and the maximum values of the covariate in the training dataset. Relaxing the mathematical formality of the orthogonal decomposition in (2), we can also use the univariate versions of popular kernels in machine learning such as the Gaussian kernel in practice for an analogous decomposition. However, some modification is necessary to ensure the identifiability of the functional components through empirical averaging operators in this case.

We choose the values of the tuning parameters λ and λ_θ so as to minimise the prediction error determined by a loss function. In simulation settings, the average expected loss can be taken as a tuning criterion, where the expectation is taken over the distribution of unobserved Y_i conditional on the observed covariates x_i, $i = 1, \ldots, n$. The misclassification rate is an example of the prediction error under the 0-1 loss $\mathcal{L}\{y, f(x)\} = I[y \neq \text{argmax}_{j=1,\ldots,k} f^j(x)]$, and the generalised comparative Kullback-Leibler distance with respect to the hinge loss (Lee et al., 2004) is another theoretically possible criterion. In practice, data-based estimation of the prediction error is necessary. Five- or ten-fold crossvalidation with either the 0-1 loss or the hinge loss is considered in this paper.

Here we summarise a one-step update procedure that alternates tuning of λ at the c-step and λ_θ at the θ-step. When there is a tunable parameter in the kernel functions, we need to tune it jointly with λ in the c-step. Let $\hat{E}$ denote a generic estimate of prediction error as a function of λ and λ_θ. It could be a theoretically available tuning measure in simulation, or a data-dependent tuning measure $\hat{E}_{\mathcal{T}}$ in practice, emphasising its dependence on the training dataset $\mathcal{T}$. The procedure consists of the following steps.

Step 1. Initialise as follows:

for the θ-step, initialise $\hat{\theta}^{(0)} = (1, \ldots, 1)^\top$;

for the c-step, find the initial multicategory support vector machine solution $(\hat{b}^{(0)}, \hat{C}^{(0)})$ that minimises $\Phi(\hat{\theta}^{(0)}, b, C)$ in (13) at $\hat{\lambda}^{(0)}$, which is a minimiser of $\hat{E}(\lambda)$.

Step 2. The first update is as follows:

> for the θ-step, find the rescaling parameters $\hat{\theta}^{(1)}$ to minimise $\Phi(\theta, \hat{b}^{(0)}, \hat{C}^{(0)})$ at $\hat{\lambda}_\theta^{(1)}$, a minimiser of $\hat{E}(\lambda_\theta)$;
>
> for the c-step, find the one-step updated solution $(\hat{b}^{(1)}, \hat{C}^{(1)})$ to minimise $\Phi(\hat{\theta}^{(1)}, b, C)$ at $\hat{\lambda}^{(1)}$, a new minimiser of $\hat{E}(\lambda)$.

Tuning in the above procedure is myopic in the sense that we choose the optimal value of one parameter at each step assuming that all the other parameters estimated from the previous step are fixed. For crossvalidated $\hat{E}$, tuning at the first θ-step needs a bit more clarification. Suppose that we use five-fold crossvalidation. For a random split of the training dataset $\mathcal{T}$ into five disjoint subsets, let $\mathcal{T}^{(-j)}$ denote the complement of the jth subset in $\mathcal{T}$ $(j = 1, \ldots, 5)$. Then $\hat{\lambda}_\theta$ is a minimiser of $\hat{E}(\lambda_\theta) = (1/5)\sum_{j=1}^{5} \hat{E}_{\mathcal{T}^{(-j)}}(\lambda_\theta)$, where $\hat{E}_{\mathcal{T}^{(-j)}}(\lambda_\theta)$ requires the fitted $(\hat{b}^{(0)}, \hat{C}^{(0)})$ based on $\mathcal{T}^{(-j)}$ only at $\hat{\lambda}^{(0)}$. Then $\hat{\theta}^{(1)}$ is obtained by using the entire training data at the chosen $\hat{\lambda}_\theta$. One may use the same split of the training dataset at each c-step and θ-step for crossvalidation.

A variation of the above one-step procedure is to simplify the first update by merging the tuning of λ and λ_θ. The idea is to tune λ_θ only by looking one step further from the first θ-step and fixing λ at the same $\hat{\lambda}^{(0)}$ of the initial c-step. This modified procedure with combined tuning has the following update scheme after initialisation; for each λ_θ, compute $\hat{\theta}^{(1)}$ and $(\hat{b}^{(1)}, \hat{C}^{(1)})$ corresponding to $\hat{\theta}^{(1)}$ at the subsequent c-step with λ fixed at $\hat{\lambda}^{(0)}$, and choose $\hat{\lambda}_\theta^{(1)}$ so that it minimises $\hat{E}(\lambda_\theta)$ as a function of $(\hat{b}^{(1)}, \hat{C}^{(1)})$.

