Inferring Sparse Kernel Combinations and Relevance Vectors: An application to subcellular localization of proteins.

Theodoros Damoulas[†], Yiming Ying[*], Mark A. Girolami[†] and Colin Campbell[*]
[†]Department of Computing Science
University of Glasgow
Sir Alwyn Williams Buidling
Lilybank Gardens
Glasgow G12 8QQ, Scotland, UK
{theo, girolami}@dcs.gla.ac.uk

[*]Department of Engineering Mathematics
University of Bristol
Queen's Building
University Walk
Bristol, BS8 1TR, England, UK
{enxyy, C.Campbell}@bris.ac.uk

Abstract

In this paper, we introduce two new formulations for multi-class multi-kernel relevance vector machines (mRVMs) that explicitly lead to sparse solutions, both in samples and in number of kernels. This enables their application to large-scale multi-feature multinomial classification problems where there is an abundance of training samples, classes and feature spaces. The proposed methods are based on an expectation-maximization (EM) framework employing a multinomial probit likelihood and explicit pruning of non-relevant training samples. We demonstrate the methods on a low-dimensional artificial dataset. We then demonstrate the accuracy and sparsity of the method when applied to the challenging bioinformatics task of predicting protein subcellular localization.

1. Introduction

Recently multi-kernel learning methods (MKL methods) have attracted great interest in the machine learning community [10, 6, 13, 14, 11]. Since many supervised learning tasks in biology involve heterogeneous data they have been successfully applied to many important bioinformatics problems [9, 12, 2], often providing state-of-the-art performance. The intuition behind these multi-kernel methods is to represent a set of heterogeneous features via different types of kernels and to combine the resulting kernels in a convex combination: this is illustrated in Figure 1. In other words, kernel functions k, with corresponding kernel parameters $\boldsymbol{\theta}$, represent the similarities between objects $\mathbf{x}_n$ based on their feature vectors $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$

Learning the kernel combination parameters $\boldsymbol{\beta}$ is therefore an important component of the learning problem. Most MKL research has been done within the popular framework of support vector machines (SVMs) with progress concentrated on finding computationally efficient algorithms via improved optimization routines [15, 20]. Such methods provide sparse solutions in samples and kernels, due to the optimisation over hyperplane normal parameters $\mathbf{w}$ and kernel combination parameters $\boldsymbol{\beta}$, but they inherit the drawbacks of the *non-probabilistic* and *binary* nature of SVMs.

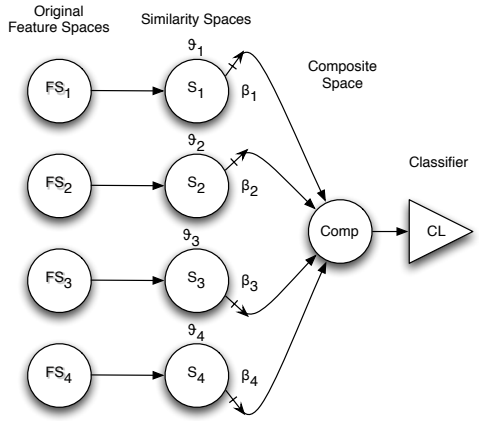

Figure 1. The intuition for MKL: From a heterogenous multitude of feature spaces, to a common metric and finally to a composite space.

In the Bayesian paradigm, the functional form analogous to SVMs is the relevance vector machine (RVM) [18] which employs sparse Bayesian learning via an appropriate prior formulation. Maximization of the marginal likelihood, a type-II maximum likelihood (ML) expression, gives sparse solutions which utilize only a subset of the basis functions: the *relevance vectors*. Compared to an SVM, there are rel-

evatively few relevance vectors and they are typically not close to the decision boundary. However, until now, the multi-class adaptation of RVMs was problematic [18] due to the bad scaling of the type-II ML procedure with respect to C, the number of classes. Furthermore, although in regression problems the maximization of the marginal likelihood is only required once, for classification this is repeated for every update of the parameter posterior statistics.

In this paper we describe two multi-class multi-kernel RVM methods which are able to address multi-kernel learning while producing both sample-wise and kernel-wise sparse solutions. In contrast to SVM approaches, they utilize the probabilistic framework of RVMs, avoid pre-computation of margin trade-off parameters or cross-validation procedures and produce posterior probabilities of class memberships without using ad-hoc post-processing methods.

