Multiclass Relevance Vector Machines: Sparsity and Accuracy

Ioannis Psorakis, Theodoros Damoulas, Mark A. Girolami

***Abstract*—In this paper we investigate the sparsity and recognition capabilities of two approximate Bayesian classification algorithms, the multi-class multi-kernel Relevance Vector Machines (mRVMs) that have been recently proposed. We provide an insight on the behavior of the mRVM models by performing a wide experimentation on a large range of real world datasets. Furthermore, we monitor various model fitting characteristics that identify the predictive nature of the proposed methods and we compare against existing classification techniques. By introducing novel convergence measures, sample selection strategies and model improvements, it is demonstrated that mRVMs can produce state of the art results on multi-class discrimination problems. In addition, this is achieved by utilizing only a very small fraction of the available observation data.**

***Index Terms*—Bayesian learning, classification, sparsity, multi-class discrimination, kernel methods**

I. Introduction

In Supervised Learning, classification or supervised discrimination is the process of categorizing samples based on available observations or past experience. We formulate a mathematical model, captured as a function $y(\mathbf{w};\mathbf{x})$ which maps an observation $\mathbf{x}^\star$ with D features to a discrete label $c \in \{1,...,C\}$, where C denotes the number of different classes. Thus given a set of N observations along with their respective labels $\{\mathbf{x}_i, t_i\}_{i=1}^N$, we infer the appropriate values for parameters $\mathbf{w}$ which give our model appropriate predictive, descriptive and generalizing properties.

The training set $\{\mathbf{x}_i, t_i\}_{i=1}^N$ captures our past experience, either as a subset of our past measurements which we consider reliable or the only available knowledge of a phenomenon. The latter is not usually the case for todays systems, where advances in sensor technology allow the collection of vast amount of measurements [2]. So, research has been driven towards formulating models which identify the key observations of a phenomenon, providing insight on its generic nature and retaining low computational complexity. These models belong to the *sparse* family of Supervised Learning methods because they utilize only a subset of the training set data, by informatively pruning out unnecessary samples or features based on a certain performance criterion . Some of the most popular sparse models are Support Vector Machines (SVMs) [11] , Informative Vector Machines (IVMs) [9] , Relevance Vector Machines (RVMs) [14] , and Lasso [13] which often provide state of the art results in many problems.

I. Psorakis is a PhD student at the Department of Engineering Science, University of Oxford, UK (email: yannis@robots.ox.ac.uk)

T. Damoulas is a post doctoral associate in the Department of Computer Science, Cornell University, USA (email: damoulas@cs.cornell.edu)

M. A. Girolami is Professor in the Department of Statistical Science, University College London, UK. (email: girolami@stats.ucl.ac.uk)

In addition to identifying the key elements of a data set, another important issue is to be able to capture predictive errors in a systematic way. For this reason, many models such as the Relevance Vector Machines employ a Bayesian treatment in order to produce *probabilistic outputs* for class membership (in classification) or continuous target value estimation (in regression). Measuring the predictive error is a critically valuable aspect in modern applications with asymmetric misclassification costs such as medicine or finance [2].

The Relevance Vector Machine (RVM) originally introduced by M. Tipping (2001), is a Bayesian learning model which provides state of the art results both in terms of accuracy and sparsity via appropriate formulation of hierarchical priors, effectively constraining the majority of the model parameters w_{nc} around zero. Thus, by maximizing the marginal likelihood using a type-II maximum likelihood (ML) procedure, we achieve solutions which utilize only a small subset of the original basis functions, named the *relevance vectors*.

Although the Relevance Vector Machine provides significantly competitive results in contrast to the traditional Support Vector Machine, its adaptation to the multi-class setting has been problematic, due to the bad scaling of the type-II ML procedure with respect to the number of classes C [6] and the dimensionality of the Hessian required for the Laplace approximation [3]. Recently, two novel classification algorithms, mRVM_1 and mRVM_2 have been introduced which expand the original Relevance Vector Machine to the multi-class multi-kernel setting [6]. These algorithms achieve sparsity without the constraint of having a binary class problem and provide probabilistic outputs for class membership instead of the hard binary decisions given by the traditional SVMs.

mRVMs expand the original RVM to the multi-class setting by introducing auxiliary variables $\mathbf{Y}$, that act as intermediate regression targets, that naturally lead to the multinomial probit likelihood [1] for the estimation of class membership probabilities. In the case of mRVM_1, the fast type-II ML is adapted to the multi-class setting while in mRVM_2 a flat prior for the hyper-parameters is explicitly employed that controls the sparsity of the resulting model. The two versions of mRVM differ on how they manipulate the kernel during the training phase; the mRVM_1 follows a *constructive* approach, incrementally adding samples to the model based on a contribution criterion while the mRVM_2 follows a *top-down* approach, loading the whole training set and pruning out uninformative samples. Adopting one of the two variants depends heavily on parameters of the problem context, such as the size of the initial training set and the available computational resources (see following sections). Additionally, mRVMs can be utilized in multiple kernel learning (MKL) problems as seen in [6].

In the present work, our intention is to provide:

- A theoretical insight on mRVMs and their convergence properties.
- Investigate the sparsity versus accuracy trade-off and prediction confidence of the probabilistic outputs.
- Propose an 'informative sample selection' methodology for mRVM$_1$, a technique to reduce its computational complexity and convergence criteria for both models.
- Provide an extensive evaluation of mRVM$_1$ and mRVM$_2$ along with a comparison against other classification models.

Initially, we provide the theoretical basis of mRVMs along with their respective pseudocodes. Then we present our experimentation results and we compare the performance of mRVMs against competing methods. Finally we conclude by analyzing our results and by providing ideas for future work.

II. Model formulation

Following the standard approach in the Machine Learning literature [3] [2], in classification we are given [1] a training set $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$ where $\mathbf{x} \in \Re^D$ our D featured observations and $t \in \{1...C\}$ their respective class labels. More conveniently, our observations can be expressed as $\mathbf{X} \in \Re^{N\times D}$ from which we derive our training kernel $\mathbf{K} \in \Re^{N\times N}$ based on a dataset-dependent kernel function.

The training kernel captures our prior knowledge over the data; each row $\mathbf{k}_n$ of the kernel $\mathbf{K}$ expresses how related, based on the selected kernel function, is the observation n to the others of the training set. The learning process involves the inference of the model parameters $\mathbf{W} \in \Re^{N\times C}$, which by the quantity $\mathbf{W}^T\mathbf{K}$ act as a voting system to express which relationships of the data are important in order for our model to have appropriate discriminative properties.

Multiple class discrimination is achieved by the introduction of auxiliary variables $\mathbf{Y} \in \Re^{C\times N}$ that act as the regression targets of $\mathbf{W}^T\mathbf{K}$ following a standardized noise model $y_{cn}|\mathbf{w}_c, \mathbf{k}_n \sim \mathcal{N}_{y_{cn}}\left(\mathbf{w}_c^T\mathbf{k}_n, 1\right)$ [1]. The auxiliary variables are endowed with independent standardized Gaussian probability distributions to ensure statistical identifiability and enable closed form iterative inference [1]. By following the intuition in [4] , as the regressors $\mathbf{W}$ express the weight with which a datapoint "votes" for a specific class, the auxiliary variables $\mathbf{Y}$ express a class membership ranking system; given a sample n, we assign it to the class c with the highest y_{cn} . The continuous nature of $\mathbf{Y}$ not only allows *multiple class discrimination* via the multinomial probit link [1] $t_n = i$ if $y_{ni} > y_{nj}\ \forall\, j \neq i$ but also a *probabilistic output* for class membership via the resulting multinomial probit likelihood function [5] [7]:

$$P\left(t_n = i|\mathbf{W}, \mathbf{k}_n\right) = \mathcal{E}_{p(u)}\left\{\prod_{j\neq i}\Phi\left(u + \left(\mathbf{w}_i - \mathbf{w}_j\right)^T\mathbf{k}_n\right)\right\} \tag{1}$$

Where $u \sim \mathcal{N}(0,1)$ and Φ the Gaussian cumulative distribution function (CDF).

[1]Throughout this paper m denotes scalar, $\mathbf{m}$ vector and $\mathbf{M}$ a matrix. Given the matrix $\mathbf{M}$, $\mathbf{m}_i$ denotes the row vector from the i-th row of $\mathbf{M}$ unless stated otherwise.