4. NUMERICAL STUDY

4·1. *A two dimensional three-class example*

All the analyses in §4 used the R packages, `quadprog` and `lpSolve`. The alternating algorithm has been implemented and is available at the first author's webpage http://www.stat.osu.edu/∼yklee. To demonstrate feature selection, we consider a three-class toy example in which two covariates $x = (x_1, x_2)$ are uniformly distributed on the unit square $[0,1] \times [0,1]$, and only x_1 is relevant to the response ($y = 1, 2, 3$). Let the conditional probabilities of each class given x be $p_1(x) = 0.97\exp(-3x_1)$, $p_3(x)\exp\{-2.5(x_1 - 1.2)^2\}$, and $p_2(x) = 1 - p_1(x) - p_3(x)$ as a function of x_1 only. The Bayes error rate for this example is approximately 0.3941. A random sample of size $n = 400$ was simulated. First, $\{x_i;\ i = 1, \ldots, n\}$ were generated from the uniform distribution and class labels $\{y_i\}$ were generated according to the specified multinomial distribution at each x_i. Simulating the presence of redundant components, we consider the two-way interaction spline kernel $K_\theta(s,t) = \theta_1 K(s_1,t_1) + \theta_2 K(s_2,t_2) + \theta_{12} K(s_1,t_1)K(s_2,t_2)$ for this example. The θ-step tuning of λ_θ with $\hat{\lambda}^{(0)}$ fixed at the Kullback-Leibler distance minimiser, 2^{-17}, shows that the Kullback-Leibler distance curve is rather flat when the penalty induced by λ_θ is not too large. This is a typical trend observed in tuning plots at the θ-step, implying that classifiers with necessary features or more may not be distinguishable in terms of prediction error once the overall complexity is appropriately controlled at the preceding c-step. Whenever there are multiple minimisers λ_θ of a chosen tuning criterion, we choose the largest of these for reasons of parsimony.

(Figure 1 is around here)

Figure 1 shows the paths of the rescaling parameters θ_1, θ_2 and θ_{12} as the component penalty λ_θ changes. The larger the penalty, the smaller are the magnitudes of the parameters and the fewer are the nonzero parameters. At $\hat{\lambda}_\theta^{(1)} = 2^{-6}$, the largest minimiser of the Kullback-Leibler distance, $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_{12}) = (1, 0, 0)$ with $\hat{\theta}_1$ being the only positive component. This θ-step lowered the Kullback-Leibler distance from 0.6176 to 0.6143 and the misclassification rate from 0.3970 to 0.3967, suggesting that it does not degrade the prediction accuracy.

(Table 1 is around here)

We note here that carrying out a few more iterations at $\hat{\lambda}^{(0)} = 2^{-17}$ and $\hat{\lambda}_\theta^{(1)} = 2^{-6}$ did not make any noticeable changes to the θ estimates, and the solutions from further iterations were virtually the same for this example. Table 1 shows how the optimal pairs $(\hat{\lambda}_\theta,\ \hat{\lambda})$ changed and stabilised as we tuned λ at each c-step and λ_θ at each θ-step in the subsequent iterations. After the relevant component was correctly chosen at the first θ-step, λ and λ_θ at the following steps needed to be retuned just once more, and they remained the same thereafter. Differences in Kullback-Leibler distance values after the first iteration were of the order of 10^{-6}, and thus negligible. Presumably this observation is a general characteristic of the proposed computational procedure according to the following heuristic argument. The first θ-step eliminates irrelevant components, and this change, if substantial, would result in change of $\hat{\lambda}$ at the first c-step that completes the first iteration. As a result of the corresponding change in $\hat{C}^{(1)}$, the second θ-step might need retuning. However, $\hat{\theta}^{(2)}$ would change little from $\hat{\theta}^{(1)}$ because the second θ-step virtually reapplies the component-selection procedure to the data with relevant features only, rendering the subsequent iterations almost redundant. This empirically justifies the one-step update procedure with sequential tuning as described in §3.4.