In contrast with the original RVM [17, 18], the proposed methods employ the multinomial probit likelihood [1], which results in multi-class classifiers via the introduction of auxiliary variables. In one case (m-RVM$_1$) we propose a multi-class extension of the fast type-II ML procedure in [16, 4] and in the second case (m-RVM$_2$) we *explicitly* employ a flat prior for the hyper-parameters that control the sparsity of the resulting model. In both cases, inference on the kernel combinatorial coefficients is enabled via a constrained QP procedure and an efficient expectation-maximization (EM) scheme is adopted. The two algorithms are suitable for different large-scale application scenarios based on the size of the initial training samples.

Within a Bayesian framework, we have pursued related work on kernel learning for binary classification [7], combination of covariance functions within a Gaussian Process (GP) methodology [8] and the variational treatment of the multinomial probit likelihood with GP priors [6]. The present work can be seen as the maximum-a-posteriori solution of previous work [2] with sparsity inducing priors and maximization of a marginal likelihood. In a summary we offer the following novel contributions:

- A fast type-II ML procedure for multi-class regression and classification problems.
- A constructive type [16] m-RVM (m-RVM$_1$).
- A bottom-down type [18] m-RVM utilizing a sparse hyper-prior to prune samples (m-RVM$_2$).
- Multi-kernel adaptations for both these methods to handle multi-feature problems.

2 Model formulation

We consider feature spaces S in which a D^s-dimensional sample $\mathbf{x}_n^s$ has an associated label $t_n \in \{1, \ldots, C\}$. We apply *kernel substitution* in each feature space and embed our features in base kernels $\mathbf{K}^s \in \Re^{N \times N}$ that can be combined into our composite kernel and so we let

$$K^{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{S} \beta_s K^s\left(\mathbf{x}_i^s, \mathbf{x}_j^s\right) \tag{1}$$

Introducing the auxiliary variables $\mathbf{Y} \in \Re^{N \times C}$ and parameters $\mathbf{W} \in \Re^{N \times C}$ we regress on $\mathbf{Y}$ with a standardized noise model, see [1, 6], thus:

$$y_{nc}|\mathbf{w}_c, \mathbf{k}_n^{\boldsymbol{\beta}} \sim \mathcal{N}_{y_{cn}}\left(\mathbf{k}_n^{\boldsymbol{\beta}}\mathbf{w}_c, 1\right). \tag{2}$$

Then we link the regression target to the classification label via the standard multinomial probit function

$$t_n = i \text{ if } y_{in} > y_{jn}\ \forall\, j \neq i. \tag{3}$$

The resulting multinomial probit likelihood (details in [6, 2]) is given by

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

Finally we introduce a zero-mean Gaussian prior distribution for the regression parameters $w_{nc} \sim \mathcal{N}\left(0, \frac{1}{\alpha_{nc}}\right)$ with scale α_{nc}, and place a Gamma prior distribution with hyper-parameters a, b on these scales in accordance with standard Bayesian approaches [3] and the RVM formalism. This hierarchical Bayesian framework results in an implicit Student-t distribution on the parameters [18] and therefore encourages sparsity. Together with appropriate inference of the scales α, this is the main focus of the RVM approach and hence it will play an important role in both m-RVM algorithms that we now propose.

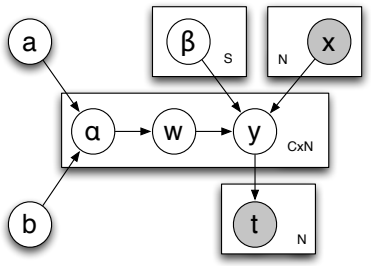

Figure 2. Plates diagram of the model.

2.1 m-RVM$_1$

The first multi-class multi-kernel RVM we consider is based on the "constructive" variant of RVMs [16, 4] which

employs a fast type-II ML procedure. The maximization of the marginal likelihood

$$p(\mathbf{Y}|\mathbf{X},\boldsymbol{\alpha}) = \int p(\mathbf{Y}|\mathbf{X},\mathbf{W})p(\mathbf{W}|\boldsymbol{\alpha})d\mathbf{W} \qquad (5)$$

