

# Learning Multi-category Classification in Bayesian Framework

Atul Kanaujia and Dimitris Metaxas

CBIM, Rutgers University  
{kanaujia, dnm}@cs.rutgers.edu

**Abstract.** We propose an algorithm for Sparse Bayesian Classification for multi-class problems using Automatic Relevance Determination(ARD). Unlike other approaches which treat multiclass problem as multiple independent binary classification problem, we propose a method to learn the multiclass predictor directly. The usual approach of “one against rest” and “pairwise coupling” are not only computationally demanding during training stage but also generates dense classifiers which have greater tendency to overfit and have higher classification cost. In this paper we discuss the algorithmic implementation of Multiclass Classification model and compare it with other multi-class classifiers. We also empirically evaluate the classifier on viewpoint learning problem using features extracted from human silhouettes. Our experiments show that our algorithm generates sparser classifiers, with performance comparable to state-of-the-art multi-class classifier.

## 1 Motivation and Related Work

Classification is a task of inferring a set of known or unknown classes based on some similarity measure, to explain an observed set of data points. Many supervised algorithms exist for classification ranging from simplest Nearest Neighbor, pairwise linear classifiers to complex RBF Networks, MLP, Tangent Distance Classifier(TDC) and Optimal Margin classifiers. However most of the classification algorithms are designed for binary classification problems. The multiclass classification can be decomposed into several independent binary classification problems. Classical approach for this decomposition had been *one against rest* and *pairwise coupling* proposed by Hastie et. al.[1]. Dietterich [2] suggested a more general approach to multi-category classification using a coding matrix that associates each row of  $l$  columns to a class label  $y \in Y$ , where  $Y$  is set of labels and  $l$  are set of hypothesis. A binary classifier is run on each column and the prediction is made based on which row of the coding matrix is closest to  $l$  hypothesis. This approach is called *error correcting output codes*. Allwein et al.[3] discusses a unifying approach for reducing multiclass to binary problems for margin classifiers. Other extensions to multi-class problems have been applied by Breiman et al. [5] using decision tree learning and by Schapire and Freund, [16] as an extension for AdaBoost classification. These approaches although powerful

and accurate, however, fail to capture relationship between different classes. The generated classifier is denser and has more tendency to overfit the training data.

A number of attempts have been made to directly approach the multi-class classification problem for optimal margin classifiers(SVM). These approaches extends the quadratic optimization for two classes to multiple classes by adding constraints for each class. The number of constraints grows exponentially with the number of classes. Bredensteiner et al.[6] and Weston [18] were among the first works on reducing multi-class learning problem to single large optimization problem.

There have been many works recently that attempt to solve this optimization in lesser time by breaking them into subproblems [4]. Tsochantaridis et al.[7] generalized the large margin method proposed by Weston et al.[18], to learning of structured response. Their algorithm is tunable to specific loss function and uses working set of active constraints that ensures sufficiently accurate solution.

However max-margin classifiers do not provide probabilistic measure for the predictions. Margin classifiers, although sparse, needs post-processing to get rid of unnecessary support vectors [17]. The smoothness parameters of margin classifiers have to be set by cross-validation.

Bayesian methods [10] do not possess above drawbacks. Sparse bayesian learning automatically embodies Occam's razor that penalizes complex models thereby smoothing the model. In this paper we propose an algorithm for learning sparse multi-category classifier in Bayesian framework as proposed by Mackay[10]. The algorithm uses multinomial distributions for multiple variables with one-of-all encoding for each class. The multiple outputs of the model is learnt as a kernel basis function with softmax as the canonical link function. The parameters are learnt using Automatic Relevance Determination(ARD). ARD is a model selection mechanism that ensures sparsity and smoothness.

To the best of our knowledge, this has not been attempted in past and our work provides complete algorithm to learn multiple class posterior probabilities directly in bayesian framework. Our work has three contributions: (1) We propose sparse bayesian classifiers for multi-class problems; (2) We empirically compare performance of our classifier with other algorithms for handling multi-class problems; (3) We use multi-class classifiers to infer viewing angle from features extracted from Human silhouettes. Section 2 gives a brief overview of the bayesian framework. In Section 3 we discuss the formulation of classification problem and our algorithm in detail. Section 4 discusses the experimental results. Theoretical proofs for convergence has been omitted from current discussion.