In accordance to the original Relevance Vector Machine [14], the regressors w_{nc} from $\mathbf{W}$ follow a standard normal distribution with zero mean and variance a_{nc}^{-1}, where a_{nc} belongs to the scales matrix $\mathbf{A} \in \Re^{N\times C}$ and follows a Gamma distribution with hyperparameters τ, υ. With sufficiently small τ, υ ($< 10^{-5}$) the scales $\mathbf{A}$ restrict $\mathbf{W}$ around its zero mean due to small variance. Thus, only a small subset of the regressors w_{nc} are non-zero, subsequently leading to a sparse solution.

The diagram of the overall model is illustrated in Fig. 1. As seen in [14], this hierarchical Bayesian framework results in an implicit Student-t distribution that encourages sparsity by restricting the regression coefficients $\mathbf{W}$ posterior distribution around zero.

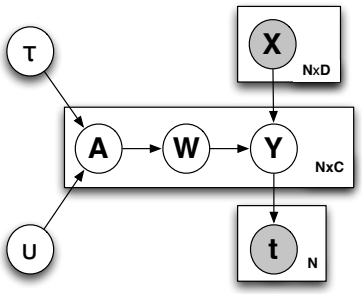


Fig. 1: Plates diagram of the model.

The training procedure involves consecutive updates of the model parameters based on a standard Expectation Maximization (E-M) scheme. Based on Fig. 1 we can derive the regressors $\mathbf{W}$ closed form posterior:

$$\begin{aligned} P(\mathbf{W}|\mathbf{Y}) &\propto P(\mathbf{Y}|\mathbf{W})P(\mathbf{W}|\mathbf{A}) \\ &\propto \prod_{c=1}^{C}\mathcal{N}((\mathbf{K}\mathbf{K}^T+\mathbf{A}_c)^{-1}\mathbf{K}\mathbf{y}_c^T,(\mathbf{K}\mathbf{K}^T+\mathbf{A}_c)^{-1}) \end{aligned}$$

Where $\mathbf{A}_c$ a diagonal matrix derived from the c column of $\mathbf{A}$ which expresses the scales α_{ic} across samples. Based on the above, the Maximum a Posteriori (MAP) estimator for the regressors is $\hat{\mathbf{W}} = \underset{W}{\operatorname{argmax}} P(\mathbf{W}|\mathbf{Y}, \mathbf{A}, \mathbf{K})$. So given a class, the parameters across samples are updated based on the maximum a posteriori value:

$$\hat{\mathbf{w_c}} = (\mathbf{K}\mathbf{K}^{\mathbf{T}} + \mathbf{A}_c)^{-1}\mathbf{K}\mathbf{y}_c^T \tag{2}$$

From (2) and by following [4], we derive the posterior distribution of the auxiliary variables, which is a product of $C \times N$ dimensional conically truncated Gaussians. So, given a class i, the E-step $\forall c \neq i$ is:

$$\widetilde{y}_{cn} \leftarrow \hat{\mathbf{w}}_c^T\mathbf{k}_n - \frac{\mathcal{E}_{p(u)}\{\mathcal{N}_u\left(\hat{\mathbf{w}}_c^T\mathbf{k}_n - \hat{\mathbf{w}}_i^T\mathbf{k}_n, 1\right)\Phi_u^{n,i,c}\}}{\mathcal{E}_{p(u)}\{\Phi\left(u + \hat{\mathbf{w}}_i^T\mathbf{k}_n - \hat{\mathbf{w}}_c^T\mathbf{k}_n\right)\Phi_u^{n,i,c}\}} \tag{3}$$

and for the i-th class:

$$\widetilde{y}_{in} \leftarrow \hat{\mathbf{w}}_i^T\mathbf{k}_n - \left(\sum_{j\neq i}\widetilde{y}_{jn} - \hat{\mathbf{w}}_j^T\mathbf{k}_n\right) \tag{4}$$

where the 'tilde' symbol above y denotes the expected value.

Finally, we define the update step for the hyperpriors α_{nc} which are again derived from the mean of a Gamma distribution, given the hyper-parameters τ, υ. Again, our closed form posterior is:

$$\begin{aligned} P(\mathbf{A}|\mathbf{W}) &\propto P(\mathbf{W}|\mathbf{A})P(\mathbf{A}|\tau,\upsilon) \\ &\propto \prod_{c=1}^{C}\prod_{n=1}^{N} \mathcal{G}(\tau+\frac{1}{2}, \frac{w_{nc}^2+2\upsilon}{2}) \end{aligned}$$

The mean of the above Gamma distribution is:

$$\widetilde{\alpha_{nc}} = \frac{2\tau+1}{w_{nc}^2+2\upsilon} \tag{5}$$

Each iteration of the learning (training) procedure involves the updates from (5), (2), (3), (4) for each model parameter, until an appropriate convergence measure is satisfied. In the following sections we describe in detail how each mRVM extends the above standard Expectation Maximization scheme in terms of sparsity induction, convergence and sample selection.

III. mRVM$_1$

A. fast type-II ML

The mRVM$_1$ is an extension of the 'new' RVM [15] [8] to the multi-class and multi-kernel setting. mRVM$_1$ achieves sparsity based on a constructive approach by starting with an empty model and adding or removing samples from the training kernel based on their contribution to the model fitness.

mRVM$_1$ employs a fast type-II Maximum Likelihood (ML) procedure, where we maximize the marginal likelihood of the model $P(\mathbf{Y}|\mathbf{K},\mathbf{A}) = \int P(\mathbf{Y}|\mathbf{K},\mathbf{W})P(\mathbf{W}|\mathbf{A})d\mathbf{W}$ with respect to the scales $\mathbf{A}$. In this model, in order to have a differentiable marginal likelihood, we follow the assumption that each sample n has a common scale α_n which is shared across classes. The procedure we follow [15] is to decompose the log-marginal likelihood into contributing terms based on each sample so we can derive criteria to add, remove or update the hyperparameter $\alpha_{\mathbf{n}}$ of an observation.

So, given the log of the marginal likelihood $\mathcal{L}(\mathbf{A}) = \log P(\mathbf{Y}|\mathbf{K},\mathbf{A}) = \log \int P(\mathbf{Y}|\mathbf{K},\mathbf{W})P(\mathbf{W}|\mathbf{A})d\mathbf{W}$ we derive:

$$\mathcal{L}(\mathbf{A}) = \sum_{c=1}^{C} -\frac{1}{2}[N\log 2\pi + \log|\boldsymbol{\mathcal{C}}| + \mathbf{y}_c^T\boldsymbol{\mathcal{C}}^{-1}\mathbf{y}_c] \tag{6}$$

where $\boldsymbol{\mathcal{C}} = \mathbf{I} + \mathbf{K}^T\mathbf{A}^{-1}\mathbf{K}$, so by decomposing as in [15]:

$$|\boldsymbol{\mathcal{C}}| = |\boldsymbol{\mathcal{C}}_{-i}|\ |1+\alpha_i^{-1}\mathbf{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i|, \tag{7}$$

where $\boldsymbol{\mathcal{C}}_{-i}$ denotes the value of $\boldsymbol{\mathcal{C}}$ with the i-th sample removed. Thus:

$$\boldsymbol{\mathcal{C}}^{-1} = \boldsymbol{\mathcal{C}}_{-i}^{-1} - \frac{\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i\mathbf{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}}{\alpha_i + \mathbf{k}_i^T\boldsymbol{\mathcal{C}}_i^{-1}\mathbf{k}_i}. \tag{8}$$

we have now expressed the quantity $\boldsymbol{\mathcal{C}}$ based on the contribution of an i-th sample. Now we can decompose the log-marginal likelihood as:

$$\mathcal{L}(\mathbf{A}) = \mathcal{L}(\mathbf{A}_{-i}) + \sum_{c=1}^{C}\frac{1}{2}\left[\log\alpha_i - log(\alpha_i+s_i) + \frac{q_{ci}^2}{\alpha_i+s_i}\right] \tag{9}$$

where we follow [15] in defining the "sparsity factor" s_i and also the new *multi-class* "quality factor" q_{ci} as:

$$s_i = \mathbf{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{k}_i \ \text{ and } \ q_{ci} = \mathbf{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}\mathbf{y}_c \tag{10}$$

The sparsity factor s_i defines the measure of overlap between a sample $\mathbf{k}_i$ and the ones already included in the model. That is, how much of the descriptive information of sample-i is already given from the existing samples. The quality factor q_{ci} measures how good the sample is in helping to describe a specific class. Thus, in an extension of the binary maximum solution proposed by [14], the descriptive quality of a sample is now assessed across classes.