with respect to $\boldsymbol{\alpha}$ results in a criterion to either add a sample, delete or update its associated hyper-parameter α_n. Therefore, the model can start with a single sample and proceed in a constructive manner as detailed below. The (log) multi-class marginal likelihood is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}) &= \log p(\mathbf{Y}|\boldsymbol{\alpha}) = \log \textstyle\int_{-\infty}^{+\infty} p(\mathbf{Y}|\mathbf{W})p(\mathbf{W}|\boldsymbol{\alpha})d\mathbf{W} \\ &= \sum_{c=1}^{C} -\frac{1}{2}[N\log 2\pi + \log|\boldsymbol{\mathcal{C}}| + \mathbf{y}_c^{\mathsf{T}}\boldsymbol{\mathcal{C}}^{-1}\mathbf{y}_c]\end{aligned}$$

where $\boldsymbol{\mathcal{C}} = \mathbf{I} + \mathbf{K}\mathbf{A}^{-1}\mathbf{K}^{\mathsf{T}}$ for composite kernel $\mathbf{K}$ and $\mathbf{A}$ is defined as $\text{diag}(\alpha_1,\ldots,\alpha_N)$. Here we have made the assumption that a common scale α_n is shared across classes for every sample n. This allows an effective type-II ML scheme based on the original binary scheme proposed by Tipping and Faul [16, 4].

The decomposition of terms in $\boldsymbol{\mathcal{C}}$ follows exactly as [16] listed as below

$$|\boldsymbol{\mathcal{C}}| = |\boldsymbol{\mathcal{C}}_{-i}|\,|1 + \alpha_i^{-1}\mathbf{k}_i^{\mathsf{T}}\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i|, \qquad (6)$$

and

$$\boldsymbol{\mathcal{C}}^{-1} = \boldsymbol{\mathcal{C}}_{-i}^{-1} - \frac{\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i\mathbf{k}_i^{\mathsf{T}}\boldsymbol{\mathcal{C}}_{-i}^{-1}}{\alpha_i + \mathbf{k}_i^{\mathsf{T}}\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i}. \qquad (7)$$

Hence the (log) marginal likelihood can be decomposed as $\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}(\boldsymbol{\alpha}_{-i}) + l(\alpha_i)$ with $l(\alpha_i)$ given by

$$\sum_{c=1}^{C}\frac{1}{2}\left[\log\alpha_i - log(\alpha_i + s_i) + \frac{q_{ci}^2}{\alpha_i + s_i}\right] \qquad (8)$$

By slightly modifying the same analysis as in [4] to the multi-class case, $\mathcal{L}(\boldsymbol{\alpha})$ has again a unique maximum with respect to α_i

$$\alpha_i = \frac{Cs_i^2}{\sum_{c=1}^{C} q_{ci}^2 - Cs_i}, \quad \text{if} \quad \sum_{c=1}^{C} q_{ci}^2 > Cs_i, \qquad (9)$$

$$\alpha_i = \infty, \qquad \text{if} \quad \sum_{c=1}^{C} q_{ci}^2 \le Cs_i. \qquad (10)$$

where we follow [16] in defining the 'sparsity factor' s_i and the now *multi-class* 'quality factor' q_{ci}:

$$s_i \triangleq \mathbf{k}_i^{\mathsf{T}}\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i \;\text{ and }\; q_{ci} \triangleq \mathbf{k}_i^{\mathsf{T}}\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{y}_c. \qquad (11)$$

It is worth noting that although the sparsity factor s_i can be still be seen as a measure of overlap between sample $\mathbf{k}_i$ and the those already included, the quality factor q_{ci} is now class-specific and the unique maximum of a retained sample's scale (Eq. 9) is an average, over classes, of the original binary maximum solution.

Furthermore, this multi-class formulation of the fast type-II ML procedure can be directly used for multinomial regression problems with little additional overhead to the original binary procedure as it only requires an extra summation over the 'quality factors' q_{ci}. Returning back to our classification framework, the M-steps for the estimates $\hat{\mathbf{W}}$ and $\hat{\boldsymbol{\beta}}$ are given by

$$\hat{\mathbf{W}}_* = \left(\mathbf{K}_*^{\mathsf{T}}\mathbf{K}_* + \mathbf{A}\right)^{-1}\mathbf{K}_*^{\mathsf{T}}\tilde{\mathbf{Y}}, \qquad (12)$$

