Pattern Anal Applic (2011) 14:47–55
DOI 10.1007/s10044-010-0177-7

THEORETICAL ADVANCES

A sparse multinomial probit model for classification

Yunfei Ding · Robert F. Harrison

Received: 20 December 2008 / Accepted: 25 February 2010 / Published online: 6 May 2010

Abstract A recent development in penalized probit modelling using a hierarchical Bayesian approach has led to a sparse binomial (two-class) probit classifier that can be trained via an EM algorithm. A key advantage of the formulation is that no tuning of hyperparameters relating to the penalty is needed thus simplifying the model selection process. The resulting model demonstrates excellent classification performance and a high degree of sparsity when used as a kernel machine. It is, however, restricted to the binary classification problem and can only be used in the multinomial situation via a one-against-all or one-against-many strategy. To overcome this, we apply the idea to the multinomial probit model. This leads to a direct multi-classification approach and is shown to give a sparse solution with accuracy and sparsity comparable with the current state-of-the-art. Comparative numerical benchmark examples are used to demonstrate the method.

Keywords Multi-classification · Sparseness · Multinomial probit · Hierarchical Bayesian

1 Introduction

The majority of machine learning methods for classifying objects into pre-determined groups consider only two-class or binary problems but many tasks involve more than two classes—so called multinomial problems. Multi-class classification methods are important in both the theory and practice of pattern recognition and present a significantly harder task than binary classification with all other things being equal [1]. The extension of existing methods for binary classification to multi-class problems is therefore of substantial and continuing interest, e.g. [1–7].

The multinomial probit (MNP) model plays an important rôle in the social, econometric and biological sciences for the analysis of multi-category response. It provides a greater degree of flexibility in modelling discrete choices (categories) over the commonly adopted multinomial logit (MNL) model. Indeed, when considered from the perspective of an underlying latent variable model, the specification for the two approaches differs only in the assumed form of error distribution (multivariate Normal and i.i.d. Gumbel, respectively) and their associated link functions. Specifically, MNP relaxes the so-called IIA (independence of irrelevant alternatives) constraint implicit in MNL by admitting a general covariance structure for the errors. In addition, it easily admits individual-specific choice and covariate sets, which is perhaps why it is preferred in modelling social phenomena, and can be readily extended to factor analysis problems as well. The price for the added flexibility is in the loss of the easily computed closed-form for the likelihood function of the MNL model. Our reason for choosing the probit form here is quite different from these putative benefits, but is, instead, one of convenience—the inherent normality of the probit approach allows us to generalize the hierarchical Bayes approach introduced by Figueiredo [8] to the multinomial case. Specifically, integrals necessary to the development can be undertaken that would otherwise be intractable. Indeed, we ultimately focus on the situation closest to MNL where category choices are independent, i.e. an identity error covariance structure.

The MNP generalizes the early work of Thurstone [9] for binary choice. Bock and Jones [10] apply the MNP

Y. Ding (✉) · R. F. Harrison
Department of Automatic Control and Systems Engineering,
The University of Sheffield, Mappin Street,
Sheffield S1 3JD, UK
e-mail: yunfeiding@hotmail.com

model to the three-class case. The MNP model formulation from utility maximization theory is described in [11]. Domencich and McFadden [12] first apply this model to the transportation analysis of Hausman and Wise [13]. Maximum likelihood estimates (MLE) and a method of simulated moments (MSM) [14] have been developed to evaluate the likelihood function. For C-class problems, these two approaches require the evaluation of $(C-1)$-dimensional Gaussian integrals: the conditional probabilities corresponding to each class. However, closed-form choice probabilities for the MNP model are not available and, in practice, numerical integration based on quadrature can feasibly estimate the general multivariate integral only when the dimension is low. While "low" has traditionally meant five or fewer [15], recent advances have extended this limit so that, for instance, the scheme devised by Genz [16] can comfortably handle up to 20 dimensions.[1] More recently, Miwa and colleagues [17] have proposed a recursive scheme which is slower than Genz's but has the advantage of being entirely deterministic. This limitation ultimately suggests resort to simulation methods. Monte Carlo simulation methods are employed to approximate high dimensional integrals of choice probabilities [18, 19]. However, simulators need to possess particular characteristics, such as continuity and differentiability, so that simulation methods are still computationally costly because of the intensive processing required by some. McCulloch and Rossi [20] give a Bayesian analysis of the MNP model (also see [21]). Chib and Greenberg [22] provide an overall Bayesian analysis of MNP models for correlated binary data. The Gibbs sampler and Markov chain Monte Carlo (MCMC) are utilized to estimate the parameters of MNP models [20, 23, 24], however, these algorithms can also be computationally very costly.

Figueiredo [8] points out that sparsity is desirable in supervised learning for the following three reasons. First, sparseness leads to a structural simplification of the estimated function. Second, obtaining a sparse estimate corresponds to performing feature/variable selection. And third, in kernel-based methods, the generalization ability improves with the degree of sparseness—a key idea behind the support vector machine (SVM). Indeed, some form of complexity control is essential for the development of kernel machines in general. Under the sparse Bayesian learning framework, the relevance vector machine (RVM) [25], variational relevance vector machine (VRVM) [26], sparse logistic regression [27] and sparse kernel Fisher discriminant algorithms [28] have been developed to solve two-class classification problems. To achieve sparseness the Bayesian approach introduces an appropriate prior distribution on the model parameters and seeks to maximize the marginalized likelihood function. More comprehensive descriptions of sparse Bayesian learning for both regression and classification can be found in [25].

Figueiredo [8] proposes a sparse Bayesian approach to learn a probit classifier for two-class responses. The method makes use of the univariate probit model—a generalized linear model with a normal c.d.f. as link function. It is well known [29] that this model can be expressed as a latent variable model that closely resembles the conventional linear regression model and thus presents a particularly convenient form. A two-level (Gaussian plus exponential) hierarchical Bayesian approach is used to represent the prior distribution of the model parameters but, instead of adopting an exponential second-level prior on the hyperparameters which would lead to an overall Laplacian prior requiring the tuning of a hyper-prior to control sparsity, Figueiredo substitutes a Jeffreys prior which has no associated parameter and therefore removes the need for hyper-prior tuning—potentially expensive in model estimation. Under this revised model the calculus necessary to construct an EM algorithm (removal of the hyperparameters via integration) can be carried out to the point at which the evaluation of the normal c.d.f. is required. This can be carried out efficiently via quadrature. Krishnapuram et al. [30] present a classifier based on this idea to promote sparsity jointly in the selection of both basis functions and covariates. Their method has been successfully applied to gene expression analysis and cancer diagnosis.