## 2 Bayesian Learning Framework

Bayesian learning intrinsically embodies regularization and model selection using Occam's razor[10] [8]. Bayesian learning is a three stage process. In the first stage the model is fit to the observed data by maximizing posterior distribution over the model parameters  $\theta$ .

$$P(\theta|D, \alpha, \beta, M) = \frac{P(D|\theta, \beta, M)P(\theta|\alpha, M)}{P(D|M)} \quad (1)$$

The normalizing constant is called the evidence of the model  $M$  and is not required for fitting a given model  $M$  to the data set  $D$ . The first term on right hand side(likelihood) is the loss function and second term(prior distribution) is the smoothing factor.  $\alpha$  and  $\beta$  are the scale parameters of these distributions. Taking the distributions as gaussians with appropriate normalization factor:

$$P(D|\theta, \beta, M) = \frac{e^{-\beta L_\theta(D)}}{(2\pi/\beta)^{N/2}} \quad (2)$$

$$P(\theta|\alpha, M) = \frac{e^{-\alpha P(\theta)}}{\int e^{-\alpha P(\theta)} d\theta} \quad (3)$$

where  $L_\theta$  is the loss function to be minimized and  $P(\theta)$  is the penalty term for penalizing complex models with larger  $|\theta|$  (smoothing). The posterior obtained is a joint function of scale parameters  $\alpha, \beta$  for the loss function and smoothing prior respectively. Given  $\alpha$  and  $\beta$ , most probable  $\theta_{MP}$  can be obtained by maximizing the posterior distribution (1). For maximum margin classifiers these parameters corresponds to error/margin tradeoff parameter 'C' and insensitivity parameter  $\epsilon$  [7] that have to be learnt using cross-validation. This is wasteful both for the data and computation.

Second stage of bayesian learning involves model selection by estimating the most probable scale parameters  $\alpha_{MP}$  and  $\beta_{MP}$  by maximizing the posterior distribution:

$$P(\alpha, \beta|D, M) \propto P(D|\alpha, \beta, M)P(\alpha|M)P(\beta|M) \quad (4)$$

For a given prior distributions of  $\alpha$  and  $\beta$ , maximizing (4) is equivalent to maximizing the evidence  $P(D|\alpha, \beta, M)$ . This evidence maximization procedure is called *Type II Maximum Likelihood* maximization and yields the equations for computing most probable  $\alpha_{MP}$  and  $\beta_{MP}$ .

The posterior of parameters  $\theta$  is approximated as

$$P(\theta|D, \alpha, \beta, M) \approx P(\theta|D, \alpha_{MP}, \beta_{MP}, M) \quad (5)$$

(5) and (1) can be used to estimate most probable  $\theta = \theta_{MP}$  (mode of the posterior distribution(1)) by substituting values for  $\alpha_{MP}$  and  $\beta_{MP}$ . The update equations for  $\theta_{MP}, \alpha_{MP}$  and  $\beta_{MP}$  can be used iteratively to estimate the model with maximum evidence.

The third stage of Bayesian Framework allows us to quatitatively rank different basis functions and the prior distributions of the scale parameters  $\alpha$  and  $\beta$ . Different priors corresponds to different hypothesis about the unknown data generation process and can be compared by evaluating evidence. [10] [19] [20] proposed Gamma distribution for the prior for scale parameters  $\alpha$  and  $\beta$ . Using gamma priors causes posterior distribution of scale parameters to concentrate at large values for inputs which contribute little towards the data interpolant to be predicted. The  $\theta$  parameters corresponding to these low relevance inputs can be pruned. The parameter set  $\theta$  so obtained is much sparser compared to those

obtained by Maximum Margin approaches. This formulation is a form of *Automatic Relevance Determination* and has been applied in different optimization methods in the past.