and

$$\begin{aligned}&\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \tfrac{1}{2}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\Omega}\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{f} \\ &\text{s.t}\quad \beta_i \ge 0\;\forall\, i \;\text{ and }\; \textstyle\sum_{s=1}^{S}\beta_s = 1\end{aligned} \qquad (13)$$

where $\boldsymbol{\Omega}_{ij} = \sum_{n,c}^{N,C}\mathbf{w}_c\mathbf{k}_n^{i\,\mathsf{T}}\mathbf{k}^j{}_n\mathbf{w}_c^{\mathsf{T}}$ is an $S\times S$ matrix, $f_i = \sum_{n,c}^{N,C}\mathbf{w}_c\mathbf{k}_n^{i\,\mathsf{T}}\tilde{y}_{nc}$, and the $*$ notation implies that currently M samples are included (i.e $\mathbf{W}_*$ is $M\times C$ and $\mathbf{K}_*$ is $N\times M$) in the model.

Finally in the E-step the posterior expectation $\mathcal{E}_{\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\mathbf{t}}\{y_{nc}\}$ of the latent variables is obtained, see [6, 2] with a closed form representation given by

$$\begin{aligned}\widetilde{y}_{nc} &\leftarrow \mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_c - \frac{\mathcal{E}_{p(u)}\{\mathcal{N}_u\left(\mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_c - \mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_i, 1\right)\Phi_u^{n,i,c}\}}{\mathcal{E}_{p(u)}\{\Phi\left(u + \mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_i - \mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_c\right)\Phi_u^{n,i,c}\}} \\ \widetilde{y}_{ni} &\leftarrow \mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_i - \left(\textstyle\sum_{j\neq i}\widetilde{y}_{nj} - \mathbf{k}_n^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{w}}_j\right)\end{aligned} \qquad (14)$$

where Φ is the cumulative distribution function and $\Phi_u^{n,i,c}$ defined as $\prod_{j\neq i,c}\Phi\left(u + \widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right)$. The resulting predictive likelihood for an unseen sample $\mathbf{x}_\dagger^s$ embedded into S base kernels $\mathbf{k}_\dagger^s$ is given by

$$\begin{aligned}p\left(t_\dagger = c|\mathbf{x}_\dagger^s,\mathbf{X},\mathbf{t}\right) &= \textstyle\int \delta_c^\dagger\mathcal{N}_{\mathbf{y}^\dagger}\left(\mathbf{k}_\dagger^{\hat{\boldsymbol{\beta}}}\hat{\mathbf{W}},\mathbf{I}\right)d\mathbf{y}^\dagger \\ &= \mathcal{E}_{p(u)}\left\{\textstyle\prod_{j\neq c}\Phi\left(u + \mathbf{k}_\dagger^{\hat{\boldsymbol{\beta}}}\left(\hat{\mathbf{w}}_c - \hat{\mathbf{w}}_j\right)\right)\right\}.\end{aligned}$$

Here the expectation $\mathcal{E}_{p(u)}$ is taken, in the usual manner, with respect to the standardized normal distribution $p(u) = \mathcal{N}(0,1)$. Typically 1,000 drawn samples give a good approximation. In Algorithm 1 a procedure for m-RVM$_1$ is given, summarizing the above section.

2.2 m-RVM$_2$

In the next multi-class multi-kernel RVM proposed we will not adopt marginal likelihood maximization but rather employ an extra E-step for the updates of the hyper-parameters $\boldsymbol{\alpha}$. This leads to a bottom-down sample pruning procedure which starts with the full model and results in a

Algorithm 1 mRVM$_1$

Initialization
Sample $\mathbf{Y} \in \Re^{C \times N}$ to follow target $\mathbf{t}$.
while Iterations $<$ max & Convergence $<$ Threshold **do**
  while Convergence **do**
    Fast Type-II ML : Similar to [16] for new updates Eq. 9, 10
  end while
  M-Step for $\mathbf{W}$: Eq. 12
  E-Step for $\mathbf{Y}$: Eq. 14
  QP program for $\boldsymbol{\beta}$: Eq. 13
end while

sparse model through constant discarding of non-relevant samples. This has the potential disadvantage that removed samples cannot be re-introduced into the model.