Having decomposed the marginal likelihood into sample specific contributions we can seek the maximum with respect to an α_i. The only term that is a function of α_i is $l(\alpha_i)$ and the only difference, in that term, with its binary definition is the extra summation over classes and the multi-class factor q_{ci}. By setting the derivative $\frac{\partial\mathcal{L}(\mathbf{A})}{\partial\alpha_i} = 0$ we obtain the following stationary points:

$$\alpha_i = \frac{Cs_i^2}{\sum_{c=1}^{C} q_{ci}^2 - Cs_i}, \quad \text{if} \quad \sum_{c=1}^{C} q_{ci}^2 > Cs_i \tag{11}$$

$$\alpha_i = \infty, \quad \text{if} \quad \sum_{c=1}^{C} q_{ci}^2 \leq Cs_i \tag{12}$$

It can be easily shown that for the above stationary points the second derivative is always negative. Thus, those solutions which maximize the marginal likelihood provide the rules for inclusion of the sample in the model (11), removal (12) or scale update (11).

The quantity:

$$\theta_i = \sum_{c=1}^{C} q_{ci}^2 - Cs_i \tag{13}$$

defines the contribution of the i-sample to the marginal likelihood, in terms of how much additional descriptive information it provides to the model. Thus during each iteration, we must have a set $\mathcal{A}$ populated by M active samples for which $\theta_i > 0\ \forall\ i\ \in\ \mathcal{A}$. Otherwise, if a sample not in $\mathcal{A}$ has a positive θ it must be included in $\mathcal{A}$ and reversely if a sample with negative θ exists in $\mathcal{A}$ it must be excluded.

So during model training, the MAP update step (2) for regressors $\hat{\mathbf{W}}$ is modified to:

$$\hat{\mathbf{W}}_\star = \left(\mathbf{K}_\star\mathbf{K}_\star^T + \mathbf{A}_\star\right)^{-1}\mathbf{K}_\star\tilde{\mathbf{Y}}^T, \tag{14}$$

where $\mathbf{K}_\star \in \Re^{M\times N}$, and $\mathbf{A}_\star \in \Re^{M\times M}$,$M << N$ the matrices now with reduced dimensions. The training phase follows the consecutive updates of $\mathbf{A}$ from (11) or (12), $\mathbf{W}$ from (14) and $\mathbf{Y}$ from (3) and (4).

Finally, from (10) and (11) and given that $\boldsymbol{\mathcal{C}} = \mathbf{I}$ we initialize the scales α_i from:

$$\alpha_i^{initial} = \frac{C||\mathbf{k}_i||^2}{\sum_{c=1}^{C}||\mathbf{k}_i^T\mathbf{y}_c||^2/||\mathbf{k}_i||^2 - C} \tag{15}$$

B. Sample selection

The selection of each sample is based on its contribution to the marginal likelihood. During each iteration, we calculate the contributions $\theta_i\ \forall i \in \{1...N\}$. Then, if a sample which is not yet included in the model has a positive contribution, it is selected for the next iteration. Otherwise, if an already included sample is found to have a negative contribution, it is selected in order to be excluded. If none of the above criteria are met, we select a random sample from inside the model. This informative sample selection is not affected by problems with heavily skewed classes (as the contribution of samples from the under-represented class is by definition high) and leads to faster convergence, see Fig. 5.

C. Convergence criteria for mRVM$_1$

We consider two convergence criteria for mRVM$_1$. The first one labeled $conv_1$, directly follows [15] while the second, $conv_2$ is an extension of $conv_1$ that produces better results both in terms of class recognition and sparsity:

- $conv_1$ terminates the algorithm under three conditions:
  1) $\theta_i > 0\ \forall i\ \in\ \mathcal{A}$ (all included samples contribute).
  2) $\theta_i\ <\ 0\ \forall i\ \notin\ \mathcal{A}$ (all excluded samples do not contribute).
  3) $|\log A^{(\kappa)} - \log A^{(\kappa-1)}| < \epsilon$ (the scales $\mathbf{A}$ update from previous iteration is insignificantly small).
- $conv_2$ follows all the above three conditions with the addition of a minimum number of iterations:
  4) $\kappa_{min} = \lambda N_{train}$, where λ is a positive integer and N_{train} the size of the training set.

As it will be demonstrated in the 'Experiments' section, applying $conv_1$ generally leads to an early termination of the training phase, achieving suboptimal solutions with more relevant vectors than needed. On the other hand, the introduction of a minimum number of iterations κ_{min}, although an empirical termination condition, allows additional updates based on Step 5 of Algorithm 1 that lead to more reconsiderations of the θ_i contributions of active samples and thus a potentially improved solution. A detailed comparison of the two criteria across a variety of datasets will be provided.

D. Initialization

We can also employ a similar informative methodology for the selection of the first sample, upon the condition that the computation of each kernel function $\mathbf{k}_i$ at the beginning of the algorithm is computationally acceptable. Given that $\boldsymbol{\mathcal{C}} = \mathbf{I}$, we follow [15] by including the $\mathbf{k}_i$ with the largest projection to the auxiliary variables $\mathbf{Y}$ normalized by $||\mathbf{k}_i||^2$ and the number of classes C:

$$\theta_i^{initial} = \frac{\sum_{c=1}^{C} q_{ci}^2}{C s_i} = \frac{||\mathbf{k}_i \mathbf{Y}^T||}{C||\mathbf{k}_i||^2} \tag{16}$$

The above requires the computation of the kernel function for every training sample. If this is not feasible due to computational constraints, then a simple random selection of the first sample must be employed.

E. Computational complexity

During the fast multi-class type II ML procedure we perform two matrix inversions ($\mathcal{O}(M^3)$, where $M << N$) per training iteration. The first one is for the calculation of $\mathcal{C}_{-i}$ in order to derive the sparsity and quality factors s_i and q_{ci}. The second one is the posterior update step of the regressor parameters $\mathbf{W}$ from (14). Both of these calculations are based on the training kernel $\mathbf{K}$ so by following [15] we propose a methodology to avoid one of the two inversions. The sparsity and quality factors of all the observations of the training set are given by the following matrices:

$$\mathbf{S} = \mathbf{K}^T \boldsymbol{\mathcal{C}}^{-1} \mathbf{K} \text{ and } \mathbf{Q} = \mathbf{K}^T \boldsymbol{\mathcal{C}}^{-1} \mathbf{Y}^T \tag{17}$$

So if during a certain iteration of the training phase, the number of active samples in our model is M, the training kernel is $\mathbf{K}_\star\ \in\ \Re^{M \times N}$ and $\boldsymbol{\mathcal{C}} = \mathbf{I} + \mathbf{K}_\star^T \mathbf{A}_\star^{-1} \mathbf{K}_\star$. By utilizing the Woodbury identity we decompose (17) to:

$$\mathbf{S} = \mathbf{K}\mathbf{K}^T - \mathbf{K}\mathbf{K}_\star^T \left(\mathbf{K}_\star \mathbf{K}_\star^T + \mathbf{A}_\star\right)^{-1} \mathbf{K}_\star \mathbf{K} \tag{18}$$

$$\mathbf{Q} = \mathbf{K}\mathbf{Y}^T - \mathbf{K}\mathbf{K}_\star^T \left(\mathbf{K}_\star \mathbf{K}_\star^T + \mathbf{A}_\star\right)^{-1} \mathbf{K}_\star \mathbf{Y}^T \tag{19}$$

Where the quantity $\left(\mathbf{K}_\star \mathbf{K}_\star^T + \mathbf{A}_\star\right)^{-1}$ can be reused for the regressor $\mathbf{W}$ posterior update in (14), thus reducing the number of matrix inversions per iteration.

If a selected sample i is not included in the model, its sparsity and quality factors are directly derived from (18) because the matrix $\boldsymbol{\mathcal{C}}^{-1}$ is in fact $\boldsymbol{\mathcal{C}}_{-i}^{-1}$ so $s_i = S_i$ and $q_{ci} = Q_{ci}$. Otherwise, we must tune the above factors in order not to include the existing information given by sample i to the $\boldsymbol{\mathcal{C}}^{-1}$:

$$s_m = \frac{\alpha_m S_m}{\alpha_m - S_m} \tag{20}$$

$$q_{cm} = \frac{\alpha_m Q_{cm}}{\alpha_m - S_m} \tag{21}$$

Algorithm 1 mRVM$_1$ and the Fast Multi-class Type-II ML procedure

1: Initialize $\mathbf{Y}$ to follow target labels $\mathbf{t}$, set all $\alpha_i = \infty$.
2: Initialize model with a single sample and set α_i from (15).
3: **while** Convergence Criteria Unsatisfied **do**
4: **if** $\theta_i > 0$ and $\alpha_i < \infty$ **then**
5: Update α_i from (11) (sample already in the model).
6: **else if** $\theta_i > 0$ and $\alpha_i = \infty$ **then**
7: Set α_i from (11) (sample added in the model).
8: **else if** $\theta_i \leq 0$ and $\alpha_i < \infty$ **then**
9: Set $\alpha_i = \infty$ from (12) (sample deleted from the model).
10: **end if**
11: M-Step for $\hat{\mathbf{W}}_\star$: (14).
12: E-Step for $\mathbf{Y}$: (3) and (4)
13: Re-calculate $\theta_i\ \forall\ i \in \{1...N\}$
14: **if** $\exists\ \theta_j > 0$ for $j\ \notin\ \mathcal{A}$ **then**
15: find the $j \notin \mathcal{A}$ for which $\theta_j > \theta_n\ \forall n\ \notin\ \mathcal{A}$
16: **else if** $\exists\ \theta_j < 0$ for $j\ \in\ \mathcal{A}$ **then**
17: find the $j \in \mathcal{A}$ for which $\theta_j < \theta_n\ \forall n\ \in\ \mathcal{A}$
18: **else**
19: Set j = one random sample from $\mathcal{A}$
20: **end if**
21: Set $i = j$
22: **end while**