### 3 Sparse Bayesian Multi-category Classification

Bayesian learning framework can be used to learn multi-class classifier which are much sparser and have low classification cost. For  $K$  classes and  $N$  observed data pairs  $(y_i, x_i)$  we use the conventional classification framework to learn the class posterior distribution as kernel basis function with canonical link function as  $\sigma_j\{\mathbf{f}\} = e^{-f_j(x)} / \sum_i^K e^{-f_i(x)}$  where  $f_i(x) = \sum_m^N \theta_{m,i} \Phi_m(x)$ , is the kernel basis functions at  $N$  training points. The likelihood can be expressed as:

$$P(\mathbf{D}|\Theta, \mathbf{M}) = \prod_{k=1}^K \prod_{n=1}^N \sigma_k \{\mathbf{f}(\mathbf{x}_n)\}^{y_{nk}} \tag{6}$$

In the classification formulation,  $\beta$  parameter has no significance as the likelihood (6) has no noise variance.

$\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$  are the weight parameters for each class and  $\mathbf{A} = \{\alpha_1, \alpha_2, \dots, \alpha_K\}$  are the scale parameters for the weight priors. We assume independent weight priors for each class,

$$P(\Theta|\mathbf{A}) = \prod_{k=1}^K P(\theta_k|\alpha_k) \tag{7}$$

In the following subsections, we discuss our algorithm for estimating model parameters  $\Theta$  and  $\mathbf{A}$  in bayesian framework.

#### 3.1 Approximating Posterior Distribution for $\Theta$

The posterior distribution can be conveniently formulated as log:

$$\log\{P(\Theta|\mathbf{D}, \mathbf{A})\} = \sum_{k=1}^K \sum_{n=1}^N c_{nk} \log\{\sigma_k\{\mathbf{f}(\mathbf{x}_n)\}\} - \left(\sum_{k=1}^K \theta_k \alpha_k \theta_k^T\right) \tag{8}$$

The  $\alpha_k = \text{diag}(\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{kN})$  are the individual prior scale parameters for each class  $k$  and  $N$  training data points. The Posterior distribution has complex non-gaussian form and cannot be estimated in usual way using (1). We use Laplace’s approximation [14] to estimate the posterior distribution as a gaussian distribution

$$P(\Theta|\mathbf{D}, \mathbf{A}) \simeq P(\Theta_{\text{MP}}|\mathbf{D}, \mathbf{A}) * \exp\left\{-\frac{1}{2}(\Theta - \Theta_{\text{MP}})\mathbf{C}^{-1}(\Theta - \Theta_{\text{MP}})^T\right\} \tag{9}$$

Laplace’s approximation assumes that the posterior distribution of  $\Theta$  has a strong peak at most probable parameters  $\Theta_{\text{MP}}$ . Training the multi-class classifier essentially becomes learning the most probable model parameters  $\Theta_{\text{MP}}$ , as the modes of approximate posterior distribution (9).

Assuming block diagonal covariance matrix  $\mathbf{C} = \text{diag}\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K\}$  for  $K$  classes we can factorize (9) as:

$$P(\Theta|\mathbf{D}, \mathbf{A}) \simeq \left\{ \prod_{k=1}^K P(\theta_{k,MP}|\mathbf{D}, \alpha_k) \right\} \exp \left\{ \sum_{k=1}^K -\frac{1}{2}(\theta_k - \theta_{k,MP})\mathbf{C}_k^{-1}(\theta_k - \theta_{k,MP})^T \right\} \quad (10)$$

$\Theta_{MP}$  can be obtained by finding  $\theta_{k,MP}$  for each class  $k$  independently, using gradient based optimization methods. The covariance matrices  $\mathbf{C}_k$  are evaluated as hessian of log-posterior of class  $k$  [14].