Revisiting our model we have $p(\mathbf{t}|\mathbf{X}, a, b) = \int p(\mathbf{t}|\mathbf{Y})p(\mathbf{Y}|\mathbf{W}, \beta, \mathbf{X})p(\mathbf{W}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|a, b)d\mathbf{Y}d\mathbf{W}d\boldsymbol{\alpha}$ and we are interested in the posterior of the hyper-parameters $p(\boldsymbol{\alpha}|\mathbf{W}, a, b) \propto p(\mathbf{W}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|a, b)$.

The prior on the parameters is a product of normal distributions $\mathbf{W}|\boldsymbol{\alpha} \sim \prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{N}_{w_{nc}}(0, \frac{1}{\alpha_{nc}})$ and the conjugate prior on the scales is a product of Gamma distributions $\boldsymbol{\alpha}|a, b \sim \prod_{c=1}^{C}\prod_{n=1}^{N}\mathcal{G}_{\alpha_{nc}}(a, b)$. Hence we are led to a closed form Gamma posterior with updated parameters, $\alpha_{nc}|w_{nc}, a, b \sim \mathcal{G}(1 + a, w_{nc}^2 + b)$. Therefore the E-step is just the expected value or mean of that distribution and we are left with the following well-known update

$$\alpha_{nc} = \frac{1 + 2a}{w_{nc}^2 + 2b} \tag{15}$$

Hence, we are now following the initial RVM formulation [17] and we simply place a flat prior (i.e $a, b \to 0$) on the scales which in the limit lead to the improper prior for the parameters $p(w_{nc}) \propto \frac{1}{|w_{nc}|}$, as given in [18]. The M-steps for the parameters $\mathbf{W}$ and the kernel combination coefficients $\boldsymbol{\beta}$ are given as before in Eq. 12 and Eq. 13 respectively, and also the E-step for the latent variables $\mathbf{Y}$ in Eq. 14. The procedure is given below in Algorithm 2.

Algorithm 2 mRVM$_2$

Initialization
Sample $\mathbf{Y} \in \Re^{C \times N}$ to follow target $\mathbf{t}$.
while Iterations $<$ max & Convergence $<$ Threshold **do**
  M-Step for $\mathbf{W}$: Eq. 12
  E-Step for $\mathbf{Y}$: Eq. 14
  E-Step for $\boldsymbol{\alpha}$: Eq. 15
  Prune $\mathbf{w}_i$, and $\mathbf{k}_i$ when $a_{ic} > 10^6 \; \forall \; c$
  QP program for $\boldsymbol{\beta}$: Eq. 13
end while

3 Multi-class relevant vectors

Due to the different inference approaches adopted for the scales $\boldsymbol{\alpha}$, which in m-RVM$_1$ is the maximization of the marginal likelihood and in m-RVM$_2$ is the E-step update, the resultant level of sparsity will slightly vary for the two methods. Furthermore, since the first is a constructive approach and the second a pruning approach, there are significant differences on the way sparse solutions are achieved.

In order to study that and visualize the "relevant" vectors retained by the models we examine a 2-D artificial dataset with 3 classes. This dataset has t=1 when $0.5 > x_1^2 + x_2^2 > 0.1$, t=2 when $1.0 > x_1^2 + x_2^2 > 0.6$ and t=3 when $[x_1, x_2]^\mathsf{T} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Convergence is monitored via the mean % change of $(\mathbf{Y} - \mathbf{KW})$.

In Figure 3 typical resulting multi-class relevant vectors are shown.

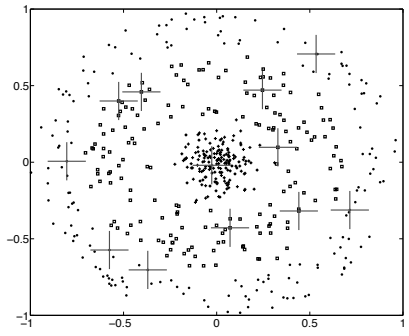

Figure 3. Typical Relevant vectors

The differences in the resulting sparsity and associated error progression can be seen from Figure 4. Results are averaged over 20 randomly initialized trials while keeping the same train/test split and varying the number of iterations.

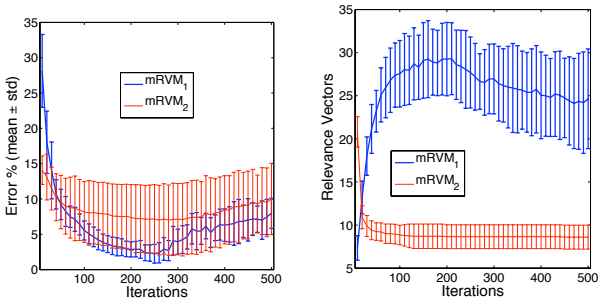

Figure 4. Error and Sparsity progression.