Naturally, it is worth thinking about how to extend the idea to multinominal probit models. Girolami and Rogers have developed a non-parametric approach—a Gaussian process (GP)-based method—to build sparse, variational multi-class GP classifiers [6]. The Gibbs sampler and variational Bayes approximation are employed to represent the joint posterior distribution via an ensemble of factored posteriors. In contrast, to the best of our knowledge, the method presented below provides the first deterministic algorithm for estimating a *sparse multinomial* probit (SMNP) model. In a natural generalization, Figueiredo's hierarchical approach with a Jeffreys hyper-prior is again adopted to encourage sparsity amongst the parameters and the outcome is an EM algorithm that can be computed for a reasonable number of classes.

The structure of the remainder of this paper is as follows. The next section describes the sparse *binary* probit (SBP) algorithm presented by Figueiredo [8] to motivate what follows. The third section introduces the proposed generalization of this to the multinomial case and the specific algorithmic steps are also provided. Section 4

[1] This method, adopted in recent versions of the Matlab Statistics Toolbox for dimensions above four, makes use of a degree of Monte Carlo simulation and so might be considered a hybrid approach.

presents comparative results from several experiments using benchmark data.

2 Sparse *binary* probit model

The development of the SBP model is now sketched out to provide a framework for the multinomial extension. Consider an underlying latent variable model, $z = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} + w$ with $p(w) = \phi(w|0, 1)$—the standard, univariate normal distribution. $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \ldots, h_M(\boldsymbol{x}))^{\mathrm{T}}$ is an M-dimensional vector of basis functions and $\boldsymbol{\beta}$ a corresponding vector of model parameters. Class membership is determined based on whether or not the value of the (unmeasured) latent variable exceeds zero, i.e. assign to the class labelled 1 if $z \geq 0$ else assign to the class labelled zero. This is expressed thus:

$$P(y = 1|\boldsymbol{x}) = P(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\beta} + w_i \geq 0) = \Phi(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\beta}) \tag{1}$$

where $\Phi(a) = \int_{-\infty}^{a} \phi(t)\ \mathrm{d}t$ is the (univariate) *probit* function.

Given a training set of input-target pairs $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$, where $\boldsymbol{x}_i$ denotes a D-dimensional input vector and y_i, its corresponding one-dimensional binary-valued target vector, we define H as the $N \times M$ design matrix with M, the number of fixed basis functions, thus

$$H = [\boldsymbol{h}(\boldsymbol{x}_1), \ldots, \boldsymbol{h}(\boldsymbol{x}_N)]^{\mathrm{T}}, \tag{2}$$

The underlying latent variable model is now given by:

$$\boldsymbol{z} = H\boldsymbol{\beta} + \boldsymbol{w} \tag{3}$$

and the likelihood function for $\boldsymbol{z}$ can be written:

$$p(\boldsymbol{z}|\boldsymbol{\beta}) = \phi(\boldsymbol{z}|H\boldsymbol{\beta}, I_N) \tag{4}$$

By placing a prior distribution on $\boldsymbol{\beta}$, an EM algorithm can then be derived to find a maximum a posteriori (MAP) estimate of $\boldsymbol{\beta}$ by treating $\boldsymbol{z}$ as missing data. To promote sparseness each β_i is given a zero-mean Gaussian prior with its own variance τ_i,

$$p(\beta_i|\tau_i) = \phi(\beta_i|0, \tau_i) \tag{5}$$

The importance of the hierarchical decomposition is that it allows the EM algorithm to estimate $\boldsymbol{\beta}$ by considering $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_M]^{\mathrm{T}}$ as missing data in addition to the latent variables, $\boldsymbol{z}$. At this stage, adopting an exponential distribution for the variance, τ_i, would be equivalent to placing Laplacian priors on the β_i but instead Figueiredo places a non-informative Jeffreys hyper-prior $p(\tau_i) \propto \frac{1}{\tau_i}$ on the variances, τ_i. This is equally tractable in the analysis but has the distinct advantage of having no associated, arbitrary parameter, thereby avoiding the need for cross-validation or other methods of selection [8].

Using (4) and (5) and the definition of the Jeffreys prior, the complete log posterior for $\boldsymbol{\beta}$ with "missing" vectors $\boldsymbol{\tau}$ and $\boldsymbol{z}$ can be written thus:

$$\begin{aligned} \log p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\tau}, \boldsymbol{z}) &\propto \log p(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{\tau}, \boldsymbol{z}) \\ &\propto \log p(\boldsymbol{z}|\boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\tau})p(\boldsymbol{\tau})p(\boldsymbol{y}|\boldsymbol{z}) \\ &\propto -\boldsymbol{\beta}^{\mathrm{T}}H^{\mathrm{T}}H\boldsymbol{\beta} + 2\boldsymbol{\beta}^{\mathrm{T}}H^{\mathrm{T}}\boldsymbol{z} - \boldsymbol{\beta}^{\mathrm{T}}\Upsilon(\boldsymbol{\tau})\boldsymbol{\beta} \end{aligned} \tag{6}$$

where $\Upsilon(\boldsymbol{\tau}) \equiv \mathrm{diag}(\tau_1^{-1}, \ldots, \tau_M^{-1})$ is a diagonal matrix containing the inverse variances of the β_i's.

For the expectation step (E-step) in the EM algorithm, the expected values of both Υ and $\boldsymbol{z}$ must be calculated at each computation step, indexed by t, by the following equations:

$$\begin{aligned} V_{(t)} &= E\left[\Upsilon(\boldsymbol{\tau})|\hat{\boldsymbol{\beta}}_{(t)}, \boldsymbol{y}\right] \\ &= \mathrm{diag}\left\{E\left[\tau_1^{-1}|\hat{\boldsymbol{\beta}}_{(t)}, \boldsymbol{y}\right], \ldots, E\left[\tau_M^{-1}|\hat{\boldsymbol{\beta}}_{(t)}, \boldsymbol{y}\right]\right\} \\ &= \mathrm{diag}\left\{|\hat{\boldsymbol{\beta}}_{1,(t)}|^{-2}, \ldots, |\hat{\boldsymbol{\beta}}_{M,(t)}|^{-2}\right\} \end{aligned} \tag{7}$$