### 3.2 Estimating Most Probable Parameters $\theta_{k,MP}$

For each class  $k$  we estimate  $\theta_{k,MP}$  as modes of the posterior distribution of  $\theta_k$ . We use iterative Newton's method to estimate  $\theta_{k,MP}$  that maximizes posterior (8). The gradient updates for the weights are:

$$\theta_k^{t+1} = \theta_k^t - \frac{\partial \log \{P(\Theta|\mathbf{D}, \mathbf{A})\}}{\partial \theta_k} \left[ \frac{\partial^2 \log \{P(\Theta|\mathbf{D}, \mathbf{A})\}}{\partial \theta_k^2} \right]^{-1} \quad (11)$$

The gradient and hessian can be evaluated as:

$$\nabla_{\theta_k}(\log\{\mathbf{P}(\Theta|\mathbf{D}, \mathbf{A})\}) = - \sum_{n=1}^N \Phi_k(x_n)(c_{nk} - \sigma_k\{f(x_n)\}) - \theta_k \alpha_k \quad (12)$$

$$\nabla_{\theta_k} \nabla_{\theta_k}(\log\{\mathbf{P}(\Theta|\mathbf{D}, \mathbf{A})\}) = -((\Phi_k^T \mathbf{B}_k \Phi_k) + \alpha_k) \quad (13)$$

where  $\mathbf{B}_k = \text{diag}(\beta_{k1}, \beta_{k2}, \dots, \beta_{kN})$ ,  $\Phi_k$  is the kernel basis function and  $\beta_{kn} = \sigma_k\{f(x_n)\}[1 - \sigma_k\{f(x_n)\}]$ . The hessian computed in (13) is used as covariance inverse  $\mathbf{C}_k^{-1}$  of the approximated posterior (10) for class  $k$ . The exact Newton's updates are expensive due to computation of hessian(13). We use quasi-newton method, limited memory BFGS [21], for approximating hessian at each iteration using  $M$  vectors  $\theta_k$  obtained from previous iterations.

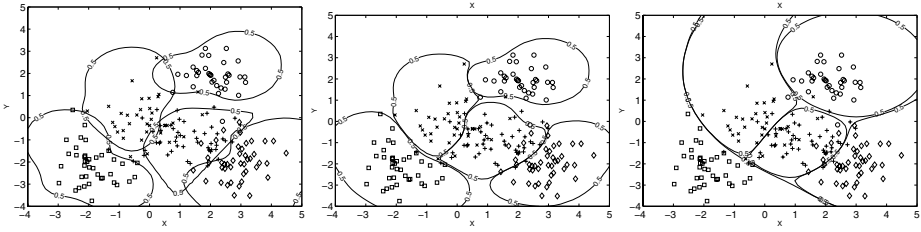
### 3.3 Estimating Most Probable Regularization Scale Parameters

$\alpha_{k,MP}$

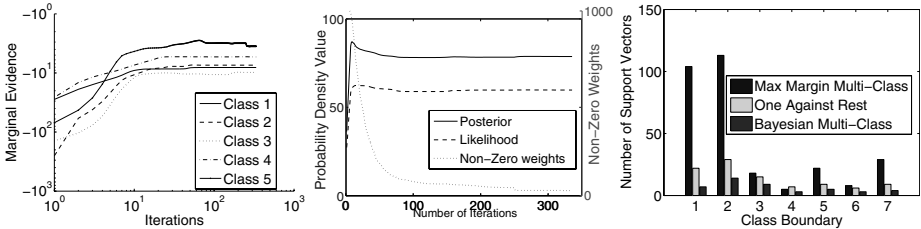
The regularization scale parameter  $\alpha_{k,MP}$  for each class  $k$  is obtained by maximizing the marginal evidence with respect to  $\alpha_k$ . The marginal evidence is obtained by marginalising evidence over the parameter  $\theta_k$  for class  $k$ . For quadratic regularizing term  $P(\theta_k)$ (3), we can approximate the marginal evidence as (10) [14]. For some initial value of  $\alpha_{k,i}$  and  $\theta_k$ , the  $\alpha_{k,MP}$  is obtained as an update equation:

$$\alpha_{k,MP} = \frac{1 - \alpha_{k,i} \text{Trace}\{\mathbf{C}_k^{-1}\}}{\theta_k^2} \quad (14)$$

