Pattern Recognition 42 (2009) 2671 – 2683

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Combining feature spaces for classification

Theodoros Damoulas*, Mark A. Girolami

Inference Research Group, Department of Computing Science, Faculty of Information and Mathematical Sciences, University of Glasgow, 18 Lilybank Gardens, Glasgow G12 8QQ, Scotland, UK

ARTICLE INFO

Article history:
Received 2 November 2007
Received in revised form 24 January 2009
Accepted 5 April 2009

Keywords:
Variational Bayes approximation
Multiclass classification
Kernel combination
Hierarchical Bayes
Bayesian inference
Ensemble learning
Multi-modal modelling
Information integration

ABSTRACT

In this paper we offer a variational Bayes approximation to the multinomial probit model for basis expansion and kernel combination. Our model is well-founded within a hierarchical Bayesian framework and is able to instructively combine available sources of information for multinomial classification. The proposed framework enables informative integration of possibly heterogeneous sources in a multitude of ways, from the simple summation of feature expansions to weighted product of kernels, and it is shown to match and in certain cases outperform the well-known ensemble learning approaches of combining individual classifiers. At the same time the approximation reduces considerably the CPU time and resources required with respect to both the ensemble learning methods and the full Markov chain Monte Carlo, Metropolis–Hastings within Gibbs solution of our model. We present our proposed framework together with extensive experimental studies on synthetic and benchmark datasets and also for the first time report a comparison between summation and product of individual kernels as possible different methods for constructing the composite kernel matrix.

1. Introduction

Classification, or supervised discrimination, has been an active research field within the pattern recognition and machine learning communities for a number of decades. The interest comes from the nature of the problems encountered within the communities, a large number of which can be expressed as discrimination tasks, i.e. to be able to learn and/or predict in which category an object belongs. In the supervised and semi-supervised learning setting, [24,6], where labeled (or partially labeled) training sets are available in order to predict labels for unseen objects, we are in the classification domain and in the unsupervised learning case, where the machine is asked to learn categories from the available unlabeled data, we fall in the domain of clustering.

In both of these domains a recurring theme is how to utilize constructively all the information that is available for the specific problem in hand. This is specifically true when a large number of features from different sources are available for the same object and there is limited or no a priori knowledge of their significance and contribution to the classification or clustering task. In such cases, concatenating all the features into a single feature space does not guarantee an optimum performance and it exacerbates the "curse of dimensionality" problem.

The problem of classification in the case of multiple feature spaces or sources has been mostly dealt with in the past by ensemble learning methods [12], namely combinations of individual classifiers. The idea behind that approach is to train one classifier in every feature space and then combine their class predictive distributions. Different ways of combining the output of the classifiers have been studied [18,34,19] and also meta-learning an overall classifier on these distributions [36] has been proposed.

The drawbacks of combining classifiers lie on their theoretical justification, on the processing loads incurred as multiple training has to be performed, and on the fact that the individual classifiers operate independently on the data. Their performance has been shown to significantly improve over the best individual classifier in many cases but the extra load of training multiple classifiers may possibly restrain their application when resources are limited. The typical combination rules for classifiers are ad hoc methods [5] that are based on the notions of the *linear* or *independent* opinion pools. However, as noted by Berger [5], "… it is usually better to approximate a "correct answer" than to adopt a completely ad hoc approach." and this has a bearing, as we shall see, on the nature of classifier combination and the motivation for kernel combination.

In this paper we offer, for the classification problem, a multiclass kernel machine [30,31] that is able to combine kernel spaces in an informative manner while at the same time learning their significance. The proposed methodology is general and can also be

* Corresponding author. Tel.: +44 141 330 2421; fax: +44 141 3308627.
E-mail addresses: theo@dcs.gla.ac.uk (T. Damoulas), girolami@dcs.gla.ac.uk (M.A. Girolami).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.04.002

employed outside the "kernel-domain", i.e. without the need to embed the features into Hilbert spaces, by allowing for combination of basis function expansions. That is useful in the case of multi-scale problems and wavelets [3]. The combination of kernels or dictionaries of basis functions as an alternative to classifier combination offers the advantages of reduced computational requirements, the ability to learn the significance of the individual sources and an improved solution based on the inferred significance.

Previous related work, includes the area of *kernel learning*, where various methodologies have been proposed in order to learn the possibly composite kernel matrix, by convex optimization methods [20,33,22,23], by the introduction of hyper-kernels [28] or by direct ad hoc construction of the composite kernel [21]. These methods operate within the context of support vector machine (SVM) learning and hence inherit the arguable drawbacks of non-probabilistic outputs and ad hoc extensions to the multiclass case. Furthermore, another related approach within the non-parametric Gaussian process (GP) methodology [27,29] has been proposed by Girolami and Zhong [16], where instead of kernel combination the integration of information is achieved via combination of the GP covariance functions.

Our work further develops the work of Girolami and Rogers [14], which we generalize to the multiclass case by employing a multinomial probit likelihood and introducing latent variables that give rise to efficient Gibbs sampling from the parameter posterior distribution. We bound the marginal likelihood or model evidence [25] and derive the variational Bayes (VB) approximation for the multinomial probit composite kernel model, providing a fast and efficient solution. We are able to combine kernels in a general way e.g. by summation, product or binary rules and learn the associated weights to infer the significance of the sources.

For the first time we offer, a VB approximation on an explicit multiclass model for kernel combination and a comparison between kernel combination methods, from summation to weighted product of kernels. Furthermore, we compare our methods against classifier combination strategies on a number of datasets and we show that the accuracy of our VB approximation is comparable to that of the full Markov chain Monte Carlo (MCMC) solution.

The paper is organized as follows. First we introduce the concepts of composite kernels, kernel combination rules and classifier combination strategies and give an insight on their theoretical underpinnings. Next we introduce the multinomial probit composite kernel model, the MCMC solution and the VB approximation. Finally, we present results on synthetic and benchmark datasets and offer discussion and concluding remarks.

2. Classifier versus kernel combination

The theoretical framework behind classifier combination strategies has been explored relatively recently[1] [12,18,19], despite the fact that ensembles of classifiers have been widely used experimentally before. We briefly review that framework in order to identify the common ground and the deviations from the classifier combination to the proposed kernel (or basis function) combination methodology.

Consider S feature spaces or sources of information.[2] An object n belonging to a dataset $\{\mathbf{X},\mathbf{t}\}$, with $\mathbf{X}$ the collection of all objects and $\mathbf{t}$ the corresponding labels, is represented by S, D^s-dimensional feature vectors $\mathbf{x}_n^s$ for $s=1,\ldots,S$ and $\mathbf{x}_n^s \in \Re^{D^s}$. Let the number of classes be C with the target variable $t_n = c = 1,\ldots,C$ and the number of objects N with $n=1,\ldots,N$. Then, the class posterior probability for object n will be $P(t_n|\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S)$ and according to Bayesian decision theory we would assign the object n with the class that has the maximum a posteriori probability.

The classifier combination methodologies are based on the following assumption, which as we shall see is not necessary for the proposed kernel combination approach. The assumption is that although "it is essential to compute the probabilities of various hypotheses by considering all the measurements simultaneously … it may not be a practical proposition." [18] and it leads to a further approximation on the (noisy) class posterior probabilities which are now broken down to individual contributions from classifiers trained on each feature space s.

The *product combination* rule, assuming equal a priori class probabilities:

$$P_{\text{prod}}(t_n|\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S) = \frac{\prod_{s=1}^{S}(P(t_n|\mathbf{x}_n^s)+\varepsilon^s)}{\sum_{t_n=1}^{C}\prod_{s=1}^{S}(P(t_n|\mathbf{x}_n^s)+\varepsilon^s)} \tag{1}$$

The *mean combination* rule:

$$P_{\text{mean}}(t_n|\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S) = \frac{1}{S}\sum_{s=1}^{S}(P(t_n|\mathbf{x}_n^s)+\varepsilon^s) \tag{2}$$

with ε^s the prediction error made by the individual classifier trained on feature space s.

The theoretical justification for the product rule comes from the *independence* assumption of the feature spaces, where $\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S$ are assumed to be uncorrelated; and the mean combination rule is derived on the opposite assumption of extreme correlation. As it can be seen from Eqs. (1) and (2), the individual errors ε^s of the classifiers are either added or multiplied together and the hope is that there will be a synergetic effect from the combination that will cancel them out and hence reduce the overall classification error.

Instead of combining classifiers we propose to combine the feature spaces and hence obtain the class posterior probabilities from the model

$$P_{\text{spaces}}(t_n|\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S) = P(t_n|\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S)+\varepsilon$$

where now we only have one error term ε from the single classifier operating on the composite feature space. The different methods to construct the composite feature space reflect different approximations to the $P(t_n|\mathbf{x}_n^1,\ldots,\mathbf{x}_n^S)$ term without incorporating individual classifier errors into the approximation. Still though, underlying assumptions of independence for the construction of the composite feature space are implied through the kernel combination rules and as we shall see this will lead to the need for *diversity* as it is commonly known for other ensemble learning approaches.

Although our approach is general and can be applied to combine basis expansions or kernels, we present here the composite kernel construction as our main example. Embedding the features into Hilbert spaces via the kernel trick [30,31] we can define[3] :

The $N \times N$ *mean composite* kernel as

$$\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}(\mathbf{x}_i,\mathbf{x}_j) = \sum_{s=1}^{S}\beta_s K^{s\boldsymbol{\theta}_s}(\mathbf{x}_i^s,\mathbf{x}_j^s) \quad \text{with} \quad \sum_{s=1}^{S}\beta_s = 1 \ \text{ and } \ \beta_s \geqslant 0 \ \ \forall s$$

[1] At least in the classification domain for machine learning and pattern recognition. The related statistical decision theory literature has studied opinion pools much earlier, see Berger [5] for a full treatment.

[2] Throughout this paper m denotes scalar, $\mathbf{m}$ vector and $\mathbf{M}$ a Matrix. If $\mathbf{M}$ is a $C \times N$ matrix then $\mathbf{m}_c$ is the cth row vector, $\mathbf{m}_n$ the nth column vector of that matrix and all other indices imply row vectors.

[3] Superscripts denote "function of", i.e. $\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}$ denotes that the kernel $\mathbf{K}$ is a function of $\boldsymbol{\beta}$ and $\boldsymbol{\Theta}$, unless otherwise specified.

The $N \times N$ *product composite* kernel[4] as

$$\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}(\mathbf{x}_i, \mathbf{x}_j) = \prod_{s=1}^{S} K^{s\boldsymbol{\theta}_s}(\mathbf{x}_i^s, \mathbf{x}_j^s)^{\beta_s} \quad \text{with } \beta_s \geqslant 0 \ \ \forall s$$

And the $N \times N$ *binary composite* kernel as

$$\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{S} \beta_s K^{s\boldsymbol{\theta}_s}(\mathbf{x}_i^s, \mathbf{x}_j^s) \quad \text{with } \beta_s \in \{0,1\} \ \ \forall s$$

where $\boldsymbol{\Theta}$ are the kernel parameters, $\boldsymbol{\beta}$ the combinatorial weights and K is the kernel function employed, typically in this study, a Gaussian or polynomial function. We can now weigh accordingly the feature spaces by the parameter $\boldsymbol{\beta}$ which, as we shall see further on, can be inferred from the observed evidence. Learning these combinatorial parameters is the main objective of multiple kernel learning methods such as the proposed ones in this article.