$$\begin{aligned} s_{i,(t)} &\equiv E\left[z_i|\hat{\boldsymbol{\beta}}_{(t)}, \boldsymbol{y}\right] \\ &= \begin{cases} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_i)\hat{\boldsymbol{\beta}}_{(t)} + \frac{\phi(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_i)\hat{\boldsymbol{\beta}}_{(t)}|0,1)}{\Phi(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_i)\hat{\boldsymbol{\beta}}_{(t)})} & \text{if } y_i = 1; \\ \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_i)\hat{\boldsymbol{\beta}}_{(t)} - \frac{\phi(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_i)\hat{\boldsymbol{\beta}}_{(t)}|0,1)}{1-\Phi(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_i)\hat{\boldsymbol{\beta}}_{(t)})} & \text{if } y_i = 0. \end{cases} \end{aligned} \tag{8}$$

where the caret indicates the estimated value. These expectations are derived analytically from the integrations employing the model assumptions and noting that z is conditionally normally distributed with mean $\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\hat{\boldsymbol{\beta}}_{(t)}$ left-truncated at zero if $y = 1$ and right-truncated at zero if $y = 0$.

Now $V_{(t)}$ and $\boldsymbol{s}_{(t)}$, can be taken into the complete log-posterior (6) to replace Υ and $\boldsymbol{z}$. Maximizing this log-posterior with respect to $\boldsymbol{\beta}$ leads to the maximization step (M-step)

$$\hat{\boldsymbol{\beta}}_{(t+1)} = (V_{(t)} + H^T H)^{-1} H^{\mathrm{T}}\boldsymbol{s}_{(t)} \tag{9}$$

Since some components of $\boldsymbol{\beta}$ are expected to become zero when sparseness is achieved, the corresponding elements of the matrix, $V_{(t)}$, in (7) will become undefined. To overcome this (9) can be rewritten as:

$$\hat{\boldsymbol{\beta}}_{(t+1)} = U_{(t)}(I + U_{(t)}H^{\mathrm{T}}HU_{(t)})^{-1}U_{(t)}H^{\mathrm{T}}\boldsymbol{s}_{(t)}, \tag{10}$$

by defining a new diagonal matrix $U_{(t)} = \mathrm{diag}(|\hat{\boldsymbol{\beta}}_{1,(t)}|, \ldots, |\hat{\boldsymbol{\beta}}_{M,(t)}|)$ thus avoiding potential divides-by-zero.

In practice, this EM algorithm produces a sequence of estimates of $\hat{\boldsymbol{\beta}}_{(t)}$ until a predefined stopping condition is satisfied. The E-step relates to (7)–(9), and the M-step is processed by (10).

3 Sparse multinomial probit model

3.1 Proposed MNP model

The extension of the above to the multinomial case follows the same procedure but now there exist C categories, leading to C-dimensional latent variable model

$$\tilde{z}^{\mathrm{T}} = \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\tilde{B} + \tilde{\boldsymbol{w}}^{\mathrm{T}} \tag{11}$$

where $\tilde{z}$ is a $C \times 1$ latent response vector, $\tilde{B}$ is an $M \times C$ parameter matrix and $p(\tilde{\boldsymbol{w}}) = \phi_C(\tilde{\boldsymbol{w}}|\boldsymbol{0}, \tilde{\Sigma})$—the C-dimensional zero mean normal density with covariance matrix $\tilde{\Sigma}$.

The MNP classification rule for the ith observation is expressed as:

$$\begin{aligned} &\tilde{y}_{ij} = 1 \quad \text{if } \tilde{z}_{ij} \geq 0, \quad \text{and} \quad \tilde{z}_{ij} = \max(\tilde{z}_i), \\ &\qquad j = 1, \ldots, (C-1) \qquad (12) \\ &\tilde{y}_{iC} = 1 \quad \text{if } \tilde{z}_{ij} < 0 \quad \text{for all} \quad j = 1, \ldots, C. \end{aligned}$$

leading to the associated probability of selecting category i given by:

$$P\left(\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\tilde{\boldsymbol{b}}_i + \tilde{\boldsymbol{w}}_i > \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\tilde{\boldsymbol{b}}_j + \tilde{\boldsymbol{w}}_j\right), \quad j \neq i \tag{13}$$

where $\tilde{\boldsymbol{b}}_k$ denotes the kth column of the matrix, $\tilde{B}$. This is equivalently expressed as:

$$P\left(\tilde{\boldsymbol{w}}_i - \tilde{\boldsymbol{w}}_j > \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})(\tilde{\boldsymbol{b}}_j - \tilde{\boldsymbol{b}}_i)\right), \quad j \neq i \tag{14}$$

Clearly, only the differences in the utilities ascribed to the latent variables are important, i.e. that choices are made only with respect to a (usually arbitrary) baseline situation. This is implicit in the binary model, where the class labelled zero takes on the baseline rôle.

Since the difference of normally distributed variables is itself normally distributed, the C-class problem can therefore be expressed in terms of $(C-1)$ latent alternatives, $\boldsymbol{z}=\left[z_1, \ldots, z_{(C-1)}\right]^{\mathrm{T}}$, thus:

$$\boldsymbol{z}^{\mathrm{T}} = \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})B + \boldsymbol{w}^{\mathrm{T}} \tag{15}$$

where B is the $M \times (C-1)$ parameter matrix and $p(\boldsymbol{w}) = \phi_{(C-1)}(\boldsymbol{w}|\boldsymbol{0}, \Sigma)$. We need not be concerned with the relationship between B and $\tilde{B}$ and between Σ and $\tilde{\Sigma}$ because (1) owing to reasons of identifiability of the latent error covariance (see, e.g. [31]) it is not possible to reconstruct $\tilde{\Sigma}$ from an estimate of Σ and, as a predictive tool, there is no reason to do so anyway, and (2) we shall ultimately focus on the case where Σ is taken to be the $(C-1)$-dimensional identity matrix. However, at this stage we continue with a general analysis and specialize later.

Once again, given a training set of N input-target pairs where the targets are now binary-valued vectors, $\boldsymbol{y}=\left[y_1, \ldots, y_{(C-1)}\right]$ and $y_{ij} = 1$ indicates that the jth class is to be preferred over the baseline, the latent variable model can be set up analogously to (3):

$$Z = \mathrm{HB} + \mathrm{W} \tag{16}$$

where $Z = [z_1, \ldots, z_N]^{\mathrm{T}}$. The associated probability of selecting the jth class in preference to the baseline class is now $P\left(\boldsymbol{w}_j > \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{b}_j\right)$.