The results in this artificial dataset demonstrate the different nature of the two approaches, as it can be seen mRVM$_1$ starts with a single sample and progresses sequentially adding and deleting vectors as it goes through the data. We can see that after a certain dataset-dependent point the model gets greedy, discarding more samples than needed and hence the average error percentag starts increasing. Furthermore, the number of relevant vectors retained varies sig-

nificantly from trial to trial as it can be witnessed by the increasing standard deviation.

On the other hand, mRVM$_2$ starts with the full model and prunes down samples. As it can be seen, the variance of the error is large due to the sensitivity to initial conditions. However, the mean retained vectors, or relevant vectors, stays almost constant. In this toy dataset, there is no statistical significant difference in zero-one loss when convergence is monitored.

4 Protein subcellular localization

In order to evaluate performance on real world datasets we consider the problem of predicting subcellular localization based on a set of disparate data sources, represented as a set of feature spaces and incorporated in the method by a set of appropriate kernels. We follow the experimental setup of Ong and Zien [20] by employing 69 feature spaces of which 64 are motif kernels computed at different sections of the protein sequence and the rest are pairwise string kernels based on BLAST E-values and phylogenetic profile kernels.

Two problems are considered: predicting subcellular localization for Gram positive (PSORT+) and Gram negative bacteria (PSORT-). Original state-of-the-art performance on this problem was given by PSORTb [5], a prediction tool utilizing multiple SVMs and a Bayesian network which provides a prediction confidence measure for the method, compensating for the non-probabilistic formulation of standard SVMs. The confidence measure can be thresholded to perform class assignment or to indicate some samples as unclassifiable.

Recently, a MKL method with SVMs [20] claimed a new state-of-the art performance, on a reduced subset of the PSORTb dataset, with reported performances of 93.8 ± 1.3 on PSORT+ and 96.1 ± 0.6 on PSORT- using an average F1 score. However due to the non-probabilistic nature of SVMs the MKL method was augmented with a post-processing criteria to create class probabilities in order to leave out the 13% lowest confidence predictions for PSORT+ and 15% for PSORT-, thus approximating the unclassifiable assignment option of PSORTb. We also compare with another multi-class multi-kernel learning algorithm proposed in [19] for regularized kernel discriminant analysis (RKDA). For this algorithm, we employ the semi-infinite linear programming (SILP) approach with a fixed regularization parameter 5×10^{-4} as suggested there.

In Table 1 we report the average test-error percentage over 10 randomly initialized 80% training and 20% test splits on the PSORT+ subset for both m-RVM methods and report the resulting average sample sparsity of the two models. Similarly Table 2 presents the results for PSORT-. We point out that there are no analogous sparse relevant vectors in the RKDA kernel learning approach and the method relies on all the training samples.

Method	Test Error%	Relevance Vectors
m-RVM$_1$	12.9 ± 3.7	27.9 ± 4.5
m-RVM$_2$	10.4 ± 3.9	60.8 ± 4.3
RKDA-MKL	8.39 ± 1.46	$--$

Table 1. Error and sparsity on PSORT+

Method	Test Error%	Relevance Vectors
m-RVM$_1$	13.8 ± 4.5	109.2 ± 19.5
m-RVM$_2$	11.9 ± 1.2	102.7 ± 7.4
RKDA-MKL	10.52 ± 2.56	$--$

Table 2. Error and sparsity on PSORT-

The sparsity of the kernel combinations for PSORT+ can be seen from Figure 5, where the average kernel combination parameters $\boldsymbol{\beta}$, over the 10 runs, is shown in reverse alphabetical order to the kernel collection provided by [20]. We are in general agreement with the selected kernels from previous studies as E-value kernels (3,4) and phylogeny kernels (68,69) are judged significant in these combinations.