(Table 2 is around here)

To demonstrate that data-adaptive tuning can be carried out without losing much efficiency, we did five-fold crossvalidation with the hinge loss. The results are shown in Table

2, which is an empirical version of Table 1. It is qualitatively in good agreement with Table 1 confirming in particular that a one-step update would be sufficient. The curve of five-fold crossvalidated hinge loss and the path of $\hat{\theta}$ values at the θ-step were quite similar to those obtained before. At the chosen tuning parameters, $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_{12}) = (1, 0, 0)$ gives the correct feature selection.

4·2. *Medical diagnosis with microarray data*

We revisited the children cancer data from Khan et al. (2001). The dataset is available from http://www.nhgri.nih.gov/DIR/Microarray/Supplement/. They classified the small round blue cell tumours of childhood into four classes based on the expression levels of 2308 genes. The data consist of 63 training cases and 20 test cases. In the presence of much larger number of variables (genes) than the sample size, filtering has been a very common and tractable approach for gene selection, where we measure the marginal association between each gene and the tumour types, and incorporate those genes with the strongest marginal association in a classifier. However, there is arbitrariness in the choice of the number of genes to be included in the classifier. Also, the fact that genes are mostly corregulated suggests that joint association could be more informative in revealing the functional relationship between gene expression levels and the tumour types.

(Figure 2 is around here)

Before the application of the method to the data, its effectiveness was tested on a miniature dataset synthesised from the original data as a working proof of the method. The miniature dataset of 100 genes, with 63 training cases and 20 test cases, was constructed as follows. First, using the F-ratio as a measure of marginal association from the training cases only, we ranked the genes and selected the top 20 genes as variables truly associated with the class. For the validity of this assumption, it is noted that rather more than 1000 genes had F-ratios greater than the 95 percentiles of F-ratios from randomly permuted data. As irrelevant variables, we included the bottom 80 genes with the class labels corresponding to the covariate vectors of 80 genes randomly jumbled, so that they are genuinely unrelated to the class, but potential correlations between those genes are intact. One hundred replicates of synthetic training data were obtained by bootstrapping samples from this miniature dataset, keeping the class proportions the same as the original data in each sample. We applied the structured multicategory support vector machine to the 100 replicates using the additive spline kernel. The combined one-step update with five-fold crossvalidation was used with either 0-1 or the hinge loss. To guard against the potential bias due to the presence of duplicate observations in the bootstrap samples, five-fold crossvalidation procedure needs to be adjusted so that for each split of a training set and a validation set, the observations that fall in both sets are not counted for validation. This adjustment resulted in the effective size of the validation set of 28 out of 63 on average for this example. The boxplots in Fig. 2(a) show the distributions of the number of genes selected by the method in 100 runs. The ideal number of relevant genes is 20 by construction. Both distributions have a sample median of 19. However, the distribution for the hinge loss is much less dispersed. A similar comparison between the two losses was made in another simulation example, which is not shown here. Figure 2(b) shows the proportion of runs in 100 bootstrap samples that included each gene in the final classifiers when the hinge loss was used for tuning. Here genes are conveniently labeled as 1-20 for the top 20 genes and 21-100 for the bottom 80 genes. The plot clearly shows that the 20 informative genes were consistently picked up over the repeated runs. Thirteen informative genes out of the 20 were selected in more than 90% of the runs while 72 noninformative genes were picked up in fewer than 5% of the runs. Improvement in classification accuracy by gene selection was also inspected based on the error rate over the 20 test cases. Use of the 0-1 loss for tuning gave a decrease of 0.007, from 0.065 to 0.058, in the test error rate on average, with standard error 0.0033, and the hinge loss gave an average decrease of 0.003, from 0.0555 to 0.0525, with standard error 0.00563. In summary, this numerical study confirms that the ℓ_1-norm-based gene selection method can be effectively used when the sample size is smaller than the number of variables.