3. The multinomial probit model

In accordance with Albert and Chib [1], we introduce auxiliary variables $\mathbf{Y} \in \Re^{C \times N}$ that we regress onto with our composite kernel and the parameters (regressors) $\mathbf{W} \in \Re^{C \times N}$. The intuition is that the regressors express the weight with which a data point "votes" for a specific class c and the auxiliary variables are continuous "target" values that are related to class membership ranking for a specific point. Following the standardized noise model $\varepsilon \sim \mathcal{N}(0,1)$ which results in $y_{cn} = \mathbf{w}_c \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}} + \varepsilon$, with $\mathbf{w}_c$ the $1 \times N$ row vector of class c regressors and $\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}$ the $N \times 1$ column vector of inner products for the nth element, leads to the following Gaussian probability distribution:

$$p(y_{cn} | \mathbf{w}_c, \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}) = \mathcal{N}_{y_{cn}}(\mathbf{w}_c \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}, 1) \tag{3}$$

The link from the auxiliary variable y_{cn} to the discrete target variable of interest $t_n \in \{1, \ldots, C\}$ is given by $t_n = i \iff y_{in} > y_{jn} \ \forall j \neq i$ and by the following marginalization $P(t_n = i | \mathbf{W}, \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}) = \int P(t_n = i | \mathbf{y}_n) P(\mathbf{y}_n | \mathbf{W}, \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}) \, d\mathbf{y}_n$, where $P(t_n = i | \mathbf{y}_n)$ is a delta function, results in the multinomial probit likelihood as

$$P(t_n = i | \mathbf{W}, \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}) = \mathscr{E}_{p(u)} \left\{ \prod_{j \neq i} \Phi(u + (\mathbf{w}_i - \mathbf{w}_j) \mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}) \right\} \tag{4}$$

where $\mathscr{E}$ is the expectation taken with respect to the standardized normal distribution $p(u) = \mathcal{N}(0,1)$ and Φ is the cumulative density function. We can now consider the prior and hyper-prior distributions to be placed on our model.

3.1. Prior probabilities

Having derived the multinomial probit likelihood we complete our Bayesian model by introducing prior distributions on the model parameters. The choice of prior distributions is justified as follows: for the *regressors* $\mathbf{W}$ we place a product of zero mean Gaussian distributions with scale ζ_{cn} that reflects independence (product) and lack of prior knowledge (zero mean normal). The only free parameter on this distribution is the *scale*, on which we place a gamma prior distribution that ensures a positive value, is a conjugate pair with the normal and is controlled via the hyper-parameters τ, υ. This prior setting propagates the uncertainty to a higher level and it is commonly employed in the statistics literature [11] for the above-mentioned reasons of conjugacy and (lack of) prior knowledge.

[4] In this case *only* the superscript β_s in $K^{s\boldsymbol{\theta}_s}(\mathbf{x}_i^s, \mathbf{x}_j^s)^{\beta_s}$ denotes power. The superscript s in $K^s(.,.)$ or generally in $\mathbf{K}^s$ indicates that this is the sth base kernel.

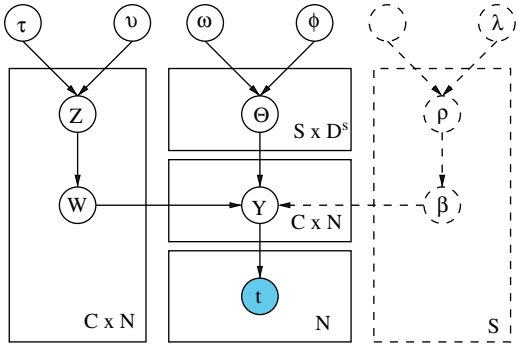

Fig. 1. Plates diagram of the model for *mean composite* kernel.

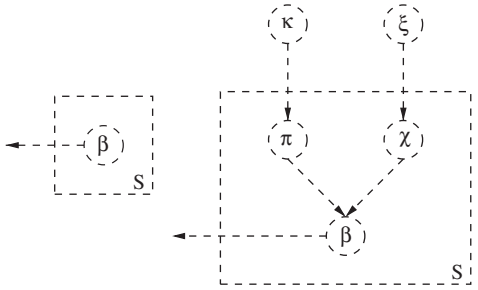

Fig. 2. Modification for *binary composite* (left) and *product composite* kernel (right).

The *hyper-parameter* prior setting of τ, υ dictates the prior form of the gamma distribution and can be set to ensure an uninformative distribution (as it will be done in this work) or we can induce sparsity via setting a flat prior ($\tau, \upsilon \to 0$) as in Damoulas et al. [9]. This leads to the empirical Bayes method of type-II maximum likelihood as employed in the construction of relevant vector machines [35].

Furthermore, we place a gamma distribution on each *kernel parameter* since $\theta_{sd} \in \Re^+$. In the case of the *mean composite* kernel, a Dirichlet distribution with parameters $\boldsymbol{\rho}$ is placed on the combinatorial weights in order to satisfy the constraints imposed on the possible values which are defined on a simplex and assists in statistical identifiability. A further gamma distribution, again to ensure positive values, is placed on each ρ_s with associated hyper-parameters μ, λ.

In the *product composite* kernel case we employ the right dashed plate in Fig. 2 which places a gamma distribution on the combinatorial weights $\boldsymbol{\beta}$, that do not need to be defined on a simplex anymore, with an exponential hyper-prior distribution on each of the parameters π_s, χ_s.

Finally, in the *binary composite* kernel case we employ the left dashed plate in Fig. 2 which places a binomial distribution on each β_s with equal probability of being 1 or zero (unless prior knowledge says otherwise). The small size of the possible 2^S states of the $\boldsymbol{\beta}$ vector allows for their explicit consideration in the inference procedure and hence there is no need to place any hyper-prior distributions.

The model, for case of the *mean composite* kernel, is depicted graphically in Fig. 1 where the conditional relation of the model parameters and associated hyper-parameters can be seen. The accompanied variations for the *binary* and *product composite* kernel are given in Fig. 2.

The intuition behind the hierarchical construction of the proposed model is that uncertainty is propagated into a higher level of prior distribution setting. In other words, we place further hyper-prior distributions on parameters of the prior distributions and we let the evidence guide us to the required posteriors. In that way the extra levels allow less sensitivity for initial settings (in the case where

fixed values are used for the hyper-parameters instead of empirical Bayes) since a specific hyper-parameter value will still lead to a (prior) *distribution* of possible parameter values which can still be "corrected" by the evidence of the data towards the appropriate final posterior distribution.

3.2. Exact Bayesian inference: Gibbs sampling

The standard statistical solution for performing Bayesian inference is via Markov chain Monte Carlo sampling methods that are powerful but computationally expensive. These approaches still approximate the desired posterior distribution but only due to the limited number of drawn samples, hence the name exact inference. From these methods, Gibbs sampling is the preferred alternative as it avoids the need for tuning acceptance ratios as is typical in the original Metropolis–Hastings (MH) algorithms [17]. Such a sampling scheme is only possible when the hard-to-sample-from joint posterior distribution can be broken down to conditional distributions from which we can easily draw samples.

In our model such a sampler naturally arises by exploiting the conditional distributions of $\mathbf{Y}|\mathbf{W},\ldots$ and $\mathbf{W}|\mathbf{Y},\ldots$, so that we can draw samples from the parameter posterior distribution $P(\boldsymbol{\Psi}|\mathbf{t},\mathbf{X},\boldsymbol{\Xi})$ where $\boldsymbol{\Psi}=\{\mathbf{Y},\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{\rho}\}$ and $\boldsymbol{\Xi}$ the aforementioned hyper-parameters of the model. In the case of non-conjugate pair of distributions, Metropolis–Hastings sub-samplers can be employed for posterior inference. More details of the Gibbs sampler are provided analytically in [8] and summarized in Appendix A.

3.3. Approximate inference: variational approximation

Exact inference via the Gibbs sampling scheme is computationally expensive and in this article we derive a deterministic variational approximation which is an efficient method with comparable accuracy to the full sampling scheme as it will be further demonstrated. The variational methodology, see Beal [4] for a recent treatment, offers a lower bound on the model evidence using an ensemble of factored posteriors to approximate the joint parameter posterior distribution. Although the factored ensemble implies independence of the approximate posteriors, which is a typical strong assumption of variational methods, there is weak coupling through the current estimates of parameters as it can be seen below.

Considering the joint likelihood of the model[5] defined as $p(\mathbf{t},\boldsymbol{\Psi}|\mathbf{X},\boldsymbol{\Xi})=p(\mathbf{t}|\mathbf{Y})p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta})p(\mathbf{W}|\mathbf{Z})p(\mathbf{Z}|\tau,\upsilon)p(\boldsymbol{\beta}|\boldsymbol{\rho})p(\boldsymbol{\Theta}|\omega,\phi)p(\boldsymbol{\rho}|\mu,\lambda)$ and the factorable ensemble approximation of the required posterior $p(\boldsymbol{\Psi}|\boldsymbol{\Xi},\mathbf{X},\mathbf{t}) \approx Q(\boldsymbol{\Psi})=Q(\mathbf{Y})Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})Q(\mathbf{Z})Q(\boldsymbol{\rho})$ we can bound the model evidence using Jensen's inequality:

$$\log p(\mathbf{t}) \geq \mathscr{E}_{Q(\boldsymbol{\Psi})}\{\log p(\mathbf{t},\boldsymbol{\Psi}|\boldsymbol{\Xi})\} - \mathscr{E}_{Q(\boldsymbol{\Psi})}\{\log Q(\boldsymbol{\Psi})\} \tag{5}$$

and minimize it with distributions $Q(\boldsymbol{\Psi}_i) \propto \exp(\mathscr{E}_{Q(\boldsymbol{\Psi}_{-i})}\{\log p(\mathbf{t},\boldsymbol{\Psi}|\boldsymbol{\Xi})\})$ where $Q(\boldsymbol{\Psi}_{-i})$ is the factorable ensemble with the ith component removed. The above standard steps describe the adoption of the *mean-field* or variational approximation theory for our specific model.

The resulting approximate posterior distributions are given below with full details of the derivations in Appendix B. First, the approximate posterior over the auxiliary variables is given by

$$Q(\mathbf{Y}) \propto \prod_{n=1}^{N} \delta(y_{i,n} > y_{k,n} \forall k \neq i)\delta(t_n = i)\mathscr{N}_{\mathbf{y}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, \mathbf{I}) \tag{6}$$

which is a product of N C-dimensional conically truncated Gaussians demonstrating independence across samples as expected from our initial i.i.d assumption. The shorthand tilde notation denotes posterior expectations in the usual manner, i.e. $\widetilde{f(\boldsymbol{\beta})} = \mathscr{E}_{Q(\boldsymbol{\beta})}\{f(\boldsymbol{\beta})\}$, and the posterior expectations (details in Appendix C) for the auxiliary variable follow as

$$\widetilde{y}_{cn} = \widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \frac{\mathscr{E}_{p(u)}\{\mathscr{N}_u(\widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, 1)\Phi_u^{n,i,c}\}}{\mathscr{E}_{p(u)}\{\Phi(u + \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}})\Phi_u^{n,i,c}\}} \tag{7}$$

$$\widetilde{y}_{in} = \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \left(\sum_{c\neq i}\widetilde{y}_{cn} - \widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right) \tag{8}$$

where Φ is the standardized cumulative distribution function (CDF) and $\Phi_u^{n,i,c} = \prod_{j\neq i,c}\Phi(u + \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}})$. Next, the approximate posterior for the regressors can be expressed as

$$Q(\mathbf{W}) \propto \prod_{c=1}^{C}\mathscr{N}_{\mathbf{w}_c}(\widetilde{\mathbf{y}}_c\mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\mathbf{V}_c, \mathbf{V}_c) \tag{9}$$

where the covariance is defined as

$$\mathbf{V}_c = \left(\sum_{i=1}^{S}\sum_{j=1}^{S}\widetilde{\beta_i\beta_j}\mathbf{K}^{\widetilde{\theta}_i\widetilde{\theta}_j} + (\widetilde{\mathbf{Z}}_c)^{-1}\right)^{-1} \tag{10}$$

and $\widetilde{\mathbf{Z}}_c$ is a diagonal matrix of the expected variances $\widetilde{\zeta}_i,\ldots,\widetilde{\zeta}_N$ for each class. The associated posterior mean for the regressors is therefore $\widetilde{\mathbf{w}}_c = \widetilde{\mathbf{y}}_c\mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\mathbf{V}_c$ and we can see the coupling between the auxiliary variable and regressor posterior expectation.