$\alpha_{k,MP}$  are computed for all the classes by substituting  $\theta_{k,MP}$  (as obtained in Section 3.2) in (14). The updated  $\alpha_k$  values are used to re-estimate classifier parameters  $\theta_k$  for each class. The iterative procedure is run till  $\alpha_k$  for all the classes



**Fig. 1. (Left)** Classification results for 5 class synthetic dataset, using “one against rest” classifier constructed using 5 RVM classifiers [8]. The contours are 0.5 probability points. The boundaries are not separated well and data points lying near boundaries are ambiguously classified. **(Middle)** Sparse Bayesian Multiclass Classifier for 5 classes. The boundaries are well demarcated as the normalization constraint is maintained throughout optimization procedure. **(Right)** Bayesian Multiclass Classifier obtained from smoother radial bases functions obtained by varying the scale parameter.



**Fig. 2. (Left)** Marginal Evidence for scale parameters of each class  $\alpha_k$  on log-scale for the artificial data in fig. 1. Notice that the marginal evidence increases with iterations for every class simultaneously. **(Middle)** *Left scale* Corresponding change of Posterior and Likelihood values with iterations. *Right Scale* Corresponding non-zero weights(model complexity) with no. of iterations. Note here that most of the change occurs in first 50 iterations only. This can be used make training faster. **(Right)** Number of support vectors for 3 multi-class classifier obtained for the Synth. dataset in Fig. 3.

do not change more than prespecified threshold. At every iteration step,  $\alpha_k$  values more than some maximum threshold can be pruned and the corresponding  $\theta_k$  values are made zero.

The critical assumption of this algorithm is the block diagonal covariance matrix in (10) which enables us to treat posterior distribution of  $\theta_k$  for each class independently. The  $\theta_k$  updates for all the classes at every iteration ensures simultaneous increase of marginal evidence at each iteration as shown in Fig. 2(Left).

### 4 Experiments

We conducted experiments to empirically evaluate and compare performance of the proposed classifier with other approaches for multi-class classification. The

more classical approach is to combine several binary classifiers in probabilistic framework to obtain multi-class classification. Classifiers obtained in this way are usually dense (due to modelling many class boundaries), have high classification cost or do not model class boundaries correctly.

The two generic approaches are, “One-against-Rest” and “Pairwise Coupling” [12],[1]. For comparison we use RVM classifier,[8] to learn “One-against-Rest” (1-REST) with logistic link function. We use RVM in “Pairwise Coupling” (PAIR) framework as proposed in [12]. We also compare the results with max margin classifier [7](MM) and generalized linear model (GLM) learnt using Iterative Reweighted Least Square (IRLS). Fig. 1 compares the classification boundaries obtained using 1-ALL classifier and Bayesian Multiclass Classifier, on 5-class synthetic dataset generated by sampling GMM. In all the experiments we used gaussian RBF kernel. For the comparisons, the global parameters of the classifiers were appropriately tuned to give best results.

#### 4.1 Benchmark Comparison Results

We performed experiments on classification benchmarks from UCI Machine Learning database. Fig. 2(Right) compares bayesian multi-class classifier with max margin multi-class classifier[7] and 1-REST classifier in terms of complexity. The histogram represents the number of support vectors(non-zero parameters in the semi-parametric class boundary interpolant) in the multi-class classifier. The number of support vectors for bayesian classifier is much lower compared to other 2 classifiers.

Fig. 3(left) compares prediction rates of Bayesian multiclass classifiers(SBC) with other classifiers. The training datasets used were of varying size and ranged from 150 to 2000. The table shows consistent good performance of SBC compared

Pred.(%)	MM	SBC	1-REST	PAIR	GLM
Synth. (300/7)	62	65.5	62.7	68.6	53.8
Dermat. (292/6)	90	96	95.3	94.6	96
Glass (160/7)	85.6	72.2	66.7	68.5	66.7
OPT (1912/10)	89.1	93.3	93.4	93.2	87.2
PEN Dig. (1500/10)	89.5	94.4	94.2	88.8	85
MFeature (2000/10)	89.6	91.8	92	88.3	82.1