(Figure 3 is around here)

The method was then applied to the original data with 2308 genes to detect important genes for classifying the children cancer, and at the same time to gauge the inherent uncertainty in estimating the effects of 2308 covariates on the response with only 63 observations. To assess variability, 100 bootstrap samples were drawn from the training data, again with the class proportions the same as for the original sample. The hinge loss was used for five-fold crossvalidation. The empirical distribution of the number of genes included in one-step updates out of 2308 had a sample median of 222, sample quartiles of 209 and 235 and a long right-hand tail. Figure 3 shows the proportion of selection of each gene in 100 replicated structured multicategory support vector machines based on the bootstrap samples. Genes are ordered by their marginal ranks in the original sample. In general, genes with high selection proportions were ranked highly by the F-ratio of marginal association. Inspection of the genes that appeared in the classifiers for more than 90% of runs showed that the joint and the marginal relevances exhibit a strong agreement up to the gene ranked 30, beyond which their agreement becomes rather weak. 1812 genes were selected on fewer than 20% of the runs, while 58 genes were consistently selected on more than 95% of the runs. We observed that the proportion of inclusion of each gene was a good proxy of the average shrinkage factor of the gene in this case. Gene selection led to reduction in test error rates by 0.0255 on average, from 0.0525 to 0.0270, with standard error of 0.00609, and it also reduced the variance of test error rates.

5. DISCUSSION

As a result of the characteristics of the support vector machine, a caveat may be necessary for proper interpretation of selected features and their functional forms. Since support vector machines directly target class codes asymptotically, the selected features are for the approximation of indicator-like functions. Although they identify covariates on which y depends, generally they are not of the simplest possible functional form to describe the relationship between x and y. For instance, polynomial approximation of $I(x_1 > a)$ requires higher order terms than a linear term x_1 despite its simple functional relationship to x_1. This is an intrinsic aspect of set estimation via function estimation.

The ℓ_1-type component penalty in the structured multicategory support vector machine treats all of the components alike. In statistical modelling, a hierarchical structure of covariates is often desired. For example, one may restrict two-way interactions to appearing only with the corresponding main effects. It would be useful to tailor the component penalty by imposing hierarchy within the model or by reflecting any a priori information about the relevance of covariates on scaling parameters for refined selection.

When the number of covariates is much higher than the sample size, it is quite common to filter covariates based on a measure of marginal association and use a subset of most highly associated covariates for model fitting. The proposed method chooses relevant features by taking into account their joint effects. It is important to understand the effect of the high dimensionality of covariates on the stability and generalisation ability of the resulting classifiers. Therefore, it would be worthwhile to investigate further the merits of this joint approach relative to the marginal feature selection approach in terms of prediction accuracy and computational complexity.

Recent developments in methods either involving ℓ_1 penalty or ℓ_1 loss (Efron et al., 2004; Hastie et al., 2004) suggest that it is feasible to characterise the entire solution path as a function of a tuning parameter in a constructive fashion. It would be attractive to devise a similar algorithm for the c-step and θ-step, thereby shortcutting the fitting and tuning procedures. Recent work in an Ohio State University technical report by Y. Lee and Z. Cui shows how the entire c-step solution path can be constructed sequentially.

ACKNOWLEDGEMENT

The research of Y. Kim, S. Lee and J.-Y. Koo was supported by the Science Research Center program of Ministry of Science & Technology and Korea Science and Engineering Foundation. We would like to thank the editor for careful reading and referees who informed us of related works and provided constructive comments that greatly improved this article.

REFERENCES

Argyriou, A., Micchelli, C. & Pontil, M. (2005). Learning convex combinations of continuously parameterized basic kernels. In *Proc. 18th Annual Conf. Learning Theory (COLT'05)*, Ed. R. Auer and R. Meir, pp. 323–37. Bertinoro, Italy: Springer.

Aronszajn, N. (1950). Theory of reproducing kernel. *Trans. Am. Math. Soc.* **68**, 3337–404.

Bellman, R. E. (1961). *Adaptive Control Processes.* Princeton University Press.

Bradley, P. S. & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *Proc. Fifteenth Int. Conf.*, Ed. J. Shavlik, pp. 82–90. San Francisco, CA: Morgan Kaufmann.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics* **37**, 373–84.

Chapelle, O., Vapnik, V., Bousquet, O. & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Mach. Learning* **46**, 131–59.

Chen, S. S., Donoho, D. L. & Saunders, M. A. (1999). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comp.* **20**, 33–61.

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004). Least angle regression (with Discussion). *Ann. Statist.* **32**, 407–99.

Evgeniou, T., Pontil, M. & Poggio, T. (2000). A unified framework for regularization networks and support vector machines. *Adv. Comp. Math.* **13**, 1–50.

Grandvalet, Y. & Canu, S. (2003). Adaptive scaling for feature selection in SVMs. In *Adv. Neural Info. Proces. Syst.*, Ed. S. T. S. Becker and K. Obermayer, pp. 553–60. Cambridge, MA: MIT Press.