The approximate posterior for the variances $\mathbf{Z}$ is an updated product of inverse-gamma distributions (gamma on the scales) and the posterior mean for the scale is given by

$$\frac{\tau + \frac{1}{2}}{\upsilon + \frac{1}{2}\widetilde{w_{cn}^2}} \tag{11}$$

for details see Appendix B.3 or Denison et al. [11]. Finally, the approximate posteriors for the kernel parameters $Q(\boldsymbol{\Theta})$, the combinatorial weights $Q(\boldsymbol{\beta})$ and the associated hyper-prior parameters $Q(\boldsymbol{\rho})$, or $Q(\boldsymbol{\pi}), Q(\boldsymbol{\chi})$ in the product composite kernel case, can be obtained by importance sampling [2] in a similar manner to Girolami and Rogers [15] since no tractable analytical solution can be offered. Details are provided in Appendix B.

Having described the approximate posterior distributions of the parameters and hence obtained the posterior expectations we turn back to our original task of making class predictions $\mathbf{t}^*$ for N_{test} new objects $\mathbf{X}^*$ that are represented by S different information sources $\mathbf{X}^{s*}$ embedded into Hilbert spaces as base kernels $\mathbf{K}^{*s\boldsymbol{\theta}_s,\beta_s}$ and combined into a composite *test* kernel $\mathbf{K}^{*\boldsymbol{\Theta},\boldsymbol{\beta}}$. The predictive distribution for a single new object $\mathbf{x}^*$ is given by $p(t^*=c|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = \int p(t^*=c|\mathbf{y}^*)p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t})\,d\mathbf{y}^* = \int \delta_c^* p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t})\,d\mathbf{y}^*$ which ends up, see Appendix D for complete derivation, as

$$p(t^*=c|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = E_{p(u)}\left\{\prod_{j\neq c}\Phi\left[\frac{1}{\widetilde{v_j^*}}(u\widetilde{v_c^*} + \widetilde{m_c^*} - \widetilde{m_j^*})\right]\right\} \tag{12}$$

where, for the general case of N_{test} objects, $\widetilde{\mathbf{m}_c^*} = \widetilde{\mathbf{y}}_c\mathbf{K}(\mathbf{K}^*\mathbf{K}^{*T} + \mathbf{V}_c^{-1})^{-1}\mathbf{K}^*\widetilde{\mathscr{V}_c^*}$ and $\widetilde{\mathscr{V}_c^*} = (\mathbf{I} + \mathbf{K}^{*T}\mathbf{V}_c\mathbf{K}^*)$ while we have dropped the notation for the dependance of the train $\mathbf{K}(N\times N)$ and test $\mathbf{K}^*(N\times N_{test})$ kernels on $\boldsymbol{\Theta},\boldsymbol{\beta}$ for clarity. In Algorithm 1 we summarize the VB approximation in a pseudo-algorithmic fashion.

[5] The *mean composite* kernel is considered as an example. Modifications for the *binary* and *product composite* kernel are trivial and respective details are given in the appendices.

Algorithm 1. VB multinomial probit composite kernel regression

Initialize $\boldsymbol{\Xi}$, sample $\boldsymbol{\Psi}$, create $\mathbf{K}_s|\beta_s, \theta_s$ and hence $\mathbf{K}|\boldsymbol{\beta}, \boldsymbol{\Theta}$
while Lower Bound changing **do**
  $\widetilde{\mathbf{w}}_c \leftarrow \widetilde{\mathbf{y}}_c \mathbf{K} \mathbf{V}_c$
  $\widetilde{y}_{cn} \leftarrow \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \frac{\mathscr{E}_{p(u)}\{\mathcal{N}_u(\widetilde{\mathbf{w}}_c \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, 1)\Phi_u^{n,i,c}\}}{\mathscr{E}_{p(u)}\{\Phi(u + \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}})\Phi_u^{n,i,c}\}}$
  $\widetilde{y}_{in} \leftarrow \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \left(\sum_{j \neq i} \widetilde{y}_{jn} - \widetilde{\mathbf{w}}_j \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right)$
  $\widetilde{\zeta}_{cn}^{-1} \leftarrow \frac{\tau + \frac{1}{2}}{\upsilon + \frac{1}{2}\widetilde{w_{cn}^2}}$
  $\widetilde{\boldsymbol{\rho}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}} \leftarrow \widetilde{\boldsymbol{\rho}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}}|\widetilde{\mathbf{w}}_c, \widetilde{\mathbf{y}}_n$ by importance sampling
  Update $\mathbf{K}|\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}}$ and $\mathbf{V}_c$
end while
Create composite test kernel $\mathbf{K}^*|\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}}$
$\widetilde{\mathscr{V}_c^*} \leftarrow (\mathbf{I} + \mathbf{K}^{*T} \mathbf{V}_c \mathbf{K}^*)$
$\widetilde{\mathbf{m}_c^*} \leftarrow \widetilde{\mathbf{y}}_c \mathbf{K}(\mathbf{K}^* \mathbf{K}^{*T} + \mathbf{V}_c^{-1})^{-1} \mathbf{K}^* \widetilde{\mathscr{V}_c^*}$
for $n = 1$ to N_{test} **do**
  for $c = 1$ to C **do**
    for $i = 1$ to K Samples **do**
      $u_i \leftarrow \mathcal{N}(0,1)$, $p_{cn}^i \leftarrow \prod_{j \neq c} \Phi\left[\frac{1}{\widetilde{v_j^*}}(u_i \widetilde{v_c^*} + \widetilde{m_c^*} - \widetilde{m_j^*})\right]$
    end for
  end for
$P(t_n^* = c|\mathbf{x}_n^*, \mathbf{X}, \mathbf{t}) = \frac{1}{K}\sum_{i=1}^{K} p_{cn}^i$
end for

4. Experimental studies

This section presents the experimental findings of our work with emphasis first on the performance of the VB approximation with respect to the full MCMC Gibbs sampling solution and next on the efficiency of the proposed feature space combination method versus the well known classifier combination strategies. We employ two artificial low-dimensional datasets, a linearly and a non-linearly separable one introduced by Neal [27], for the comparison between the approximation and the full solution; standard UCI[6] multinomial datasets for an assessment of the VB performance; and finally, two well known large benchmark datasets to demonstrate the utility of the proposed feature space combination against ensemble learning and assess the performance of the different kernel combination methods.

Unless otherwise specified, all the hyper-parameters were set to uninformative values and the Gaussian kernel parameters were fixed to $1/D$ where D the dimensionality of the vectors. Furthermore, the convergence of the VB approximation was determined by monitoring the lower bound and the convergence occurred when there was less than 0.1% increase in the bound or when the maximum number of VB iterations was reached. The burn-in period for the Gibbs sampler was set to 10% of the total 100,000 of samples. Finally, all the CPU times reported in this study are for a 1.6 GHz Intel based PC with 2Gb RAM running unoptimized Matlab codes and all the p-values reported are for a two sample *t-test* with the null hypothesis that the distributions have a common mean.

4.1. Synthetic datasets

In order to illustrate the performance of the VB approximation against the full Gibbs sampling solution, we employ two low-dimensional datasets which enable us to visualize the decision boundaries and posterior distributions produced by either method. First we consider a linearly separable case in which we construct the dataset by fixing our regressors $\mathbf{W} \in \Re^{C \times D}$, with $C = 3$ and $D = 3$, to known values and sample two-dimensional covariates $\mathbf{X}$ plus a constant term. In that way, by knowing the true values of our regressors, we can examine the accuracy of both the Gibbs posterior distribution and the approximate posterior estimate of the VB. In Fig. 3 the dataset together with the optimal decision boundaries constructed by the known regressor values can be seen.

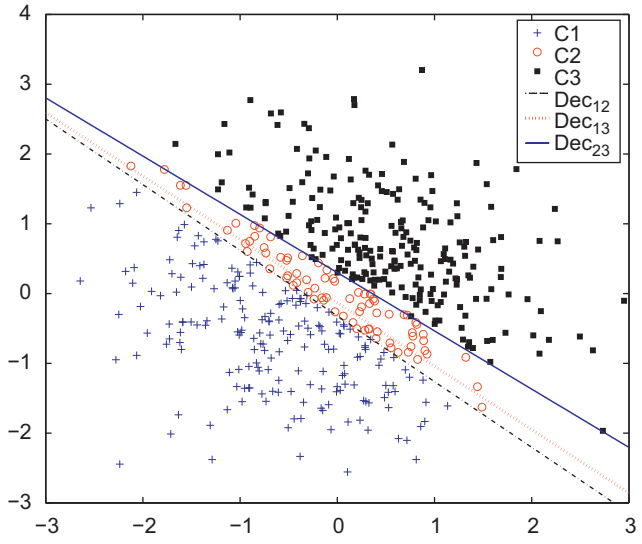

Fig. 3. Linearly separable dataset with known regressors defining the decision boundaries. C_n denotes the members of class n and Dec_{ij} is the decision boundary between classes i and j.

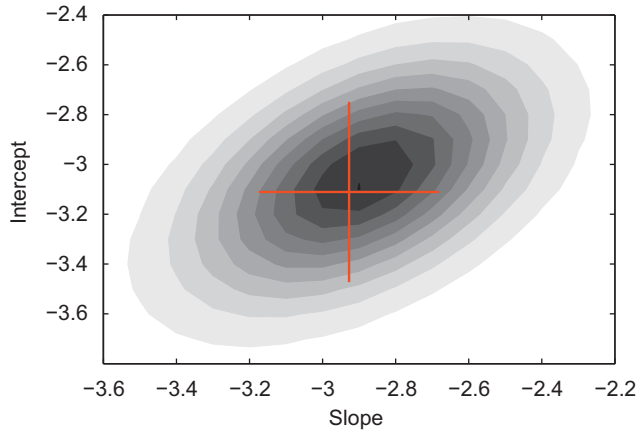

Fig. 4. Gibbs posterior distribution of a decision boundary's (Dec_{12}) slope and intercept for a chain of 100,000 samples. The cross shows the original decision boundary employed to sample the dataset.

In Figs. 4 and 5 we present the posterior distributions of one decision boundary's (Dec_{12}) slope and intercept based on both our obtained Gibbs samples and the approximate posterior of the regressors $\mathbf{W}$. As we can see, the variational approximation is in agreement with the mass of the Gibbs posterior and it successfully captures the pre-determined regressors values.

However, as it can be observed the approximation is overconfident in the prediction and produces a smaller covariance for the posterior distribution as expected [10]. Furthermore, the

[6] A. Asuncion, D.J. Newman, 2007. UCI Machine Learning Repository, Department of Information and Computer Science, University of California, Irvine, CA [http://www.ics.uci.edu/~mlearn/MLRepository.html]

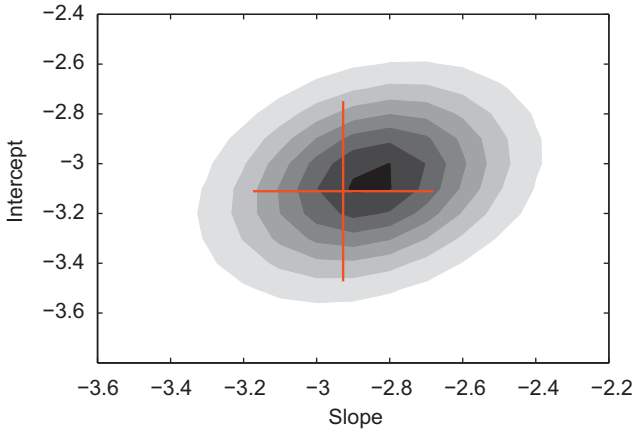

Fig. 5. The variational approximate posterior distribution for the same case as above. Employing 100,000 samples from the approximate posterior of the regressors **W** in order to estimate the approximate posterior of the slope and intercept.