We are now in a position to re-write the model in a more convenient form for the subsequent analysis. To do this, apply the vec[2] operation to (16), define $\boldsymbol{z} = \mathrm{vec}(Z)$, $\boldsymbol{\beta} = \mathrm{vec}(B)$, $\boldsymbol{w} = \mathrm{vec}(W)$ giving:

$$\boldsymbol{z} = \left(I_{(C-1)} \otimes H\right)\boldsymbol{\beta} + \boldsymbol{w} \tag{17}$$

$$\boldsymbol{z} \triangleq \mathbf{H}\boldsymbol{\beta} + \boldsymbol{w} \tag{18}$$

where $\otimes$ denotes the Kronecker product, $\dim(\boldsymbol{z}) = \dim(\boldsymbol{w}) = (C-1)N \times 1$ and $\dim(\mathbf{H}) = (C-1)N \times (C-1)M$ and the new design matrix, $\mathbf{H} = [H_1, H_2, \ldots, H_N]$, The ith design matrix, H_i is given by $H_i = I_{C-1} \otimes \boldsymbol{h}_i^{\mathrm{T}}$.

3.2 An EM algorithm for SMNP

As in the binary case (see Sect. 2) a hierarchical structure is again used, placing independent Gaussian priors on the β_i and Jeffreys' hyper-priors on their associated variances, leading to an identical situation (notwithstanding the increase in dimensionality of $\boldsymbol{\beta}$ from M to $(C-1)M$ with the attendant advantages of being parameter free yet analytically tractable. The related part of the derivation of the EM Algorithm remains, therefore, unchanged. To motivate the development, consider first the introduction of a *Laplacian* prior on $\boldsymbol{\beta}$

$$p(\boldsymbol{\beta}|\alpha) = \prod_{i=1}^{M(C-1)} \frac{\alpha}{2}\exp\{-\alpha|\beta_i|\} = \left(\frac{\alpha}{2}\right)^{M(C-1)} \exp\left\{-\alpha\|\boldsymbol{\beta}\|_1\right\} \tag{19}$$

where the hyper-parameter, α, defines its precision. A particularly convenient way to structure the prior distribution is through its decomposition into several conditional levels by repeated application of Bayes' theorem and can improve the robustness of resulting Bayes estimators [32]. Adopting a two-level hierarchy [8], the first-level distribution is chosen to be a zero-mean Gaussian prior, $p(\beta_i|\tau_i) = \phi(\beta_i|0, \tau_i)$, for each β_i, each having its own variance (inverse precision), τ_i. For the second stage, an exponential distribution is used as a hyper-prior for the variances, τ_i

$$p(\tau_i|\gamma) = \frac{\gamma}{2}\exp\left\{-\frac{\gamma}{2}\tau_i\right\}, \qquad \text{for } \tau_{\mathrm{i}} \geq 0. \tag{20}$$

Taking the product of these distributions and integrating with respect to τ_i gives

[2] Vec is the operation that stacks the columns of a matrix one upon the other from left to right.

$$p(\beta_i|\gamma) = \int_0^\infty p(\beta_i|\tau_i)p(\tau_i|\gamma)\mathrm{d}\tau_i = \frac{\sqrt{\gamma}}{2}\exp\left\{-\frac{\gamma}{2}|\beta_i|\right\}. \tag{21}$$

demonstrating that the Laplacian prior on $\boldsymbol{\beta}$ is equivalent to this two-level hierarchical Bayes model [8]. However, this introduces an arbitrary parameter into the problem, γ, which controls the trade-off between the degree of sparseness in $\boldsymbol{\beta}$. To remove this, Figueiredo [8] uses the non-informative Jeffreys prior

$$p(\tau) \propto \frac{1}{\tau} \tag{22}$$

to remove the dependence on γ. The Jeffreys prior replaces the exponential hyper-prior in (20) and so removes the need to conduct a search for a good value of its parameter.

As before, $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{(C-1)M}]^{\mathrm{T}}$ is treated as missing data alongside $\boldsymbol{z}$. The EM algorithm generates a sequence of estimates $\hat{\boldsymbol{\beta}}_{(t)}$ and $\hat{\Sigma}_{(t)}$ at different iteration times, t, by applying the expectation (E) and maximization (M) steps, sequentially. For the M-step, let the function, Q, express the expected log posterior,

$$\begin{aligned} &Q(\boldsymbol{\beta}, \Sigma|\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)}) \\ &\quad = \int \log p(\boldsymbol{\beta}, \Sigma|\boldsymbol{y}, \boldsymbol{\tau}, \boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{y}, \hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)}, \hat{\boldsymbol{\tau}}_{(t)})\mathrm{d}\boldsymbol{z}. \end{aligned} \tag{23}$$

The maximization step (M-step) then updates the parameter estimates according to

$$(\hat{\boldsymbol{\beta}}_{(t+1)}, \hat{\Sigma}_{(t+1)}) = \arg\max_{\boldsymbol{\beta},\Sigma} Q(\boldsymbol{\beta}, \Sigma|\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)}). \tag{24}$$

This provides a MAP estimate of $\boldsymbol{\beta}$, i.e. it finds a local maximum of the log-posterior function given by

$$\begin{aligned} \log p(\boldsymbol{\beta}, \Sigma|\boldsymbol{y}, \boldsymbol{\tau}, \boldsymbol{z}) &\propto \log p(\boldsymbol{z}|\boldsymbol{\beta}, \Sigma)p(\boldsymbol{\beta}|\boldsymbol{\tau}) \\ &\propto -\log\det(\Sigma) - (\boldsymbol{z} - H\boldsymbol{\beta})^{\mathrm{T}}\Sigma(\boldsymbol{z} - H\boldsymbol{\beta}) \\ &\quad - \boldsymbol{\beta}^{\mathrm{T}}\Upsilon(\boldsymbol{\tau})\boldsymbol{\beta}, \end{aligned} \tag{25}$$

where $\Upsilon(\boldsymbol{\tau}) = \mathrm{diag}\left(\tau_1^{-1}, \ldots, \tau_{(C-1)M}\right)$ is a diagonal matrix with the inverse variances related to $\boldsymbol{\beta}$. In (25), because the influence of the prior on the estimate of the variances is very small for large N, $p(\Sigma)$ is set to a constant that can be ignored in the log-posterior function [8]. Thus it should then be easier to compute the MAP estimate of model parameters, $\boldsymbol{\beta}$ and Σ. Clearly we have to execute the M-step to gain the update relationships for the two parameters Σ and $\boldsymbol{\beta}$ in (25) by, respectively, maximizing $Q(\boldsymbol{\beta}, \Sigma|\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)})$ with respect to Σ and $\boldsymbol{\beta}$. The two update equations are given by