**Fig. 3. (left)** Comparison of prediction rates for different multi-class classifiers. The value in the brackets shows (training dataset size/Classes). All the recognition rates are in (%). Notice that SBC consistently performs good (**center**) Row-wise ordered, 8 classes of viewpoints at rotation angles of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ,  $180^\circ$ ,  $225^\circ$ ,  $270^\circ$  and  $315^\circ$  around Z-axis, (**right**) Class 1, Class 3, Class 4 and Class 5 Real motion sequences.

to other classifiers. Max Margin classifier performs good for some datasets but worse for others. Prediction rate of SBC is always slightly more than 1-REST due to inaccurate boundary modeling in 1-REST.

We also compare the classification time of different classifiers for OPT Digits recognition dataset. Classification of 2000 points for bayesian multi-class classifier was 6 times faster than pairwise classifier and 100 times faster than max-margin classifier. This is due to very sparse model obtained for SBC with typically less than 1% – 2% of number of training points. The "One-against-Rest" classifier was denser as the number of support vectors were more compared to SBC. The pairwise classifier obtained from multiple RVM classifiers, although were sparser, required computing posterior classification probabilities from  $\frac{K(K-1)}{2}$  classifiers. This takes time which increase with the number of classes quadratically. MM classification time largely depended on the constraints' working set size, which was tuned to maximize the prediction rate.

## 4.2 Estimating Viewpoint from Human Silhouettes

We use Bayesian multi-class classifier to learn viewing angle from the human silhouettes. Estimating viewpoint directly has direct application in the context of human body tracking and 3D pose reconstruction. Several human motions are difficult to track from a viewpoint but are easier to track from other. Knowledge of viewpoint can be used to dynamically modify the tracker parameters and adjust to current viewing conditions.

We formulate the problem in classification framework by defining 8 classes based on viewing angles around vertical Z axis(at regular rotation angles of  $45^\circ$ ). The framework can be extended to consider rotation around X and Y axes. However these variations are not relevant in the context of tracking human motion which seldom involves rotation around X and Y axes.

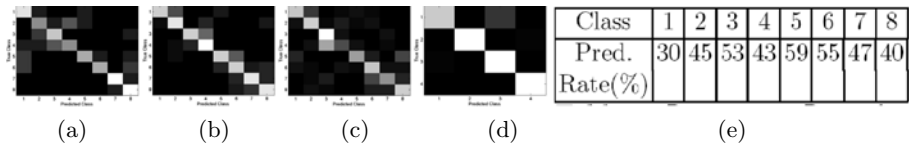
We train the SBC on 2D images rendered using MAYA. The motion capture data for generic motion[13] is imported to MAYA model constructed using standard human specification for joints and segments. The silhouettes extracted from the rendered images(Fig. 3(Right)) are used to generate shape context histogram[15](12 angular bins, 5 radial bins). The overall dimension is reduced to 60 by clustering to few bases means.

We train our classifier on 4 different activities' images rendered from 8 or 4 different viewpoints. We tested the classifier on both artificial and real motion sequences. For artificial test sequence, we rendered images from similar but unseen sequences at angles  $\pm 15^\circ$  of the 8 class viewing angles. Table 1 gives the details of each sequence and the confusion matrix obtained from predictions using bayesian multi-class classifier.

We also tested on a real sequence captured from 8 different viewpoints. Fig. 3(right) shows 4 of these viewpoints. The viewing angle of the motion capture was changed by performing the motion at different angles with respect to camera plane. For extracting silhouettes from real images, we use non-parametric background subtraction and assumed stationary background with single foreground object.

Notice that training silhouettes are quite different from the testing silhouettes and also contained variations due to multiple subjects. The class boundaries are also not very well defined as the shape contexts[15] are invariant to small rotations. We used the artificial walking sequence in Table 1 for training. The test sequences contained varying number of frames for 8 classes of viewpoints. Table 1(e) shows the classification rate of real data set for 8 classes of viewpoint using sparse multi-class classifier. The classification rate is not encouraging due to forward backward ambiguities and the invariability of the shape context to small rotations.