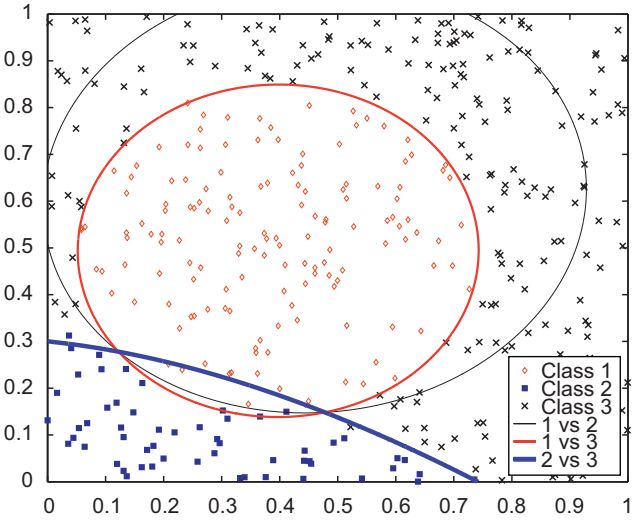

Fig. 6. Decision boundaries from the Gibbs sampling solution.

probability mass is concentrated in a very small area due to the very nature of VB approximation and similar mean field methods that make extreme "judgments" as they do not explore the posterior space by Markov chains:

$$C_{\text{Gibbs}} = \begin{bmatrix} 0.16 & 0.18 \\ 0.18 & 0.22 \end{bmatrix}, \quad C_{\text{VB}} = \begin{bmatrix} 0.015 & 0.015 \\ 0.015 & 0.018 \end{bmatrix} \tag{13}$$

The second synthetic dataset we employ is a four-dimensional three-class dataset $\{\mathbf{X}, \mathbf{t}\}$ with $N = 400$, first described by Neal [27], which defines the first class as points in an ellipse $\alpha > x_1^2 + x_2^2 > \beta$, the second class as points below a line $\alpha x_1 + \beta x_2 < \gamma$ and the third class as points surrounding these areas, see Fig. 6.

We approach the problem by: (1) introducing a second order polynomial expansion on the original dataset $F(\mathbf{x}_n) = [1\ x_{n1}\ x_{n2}\ x_{n1}^2\ x_{n1}x_{n2}\ x_{n2}^2]$ while disregarding the uninformative dimensions x_3, x_4, (2) modifying the dimensionality of our regressors **W** to $C \times D$ and analogously the corresponding covariance and (3) substituting $F(\mathbf{X})$ for **K** in our derived methodology. Due to our expansion we now have a $2-D$ decision plane that we can plot and a six-dimensional regressor **w** per class. In Fig. 6 we plot the decision boundaries produced from the full Gibbs solution by averaging over the posterior parameters after 100,000 samples and in Fig. 7

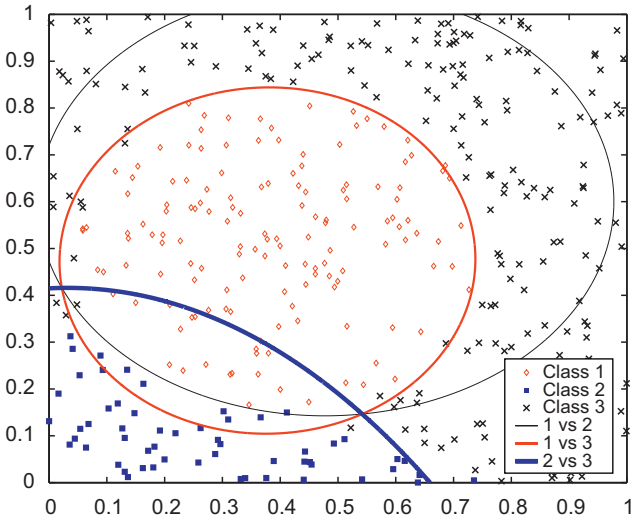

Fig. 7. Decision boundaries from the VB approximation.

Table 1
CPU time (s) comparison for 100,000 Gibbs samples versus 100 VB iterations.

Gibbs	VB
41,720 (s)	120.3 (s)

Notice that the number of VB iterations needed for the lower bound to converge is typically less than 100.

the corresponding decision boundaries from the VB approximation after 100 iterations.

As it can be seen, both the VB approximation and the MCMC solution produce similar decision boundaries leading to good classification performances—2% error for both Gibbs and VB for the above decision boundaries—depending on training size. However, the Gibbs sampler produces tighter boundaries due to the Markov chain exploring the parameter posterior space more efficiently than the VB approximation.

The corresponding CPU times are given in Table 1.

4.2. Multinomial UCI datasets

To further assess the VB approximation of the proposed multinomial probit classifier, we explore a selection of UCI multinomial datasets. The performances are compared against reported results [26] from well-known methods in the pattern recognition and machine learning literature. We employ an RBF (VB RBF), a second order polynomial (VB P) and a linear kernel (VB L) with the VB approximation and report 10-fold cross-validated (CV) error percentages, in Table 2, and CPU times, in Table 3, for a maximum of 50 VB iterations unless convergence has already occurred. The comparison with the *K*-nn and P*K*-nn is for standard implementations of these methods, see Manocha and Girolami [26] for details.

As we can see the VB approximation to the multinomial probit outperforms in most cases both the *K*-nn and P*K*-nn although not offering statistical significant improvements as the variance of the 10-fold CV errors is quite large in most cases.

4.3. Benchmark datasets

4.3.1. Handwritten numerals classification

In this section we report results demonstrating the efficiency of our kernel combination approach when multiple sources of

Table 2
Ten-fold cross-validated error percentages (mean ± std.) on standard UCI multinomial datasets.

Dataset	VB RBF	VB L	VB P	*K*-nn	P*K*-nn
Balance	8.8 ± 3.6	12.2 ± 4.2	**7.0 ± 3.3**	11.5 ± 3.0	10.2 ± 3.0
Crabs	23.5 ± 11.3	**13.5 ± 8.2**	21.5 ± 9.1	15.0 ± 8.8	19.5 ± 6.8
Glass	27.9 ± 10.1	35.8 ± 11.8	28.4 ± 8.9	29.9 ± 9.2	**26.7 ± 8.8**
Iris	**2.7 ± 5.6**	11.3 ± 9.9	4.7 ± 6.3	5.3 ± 5.2	4.0 ± 5.6
Soybean	6.5 ± 10.5	6 ± 9.7	**4 ± 8.4**	14.5 ± 16.7	4.5 ± 9.6
Vehicle	**25.6 ± 4.0**	29.6 ± 3.3	26 ± 6.1	36.3 ± 5.2	37.2 ± 4.5
Wine	4.5 ± 5.1	2.8 ± 4.7	**1.1 ± 2.3**	3.9 ± 3.8	3.4 ± 2.9

THe bold letters indicate top mean accuracy.

Table 3
Running times (seconds) for computing 10-fold cross-validation results.

Dataset	Balance	Crabs	Glass	Iris	Soybean	Vehicle	Wine
VB CPU time (s)	2285	270	380	89	19	3420	105

Table 4
Classification percentage error (mean ± std.) of individual classifiers trained on each feature space.

FR	KL	PX	ZM
Full MCMC Gibbs sampling—single FS			
27.3 ± 3.3	11.0 ± 2.3	7.3 ± 2	25.2 ± 3

Table 5
Combinations of the individual classifiers based on four widely used rules.

Prod	Sum	Max	Maj
Full MCMC Gibbs sampling—Comb. classifiers			
5.1 ± 1.7	5.3 ± 2	8.4 ± 2.3	8.45 ± 2.2

Product (Prod), Mean (Sum), Majority voting (Maj) and Maximum (Max).

Table 6
Combination of feature spaces.

Bin	FixSum	WSum	FixProd	WProd
Full MCMC Gibbs sampling—Comb. FS				
5.7 ± 2	5.5 ± 2	5.8 ± 2.1	5.2 ± 1.8	5.9 ± 1.2

Gibbs sampling results for four feature space combination methods: Binary (Bin), Mean with fixed weights (FixSum), Mean with inferred weights (WSum), product with fixed weights (FixProd) and product with inferred weights (WProd).

Table 7
Combination of feature spaces.

Bin	FixSum	WSum	FixProd	WProd
VB approx.—Comb. FS				
5.53 ± 1.7	4.85 ± 1.5	6.1 ± 1.6	5.35 ± 1.4	6.43 ± 1.8

VB approximation.

information are available. Comparisons are made against combination of classifiers and also between different ways to combine the kernels as we have described previously. In order to assess the above, we make use of two datasets that have multiple feature spaces describing the objects to be classified. The first one is a large $N=2000$, multinomial $C=10$ and "multi-featured" $S=4$ UCI dataset, named "Multiple Features". The objects are handwritten numerals, from 0 to 9, and the available features are the Fourier descriptors (FR), the Karhunen–Loéve features (KL), the pixel averages (PX) and the Zernike moments (ZM). Tax et al. [34] have previously reported results on this problem by combining classifiers but have employed a different test set which is not publicly available. Furthermore, we allow the rotation invariance property of the ZM features to cause problems in the distinction between digits 6 and 9. The hope is that the remaining feature spaces can compensate on the discrimination.

In Tables 4–7, we report experimental results over 50 repeated trials where we have randomly selected 20 training and 20 testing objects from each class. For each trial we employ (1) a single classifier on each feature space, (2) the proposed classifier on the composite feature space. This allows us to examine the performance of combinations of classifiers versus combination of feature spaces. It is worth noting that a concatenation of feature spaces for this specific problem has been found to perform as good as ensemble learning methods [8]. We report results from both the full MCMC solution and the VB approximation in order to analyse the possible degradation of the performance for the approximation. In all cases we employ the multinomial probit kernel machine using a Gaussian kernel with fixed parameters.

As we can see from Tables 5 and 4 the two best performing ensemble learning methods outperform all of the individual classifiers trained on separate feature spaces. With a p-value of $1.5e^{-07}$ between the Pixel classifier and the Product rule, it is a statistical significant difference. At the same time, all the kernel combination methods in Table 6 match the best performing classifier combination approaches, with a p-value of 0.91 between the Product classifier combination rule and the product kernel combination method. Finally, from Table 7 it is obvious that the VB approximation performs very well compared with the full Gibbs solution and even when combinatorial weights are inferred, which expands the parameter space, there is no statistical difference, with a p-value of 0.47, between the MCMC and the VB solution.

Furthermore, the variants of our method that employ combinatorial weights offer the advantage of inferring the significance of the contributing sources of information. In Fig. 8, we can see that the pixel (PX) and Zernike moments (ZM) feature spaces receive large weights and hence contribute significantly in the composite feature space. This is in accordance with our expectations, as the pixel feature space seems to be the best performing individual classifier and complementary predictive power mainly from the ZM channel is improving on that.

The results indicate that there is no benefit in the classification error performance when weighted combinations of feature spaces are employed. This is in agreement with past work by Lewis et al. [23] and Girolami and Zhong [16]. The clear benefit, however, remains the ability to infer the relative significance of the sources and hence gain a better understanding of the problem.

4.3.2. Protein fold classification

The second multi-feature dataset we examine was first described by Ding and Dubchak [13] and is a protein fold classification problem with $C=27$ classes and $S=6$ available feature spaces (approx.