$$\hat{\Sigma}_{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} E\left[\left(\boldsymbol{z}_{i(t)} - H_i\hat{\boldsymbol{\beta}}_{(t)}\right)\left(\boldsymbol{z}_{i(t)} - H_i\hat{\boldsymbol{\beta}}_{(t)}\right)^{\mathrm{T}}\right] \tag{26}$$

and

$$\hat{\boldsymbol{\beta}}_{(t+1)} = \left(V_{(t)} + \sum_{i=1}^{N} H_i^{\mathrm{T}}\hat{\Sigma}_{(t+1)}^{-1}H_i\right)^{-1} \sum_{i=1}^{N} H_i^{\mathrm{T}}\hat{\Sigma}_{(t+1)}^{-1}\boldsymbol{s}_{i(t)}, \tag{27}$$

where $\boldsymbol{s}_{i(t)}$ and $V_{(t)}$ are the expected values of the corresponding latent vector, $\boldsymbol{z}_{i(t)}$, and the hyper-parameter matrix, $\Upsilon(\tau)$, which can be estimated from observations and the tth results for $\boldsymbol{\beta}$ and Σ. $V_{(t)}$ is given by

$$\begin{aligned} V_{(t)} &= E(\Upsilon(\tau)|\boldsymbol{y}, \hat{\Sigma}_{(t)}, \hat{\boldsymbol{\beta}}_{(t)}) \\ &= \mathrm{diag}\{E(\tau_1^{-1}|\boldsymbol{y}, \hat{\Sigma}_{(t)}, \hat{\boldsymbol{\beta}}_{(t)}), \ldots, E(\tau_{(C-1)M}^{-1}|\boldsymbol{y}, \hat{\Sigma}_{(t)}, \hat{\boldsymbol{\beta}}_{(t)})\}. \end{aligned} \tag{28}$$

Noting, as before, replacing the subscript M with $(C-1)M$, that $p(\tau_i|\boldsymbol{y}, \hat{\boldsymbol{\beta}}_{(t)}, \hat{\boldsymbol{\Sigma}}_{(t)}, \boldsymbol{z}_{(t)}) \propto p(\hat{\beta}_{i(t)}|\tau_i)p(\tau_i)$, where $p(\hat{\beta}_{i(t)}|\tau_i) = \phi(\hat{\beta}_{i(t)}|0, \tau_i)$ and $p(\tau_i)$ is the Jeffreys hyper-prior, $\frac{1}{\tau_i}$. The expected value of τ_i^{-1} in (28), given $\boldsymbol{y}$,$\hat{\boldsymbol{\beta}}_{(t)}$, and $\hat{\Sigma}$, is expressed as

$$\begin{aligned} E\left(\tau_i^{-1}|\boldsymbol{y}, \hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)}\right) &= \frac{\int_0^{+\infty} \frac{1}{\tau_i}p(\beta_{i(t)}|\tau_i)p(\tau_i)\mathrm{d}\tau_i}{\int_0^{+\infty} p(\beta_{i(t)}|\tau_i)p(\tau_i)\mathrm{d}\tau_i} \\ &= \frac{1}{\left|\beta_{i(t)}\right|^2} \end{aligned} \tag{29}$$

so that

$$V_{(t)} = \mathrm{diag}\left(\left|\hat{\beta}_{1(t)}\right|^{-2}, \ldots, \left|\hat{\beta}_{(C-1)M(t)}\right|^{-2}\right). \tag{30}$$

c.f. (7).

In addition, we also need the expected value of $\boldsymbol{z}_i$, which should take two situations into account according to class label. First, when $j = 1, \ldots, (C-1)$ where, for the ith sample, the choice $y_{ij} = 1$ would be made if $z_{ij} > 0$ and $z_{ij} = \max_m\{z_{im}\}$, $m = 1, \ldots, (C-1)$,

$$\begin{aligned} s_{im} &= E\left(z_{im}|\boldsymbol{y}, \hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)}\right) \\ &= \frac{\int_{\Omega_{z_i}} z_{im}\phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i}{\int_{\Omega_{z_i}} \phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i} \\ &= \begin{cases} \dfrac{\int_0^\infty z_{ij}\int_{-\infty}^{z_{ij}}\cdots\int_{-\infty}^{z_{ij}} \phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)},\hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i}{\int_0^\infty \int_{-\infty}^{z_{ij}}\cdots\int_{-\infty}^{z_{ij}} \phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)},\hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i}, & \text{if } m = j \\ \dfrac{\int_0^\infty \int_{-\infty}^{z_{ij}}\cdots\int_{-\infty}^{z_{ij}} z_{im}\phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)},\hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i}{\int_0^\infty \int_{-\infty}^{z_{ij}}\cdots\int_{-\infty}^{z_{ij}} \phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)},\hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i}, & \text{if } m \neq j \end{cases} \end{aligned} \tag{31}$$

When $j = C$, i.e. the ith sample belongs to the baseline class, C, so $y_{iC} = 1$, $z_{im} < 0$ and the expected value of z_{im} is given by

$$\begin{aligned} s_{im} &= E\left(z_{im}|\boldsymbol{y}, \hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)}\right) \\ &= \frac{\int_{-\infty}^0 \int_{-\infty}^0 \cdots \int_{-\infty}^0 z_{im}\phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i}{\int_{-\infty}^0 \int_{-\infty}^0 \cdots \int_{-\infty}^0 \phi_{(C-1)}(\boldsymbol{z}_i|H_i\hat{\boldsymbol{\beta}}_{(t)}, \hat{\Sigma}_{(t)})\mathrm{d}\boldsymbol{z}_i} \end{aligned} \tag{32}$$

The SMNP algorithm is described by the general forms given in (26)–(32). The E-step uses (30), (31) and (32) to produce the expected values of $\boldsymbol{\tau}$ and $\boldsymbol{z}$ and the M-step uses (26) and (27) to update the estimates of Σ and $\boldsymbol{\beta}$.