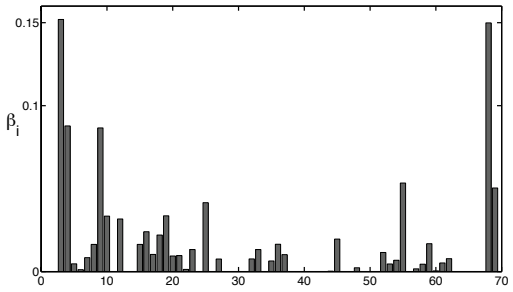

Figure 5. Average kernel usage: PSORT+

Similarly for PSORT-, Figure 6 indicates that the E-value and phylogeny kernels are significant contributors. We now have sample-wise and kernel-wise sparse solutions for the problem under consideration.

5 Conclusion

In this contribution we have described two multi-class and multi-kernel extensions of relevance vector machines and their application to a significant multi-feature problem in the area of subcellular localization prediction. Following the original derivation of the fast type-II ML we present a

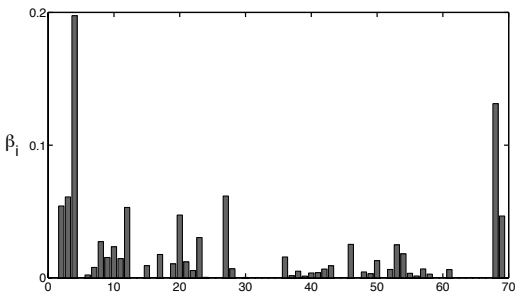

Figure 6. Average kernel usage: PSORT-

multi-class extension. The additional computational overhead to the binary case is minimal and given only by a summation of the 'quality factor' over classes. This renders the multi-class extension very efficient for large multinomial problems following the already established benefits of sparse Bayesian learning.

The application of m-RVMs to subcellular localization offers the ability to integrate heterogenous feature spaces while imposing sparse solutions and being able to cope with a large number of training samples. Depending on the requirements, either a constructive approach (m-RVM_1), which has less bias on initialization, or a bottom-down one (m-RVM_2), which increases learning rate as basis/samples are removed, can be used to tackle multi-feature multi-class problems.

6. Acknowledgments

NCR Financial Solutions Group Ltd provided a Scholarship to T.D. An EPSRC Advanced Research Fellowship (EP/E052029/1) was awarded to M.A.G. T.D acknowledges the support received from the Advanced Technology & Research Group within the NCR Financial Solutions Group Ltd company and especially the help and support of Dr. Gary Ross and Dr. Chao He.

References

[1] J. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.

[2] T. Damoulas and M. A. Girolami. Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection. *Bioinformatics*, 24(10):1264–1270, 2008.

[3] D. G. T. Denison, C. C. Holmes, B. K. Mallick, and A. F. M. Smith. *Bayesian Methods for Nonlinear Classification and Regression*. Wiley Series in Probability and Statistics, West Sussex, UK, 2002.

[4] A. Faul and M. Tipping. Analysis of sparse bayesian learning. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[5] J. L. Gardy, M. R. Laird, F. Chen, S. Rey, C. J. Walsh, M. Ester, and F. S. L. Brinkman. Psortb v.2.0: Expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. *Bioinformatics*, 21(5):617–623, 2005.

[6] M. Girolami and S. Rogers. Hierarchic Bayesian models for kernel learning. In *Proceedings of the* 22^{nd} *International Conference on Machine Learning*, pages 241–248, 2005.

[7] M. Girolami and S. Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18(8):1790–1817, 2006.

[8] M. Girolami and M. Zhong. Data integration for classification problems employing Gaussian process priors. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 465–472, Cambridge, MA, 2007. MIT Press.

[9] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[10] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[11] D. P. Lewis, T. Jebara, and W. S. Noble. Nonstationary kernel combination. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[12] D. P. Lewis, T. Jebara, and W. S. Noble. Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics*, 22(22):2753–2760, 2006.

[13] C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[14] S. Sonnenburg, G. Rätsch, and C. Schäfer. A general and efficient multiple kernel learning algorithm. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1273–1280, Cambridge, MA, 2006. MIT Press.

[15] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *JMLR*, 1:1–18, 2006.

[16] M. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *Proceedings of 9th AISTATS Workshop*, pages 3–6, 2003.

[17] M. E. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems 12*, pages 652–658, 1999.

[18] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *JMLR*, 1:211–244, 2001.

[19] J. Ye, S. Ji, and J. Chen. Multi-class discriminant kernel learning via convex programming. *JMLR*, 9:719–758, 2008.

[20] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, New York, NY, USA, 2007. ACM.