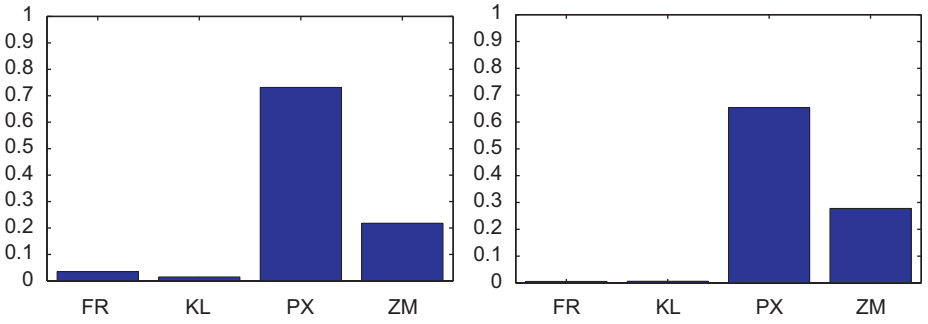

Fig. 8. Typical combinatorial weights from Mean (left) and Product (right) composite kernel.

Table 8
Classification error percentage on the individual feature spaces.

Comp	Hydro	Pol	Polz	Str	Vol
VB approximation—Single FS					
49.8 ± 0.6	63.4 ± 0.6	65.2 ± 0.4	66.3 ± 0.4	60.0 ± 0.25	65.6 ± 0.3

(1) Amino Acid composition (Comp); (2) Hydrophobicity profile (HP); (3) Polarity (Pol); (4) Polarizability (Polz); (5) Secondary structure (Str); and (6) van der Waals volume profile (Vol).

The bold letters indicate statistical significant improvements.

Table 9
Classification error percentages by combining the predictive distributions of the individual classifiers.

Prod	Sum	Max	Maj
VB approximation—Comb. classifiers			
49.8 ± 0.6	**47.6 ± 0.4**	53.7 ± 1.4	54.1 ± 0.5

The bold letters indicate statistical significant improvements.

Table 10
Classification error percentages by feature space combination.

Bin	FixSum	WSum	FixProd	WProd
VB approximation—Comb. FS				
40.7 ± 1.2	**40.1 ± 0.3**	44.4 ± 0.6	43.8 ± 0.14	43.2 ± 1.1

The bold letters indicate statistical significant improvements.

Table 11
Typical running times (seconds).

Method	Classifier combination	FS combination
CPU time (s)	11,519	1256

Ten-fold reduction in CPU time by combining feature spaces.

20-D each). The dataset is available online[7] and it is divided into a train set of $N=313$ size and an independent test set with $N_{test}=385$. We use the original dataset and do not consider the feature modifications (extra features, modification of existing ones and omission of four objects) suggested recently in the work by Shen and Chou [32] in order to increase prediction accuracy. In Table 8 we report the performances of the individual classifiers trained on each feature space, in Table 9 the classifier combination performances and in Table 10 the kernel combination approaches. Furthermore, in Table 11 we give the corresponding CPU time requirements between classifier and kernel combination methods for comparison. It is worth noting that a possible concatenation of all features into a single representation has been examined by Ding and Dubchak [13] and was found not to be performing competitively.

The experiments are repeated over five randomly initialized trials for a maximum of 100 VB iterations by monitoring convergence with a 0.1% increase threshold on the lower bound progression. We employ second order polynomial kernels as they were found to give the best performances across all methods.

As it can be observed the best of the classifier combination methods outperform any individual classifier trained on a single feature space with a p-value between the Sum combination rule and the composition classifier of $2.3e-02$. However, all the kernel combination methods perform better than the best classifier combination with a p-value of $1.5e-08$ between the latter (FixSum) and the best performing classifier combination method (Sum). The corresponding CPU times show a 10-fold reduction by the proposed feature space combination method from the classifier combination approaches.

Furthermore, with a top performance of 38.3 error % for a single binary combination run (Bin) we match the state-of-the-art reported on the original dataset by Girolami and Zhong [16] employing Gaussian process priors. The best reported performance by Ding and Dubchak [13] was a 56% accuracy (44%) error employing an all-vs-all method with $S\times C\times(C-1)/2=8240$ binary SVM classifiers. We offer instead a single multiclass kernel machine able to combine the feature spaces and achieve a 60–62% accuracy in reduced computational time. It is worth noting that recent work by the authors [7] has achieved a 70% accuracy by considering additional state-of-the-art string kernels.

Finally, in Fig. 9 the combinatorial weights can be seen for the case of the mean and the product[8] composite kernel. The results are in agreement with the past work identifying the significance of the amino acid composition, hydrophobicity profile and secondary structure feature spaces for the classification task.

5. Discussion

A multinomial probit composite kernel classifier, based on a well-founded hierarchical Bayesian framework and able to instructively combine feature spaces has been presented in this work. The full MCMC MH within Gibbs sampling approach allows for exact inference to be performed and the proposed variational Bayes approximation reduces significantly computational requirements while retaining the same levels of performance. The proposed methodology enables inference to be performed in three significant levels by

[7] http://crd.lbl.gov/~cding/protein/

[8] Notice that for the product composite kernel the combinatorial weights are not restricted in the simplex.

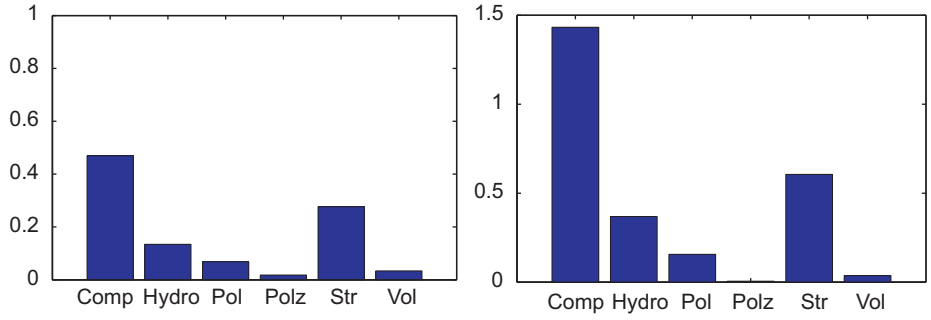

Fig. 9. Typical combinatorial weights from Mean (left) and Product (right) composite kernel for the protein fold dataset.

learning the combinatorial weights associated to each feature space, the kernel parameters associated to each dimension of the feature spaces and finally the parameters/regressors associated with the elements of the composite feature space.

An explicit comparison between the Gibbs sampling and the VB approximation has been presented. The approximate solution was found not to worsen significantly the classification performance and the resulting decision boundaries were similar to the MCMC approach, though the latter is producing tighter descriptions of the classes. The VB posteriors over the parameters were observed to be narrow as expected and all the probability mass was concentrated on a small area, however, within the posterior distributions produced by the Gibbs solution.

The proposed feature space combination approach was found to be as good as, or outperforming the best classifier combination rule examined and achieving the state-of-the-art while at the same time offering a 10-fold reduction in computational time and a better scaling when multiple spaces are available. At the same time, contrary to the classifier combination approach, inference can be performed on the combinatorial weights of the composite feature space, enabling a better understanding of the significance of the sources. In comparison with previously employed SVM combinations for the multiclass problem of protein fold prediction, our method offers a significant improvement of performance while employing a single classifier instead of thousands, reducing the computational time from approximately 12 to 0.3 CPU hours.

Finally, with respect to the two main problems considered, we can conclude that there is greater improvement on zero–one loss when diverse sources of information are used. This is a well known observation in ensemble learning methodology and kernel combination methods follow the same phenomenon. Theoretical justification for this can be easily provided via a simple bias-variance–covariance decomposition of the loss which follows by the same analysis as the one used for an ensemble of regressors. We are currently investigating a theoretical justification for another well known phenomenon in multiple kernel learning, that of zero–one loss equivalence between the average and the weighted kernel combination rules on specific problems. Recently we have offered in Damoulas et al. [9] a further deterministic approximation based on an EM update scheme which induces sparsity and in fact generalizes the relevance vector machine to the multi-class and multi-kernel setting.

Acknowledgments

This work is sponsored by NCR Financial Solutions Group Ltd. The authors would particularly like to acknowledge the help and support from Dr. Gary Ross and Dr. Chao He of NCR Labs. The second author is supported by an EPSRC Advanced Research Fellowship (EP/E052029/1).

Appendix A. Gibbs sampler

The conditional distribution for the auxiliary variables $\mathbf{Y}$ is a product of N C-dimensional conically truncated Gaussians given by

$$\prod_{n=1}^{N} \delta(y_{in} > y_{jn} \forall j \neq i)\delta(t_n = i)\mathcal{N}_{\mathbf{y}_n}(\mathbf{W}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}, \mathbf{I})$$

and for the regressors $\mathbf{W}$ is a product of Gaussian distributions $\prod_{c=1}^{C}\mathcal{N}_{\mathbf{w}_c}(\mathbf{m}_c, \mathbf{V}_c)$ where

$$\mathbf{m}_c = \mathbf{y}_c\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{V}_c \text{ (row vector)} \quad \text{and} \quad \mathbf{V}_c = (\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}} + \mathbf{Z}_c^{-1})^{-1}$$

hence by iteratively sampling from these distributions will lead the Markov chain to sample from the desired posterior distribution. The typical MH subsamplers employed, in the case of the *mean composite* kernel model have acceptance ratios

$$A(\boldsymbol{\beta}^i, \boldsymbol{\beta}^*) = \min\left(1, \frac{\prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{N}_{\mathbf{y}_{cn}}(\mathbf{w}_c\mathbf{k}_n^{\boldsymbol{\beta}^*\boldsymbol{\Theta}}, 1)\prod_{s=1}^{S}\beta_s^{*\rho_s - 1}}{\prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{N}_{\mathbf{y}_{cn}}(\mathbf{w}_c\mathbf{k}_n^{\boldsymbol{\beta}^i\boldsymbol{\Theta}}, 1)\prod_{s=1}^{S}\beta_s^{i\,\rho_s - 1}}\right)$$

$$A(\boldsymbol{\rho}^i, \boldsymbol{\rho}^*) = \min\left(1, \frac{\Phi(\boldsymbol{\rho}^*)\prod_{s=1}^{S}\beta_s^{\rho_s^* - 1}\prod_{s=1}^{S}\rho_s^{*\mu - 1}e^{-\lambda\rho_s^*}}{\Phi(\boldsymbol{\rho}^i)\prod_{s=1}^{S}\beta_s^{\rho_s^i - 1}\prod_{s=1}^{S}\rho_s^{i\,\mu - 1}e^{-\lambda\rho_s^i}}\right)$$

$$\text{where} \quad \Phi(\boldsymbol{\rho}) = \frac{\Gamma(\sum_{s=1}^{S}\rho_s)}{\prod_{s=1}^{S}\Gamma(\rho_s)}$$

with the proposed move symbolized by $*$ and the current state with i.

For the *binary composite* kernel, an extra Gibbs step is introduced in our model as $p(\beta_i = 0|\boldsymbol{\beta}_{-i}, \mathbf{Y}, \mathbf{K}^{s\theta_s}\ \forall s \in \{1, \ldots, S\})$ which depends on the marginal likelihood given by $p(\mathbf{Y}|\boldsymbol{\beta}, \mathbf{K}^{s\theta_s}\ \forall s \in \{1, \ldots, S\}) = \prod_{c=1}^{C}(2\pi)^{-N/2}|\boldsymbol{\Omega}_c|^{-1/2}\exp\{-\frac{1}{2}\mathbf{y}_c\boldsymbol{\Omega}_c\mathbf{y}_c^T\}$ with $\boldsymbol{\Omega}_c = \mathbf{I} + \mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{Z}_c^{-1}\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}$.