IV. mRVM$_2$

The training phase in mRVM$_2$ consists of subsequent updates of the parameters $\mathbf{A}$ $\mathbf{W}$ $\mathbf{Y}$ from (5), (2), (3) and (4). The only difference from the standard Expectation Maximization scheme is that we explicitly remove samples with scales α_{ic} large enough to "switch off" their respective regressors w_{ic}. In other words, if for the i-th sample we have $\alpha_{ic} > 10^5 \ \forall \ c \in \{1, ..., C\}$ then it is removed from $\mathcal{A}$.

mRVM$_2$ follows a 'top-down' approach by loading the whole training kernel into memory and iteratively removing insignificant samples. Although relatively more expressive than mRVM$_1$ because each sample i has a different scale α_{ic} across classes, if mRVM$_2$ prunes a sample it can not be reintroduced in the model.

Convergence criteria for mRVM$_2$

For mRVM$_2$ we used two different convergence criteria:

- $conv_A$ that terminates the model when $|\log A^{(k)} - \log A^{(k-1)}| < \epsilon$ (insignificant change in the scales $\mathbf{A}$)
- $conv_N$ when the number of iterations are λN_{train}.

The intuition behind $conv_A$ is that we stop model training when we have insignificant changes in the hyperpriors that control the sparsity, thus the relevant vectors of the model. The second measure $conv_N$ is an alternative termination decision that (as $conv_2$ in mRVM$_1$) is found to yield better results.

Algorithm 2 mRVM$_2$

```
while Convergence Criteria Unsatisfied do
    E-Step for α_ic ∀ i ∈ A and c ∈ {1,...,C}: (5).
    if ∃ i for which a_ic > 10^5 ∀ c ∈ {1,...,C} then
        Remove i from A
        Prune w_i, k_i α_i
    end if
    M-Step for Ŵ_⋆ : (14)
    E-Step for Y : (3) and (4)
end while
```

V. Quadrature approximation

As mentioned previously, the estimation of (1) can not be computed analytically, so a numerical estimation must be employed, like the Monte Carlo estimation [6] [5] with sampling of the random variable u. In the present work, we follow a different approach using a Quadrature approximation. As we take the expected value of (1) and for the random variable u we have $u \sim \mathcal{N}(0, 1)$, we can write (1) as:

$$P\left(t_n = i|\mathbf{W}, \mathbf{k}_n\right) = \mathcal{E}_{p(u)}\left\{\mathcal{F}(u)\right\} = \frac{1}{\sqrt{2\pi}} \int \mathcal{F}(u)\, e^{-u^2}\, du \tag{22}$$

Where the quantity e^{-u^2} is the standard Gauss-Hermite weight function $W(x)$. Typically, 2 roots are enough for a good approximation and provide as accurate results as the previous Monte Carlo simulation. The advantage of this methodology is that it is computationally faster than sampling.

VI. Predictive Likelihood

In most cases, apart from a high recognition rate, we are also interested that our model has an acceptable prediction confidence i.e we want our class membership probabilities to be as much diverged from a random guess as possible. In order to measure that characteristic, we define the *predictive likelihood* as the quantity $\mathcal{P}_1$ derived from the logarithm of the probability p_{nc} of a sample n belonging to the correct class c specified by our target label during the model training:

$$\mathcal{P}_1 = \log p_{nc} \tag{23}$$

The predictive likelihood measures the model confidence for the prediction of the 'correct' (based on the target label) class ignoring all other class memberships.

VII. Illustration using an artificial dataset

In this section we will demonstrate the two models in an artificial problem. This dataset was designed specifically to illustrate the behavior of mRVM$_1$ and mRVM$_2$ in terms of sample selection and level of sparsity. This toy example consists of $N = 140$ observations that belong to $C = 3$ different classes, represented as "asterisks" ($*$), "dots" ($\bullet$) and "crosses" ($+$). The data was randomly generated by five different Gaussian distributions.

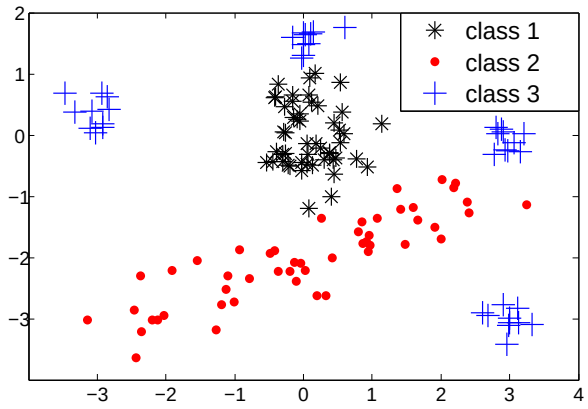


Fig. 2: A toy dataset with three classes.

In Fig. 3, 4 we illustrate the progression of each model into sample selection. Circles around a point mark it as a "relevant vector" while the labels on the decision boundaries represent which classes they separate. We selected three random iterations from the beginning, the middle and the end of the training phase. In Fig. 3 we see that mRVM$_1$ starts by constructing an initial collection of relevant vectors and then prunes out the unnecessary ones, while in Fig 4 mRVM$_2$ begins with the whole training set and proceeds in a deconstructive manner.

In Fig. 5 we show the advantage of informative sample selection in mRVM$_1$ versus random choice. We monitor the number of relevant vectors per iteration across a 10 times the size of the training set run. We see that using informative sample selection our model quickly converges to the actual solution where otherwise, the model does not reach the optimal solution even after reaching the maximum iterations.

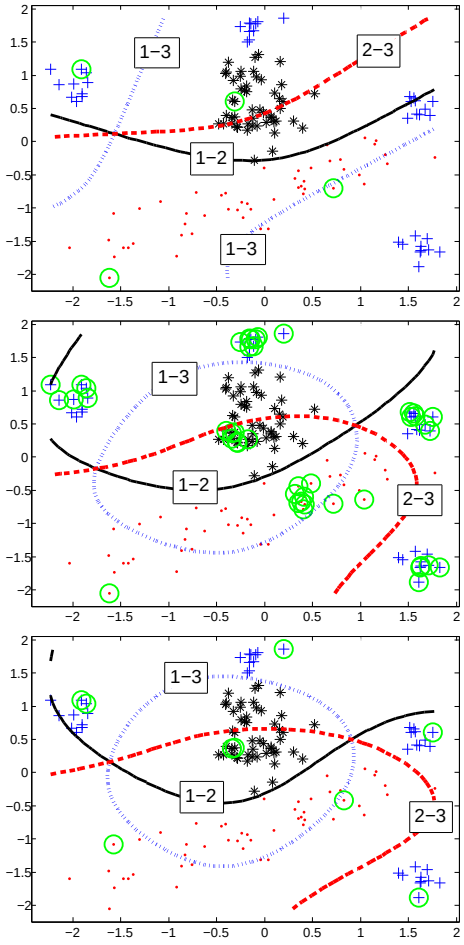


Fig. 3: The sample selection scheme of mRVM$_1$.