Gu, C. (2002). *Smoothing Spline ANOVA Models.* New York: Springer.

Gunn, S. R. & Kandola, J. S. (2002). Structural modelling with sparse kernels. *Mach. Learning* **48**, 137–63.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Mach. Learning* **46**, 389–422.

Hastie, T., Rosset, S., Tibshirani, R. & Zhu, J. (2004). The entire regularization path for the support vector machine. *J. Mach. Learning Res.* **5**, 1391–415.

Hastie, T. & Tibshirani, R. (1986). Generalized additive models (with Discussion). *Statist. Sci.* **1**, 297–318.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Atonescu, C., Peterson, C. & Meltzer, P. (2001).

Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Med.* **7**, 673–79.

Kohavi, R. & John, G. H. (1997). Wrappers for feature subset selection. *Artif. Intel.* **97**, 273–324.

Lee, Y., Lin, Y. & Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *J. Am. Statist. Assoc.* **99**, 67–81.

Micchelli, C. & Pontil, M. (2005). Learning the kernel function via regularization. *J. Mach. Learning Res.* **6**, 1099–125.

Schölkopf, B. & Smola, A. (2002). *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press.

Tibshirani, R. (1996). Regression selection and shrinkage via the lasso. *J. R. Statist. Soc.* B **58**, 267–88.

Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.

Wahba, G. (1990). *Spline Models for Observational Data*. Philadelphia: SIAM.

Wahba, G. (1998). Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV. In *Adv. Kernel Methods: Support Vector Learning*, Ed. B. Schölkopf, C. J. C. Burges and A. J. Smola, pp. 69–87. Cambridge, MA: MIT Press.

Wahba, G. (2002). Soft and hard classification by reproducing kernel Hilbert space methods. *Proc. Nat. Acad. Sci.* **99**, 16524–30.

Wahba, G., Wang, Y., Gu, C., Klein, R. & Klein, B. (1994). Structured machine learning for "soft" classification with smoothing spline ANOVA and stacked tuning, testing, and evaluation. In *Adv. Neural Info. Proces. Syst.*, vol. 6, Ed. J. D. Cowan, G. Tesauro and J. Alspector, pp. 415–22. San Fransisco, CA: Morgan Kaufmann.

Weston, J., Elisseff, A., Schölkopf, B. & Tipping, M. (2003). Use of the zero-norm with linear models and kernel methods. *J. Mach. Learning Res.* **3**, 1439–61.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. & Vapnik, V. (2001). Feature selection for SVMs. In *Adv. Neural Info. Proces. Syst.*, Ed. S. A. Solla, T. K. Leen, and K.-R. Muller, pp. 668–74. Cambridge, MA: MIT Press.

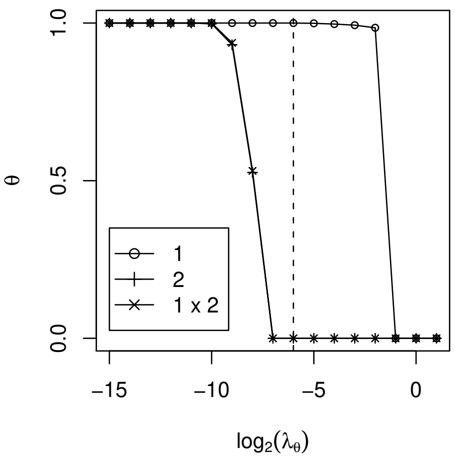

Figure 1: The trajectory of $\hat{\theta}$ as a function of λ_θ with the two-way interaction spline. The trajectory for θ_1 is denoted by $\circ$, that for θ_2 by $+$ and that for θ_{12} by $\times$. The overlap of $+$ and $\times$ indicates that the trajectories for θ_2 and θ_{12} are almost indistinguishable. $(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_{12}) = (1, 0, 0)$ at the Kullback-Leibler distance minimiser $\hat{\lambda}_\theta = 2^{-6}$ is indicated by the dotted vertical line.

Table 1: Kullback-Leibler distance minimising values of λ_θ at the θ-step and λ at the c-step. In the table, 0* denotes values that are not exactly zero but less than 10^{-5}.