The *product composite* kernel employs two MH subsamplers to sample $\boldsymbol{\beta}$ (from a gamma distribution this time) and the hyper-parameters $\boldsymbol{\pi}, \boldsymbol{\chi}$. The kernel parameters $\boldsymbol{\Theta}$ are inferred in all cases via an extra MH subsampler with acceptance ratio

$$A(\boldsymbol{\Theta}^i, \boldsymbol{\Theta}^*) = \min\left(1, \frac{\prod_{s=1}^{S}\prod_{d=1}^{D}\theta_{sd}^{*}\prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{N}_{\mathbf{y}_{cn}}(\mathbf{w}_c\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}^*}, 1)\theta_{sd}^{*\,\omega - 1}e^{-\phi\theta_{sd}^*}}{\prod_{s=1}^{S}\prod_{d=1}^{D}\theta_{sd}^{i}\prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{N}_{\mathbf{y}_{cn}}(\mathbf{w}_c\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}^i}, 1)\theta_{sd}^{i\,\omega - 1}e^{-\phi\theta_{sd}^i}}\right)$$

Finally, the Monte Carlo estimate of the predictive distribution is used to assign the class probabilities according to a number of samples L drawn from the predictive distribution which, considering

Eq. (4) and substituting for the test composite kernel $\mathbf{k}_n^{*\boldsymbol{\beta}\boldsymbol{\Theta}}$ defines the estimate of the predictive distribution as

$$P(t_*=c|\mathbf{x}_*)=\frac{1}{L}\sum_{l=1}^{L}\mathscr{E}_{p(u)}\left\{\prod_{j\neq c}\Phi(u+(\mathbf{w}_c^l-\mathbf{w}_j^l)\mathbf{k}^{*\beta^l\boldsymbol{\Theta}^l})\right\}$$

Appendix B. Approximate posterior distributions

We give here the full derivations of the approximate posteriors for the model under the VB approximation. We employ a first order approximation for the kernel parameters $\boldsymbol{\Theta}$, i.e. $\mathscr{E}_{Q(\boldsymbol{\theta})}\{\mathbf{K}^{i\boldsymbol{\theta}_i}\mathbf{K}^{j\boldsymbol{\theta}_j}\}\approx\mathbf{K}^{i\widetilde{\boldsymbol{\theta}}_i}\mathbf{K}^{j\widetilde{\boldsymbol{\theta}}_j}$, to avoid nonlinear contributions to the expectation. The same approximation is applied to the *product composite* kernel case for the combinatorial parameters $\boldsymbol{\beta}$ where $\mathscr{E}_{Q(\boldsymbol{\beta})}\{\mathbf{K}^{i\beta_i}\mathbf{K}^{j\beta_j}\}\approx\mathbf{K}^{i\widetilde{\beta}_i}\mathbf{K}^{j\widetilde{\beta}_j}$

B.1. $Q(\mathbf{Y})$

$$\begin{aligned}Q(\mathbf{Y}) &\propto \exp\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\{\log J\cdot L\}\}\\ &\propto \exp\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\{\log p(\mathbf{t}|\mathbf{Y})+\log p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta})\}\}\\ &\propto \prod_{n=1}^{N}\delta(y_{i,n}>y_{k,n}\ \forall k\neq i)\delta(t_n=i)\exp\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\log p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta})\}\end{aligned}$$

where the exponential term can be analysed as follows:

$$\begin{aligned}&\exp\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\log p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta})\}\\ &=\exp\left\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\log\prod_{n=1}^{N}\mathcal{N}_{\mathbf{y}_n}(\mathbf{W}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}},\mathbf{I})\right\}\\ &=\exp\left\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\sum_{n=1}^{N}\log\frac{1}{(2\pi)^{C/2}|\mathbf{I}|^{1/2}}\right.\right.\\ &\quad\left.\left.\times\exp\left(-\frac{1}{2}(\mathbf{y}_n-\mathbf{W}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}})^T(\mathbf{y}_n-\mathbf{W}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}})\right)\right\}\right\}\\ &=\exp\left\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\sum_{n=1}^{N}\left(-\frac{1}{2}(\mathbf{y}_n^T\mathbf{y}_n-2\mathbf{y}_n^T\mathbf{W}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}\right.\right.\right.\\ &\quad\left.\left.\left.+(\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}})^T\mathbf{W}^T\mathbf{W}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}})\right)\right\}\right\}\\ &=\exp\left\{\sum_{n=1}^{N}-\frac{1}{2}\left(\mathbf{y}_n^T\mathbf{y}_n-2\mathbf{y}_n^T\widetilde{\mathbf{W}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right.\right.\\ &\quad\left.\left.+\sum_{i=1}^{S}\sum_{j=1}^{S}\widetilde{\boldsymbol{\beta}_i}\widetilde{\boldsymbol{\beta}_j}(\mathbf{k}_n^{i\widetilde{\boldsymbol{\theta}}_i})^T\widetilde{\mathbf{W}^T\mathbf{W}}\mathbf{k}_n^{j\widetilde{\boldsymbol{\theta}}_j}\right)\right\}\end{aligned}$$

where $\mathbf{k}_n^{i\widetilde{\boldsymbol{\theta}}_i}$ is the nth N-dimensional column vector of the ith base kernel with kernel parameters $\widetilde{\boldsymbol{\theta}}_i$. Now from this exponential term we can form the posterior distribution as a Gaussian and reach to the final expression:

$$Q(\mathbf{Y})\propto\prod_{n=1}^{N}\delta(y_{i,n}>y_{k,n}\ \forall k\neq i)\delta(t_n=i)\mathcal{N}_{\mathbf{y}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}},\mathbf{I}) \tag{B.1}$$

which is a C-dimensional conically truncated Gaussian.

B.2. $Q(\mathbf{W})$

$$\begin{aligned}Q(\mathbf{W}) &\propto \exp\{E_{Q(\mathbf{y})Q(\boldsymbol{\beta})Q(\zeta)Q(\boldsymbol{\Theta})}\{\log p(\mathbf{y}|\mathbf{W},\boldsymbol{\beta})+\log p(\mathbf{W}|\boldsymbol{\zeta})\}\}\\ &=\exp\left\{E_{Q(\mathbf{y})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\log\prod_{c=1}^{C}\mathcal{N}_{\mathbf{y}_c}(\mathbf{w}_c\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}},\mathbf{I})\right\}\right.\\ &\quad\left.+E_{Q(\zeta)}\left\{\log\prod_{c=1}^{C}\mathcal{N}_{\mathbf{w}_c}(\mathbf{0},\mathbf{Z}_c)\right\}\right\}\\ &=\exp\left\{E_{Q(\mathbf{y})Q(\boldsymbol{\beta}Q(\boldsymbol{\Theta}))}\left\{\sum_{c=1}^{C}\log\frac{1}{(2\pi)^{N/2}|\mathbf{I}|^{1/2}}\right.\right.\\ &\quad\left.\times\exp\left\{-\frac{1}{2}(\mathbf{y}_c-\mathbf{w}_c\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}})(\mathbf{y}_c-\mathbf{w}_c\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}})^T\right\}\right\}\\ &\quad\left.+E_{Q(\zeta)}\left\{\sum_{c=1}^{C}\log\frac{1}{(2\pi)^{N/2}|\mathbf{Z}_c|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{w}_c(\mathbf{Z}_c)^{-1}\mathbf{w}_c^T)\right\}\right\}\right\}\\ &=\exp\left\{E_{Q(\mathbf{y})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\sum_{c=1}^{C}-\frac{1}{2}(\mathbf{y}_c\mathbf{y}_c^T\right.\right.\\ &\quad\left.-2\mathbf{y}_c\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{w}_c^T+\mathbf{w}_c\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{w}_c^T)\right\}\\ &\quad\left.+E_{Q(\zeta)}\left\{\sum_{c=1}^{C}-\frac{1}{2}\log\prod_{n=1}^{N}\zeta_{cn}-\frac{1}{2}\mathbf{w}_c(\mathbf{Z}_c)^{-1}\mathbf{w}_c^T\right\}\right\}\\ &=\exp\left\{\sum_{c=1}^{C}-\frac{1}{2}\left\{\widetilde{\mathbf{y}_c\mathbf{y}_c^T}-2\widetilde{\mathbf{y}_c}\mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\mathbf{w}_c^T\right.\right.\\ &\quad\left.\left.+\mathbf{w}_c\sum_{i=1}^{S}\sum_{j=1}^{S}\widetilde{\beta_i}\widetilde{\beta_j}\mathbf{K}^{i\widetilde{\boldsymbol{\theta}}_i}\mathbf{K}^{j\widetilde{\boldsymbol{\theta}}_j}\mathbf{w}_c^T+\sum_{n=1}^{N}\widetilde{\log\zeta_{cn}}+\mathbf{w}_c(\widetilde{\mathbf{Z}}_c)^{-1}\mathbf{w}_c^T\right\}\right\}\end{aligned}$$

Again we can form the posterior expectation as a new Gaussian:

$$Q(\mathbf{W})\propto\prod_{c=1}^{C}\mathcal{N}_{\mathbf{w}_c}(\widetilde{\mathbf{y}_c}\mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\mathbf{V}_c,\mathbf{V}_c) \tag{B.2}$$

where $\mathbf{V}_c$ is the covariance matrix defined as

$$\mathbf{V}_c=\left(\sum_{i=1}^{S}\sum_{j=1}^{S}\widetilde{\beta_i}\widetilde{\beta_j}\mathbf{K}^{i\widetilde{\boldsymbol{\theta}}_i}\mathbf{K}^{j\widetilde{\boldsymbol{\theta}}_j}+(\widetilde{\mathbf{Z}}_c)^{-1}\right)^{-1} \tag{B.3}$$

and $\widetilde{\mathbf{Z}}_c$ is a diagonal matrix of the expected variances $\widetilde{\zeta_{ci}},\ldots,\widetilde{\zeta_{cN}}$ for each class.

B.3. $Q(\mathbf{Z})$

$$\begin{aligned}Q(\mathbf{Z}) &\propto \exp\{E_{Q(\mathbf{W})}(\log p(\mathbf{W}|\mathbf{Z})+\log p(\mathbf{Z}|\tau,\upsilon))\}\\ &= \exp\{E_{Q(\mathbf{W})}(\log p(\mathbf{W}|\mathbf{Z}))\}p(\mathbf{Z}|\tau,\upsilon)\end{aligned}$$

Analysing the exponential term only:

$$\begin{aligned}&\exp\left\{E_{Q(\mathbf{W})}\left(\log\prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{N}_{w_{cn}}(0,\zeta_{cn})\right)\right\}\\ &=\exp\left\{E_{Q(\mathbf{W})}\left(\sum_{c=1}^{C}\sum_{n=1}^{N}-\frac{1}{2}\log\zeta_{cn}-\frac{1}{2}w_{cn}^2\zeta_{cn}^{-1}\right)\right\}\\ &=\prod_{c=1}^{C}\prod_{n=1}^{N}\zeta_{cn}^{-1/2}\exp\left(-\frac{1}{2}\frac{\widetilde{w_{cn}^2}}{\zeta_{cn}}\right)\end{aligned}$$

which combined with the $p(\mathbf{Z}|\tau,\upsilon)$ prior Gamma distribution leads to our expected posterior distribution:

$$Q(\mathbf{Z}) \propto \prod_{c=1}^{C}\prod_{n=1}^{N} \zeta_{cn}^{-(\tau+1/2)+1} \frac{\upsilon^{\tau} e^{-\zeta_{cn}^{-1}(\upsilon+(1/2)\widetilde{w}_{cn}^2)}}{\Gamma(\tau)} = \prod_{c=1}^{C}\prod_{n=1}^{N} Gamma\left(\zeta_{cn}^{-1}|\tau+\frac{1}{2}, \upsilon+\frac{1}{2}\widetilde{w}_{cn}^2\right)$$

B.4. $Q(\boldsymbol{\beta}), Q(\boldsymbol{\rho}), Q(\boldsymbol{\Theta}), Q(\boldsymbol{\pi}), Q(\boldsymbol{\chi})$

Importance sampling techniques [2] are used to approximate these posterior distributions as they are intractable. We present the case of the *mean composite* kernel analytically and leave the cases of the *product* and *binary composite* kernel as a straightforward modifications.