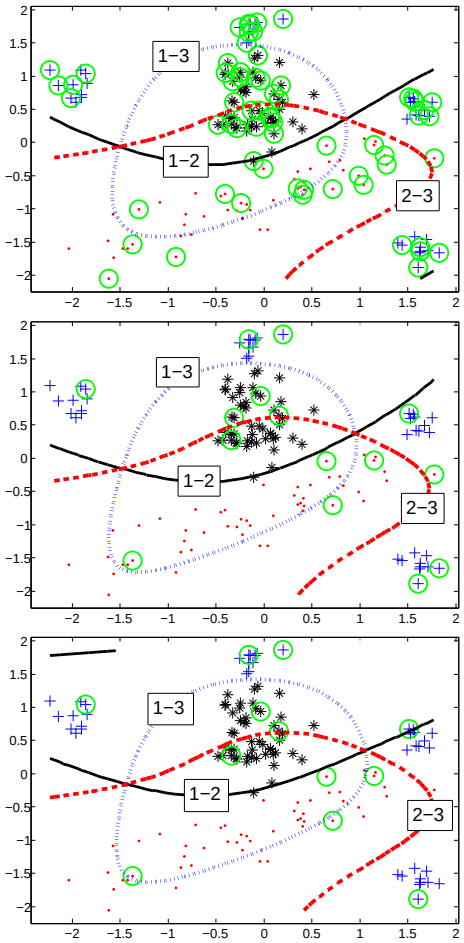


Fig. 4: The sample selection scheme of mRVM$_2$.

VIII. Experiments

A. Set-up

The study on mRVMs involved large scale experimentation on a range of different data-sets, which we selected in order to test the models on a variety of real world problems. Our source for the data-sets, apart from 'Crabs' [18], was the University of California Irvine (UCI) Machine Learning Repository [17].

TABLE I: Datasets used for experimentation

Dataset	*N*	*C*	*D*	Kernel used
Breast Cancer	569	2	30	Gaussian
Ecoli	336	8	7	Gaussian
Glass	214	6	9	Polynomial
Haberman	306	2	3	Gaussian
Ionosphere	351	2	34	Polynomial
Iris	150	3	4	Gaussian
Liver	345	2	6	Polynomial
Parkinsons	195	2	22	Polynomial
Pima	768	2	8	Gaussian
Wine	178	3	13	Linear
Soybean(small)	47	4	35	Linear
Vehicle	846	4	18	Polynomial
Balance	625	3	4	Polynomial
Crabs	200	4	5	Linear

For each of the above data-sets, we ran both mRVM$_1$ and mRVM$_2$ by performing a 10 times 10 fold cross-validation procedure, in order to minimize any result variance produced by 'improper' folds. As our models do not employ any automated kernel parameter learning scheme, we selected a

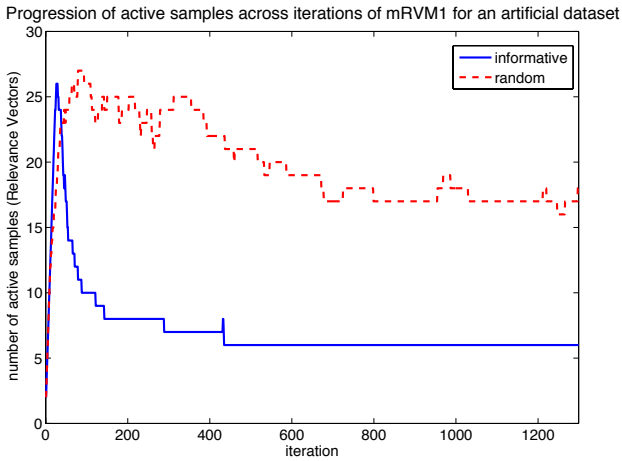


Fig. 5: We demonstrate the advantage of our informative sample selection scheme against random choice, in terms of algorithm convergence.

bandwidth of $1/D$ for the Gaussian kernels as employed in previous works [10]. Cross-validation or gradient ascent methods may be used to further improve performance but its beyond the scope of this paper and interesting future directions. During each run, we monitored:

- the % accurate recognition rate i.e the number of correctly classified test samples versus the size of the test set.
- the marginal likelihood for mRVM$_1$ and the joint likelihood for mRVM$_2$.
- the model confidence, captured as the predictive likelihood of the class membership probabilities.
- the number of relevant vectors, i.e active samples used for the model training.
- the number of iterations upon which each convergence condition was satisfied.
- other auxiliary quantities, such as the % change in the model parameters W, Y and A.

The models were implemented using MATLAB and the scripts [2] were run on a 416 core compute cluster.

B. Results of mRVM$_1$

In this section we provide the results for each dataset, in terms of % accurate recognition rate and number of relevance vectors for each of the proposed convergence measures. In Table II the second column of the table specifies the % accurate recognition rate when the model has reached the maximum number of iterations (in our experiments is 6 times the size of the training set), while the third and fourth columns represent the accuracies achieved by the two convergence measures described in the relevant section (for $\lambda = 1$). Additionally, in Table III we provide the number of relevant vectors, again for each of the termination criteria we described. With bold text we depict the top performance achieved that in most cases is not statistically significant due to large variance of the cross validation scheme.

In the graphs, we monitor the % accurate recognition rate, the predictive likelihood $\mathcal{P}_1$, the log-marginal likelihood and the number of relevance vectors. The points where we achieve convergence are identified by the symbols '1' and '2', for $conv_1$ and $conv_2$ respectively. The horizonal axis in the graphs represents the number of iterations during the training phase.

TABLE II: % recognition rate of mRVM$_1$

Dataset	max it	$conv_1$	$conv_2$
Breast c.	97.07 ± 0.85	**97.54 ± 1.98**	97.29 ± 2.04
Ecoli	83.33 ± 2.56	83.48 ± 5.99	**83.76 ± 5.99**
Glass	64.14 ± 3.68	**64.19 ± 8.57**	64.10 ± 9.02
Haberman	75.10 ± 2.45	74.63 ± 8.09	**75.23 ± 7.66**
Ionosphere	90.14 ± 1.34	89.74 ± 4.63	**90.17 ± 4.72**
Iris	93.47 ± 1.74	93.33 ± 6.77	**93.80 ± 6.01**
Liver	**58.85 ± 2.21**	58.65 ± 7.94	58.82 ± 8.03
Parkinsons	**84.63 ± 2.39**	83.79 ± 8.78	84.58 ± 8.57
Pima	77.11 ± 1.72	**77.17 ± 4.38**	77.14 ± 4.09
Wine	**96.00 ± 1.86**	95.71 ± 4.72	95.94 ± 4.71
Soybean	89.25 ± 5.53	88.25 ± 19.93	**91.75 ± 16.30**
Vehicle	**73.82 ± 1.42**	73.07 ± 4.47	73.77 ± 4.93
Balance	**96.63 ± 0.53**	92.35 ± 3.52	95.03 ± 3.12
Crabs	94.70 ± 1.75	94.49 ± 5.78	**94.80 ± 5.71**

[2] Available in http://www.dcs.gla.ac.uk/inference/pMKL

TABLE III: Number of relevant vectors mRVM$_1$

Dataset	N_{train}	max it.	$conv_1$	$conv_2$
Breast c.	513	4 ± 0	9 ± 5	5 ± 1
Ecoli	303	7 ± 0	16 ± 7	9 ± 5
Glass	193	7 ± 0	13 ± 3	9 ± 1
Haberman	276	4 ± 0	10 ± 3	5 ± 1
Ionosphere	316	9 ± 0	17 ± 5	10 ± 2
Iris	135	4 ± 0	8 ± 2	5 ± 1
Liver	311	2 ± 0	3 ± 1	2 ± 1
Parkinsons	176	6 ± 0	10 ± 3	7 ± 1
Pima	692	8 ± 0	16 ± 4	8 ± 1
Wine	161	3 ± 0	5 ± 2	3 ± 1
Soybean	43	3 ± 0	5 ± 2	4 ± 2
Vehicle	762	14 ± 1	38 ± 15	15 ± 3
Balance	563	8 ± 0	13 ± 5	8 ± 1
Crabs	180	4 ± 1	5 ± 2	4 ± 2

C. Results of mRVM$_2$

Similarly to mRVM$_1$, in Table IV we demonstrate the predictive power and in Table V the sparsity inducing capabilities of mRVM$_2$ across different datasets. In the graphs we monitor the % accurate recognition rate, the predictive likelihood $\mathcal{P}_1$, the log-joint likelihood and the number of relevance vectors. The points where we achieve convergence are identified by the symbols 'A' and 'N', for each of the criteria $conv_A$ and $conv_N$ described in the mRVM$_2$ model formulation section.