Iteration	Optimal parameters		θ estimates		
	$\log_2(\hat{\lambda}_\theta)$ (GCKL)	$\log_2(\hat{\lambda})$ (GCKL)	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_{12}$
0		-17 (0.6176)	1	1	1
1	-6 (0.6143)	-19 (0.6016)	1	0	0
2	-2 (0.6016)	-19 (0.6016)	1	0	0*
3	-2 (0.6016)	-19 (0.6016)	1	0	0*

GCKL, value of generalised comparative Kullback-Leibler distance

Table 2: Five-fold crossvalidated minimising values of λ_θ at the θ-step and λ at the c-step with the hinge loss.

Iteration	Optimal parameters		θ estimates		
	$\log_2(\hat{\lambda}_\theta)$ (CV: hinge)	$\log_2(\hat{\lambda})$ (CV: hinge)	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_{12}$
0		-16 (0.5953)	1	1	1
1	-6 (0.5937)	-20 (0.5805)	1	0	0
2	-2 (0.5807)	-20 (0.5807)	0.9842	0	0
3	-4 (0.5771)	-20 (0.5805)	1	0	0

CV, crossvalidation

(a)　(b)

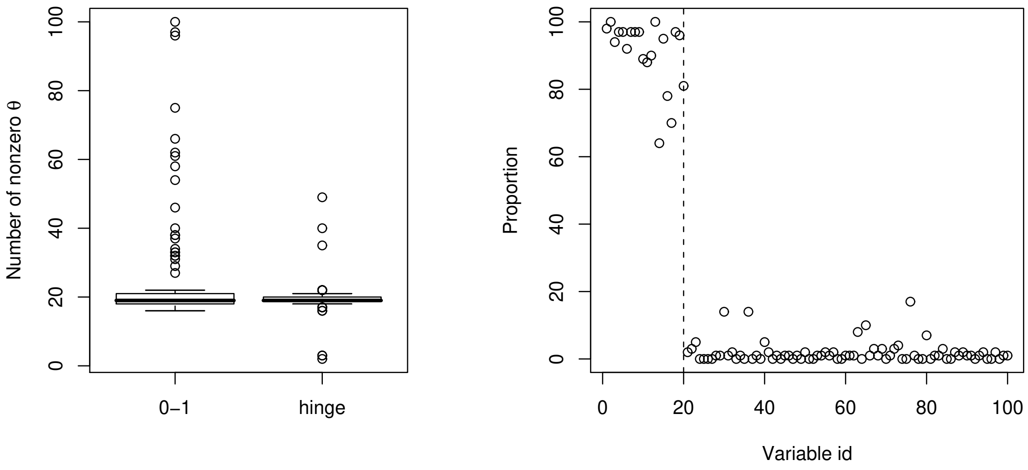

Figure 2: Miniature child cancer dataset. (a) boxplots of the number of genes with nonzero rescaling parameters in 100 runs for the 0-1 loss and hinge loss. (b) the proportion of inclusion (%) of each gene in the final classifiers over 100 runs when the hinge loss was used for tuning. The dotted line separates informative variables from noninformative ones.

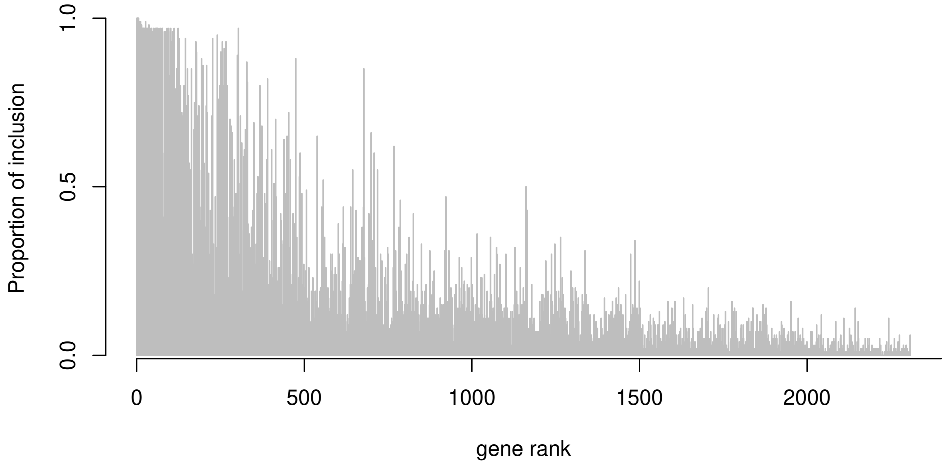

Figure 3: Child cancer dataset. The proportion of selection of each gene in one-step updated structured multicategory support vector machines for 100 bootstrap samples. Genes are presented in the order of marginal rank in the original sample.