For $Q(\boldsymbol{\rho})$ we have $p(\boldsymbol{\rho}|\boldsymbol{\beta}) \propto p(\boldsymbol{\beta}|\boldsymbol{\rho})p(\boldsymbol{\rho}|\mu,\lambda)$

The unnormalized posterior is $Q^*(\boldsymbol{\rho}) = p(\boldsymbol{\beta}|\boldsymbol{\rho})p(\boldsymbol{\rho}|\mu,\lambda)$ and hence the importance weights are

$$\mathscr{W}(\boldsymbol{\rho}_i) = \frac{\frac{Q^*(\boldsymbol{\rho}^{i'})}{\prod_{s=1}^{S} Gamma_{\rho_s^{i'}}(\mu,\lambda)}}{\sum_{i=1}^{I} \frac{Q^*(\boldsymbol{\rho}^{i})}{\prod_{s=1}^{S} Gamma_{\rho_s^{i}}(\mu,\lambda)}} = \frac{Dir(\boldsymbol{\rho}^{i'})}{\sum_{i=1}^{I} Dir(\boldsymbol{\rho}^{i})}$$

where *Gamma* and *Dir* are the Gamma and Dirichlet distributions, while I is the total number of samples of $\boldsymbol{\rho}$ taken until now from the product gamma distributions and i' denotes the current (last) sample. So now we can estimate any function f of $\boldsymbol{\rho}$ based on

$$\widetilde{f}(\boldsymbol{\rho}) = \sum_{i=1}^{I} f(\boldsymbol{\rho})\mathscr{W}(\boldsymbol{\rho}^i)$$

In the same manner as above but now for $Q(\boldsymbol{\beta})$ and $Q(\boldsymbol{\Theta})$ we can use the unnormalized posteriors $Q^*(\boldsymbol{\beta})$ and $Q^*(\boldsymbol{\Theta})$, where $p(\boldsymbol{\beta}|\boldsymbol{\rho},\mathbf{Y},\mathbf{W}) \propto p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\rho})$ and $p(\boldsymbol{\Theta}|\omega,\phi,\mathbf{Y},\mathbf{W}) \propto p(\mathbf{Y}|\mathbf{W},\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\omega,\phi)$ with importance weights defined as

$$\mathscr{W}(\boldsymbol{\beta}^i) = \frac{\frac{Q^*(\boldsymbol{\beta}^{i'})}{Dir(\boldsymbol{\beta}^{i'})}}{\sum_{i=1}^{I} \frac{Q^*(\boldsymbol{\beta}^{i})}{Dir(\boldsymbol{\beta}^{i})}} = \frac{\prod_{n=1}^{N} \mathscr{N}_{\widetilde{\mathbf{y}}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\boldsymbol{\beta}^{i'}}, \mathbf{I})}{\sum_{i=1}^{I}\prod_{n=1}^{N} \mathscr{N}_{\widetilde{\mathbf{y}}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\boldsymbol{\beta}^{i}}, \mathbf{I})}$$

and

$$\mathscr{W}(\boldsymbol{\Theta}^i) = \frac{\frac{Q^*(\boldsymbol{\Theta}^{i'})}{Dir(\boldsymbol{\Theta}^{i'})}}{\sum_{i=1}^{I} \frac{Q^*(\boldsymbol{\Theta}^{i})}{Dir(\boldsymbol{\Theta}^{i})}} = \frac{\prod_{n=1}^{N} \mathscr{N}_{\widetilde{\mathbf{y}}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\boldsymbol{\Theta}^{i'}}, \mathbf{I})}{\sum_{i=1}^{I}\prod_{n=1}^{N} \mathscr{N}_{\widetilde{\mathbf{y}}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\boldsymbol{\Theta}^{i}}, \mathbf{I})}$$

and again we can estimate any function g of $\boldsymbol{\beta}$ and h of $\boldsymbol{\Theta}$ as

$$\widetilde{g}(\boldsymbol{\beta}) = \sum_{i=1}^{I} g(\boldsymbol{\beta})\mathscr{W}(\boldsymbol{\beta}^i) \quad \text{and} \quad \widetilde{h}(\boldsymbol{\Theta}) = \sum_{i=1}^{I} g(\boldsymbol{\Theta})\mathscr{W}(\boldsymbol{\Theta}^i)$$

For the *product composite* kernel case we proceed in the same manner by importance sampling for $Q(\boldsymbol{\pi})$ and $Q(\boldsymbol{\chi})$.

Appendix C. Posterior expectations of each $\mathbf{y}_n$

As we have shown $Q(\mathbf{Y}) \propto \prod_{n=1}^{N}\delta(y_{i,n} > y_{k,n}\ \forall k \neq i)\delta(t_n = i)\mathscr{N}_{\mathbf{y}_n}(\widetilde{\mathbf{W}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, \mathbf{I})$. Hence $Q(\mathbf{y}_n)$ is a truncated multivariate Gaussian distribution and we need to calculate the correction to the normalizing term $\mathscr{Z}_n$ caused by the truncation. Thus, the posterior expectation can be expressed as

$$Q(\mathbf{y}_n) = \mathscr{Z}_n^{-1} \prod_{c=1}^{C} \mathscr{N}_{y_{cn}}^{t_n}(\widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1)$$

where the superscript t_n indicates the truncation needed so that the appropriate dimension i (since $t_n = i \iff y_{in} > y_{jn}\ \forall j \neq i$) is the largest.

Now, $\mathscr{Z}_n = P(\mathbf{y}_n \in \mathscr{C})$ where $\mathscr{C} = \{\mathbf{y}_n : y_{in} > y_{jn}\}$ hence

$$\mathscr{Z}_n = \int_{-\infty}^{+\infty} \mathscr{N}_{y_{in}}(\widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1)\prod_{j\neq i}\int_{-\infty}^{y_{in}} \mathscr{N}_{y_{jn}}(\widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1)\, dy_{jn}\, dy_{in} = \mathscr{E}_{p(u)}\left\{\prod_{j\neq i}\Phi(u + \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\right\}$$

with $p(u) = \mathscr{N}_u(0,1)$. The posterior expectation of y_{cn} for all $c \neq i$ (the auxiliary variables associated with the rest of the classes except the one that object n belongs to) is given by

$$\begin{aligned}\widetilde{y}_{cn} &= \mathscr{Z}_n^{-1}\int_{-\infty}^{+\infty} y_{cn}\prod_{j=1}^{C}\mathscr{N}_{y_{jn}}(\widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\, dy_{jn}\\ &= \mathscr{Z}_n^{-1}\int_{-\infty}^{+\infty}\int_{-\infty}^{y_{in}} y_{cn}\mathscr{N}_{y_{cn}}(\widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\\ &\quad \prod_{j\neq i,c}\mathscr{N}_{y_{in}}(\widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1)\Phi(y_{in} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\, dy_{cn}\, dy_{in}\\ &= \widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \mathscr{Z}_n^{-1}\mathscr{E}_{p(u)}\left\{\mathscr{N}_u(\widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1)\right.\\ &\quad \left.\prod_{j\neq i,c}\Phi(u + \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\right\}\end{aligned}$$

For the ith class the posterior expectation y_{in} (the auxiliary variable associated with the known class of the nth object) is given by

$$\begin{aligned}\widetilde{y}_{in} &= \mathscr{Z}_n^{-1}\int_{-\infty}^{+\infty} y_{in}\mathscr{N}_{y_{in}}(\widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1)\prod_{j\neq i}\Phi(y_{in} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\, dy_{in}\\ &= \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} + \mathscr{Z}_n^{-1}\mathscr{E}_{p(u)}\left\{u\prod_{j\neq i}\Phi(u + \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}})\right\}\\ &= \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} + \sum_{c\neq i}(\widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{y}_{cn})\end{aligned}$$

where we have made use of the fact that for a variable $u\mathscr{N}(0,1)$ and any differentiable function $g(u)$, $\mathscr{E}\{ug(u)\} = \mathscr{E}\{g'(u)\}$.

Appendix D. Predictive distribution

In order to make a prediction t^* for a new point $\mathbf{x}^*$ we need to know:

$$p(t^* = c|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = \int p(t^* = c|\mathbf{y}^*)p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t})\, d\mathbf{y}^* = \int \delta_c^* p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t})\, d\mathbf{y}^*$$

Hence we need to evaluate

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = \int p(\mathbf{y}^*|\mathbf{W},\mathbf{x}^*)p(\mathbf{W}|\mathbf{X},\mathbf{t})\, d\mathbf{W} = \prod_{c=1}^{C}\int \mathscr{N}_{\mathbf{w}_c\mathbf{K}^*}(\mathbf{y}_c^*, \mathbf{I})\mathscr{N}_{\mathbf{w}_c}(\widetilde{\mathbf{y}}_c\mathbf{K}\mathbf{V}_c, \mathbf{V}_c)\, d\mathbf{w}_c$$

We proceed by analysing the integral, gathering all the terms depending on $\mathbf{w}_c$, completing the square twice and reforming to

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = \prod_{c=1}^{C}\int \mathcal{N}_{\mathbf{y}_c^*}(\widetilde{\mathbf{y}}_c\mathbf{K}\mathbf{\Lambda}_c\mathbf{K}^*\widetilde{\mathscr{V}}_c^*, \widetilde{\mathscr{V}}_c^*) \times \mathcal{N}_{\mathbf{w}_c}((\mathbf{y}_c^*\mathbf{K}^{*T}+\widetilde{\mathbf{y}}_c\mathbf{K})\mathbf{\Lambda}_c, \mathbf{\Lambda}_c) \quad \text{(D.1)}$$

with

$$\widetilde{\mathscr{V}}_c^* = (\mathbf{I}-\mathbf{K}^{*T}\mathbf{\Theta}_c\mathbf{K}^*)^{-1} \quad (Ntest\times Ntest)$$

and

$$\mathbf{\Lambda}_c = (\mathbf{K}^*\mathbf{K}^{*T}+\mathbf{V}_c^{-1})^{-1} \quad (N\times N) \quad \text{(D.2)}$$

Finally we can simplify $\widetilde{\mathscr{V}}_c^*$ by applying the Woodbury identity and reduce its form to

$$\widetilde{\mathscr{V}}_c^* = (\mathbf{I}+\mathbf{K}^{*T}\mathbf{V}_c\mathbf{K}^*) \quad (Ntest\times Ntest)$$

Now the Gaussian distribution wrt $\mathbf{w}_c$ integrates to one and we are left with

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = \prod_{c=1}^{C}\mathcal{N}_{\mathbf{y}_c^*}(\widetilde{\mathbf{m}}_c^*, \widetilde{\mathscr{V}}_c^*) \quad \text{(D.3)}$$

where $\widetilde{\mathbf{m}}_c^* = \widetilde{\mathbf{y}}_c\mathbf{K}\mathbf{\Lambda}_c\mathbf{K}^*\widetilde{\mathscr{V}}_c^*$ $(1\times Ntest)$