TABLE IV: % recognition rate of mRVM$_2$

Dataset	max it	$conv_A$	$conv_N$
Breast c.	97.07 ± 0.55	**97.20 ± 2.13**	97.14 ± 0.72
Ecoli	84.73 ± 2.98	**85.00 ± 6.22**	84.85 ± 2.66
Glass	**67.49 ± 2.33**	67.21 ± 27.10	67.37 ± 2.38
Haberman	74.97 ± 2.13	**75.34 ± 7.78**	74.87 ± 2.45
Ionosphere	90.49 ± 1.88	**90.63 ± 4.60**	90.54 ± 1.32
Iris	93.80 ± 1.75	93.87 ± 6.04	**93.87 ± 1.80**
Liver	68.71 ± 3.10	68.65 ± 7.79	**68.74 ± 3.11**
Parkinsons	**84.11 ± 1.31**	83.95 ± 7.34	84.00 ± 2.12
Pima	77.13 ± 1.47	**77.22 ± 4.86**	77.18 ± 1.53
Wine	95.94 ± 1.02	96.24 ± 4.77	**96.24 ± 0.97**
Soybean	96.50 ± 2.11	96.21 ± 9.01	**97.00 ± 1.58**
Vehicle	75.88 ± 2.03	76.26 ± 5.08	**76.30 ± 1.72**
Balance	**92.71 ± 0.69**	92.26 ± 3.52	92.63 ± 0.69
Crabs	**94.85 ± 1.33**	93.70 ± 5.53	93.85 ± 1.55

TABLE V: Number of relevant vectors mRVM$_2$

Dataset	N_{train}	max it.	$conv_A$	$conv_N$
Breast c.	513	7 ± 0	10 ± 2	8 ± 0
Ecoli	303	11 ± 0	11 ± 1	11 ± 0
Glass	193	11 ± 1	11 ± 6	11 ± 1
Haberman	276	5 ± 0	6 ± 1	5 ± 0
Ionosphere	316	12 ± 0	13 ± 2	13 ± 1
Iris	135	6 ± 0	6 ± 1	6 ± 0
Liver	311	5 ± 0	5 ± 1	5 ± 0
Parkinsons	176	9 ± 0	9 ± 2	9 ± 0
Pima	692	11 ± 1	13 ± 2	12 ± 1
Wine	161	5 ± 0	5 ± 1	5 ± 0
Soybean	43	5 ± 0	6 ± 1	6 ± 0
Vehicle	762	36 ± 1	41 ± 3	38 ± 1
Balance	563	14 ± 0	15 ± 2	14 ± 0
Crabs	180	8 ± 0	9 ± 1	9 ± 0

D. Result interpretation of mRVM$_1$

As mentioned previously, mRVM$_1$ incrementally builds up the training kernel based on each sample contribution. It can be seen from Fig. 6, 7 and 8, that during the initial iterations there is a massive build-up in the training kernel. At this point, the quality factor q_{ci} of the samples plays the most important role to the contribution θ_i, because in an initially low populated $\mathcal{A}$ the descriptive overlap s_i is small due to the low dimensionality of $\boldsymbol{\mathcal{C}}$. Thus, any sample which can describe a class is included during this phase.

Then the model reaches a point that is illustrated as the "peak" in the number of relevance vector diagram, where all class descriptive samples have been included. From this point on, the matrix $\mathcal{C}$ reaches its highest dimensionality thus the sparsity factor s_i becomes significant and samples which provide redundant descriptive information are pruned out. Because in some cases a sample contributes to the solution in combination with another, there might be some small fluctuations in the % recognition rate until the best sample combination is correctly tuned. This particular region, around the peak of the relevant vectors graph is usually where dataset dependent phenomena are observed, e.g small peaks or fluctuations in the predictive or marginal likelihood. The model continues to prune out samples, until it reaches a very small subset of the training set, sometimes with size the same as the number of classes.

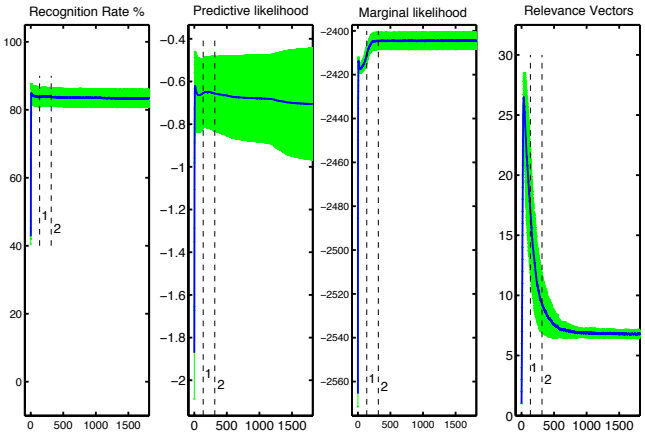


Fig. 6: Results of mRVM$_1$ for Ecoli dataset.

In terms of convergence, $conv_1$ terminates the model training when a first 'good' solution is found (conditions 1+2 of the 'Convergence' section), as it evaluates the change in scales $\mathbf{A}$ only based on the previous iteration (condition 3). On the other hand, $conv_2$ is less prone in falling into local maxima, as it imposes more updates on the scales of active samples (condition 4), changing the overall solution if some these observations turn out to be uniformative. It can be seen from the mRVM$_1$ result tables II, III that $conv_2$ generally gives better solutions, both recognition and sparsity-wise.

There are also datasets for which we have a fall in the confidence as defined by the predictive likelihood $\mathcal{P}_1$ such as Ecoli in Fig. 6. For those datasets, we do not have only a trade-off between sparsity and accuracy but also between sparsity and model confidence. For those datasets, during the initial relevance vector collection build-up the predictive likelihood increases until a certain problem dependent point, which is not the same as the one where we have maximum number of active samples. It is important to mention that this fall of the mean predictive likelihood does not compromise the predictive power of the model, i.e does not neither align with a significant fall in the % accurate recognition rate nor represents a significant fall in the membership probability for the correct class. The dominant characteristic for problems with decreasing predictive likelihood is the significant variance

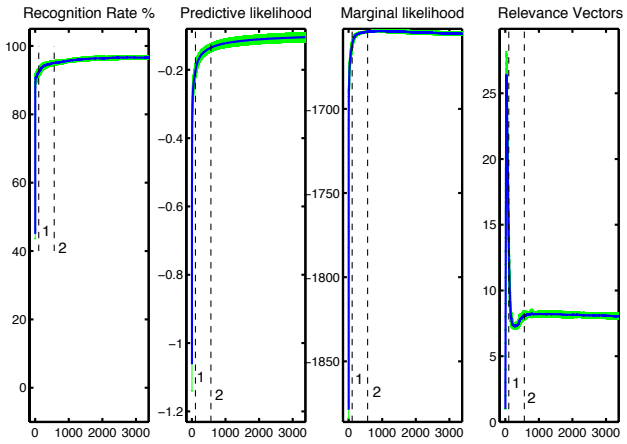


Fig. 7: Results of mRVM$_1$ for Balance dataset.

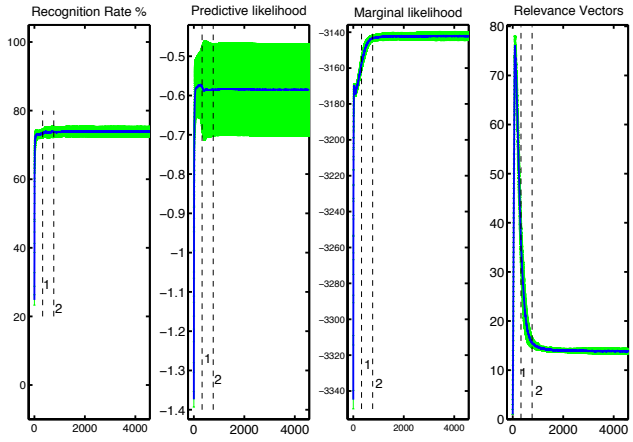


Fig. 8: Results of mRVM$_1$ for Vehicle dataset.

of the model confidence.

E. Result intepretation of mRVM$_2$

For mRVM$_2$, it can be seen from Fig. 9, 10 and 11 that during the initial iterations the sample removal is cataclysmic and the model ends up very early to the final solution. So in contrast to mRVM$_1$, mRVM$_2$ speeds up as the training phase progresses (due to the decreasing dimensionality of the training kernel) and the final collection of samples is built up in considerably less iterations. Another difference from mRVM$_1$ is that the performance of mRVM$_2$ in terms of % accurate recognition rate is more stable across iterations. This is very natural for the earlier iterations, as the algorithm starts with the whole training set while during the subsequent iterations, we see only small fluctuations of test accuracy, for example in 'Ionosphere' (see Fig. 11) or 'Parkinsons' (see Fig. 9).