There are two main difficulties in realizing the above steps. For a full covariance matrix, Σ, in the MNP model there are $\frac{C\times(C-1)}{2}$ parameters to be estimated. However, it is clear from (14) that only the relative values of the latent variables are important in assigning class membership, therefore an arbitrary change of scale leaves the model unaffected and the values of the elements of σ_{ij} are not unique. In the binary case, this problem of " identifiability" is dealt with by adopting unit variance. In the multinomial case, numerous authors have proposed solutions, such as, arbitrarily setting, e.g. $\sigma_{11}=1$, imposing a "correlation" structure, i.e. $\sigma_{ii}=1$, $\sigma_{ij}\leq 1 i\neq j$ or simply estimating Σ directly and re-scaling [33]. To avoid the problem we adopt an identity covariance structure. This removes the need to estimate Σ at all, but the price of doing this is a reversion to the IIA constraint inherent in, e.g. the MNL model. We regard the benefit of facilitating a simple sparse algorithm for objective pattern classification tasks as more than compensating for the inability fully to model more subjective, choice problems. Nonetheless, it would be interesting to pursue this question in future work. The resulting algorithm is therefore appropriate for the kind of classification tasks usually addressed by the MNL model but has the advantage of a simple approach to sparsity—we do not regard this as overly restrictive. Other work, e.g. [34, 35] have used this assumption and have made successful applications in practice.

The second difficulty is that there is no closed form available for calculating the integrals required in (31) and (32) to acquire the expectations of z_i. As discussed earlier, low dimensional (up to 20) numerical methods are available but with the obvious exponential increase in computational burden—undesirable in an iterative method. A second advantage of choosing the identity covariance structure is that the multi-dimensional Gaussian integrals now decouple into products of one-dimensional integrals for which efficient quadratures do exist, permitting the solution to the SMNP problem with a reasonable amount of computing resource.

Here we express the jth row vector of the ith design matrix H_i in (18) as $\boldsymbol{h}_{ij}$. Accordingly, the E-step becomes a closed form for z that when $j=1,\ldots,(C-1)$, $m=1,\ldots,(C-1)$

$$s_{im(t)} = E(z_{im}|\boldsymbol{y},\hat{\boldsymbol{\beta}}_{(t)}) = \begin{cases} \boldsymbol{h}_{ij}\hat{\boldsymbol{\beta}}_{(t)} + \frac{\phi(\boldsymbol{h}_{ij}\hat{\boldsymbol{\beta}}_{(t)}|0,1)}{\Phi(\boldsymbol{h}_{ij}\hat{\boldsymbol{\beta}}_{(t)})}, & \text{if } m=j; \\ \boldsymbol{h}_{im}\hat{\boldsymbol{\beta}}_{(t)} - \frac{\phi(s_{ij}-\boldsymbol{h}_{im}\hat{\boldsymbol{\beta}}_{(t)}|0,1)}{\Phi(s_{ij}-\boldsymbol{h}_{im}\hat{\boldsymbol{\beta}}_{(t)})}, & \text{if } m\neq j; \end{cases} \quad (33)$$

and when $j=C$, $m=1,\ldots,(C-1)$

$$s_{im(t)} = \boldsymbol{h}_{im}\hat{\boldsymbol{\beta}}_{(t)} - \frac{N(\boldsymbol{h}_{im}\hat{\boldsymbol{\beta}}_{(t)}|0,1)}{1-\Phi(\boldsymbol{h}_{im}\hat{\boldsymbol{\beta}}_{(t)})}, \quad (34)$$

The estimate of $\boldsymbol{\tau}$ is the same as in (29) since it is independent of Σ so the expected value of $\Upsilon(\tau)$ is still V. The M-step now only needs to update the parameter vector $\boldsymbol{\beta}$ thus:

$$\hat{\boldsymbol{\beta}}_{(t+1)} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + V_{(t)}\right)^{-1}\mathbf{H}^{\mathrm{T}}\boldsymbol{s}_{(t)} \quad (35)$$

and again, to avoid any divides-by-zero in computation, define:

$$U_{(t)} = \operatorname{diag}\left(\left|\hat{\beta}_{i(t)}\right|\right), \quad i=1,2,\ldots,(C-1)M \quad (36)$$

and re-write (35) as

$$\hat{\boldsymbol{\beta}}_{(t+1)} = U_{(t)}\left(U_{(t)}\mathbf{H}^{\mathrm{T}}\mathbf{H}U_{(t)} + I\right)^{-1}U_{(t)}\mathbf{H}^{\mathrm{T}}\boldsymbol{s}_{(t)}, \quad (37)$$

In summary, we give the detailed SMNP learning algorithm as follows:

Step 1 Compute the design matrix $\mathbf{H}$ for the training data, $\mathcal{D}$. Set the initial value for the $\boldsymbol{\beta}$.

Step 2 Calculate a current estimate for $\hat{\boldsymbol{\beta}}_{(t)}$ according to (37).

Step 3 (E-step) Calculate the diagonal matrix $U_{(t)}$ from (36) and the expected value of latent vector $\boldsymbol{s}_{(t)}$ from (33) and (34) according to the current estimate, $\hat{\boldsymbol{\beta}}_{(t)}$.

Step 4 (M-step) Update $\hat{\boldsymbol{\beta}}_{(t)}$ to $\hat{\boldsymbol{\beta}}_{(t+1)}$ using (37).

Step 5 Check for convergence through, e.g. $\delta = \frac{\left\|\hat{\boldsymbol{\beta}}_{(t+1)}-\hat{\boldsymbol{\beta}}_{(t)}\right\|}{\left\|\hat{\boldsymbol{\beta}}_{(t)}\right\|}$. If $\delta \ll 1$ then stop; otherwise set $t=t+1$ and return to the Step 2.