Hence we can go back to the predictive distribution and consider the case of a single test point (i.e. $Ntest=1$) with associated scalars $\widetilde{m}_c^*$ and $\widetilde{v}_c^*$

$$\begin{aligned} p(t^*=c|\mathbf{x}^*,\mathbf{X},\mathbf{t}) &= \int \delta_c^*\prod_{c=1}^{C}\mathcal{N}_{y_c^*}(\widetilde{m}_c^*,\widetilde{v}_c^*)\,dy_c^* \\ &= \int_{-\infty}^{+\infty}\mathcal{N}_{y_c^*}(\widetilde{m}_c^*,\widetilde{v}_c^*)\prod_{j\neq c}\int_{-\infty}^{y_c^*}\mathcal{N}_{y_j^*}(\widetilde{m}_j^*,\widetilde{v}_j^*)\,dy_j^*\,dy_c^* \\ &= \int_{-\infty}^{+\infty}\mathcal{N}_{y_c^*-\widetilde{m}_c^*}(0,\widetilde{v}_c^*)\prod_{j\neq c}\int_{-\infty}^{y_c^*-\widetilde{m}_j^*}\mathcal{N}_{y_j^*-\widetilde{m}_j^*}(0,\widetilde{v}_j^*)\,dy_j^*\,dy_c^* \end{aligned}$$

Setting $u=(y_c^*-\widetilde{m}_c^*)\widetilde{v}_c^{*-1}$ and $x=(y_c^*-\widetilde{m}_j^*)\widetilde{v}_j^{*-1}$ we have

$$\begin{aligned} p(t^*=c|\mathbf{x}^*,\mathbf{X},\mathbf{t}) &= \int_{-\infty}^{+\infty}\mathcal{N}_u(0,1)\prod_{j\neq c}\int_{-\infty}^{(u\widetilde{v}_c^*+\widetilde{m}_c^*-\widetilde{m}_j^*)\widetilde{v}_j^{*-1}}\mathcal{N}_x(0,1)\,dx\,du \\ &= E_{p(u)}\left\{\prod_{j\neq c}\Phi\left[\frac{1}{\widetilde{v}_j^*}(u\widetilde{v}_c^*+\widetilde{m}_c^*-\widetilde{m}_j^*)\right]\right\} \end{aligned}$$

Appendix E. Lower bound

From Jensen's inequality in Eq. (5) and by conditioning on current values of $\boldsymbol{\beta},\mathbf{\Theta},\mathbf{Z},\boldsymbol{\rho}$ we can derive the variational lower bound using the relevant components

$$\begin{aligned} &\mathscr{E}_{Q(\mathbf{Y})Q(\mathbf{W})}\{\log p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\mathbf{X})\} && \text{(E.1)}\\ &+\mathscr{E}_{Q(\mathbf{Y})Q(\mathbf{W})}\{\log p(\mathbf{W}|\mathbf{Z},\mathbf{X})\} && \text{(E.2)}\\ &-\mathscr{E}_{Q(\mathbf{Y})}\{\log Q(\mathbf{Y})\} && \text{(E.3)}\\ &-\mathscr{E}_{Q(\mathbf{W})}\{\log Q(\mathbf{W})\} && \text{(E.4)} \end{aligned}$$

which, by noting that the expectation of a quadratic form under a Gaussian is another quadratic form plus a constant, leads to the following expression for the lower bound:

$$\begin{aligned} &-\frac{NC}{2}\log 2\pi - \frac{1}{2}\sum_{c=1}^{C}\sum_{n=1}^{N}\{\widetilde{y_{cn}^2}+\mathbf{k}_n^T\widetilde{\mathbf{w}_c^T\mathbf{w}_c}\mathbf{k}_n-2\widetilde{y}_{cn}\widetilde{\mathbf{w}}_c\mathbf{k}_n\}\\ &-\frac{NC}{2}\log 2\pi - \frac{1}{2}\sum_{c=1}^{C}\log|\mathbf{Z}_c| - \frac{1}{2}\sum_{c=1}^{C}\widetilde{\mathbf{w}}_c\mathbf{Z}_c^{-1}\widetilde{\mathbf{w}}_c^T\\ &-\frac{1}{2}\sum_{c=1}^{C}\mathrm{Tr}[\mathbf{Z}_c^{-1}\mathbf{V}_c] + \sum_{n=1}^{N}\log\mathscr{Z}_n + \frac{NC}{2}\log 2\pi\\ &+\frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{C}(\widetilde{y^2}_{cn}-2\widetilde{y}_{cn}\widetilde{\mathbf{w}}_c\mathbf{k}_n+\mathbf{k}_n^T\widetilde{\mathbf{w}}_c^T\widetilde{\mathbf{w}}_c\mathbf{k}_n)\\ &+\frac{NC}{2}\log 2\pi + \frac{1}{2}\sum_{c=1}^{C}\log|\mathbf{V}_c| + \frac{NC}{2} \end{aligned}$$

which simplifies to our final expression

$$\begin{aligned} \text{Lower Bound} = &\frac{NC}{2} + \frac{1}{2}\sum_{c=1}^{C}\log|\mathbf{V}_c| + \sum_{n=1}^{N}\log\mathscr{Z}_n && \text{(E.5)}\\ &-\frac{1}{2}\sum_{c=1}^{C}\mathrm{Tr}[\mathbf{Z}_c^{-1}\mathbf{V}_c] - \frac{1}{2}\sum_{c=1}^{C}\widetilde{\mathbf{w}}_c\mathbf{Z}_c^{-1}\widetilde{\mathbf{w}}_c^T && \text{(E.6)}\\ &-\frac{1}{2}\sum_{c=1}^{C}\log|\mathbf{Z}_c| - \frac{1}{2}\sum_{c=1}^{C}\sum_{n=1}^{N}\mathbf{k}_n^T\mathbf{V}_c\mathbf{k}_n && \text{(E.7)} \end{aligned}$$

References

[1] J. Albert, S. Chib, Bayesian analysis of binary and polychotomous response data, Journal of the American Statistical Association 88 (1993) 669–679.
[2] C. Andrieu, An introduction to MCMC for machine learning, Machine Learning 50 (2003) 5–43.
[3] L. Bai, L. Shen, Combining wavelets with HMM for face recognition, in: Proceedings of the 23rd Artificial Intelligence Conference, 2003.
[4] M.J. Beal, Variational algorithms for approximate Bayesian inference, Ph.D. Thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.
[5] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer Series in Statistics, Springer, Berlin, 1985.
[6] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, USA, 2006.
[7] T. Damoulas, M.A. Girolami, Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection, Bioinformatics 24 (10) (2008) 1264–1270.
[8] T. Damoulas, M.A. Girolami, Pattern recognition with a Bayesian kernel combination machine, Pattern Recognition Letters 30 (1) (2009) 46–54.
[9] T. Damoulas, Y. Ying, M.A. Girolami, C. Campbel, Inferring sparse kernel combinations and relevant vectors: an application to subcellular localization of proteins, in: IEEE, International Conference on Machine Learning and Applications (ICMLA '08), 2008.
[10] N. de Freitas, P. Højen-Sørensen, M. Jordan, S. Russell, Variational MCMC, in: Proceedings of the 17th conference in Uncertainty in Artificial Intelligence, 2001.
[11] D.G.T. Denison, C.C. Holmes, B.K. Mallick, A.F.M. Smith, Bayesian Methods for Nonlinear Classification and Regression, Wiley Series in Probability and Statistics, West Sussex, UK, 2002.
[12] T.G. Dietterich, Ensemble methods in machine learning, in: Proceedings of the 1st International Workshop on Multiple Classifier Systems, 2000, pp. 1–15.
[13] C. Ding, I. Dubchak, Multi-class protein fold recognition using support vector machines and neural networks, Bioinformatics 17 (4) (2001) 349–358.
[14] M. Girolami, S. Rogers, Hierarchic Bayesian models for kernel learning, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 241–248.
[15] M. Girolami, S. Rogers, Variational Bayesian multinomial probit regression with Gaussian process priors, Neural Computation 18 (8) (2006) 1790–1817.
[16] M. Girolami, M. Zhong, Data integration for classification problems employing Gaussian process priors, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, vol. 19, MIT Press, Cambridge, MA, 2007, pp. 465–472.
[17] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97–109.
[18] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.
[19] L.I. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, Wiley, New York, 2004.

[20] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, M.I. Jordan, Learning the kernel matrix with semidefinite programming, Journal of Machine Learning Research 5 (2004) 27–72.
[21] W.-J. Lee, S. Verzakov, R.P. Duin, Kernel combination versus classifier combination, in: 7th International Workshop on Multiple Classifier Systems, 2007.
[22] D.P. Lewis, T. Jebara, W.S. Noble, Nonstationary kernel combination, in: 23rd International Conference on Machine Learning, 2006.
[23] D.P. Lewis, T. Jebara, W.S. Noble, Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure, Bioinformatics 22 (22) (2006) 2753–2760.
[24] D. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, Cambridge, 2003.
[25] D.J.C. MacKay, The evidence framework applied to classification networks, Neural Computation 4 (5) (1992) 698–714.
[26] S. Manocha, M.A. Girolami, An empirical analysis of the probabilistic k-nearest neighbour classifier, Pattern Recognition Letters 28 (13) (2007) 1818–1824.
[27] R.M. Neal, Regression and classification using Gaussian process priors, Bayesian Statistics 6 (1998) 475–501.
[28] C.S. Ong, A.J. Smola, R.C. Williamson, Learning the kernel with hyperkernels, Journal of Machine Learning Research 6 (2005) 1043–1071.
[29] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, USA, 2006.
[30] B. Schölkopf, A. Smola, Learning with Kernels, The MIT Press, Cambridge, MA, USA, 2002.
[31] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, England, UK, 2004.
[32] H.-B. Shen, K.-C. Chou, Ensemble classifier for protein fold pattern recognition, Bioinformatics 22 (14) (2006) 1717–1722.
[33] S. Sonnenburg, G. Rätsch, C. Schäfer, A general and efficient multiple kernel learning algorithm, in: Y. Weiss, B. Schölkopf, J. Platt (Eds.), Advances in Neural Information Processing Systems, vol. 18, MIT Press, Cambridge, MA, 2006, pp. 1273–1280.
[34] D.M.J. Tax, M. van Breukelen, R.P.W. Duin, J. Kittler, Combining multiple classifiers by averaging or by multiplying?, Pattern Recognition 33 (2000) 1475–1485.
[35] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, Journal of Machine Learning Research 1 (2001) 211–244.
[36] D.H. Wolpert, Stacked generalization, Neural Networks 5 (2) (1992) 241–259.

About the Author—THEODOROS DAMOULAS was awarded the NCR Ltd. Ph.D. Scholarship in 2006 in the Department of Computing Science at the University of Glasgow. He is a member of the Inference Research Group (http://www.dcs.gla.ac.uk/inference) and his research interests are in statistical machine learning and pattern recognition. He holds a 1st class M.Eng. degree in Mechanical Engineering from the University of Manchester and an M.Sc. in Informatics (Distinction) from the University of Edinburgh.

About the Author—MARK A. GIROLAMI is Professor of Computing and Inferential Science and holds a joint appointment in the Department of Computing Science and the Department of Statistics at the University of Glasgow. His various research interests lie at the boundary between computing, statistical, and biological science. He currently holds a prestigious five year long (2007 until 2012) Advanced Research Fellowship from the Engineering & Physical Sciences Research Council (EPSRC). In 2009 the International Society of Photo-Optical Instrumentation Engineers (SPIE) presented him with their Pioneer Award for his contributions to Biomedical Signal Processing and the development of analytical tools for EEG and fMRI which revolutionized the automated processing of such signals. In 2005 he was awarded an MRC Discipline Hopping Award in the Department of Biochemistry and during 2000 he was the TEKES visiting professor at the Laboratory of Computing and Information Science in Helsinki University of Technology. In 1998 and 1999 Professor Girolami was a research fellow at the Laboratory for Advanced Brain Signal Processing in the Brain Science Institute, RIKEN, Wako-Shi, Japan. He has been a visiting researcher at the Computational Neurobiology Laboratory (CNL) of the Salk Institute. The main focus of the seven post-doctoral researchers and eight Ph.D. students in his group is in the development of statistical inferential methodology to support reasoning about biological mechanisms within a systems biology context.