It can be seen from the result tables and graphs that for the maximum number of iterations mRVM$_1$ leads to more sparse solutions than mRVM$_2$ and more confident predictions. From the perspective of a generative model, mRVM$_1$ builds the class membership distribution more sharply peaked around a small collection of prototypical samples. Thus, points which are near to the probability mass are classified with more confidence (higher probability of class membership) in contrast to samples

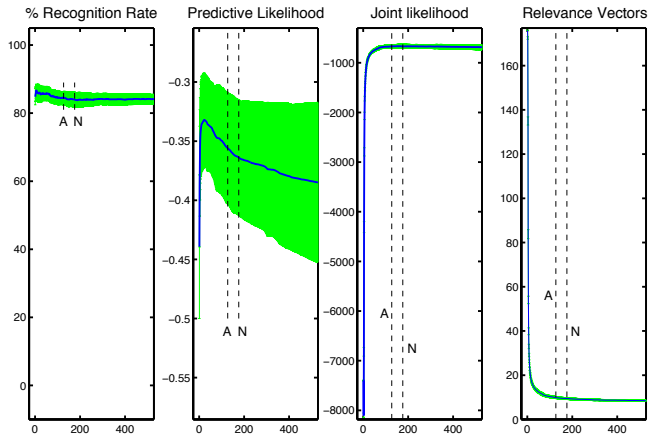


Fig. 9: Results of mRVM_2 for Parkinsons dataset.

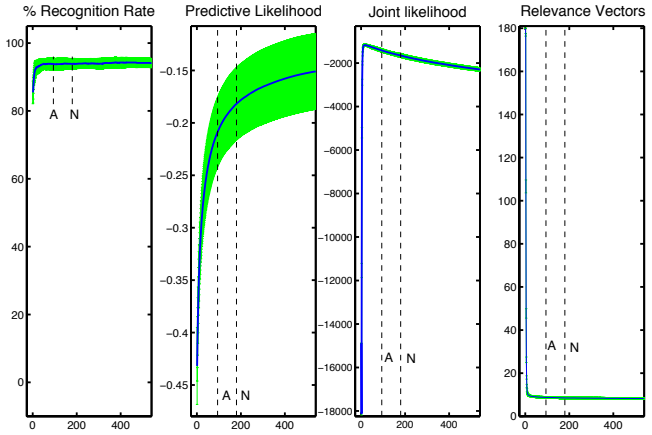


Fig. 10: Results of mRVM_2 for Crabs dataset.

which are near to the class boundary. So, although we have very confident predictions for samples which are very typical to their class, more outlying observations are more prone to being misclassified, due to the significantly low probability mass in their area. On the other hand, mRVM_2 keeps a larger subset of training samples, which spread the class conditional probability mass to a larger area. This leads to better identification of boundary samples, as they take a significantly higher class membership probability than mRVM_1, but with lower prediction confidence.

F. Solution Stability

The solution produced by each of the two sparse models is a collection of relevant vectors which describe each class of the problem. In this section we discuss the stability of that solution, i.e if the prototypical observations identified by our models appear frequently across individual runs. As mentioned in the experiments section we performed a 10 times 10 fold cross validation procedure, so we study the appearance of each observation to 100 solutions for each model. Although sometimes a sample may appear in the test set due to our cross validation scheme, it is more important to assess the importance of an prototypical sample when belonging to different training sets of the same problem rather than perform multiple runs of the same training set.

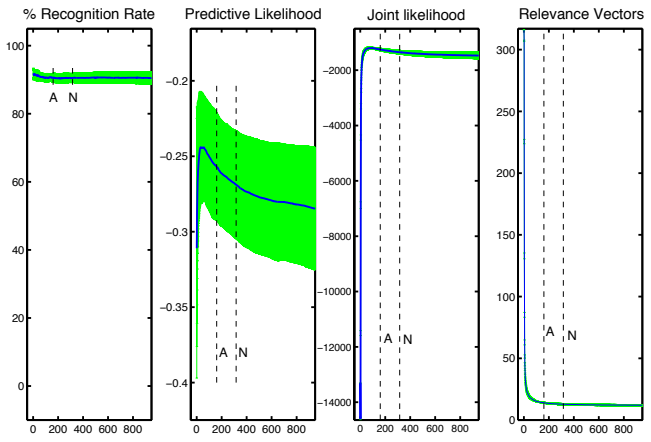


Fig. 11: Results of mRVM_2 for Ionosphere dataset.

In Fig. 12 we see the histogram where the horizontal axis represents the indices of samples and the vertical bar the number of occurrences of each sample in our model solution, for maximum number of iterations. The dashed vertical lines represent the class boundary. It can be seen that mRVM_1 holds a smaller number of relevant vectors and has better identification properties as the same samples appear more frequently in the solution. On the other hand, mRVM_2 has smaller prototypical identification power as nearly the majority of samples appear at least a couple of times in the model solution. Similar observations occurred in other datasets from our collection.

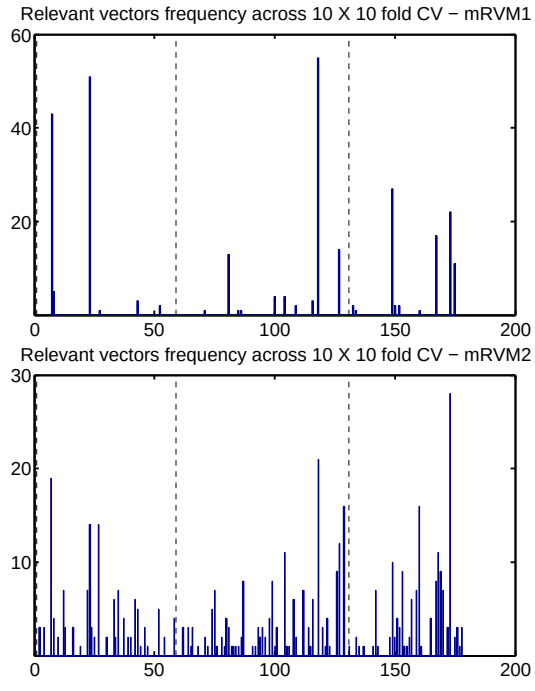


Fig. 12: Solution stability for the Wine dataset.

TABLE VI: Comparison with the non-sparse Expectation Maximization model

	mRVM$_1$		mRVM$_2$		E-M	
Dataset	% recognition rate	RVs used	% recognition rate	RVs used	% recognition rate	N$_{train}$
Breast c.	**97.54 ± 1.98**	9 ± 5	97.20 ± 2.13	10 ± 2	96.96 ± 1.89	513
Ecoli	83.76 ± 5.99	9 ± 5	85.00 ± 6.22	11 ± 1	**85.76 ± 6.23**	303
Glass	64.19 ± 8.57	13 ± 3	67.49 ± 2.33	11 ± 1	**70.00 ± 13.48**	193
Haberman	75.23 ± 7.66	5 ± 1	**75.34 ± 7.78**	6 ± 1	73.33 ± 7.37	276
Ionosphere	90.17 ± 4.72	10 ± 2	90.63 ± 4.60	13 ± 2	**93.14 ± 5.75**	316
Iris	93.80 ± 6.01	5 ± 1	**93.87 ± 1.80**	6 ± 0	93.33 ± 5.44	135
Liver	58.85 ± 2.21	2 ± 0	**68.74 ± 3.11**	5 ± 0	68.53 ± 7.73	311
Parkinsons	84.63 ± 2.39	6 ± 0	84.11 ± 1.31	9 ± 0	**89.47 ± 6.56**	176
Pima	77.17 ± 4.38	16 ± 4	**77.22 ± 4.86**	13 ± 2	75.79 ± 5.20	692
Wine	96.00 ± 1.86	3 ± 0	**96.24 ± 0.97**	5 ± 0	95.88 ± 3.97	161
Soybean	91.75 ± 16.30	4 ± 2	97.00 ± 1.58	6 ± 0	**97.50 ± 7.91**	43
Vehicle	73.82 ± 1.42	14 ± 1	**76.30 ± 1.72**	38 ± 1	75.95 ± 5.38	762
Balance	**96.63 ±0.53**	8 ± 0	92.71 ± 0.69	14 ± 0	95.00 ± 3.60	563
Crabs	94.80 ± 5.71	4 ± 2	**94.85 ± 1.33**	8 ± 0	86.50 ± 7.09	180

IX. Competing methods

In this section we compare the performance of our sparse models against published results from other machine learning algorithms: the standard non-sparse Expectation-Maximization (E-M) model, the Variational Bayes approximation for the kernel-based multinomial probit likelihood model (VBpMKL) [4], the K-nearest neighbors (KNN) classifier [10] along with its probabilistic version PK-nn [10]. Similarly to mRVMs, we followed a 10 times 10 fold cross-validation methodology. In Tables VI, VIII and VII we can see that our models produce very competitive results using only a fraction of the original training set while possessing prototypical sample identification capabilities.