4 Numerical examples

Until a standard protocol is agreed for training/testing methodology and the reporting of results in machine learning classification experiments, the conduct of comparative studies presents a problem. The need to compare any new method with as large a cohort as possible of alternative techniques means that it is frequently impossible to make like-for-like comparisons in terms of say, number of cross-validatory folds for hyper-parameter selection, number of random data splits, etc. An alternative is to match methodology as closely as possible but this is not always possible because authors report a greater or lesser degree of detail. Another possibility is to replicate all other techniques with a common methodology. While this might be considered ideal, the potential for error, e.g. in coding, and the loss of objectivity—the author would be in

Table 1 Details of datasets used in comparative experiments available from the UCI machine learning repository [37]

Dataset	No. samples	No. classes	No. covariates
Iris	150	3	4
Wine	178	3	13
Glass	214	6	9
Thyroid 1	215	3	5
Dermatology	358	6	34
Balance scale	625	3	4
Vehicle	846	4	18
Vowel	990	11	11
Contraceptive method choice (CMC)	1,473	3	9
Car evaluation	1,728	4	6
Image segment	2,310	7	18
Letters	2,323	3	16
Waveform	5,000	3	21
Thyroid 2	7,200	3	21

charge of the competing methods—makes this less than satisfactory, notwithstanding the amount of additional work involved. Here we have sought a reasonably wide-ranging comparison with currently best performing techniques. This inevitably introduces some of the problems mentioned above. We have therefore tried to match the protocol of each comparative method as closely as possible but have used five-fold cross validation to optimize hyper-parameters and 20 replications with different, arbitrary splits into training and testing sets to provide a measure of spread.[3] All real-valued covariates are standardized and the MAP decision is taken. In each of our experiments, the SMNP algorithm is used as a kernel-based classifier, i.e. the design matrix, H corresponds to a kernel Gram matrix whose elements, h_{ij}, are given by $k(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\delta^2}\}$ and which is augmented by a unit column to represent any offset. δ represents the kernel width parameter. The subject of kernel machines has been widely explored in the literature and so no details are given here—the reader is instead directed to e.g. [36]. The 14 datasets used in the comparison are taken from the UCI Machine Learning Repository [37] and details are given in Table 1.

Before conducting the study, the SMNP code was tested against the SBP model of [8] using the settings and data published therein. The results obtained were identical, demonstrating that the multinomial code specializes to the binary situation and provides a degree of confidence in the new code. We then applied SMNP to the 14 datasets and have compared these with the best, to the best of our knowledge, reported results in the literature to date. The results are shown in Table 2.

We consider that, where there is an overlap between the intervals defined by mean ± standard error, such entries should be taken to be indistinguishable. Where no interval information is provided, we assume zero standard error for the deficient quantity. This generally militates against the proposed method in a “which method is best?” sense. However, the purpose here is simply to demonstrate that SMNP is a valid contender among current state-of-the-art techniques. Note also that the methods, REFNE [38], BAN [39] and VBGP [6] are not by their nature “sparse” hence the concept is not applicable (N/A).

In summary, examination of Table 2 shows that SMNP equals or betters (marginally) the classification accuracies of current best performers on these datasets in 11 of the 14 cases. It is, however, substantially worse in the “Glass” experiment, for reasons we are unable to explain but may be related to the severe imbalance in class priors in this sample. Focussing now on the level of sparsity achieved, first it is important to be clear that here “sparsity” is related to the number of data samples that must be retained to construct the trained classifier, i.e. a complete row of the matrix, B, must be eliminated. This differs from many authors’ usage which counts the number of zero entries in B (of course the two quantities are identical in the binary situation). We do not consider this latter to be useful since it may be that good sparsity can be achieved under that definition while still requiring all data to be retained in the final classifier. Examination of Table 2 shows that SMNP equals or betters the performance of other leading classifiers in seven of the eight eligible comparisons. The only failure takes place in the non-replicated experiment, “Thyroid 2” and here the difference is small given that the numbers represent only approximately 3% of the training sample.

5 Discussion and conclusion

In this paper a classification method for the multi-class problem is described based on the SBP method of Figueiredo [8]. We extend the main idea of SBP to the MNP model aiming to solve multi-classification by considering all classes at once and not by combining a number of binary classifiers. A hierarchical prior structure making use of Jeffreys’ non-informative hyper-prior is used to introduce sparseness and eliminate the need to adjust or estimate the hyper-parameter associated with the prior. The SMNP parameters are estimated via an EM algorithm. For convenience of implementation, a specialization of the SMNP model is constructed based on an identity covariance structure for the underlying latent variable model. We

[3] Except in the case of Thyroid 2, for reasons of runtime, owing to its large size.

Table 2 Mean error rates (MER (%)) ± standard error, and number of retained support vectors (N_{SV}) ± standard error on a sample of 14 datasets from the UCI machine learning repository [37]

Dataset	SMNP		Published best		Method
	MER (%)	N_{SV}	N_{SV}	MER (%)	
Iris	1.33 ± 0.61	7.67 ± 2.05	50.37	0.67	SMLR(l_1) [40]
Wine	1.15 ± 0.48	5.96 ± 1.17	9.55 ± 3.05	0.22 ± 0.31	sMKDA [4]
Glass	30.31 ± 1.91	16.24 ± 2.01	93.37	23.36	SMLR(l_1) [40]
Thyroid 1	2.77 ± 0.87	6.00 ± 1.14	8.95 ± 1.87	2.79 ± 0.33	sMKDA [4]
Dermatology	1.64 ± 0.41	13 ± 3.09	18.30 ± 4.52	1.51 ± 0.15	sMKDA [4]
Balance scale	6.72 ± 0.26	14.67 ± 2.14	N/A	6.72	REFNE [38]
Vehicle	14.7 ± 1.72	17.57 ± 2.58	45	12.53	SVM [41]
Vowel	3.08 ± 0.55	25.53 ± 4.91	23.91 ± 0.99	2.59 ± 0.43	sMKDA [4]
CMC	29.93 ± 0.20	21.5 ± 2.22	N/A	30.21	GS [42]
Car evaluation	7.91 ± 0.64	15.8 ± 2.1075	N/A	5.96 ± 0.44	BAN [39]
Image segment	7.72 ± 1.14	21.37 ± 3.61	N/A	7.8 ± 1.5	VBGP [7]
Letters	1.78 ± 0.93	15 ± 2.72	N/A	1.8 ± 0.8	VBGP [7]
Waveform	15.73 ± 0.83	12.37 ± 3.64	N/A	15.6 ± 0.7	VBGP [7]
Thyroid 2	2.04	115	111	3.28	sMKDA [4]

do not consider this restrictive for conventional use as a classifier—it provides a close approximation to the widely used multinomial logistic model. This reduces the need to perform multivariate Gaussian integrals and hence facilitates the solution of sizeable problems.

Several benchmark data sets are used to test the proposed method and they broadly indicate performance competitive with other state-of-the-art multi-class classifiers and reflect Figueiredo's findings for the binary model: that good classification accuracy is achieved whilst simultaneously providing excellent levels of sparsity. This makes the method particularly suited to its use, as here, as a kernel machine. Work to be considered for the future is the relaxation of the identity covariance condition to increase generality and the use of the technique for covariate selection.