**Table 1.** Confusion Matrix for the artificial test. Notice the bright tridiagonal band due to inaccuracy in classifying bordering points of adjoining classes. Also class 1 and class 5 have larger inaccuracies due to forward backward ambiguities. (a) Walking - 1000 training, 125 samples for each class. Recognition rate - 70% (b) Running - 1000 training points, 125 samples for each class. Recognition rate - 73.67% (c) Jumping - 1000 training points, 125 samples for each class. Recognition rate - 61.25% (d) Bending - 4 Classes at viewing angles  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ , 1000 training samples, 250 samples for each class, recognition rate - 92.5%, Notice the bright cell in row 1, column 3 due to misclassification of forward facing pose as backward facing pose. (e) Recognition Rates for Classes on Real Walking sequence. Notice the very low recognition rates for class 1 (person facing the camera) due to forward-backward ambiguities. Overall recognition rate was 51.3%.



Nevertheless, the proposed bayesian classifier, in general, gives consistently good performance compared to other approaches for multi-class classification and can be used effectively for other machine learning problems. Although the training time for the classifier is more, the classification time is extremely low compared to other multi-class classifiers.

### 5 Conclusions and Future Work

In this paper we propose an extension for bayesian classification [10] to multi-class problems which gives improvement both in classification accuracy and time. The improvement occurs essentially due to sparse non-linear modeling of the class boundaries and maintaining the normalization constraint during the optimization procedure. The future work would involve making the training algorithm faster. The training time for bayesian multi-class classifier is comparable to max margin classifier and GLM but more than “pairwise coupling” and “One-against-Rest” implementation.

## References

1. T. Hastie, R. Tibshirani, "Classification by pairwise coupling" *The Annals of Statistics*, 1998
2. T. G. Dietterich, G. Bakiri, "Solving multiclass learning problems via error-correcting output codes" *Journal of Artificial Intelligence Research*, 1995
3. E. L. Allwein, R. E. Schapire and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers" *ICML*, 2000
4. B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm for optimal margin classifiers" *Computational Learning Theory*, 1992
5. Leo Breinman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, "Classification and Regression Trees" *Wadsworth and Brooks*, 1984
6. E. J. Bredensteiner and K. P. Bennet, "Multicategory classification using Support Vector Machines" *Computational Optimization and Applications*, 1999
7. I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support Vector Learning for Interdependent and Structured Output Spaces," *ICML*, 2004.
8. M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine" *JMLR*, 2001.
9. Ian T. Nabney, "Efficient Training of RBF Networks for Classification" *International Journal of Neural Systems*, 2004
10. D. J.C. Mackay, "Bayesian Interpolation" *Neural Computation*, 1991
11. K. Crammer, Y. Singer "On Algorithmic Implementation of Multiclass Kernel-bases Vector Machines, *JMLR*, 2001
12. T. Hamamura, H. Mizutani, B. Irie, "A Multiclass classification method based on multiple pairwise classifiers" *ICDAR*, 2003
13. "CMU Human Motion Capture Database" <http://mocap.cs.cmu.edu>
14. D. J. C. MacKay. "Choice of basis for Laplace approximation", *Machine Learning*, 1998
15. S. Belongie, J. Malik, and J. Puzicha. "Shape Matching and Object Recognition Using Shape Contexts", *PAMI* 2002
16. Y. Freund, R. E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of Computer and System Sciences*, August 1997
17. C. Burges. "A tutorial on support vector machines for pattern recognition." *KDD*, 1998
18. J. Weston, C. Watkins "Support vector machines for multi-class pattern recognition" *European Symposium on ANN*, April 1999
19. R.M. Neal, "Bayesian Learning for Neural Networks", *Springer*, 1996
20. J. O. Berger, "Statistical decision theory and Bayesian analysis", *Springer*, 1985
21. J. Nocedal, "Updating quasi-Newton matrices with limited storage" *Mathematics of Computation*, 1980