TABLE VII: Results comparison against K-nearest neighbors methods [10]

Dataset	mRVM$_1$	mRVM$_2$	K-nn	PK-nn
Glass	64.19 ± 8.57	67.49 ± 2.33	70.09 ± 9.22	**73.33 ± 8.81**
Iris	93.80 ± 6.01	93.87 ± 1.8	94.67 ± 5.25	**96 ± 5.62**
Crabs	94.80 ± 5.71	**94.85 ± 1.33**	85 ± 8.82	80.5 ± 6.85
Pima	77.17 ± 4.38	**77.22 ± 4.86**	73 ± 8.88	76 ± 14.68
Soybean	91.75 ± 16.30	**97.00 ± 1.58**	85.5 ± 16.74	95.5 ± 9.56
Wine	96.00 ± 1.86	96.24 ± 0.97	96.08 ± 3.77	**96.63 ± 2.89**
Balance	**96.63 ± 0.53**	92.71 ± 0.69	88.48 ± 2.99	89.77 ± 3.02
Liver	58.85 ± 2.21	**68.74 ± 3.11**	66.4 ± 6.98	63.74 ± 12.93
Vehicle	73.82 ± 1.42	**76.30 ± 1.72**	63.72 ± 5.16	62.78 ± 4.53

TABLE VIII: Results comparison against the Variational Bayes method VBpMKL [4]

Dataset	mRVM$_1$	mRVM$_2$	VBpMKL
Balance	**96.63 ± 0.53**	92.71 ± 0.69	93 ± 3.3
Crabs	94.80 ± 5.71	**94.85± 1.33**	86.5 ± 8.2
Glass	64.19 ± 8.57	67.49 ± 2.33	**72.1 ± 10.1**
Iris	93.80 ± 6.01	93.87 ± 1.80	**97.3 ± 5.6**
Soybean	91.75 ± 16.30	**97.00 ± 1.58**	95.16 ± 8.4
Vehicle	73.82 ± 1.42	**76.30 ± 1.72**	74.4 ± 4
Wine	96.00 ± 1.86	96.24 ± 0.97	**98.9 ± 2.3**

X. Conclusion

In this work we introduced and provided the theoretical background of the two multi-class multi-kernel Relevance Vector Machines, focusing on their multi-class discrimination aspect. Additionally, we proposed a collection of methodologies that boost the performance of mRVM$_1$ both in terms of computational complexity and discrimination power. Following wide experimentation on real world datasets, we showed that mRVM$_1$ has better prototypical sample identification properties and leads to more confident predictions. On the other hand, mRVM$_2$ is more accurate in terms of predictive power and has better outlier detection capabilities. Using the fast type-II ML procedure, mRVM$_1$ allows the incremental building of the training kernel, making the method very suitable for large scale problems. From the other hand, the assumption of a common scale α across classes makes the model less expressive than mRVM$_2$, providing lower class recognition rates. In terms of sparsity, our experiments showed that we can retain a significant amount of class recognition accuracy, using only a small fraction of the overall training set, *sometimes with size the same as the number of classes*.

mRVMs have the profound advantage of introducing sparsity to the multi-class setting, with all the well recognized properties of the original Relevance Vector Machine and Bayesian probabilistic models in general. Extensions to the binary type-II ML RVM such as the smoothing prior proposed in [12] and further adopted in [16] can be now applied to the multi-class setting for regression problems. As a future work, it will be very interesting to extend mRVMs to the joint feature and sample sparsity setting so that our solution can not only identify prototypical class observations, but also the most important sample features. As mRVMs have multi-kernel adaptation capabilities, it would be an interesting starting point to map the observations $\mathbf{X} \in \Re^{N\times D}$ into D number of kernels, one for each feature. Then, assuming the process in computationally scalable for large problems, using the informative kernel combination scheme proposed in [6] infer the important features for each sample while at the same time prune insignificant samples. Additionally, a very useful extension to the mRVMs would be a kernel parameter learning scheme, as shown in [16]. In the case of large scale applications in high performance computing, scalability can be improved by adapting the incremental formulae provided in [15]. Finally, a very interesting area of research would be to evaluate the qualitative properties of mRVM solutions in terms of the problem context. For example, for Bioinformatics problems, prototypical sample identification might be more interesting than sheer class recognition accuracy.

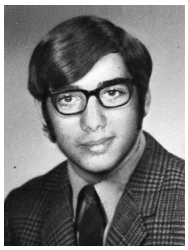
Acknowledgment

Ioannis Psorakis conducted his MSc in the Inference Research Group and acknowledges technical support and residence. In addition he acknowledges Microsoft Research for

supporting his PhD work. Theodoros Damoulas was supported by a scholarship grant awarded by NCR Financial Solutions Group Ltd and acknowledges the help and support of NCR Labs and especially Dr. Gary Ross and Dr. Chao He. In addition he acknowledges funding from the NSF Expeditions in Computing grant on Computational Sustainability (Award Number 0832782). Mark A. Girolami is supported by an EPSRC Advanced Research Fellowship (EP/E052029/1).

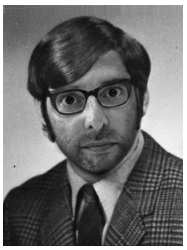
REFERENCES

[1] J. H. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *Journal of American Statistical Association*, vol. 88, pp. 669-679, June 1993.
[2] E. Alpaydin, *Introduction to Machine Learning*, Cambridge, MA: The MIT Press, 2004.
[3] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer, 2006.
[4] T. Damoulas and M. A. Girolami, "Combining feature spaces for classification," *Pattern Recognition*, vol. 42, no. 11, pp. 2671-2683, Nov. 2009.
[5] T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection," *Bioinformatics*, vol. 24, no. 10, pp. 1264-1270, 2008.
[6] T. Damoulas, M. A. Girolami, Y. Ying, and C. Campbell, "Inferring Sparse Kernel Combinations and Relevance Vectors: An application to subcellular localization of proteins," in *Proceedings of the 7th International Conference on Machine Learning and Applications*, San Diego, CA, Dec. 2008, pp. 577-582.
[7] M. A. Girolami and S. Rogers, "Hierarchic Bayesian models for kernel learning," in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, Aug. 2005, pp. 241-248.
[8] A.C. Faul and M. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems 14: proceedings of the 2001 conference*, pp. 383–389, Cambridge MA: The MIT Press, 2002.
[9] N. D. Lawrence and R. Herbrich, "A sparse Bayesian compression scheme - the informative vector machine," in *Neural Information Processing Systems Workshop on Kernel Methods*, Vancouver CA, Dec. 2001
[10] S. Manocha and M. A. Girolami, "An empirical analysis of the probabilistic k-nearest neighbour classifier," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1818-1824, Oct. 2007.
[11] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169-186, Sept. 2003.
[12] A. Schmolck and R. Everson, "Smooth relevance vector machine: a smoothness prior extension of the RVM," *Machine Learning*, vol. 68, no. 2, pp. 107135, August 2007.
[13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267-288, 1994.
[14] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211-244, Sept. 2001.
[15] M. Tipping and A. Faul, "Fast marginal likelihood maximization for sparse Bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, Florida, Jan. 2003, pp. 3-6.
[16] D. Tzikas, A. Likas, and N. Galatsanos, "Sparse Bayesian modeling with adaptive kernel learning," *IEEE Trans. Neural Networks*, vol. 20, no. 6, pp. 926-937, June 2009.
[17] University of California Irvine Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml/index.html
[18] Crabs dataset. [Online]. Available: http://www.ens.gu.edu.au/STATS/env2291/DATASETS/CRABS/crabs.htm



Ioannis Psorakis is a Microsoft Research sponsored PhD student at the Department of Engineering Science of Oxford University (UK), working under the supervision of Prof Stephen Roberts and Prof Ben Sheldon. He received his Engineering degree from the Department of Production Engineering and Management, Technical University of Crete (Greece) and his MSc (Distinction) from the Department of Computing Science, University of Glasgow (UK).

Theodoros Damoulas is a Postdoctoral Associate in the Faculty of Computing and Information Science at Cornell University (USA). In 2009 he completed his PhD at the Department of Computing Science, University of Glasgow (UK) where he was a member of the Inference Research Group. He holds an M.Eng (1st Class) in Mechanical Engineering from the University of Manchester (UK) and an M.Sc in Informatics (Distinction) from the University of Edinburgh (UK).



Mark A. Girolami is Professor of Statistics in the Department of Statistical Science at University College London. He was awarded an EPSRC Advanced Research Fellowship in 2007. He obtained his PhD in 1998 with a thesis on Independent Component Analysis (ICA), there are currently in excess of 1500 citations to the papers published during his PhD studies and in 2009 he was awarded a Pioneer Award from SPIE (International Society of Photo-Optical Engineers) for the impact his contributions to ICA has had on advancing neuro-imaging technology.