References

1. Mukherjee S (2003) Classifying microarray data using support vector machines. In: Berrar DP, Dubitzky W, Granzow M (eds) A practical approach to microarray data analysis. Kluwer, Boston
2. Weston J, Watkins C (1998) Multi-class support vector machines. Tech Rep CSD-TR-98-04, Department of Computer Science, Royal Holloway University of London
3. Lee Y, Lin Y, Wahba G (2001) Multicategory support vector machines. Tech Rep 1043, Department of Statistics, University of Wisconsin, Madison, WI
4. Harrison RF, Pasupa K, (2009) Sparse multinomial kernel discriminant analysis (sMKDA). Pattern Recognit 42:1795–1802
5. Abe S, (2007) Sparse least-squares support vector training in the reduced empirical feature space. Pattern Anal Appl 10:203–214
6. Girolami M, Rogers S (2006) Variational Bayesian multinomial probit regression with Gaussian process priors. Neural Comput 18(8):1790–1817
7. Cawley GC, Talbot NLC, Girolami M (2007) Sparse multinomial logistic regression via Bayesian L1 regularisation. In: Schölkopf B, Platt JC, Hoffmann T (eds) Advances in neural information processing systems (NIPS), vol 19. MIT Press, Cambridge
8. Figueiredo MAT (2003) Adaptive sparseness for supervised learning. IEEE Trans Pattern Anal Mach Intell 25(9):1150–1159
9. Thurstone L (1927) A law of comparative judgement. Psychol Rev 34(4):273–286
10. Bock RD, Jones LV (1968) The measurement and prediction of judgment and choice. Holden-Day, San Francisco
11. McFadden D (1974) Conditional logit analysis of qualitative choice behavior. In: Zarembka P (ed) Frontiers in econometrics. Academic Press, New York
12. Domencich T, McFadden DL (1975) Urban travel demand: a behavioral analysis. North-Holland, Amsterdam
13. Hausman JA, Wise DA (1978) A conditional probit model for qualitative choice: discrete decisions recognizing interdependence and heterogeneous preferences. Econometrica 46(2):403–426
14. McFadden D (1989) A method of simulated moments for estimation of discrete response models without numerical integration. Econometrica 57:995–1026
15. Miguel J, Abito M (2005) A MATLAB implementation of a Halton sequence-based GHK simulator for multinomial probit models. Department of Economics National University of Singapore
16. Genz A, (1992) Numerical computation of multivariate normal probabilities. J Comput Graph Stat 1:141–149
17. Miwa T, Hayter AJ, Kuriki S (2003) The evaluation of general non-centred orthant proabilties. J R Stat Soc B 65:223–234
18. Manski C (1977) The structure of random utility models. Theory Decis 8:229–254
19. Lerman S, Manski C (1981) On the use of simulated frequencies to approximate choice probabilities. In: Manski C, McFadden D (eds) Structural analysis of discrete data with econometric applications. MIT Press, Cambridge

20. McCulloch R, Rossi PE (1994) An exact likelihood analysis of the multinomial probit model. J Econom 64(1–2):207–240
21. Nobile A (1998) A hybrid markov chain for the bayesian analysis of the multinomial probit model. Stat Comput 8:229–242
22. Chib S, Greenberg E (1995) Hierarchical analysis of SUR models with extensions to correlated serial errors and time varying parameter models. J Econom 68:339–360
23. McCulloch RE, Polson NG, Rossi PE (2000) A Bayesian analysis of the multinomial probit model with fully identified parameters. J Econom 99:173–193
24. Imai K, van Dyk DA (2005) A Bayesian analysis of the multinomial probit model using marginal data augmentation. J Econom 124(2):311–334
25. Tipping ME (2001) Sparse bayesian learning and the relevance vector machine. J Mach Learn Res 1:211–244
26. Bishop C, Tipping M (2000) Variational relevance vector machines. In: Proceedings of the 16th conference on uncertainty in artificial intelligence, pp 46–53
27. Shevade SK, Keerthi SS, (2003) A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics 19:2246–2253
28. Harrison RF, Pasupa K (2008) A simple iterative algorithm for parsimonious binary kernel Fisher discrimination. Patt Anal Appl. doi:10.1007/s10044-009-0162-1
29. Albert J, Chib S (1993) Bayesian analysis of binary and polychotomous response data. J Am Stat Assoc 88:669–679
30. Krishnapuram B, Hartemink AJ, Carin L, Figueiredo MAT (2004) A Bayesian approach to joint feature selection and classifier design. IEEE Trans Pattern Anal Mach Intell 26(9):1105-1111
31. Keane MP (1992) A note on identification in the multinomial probit model. American statistical association. Am Stat Assoc J Bus Econom Stat 10(2):193-200
32. Robert CP (2001) The Bayesian choice. Springer, New York
33. Edwards YD, Allenby GM (2003) Multivariate analysis of multiple response data. J Mark Res 40:321–334
34. Zhou X, Wang X, Dougherty ER (2006) Multi-class cancer classification using multinomial probit regression with Bayesian gene selection. IEE Proc Syst Biol 153(2):70–78
35. Yau P, Kohn R, Wood S (2003) Bayesian variable selection and model avergaing in high-dimensional multinomial nonparametric regression. J Comput Graph Stat 12:23–54
36. Schölkopf B, Smola A (2002) Learning with kernels. MIT Press, Cambridge
37. Asuncion A, Newman DJ (2007) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. http://www.ics.uci.edu/~mlearn/MLRepository.html
38. Zhou ZH, Jiang Y, Chen SF (2003) Extracting symbolic rules from trained neural network ensembles. AI Commun 16:3–15
39. Cheng J, Greiner R (1999) Comparing Bayesian networks classifiers. In: Proceedings of the 15th conference on uncertainty in artificial intelligence (UAI 1999), pp 101–108
40. Krishnapuram B, Carin L, Figueiredo MAT, Hartemink AJ (2005) Sparse multinomial logistic regression: fast algorithms and generalization bounds. IEEE Trans Pattern Anal Mach Intell 27(6):957–968
41. Hsu CW, Lin CJ (2002) A comparison of methods for multi-class support vector machines. IEEE Trans Neural Netw 13(2):415–425
42. Ray S, Page D (2005) Generalized skewing for functions with continuous and nominal attributes. In: The 22nd international conference on machine learning (ICML 2005), pp 705–712