Adaptive Gaussian Kernel Learning for Sparse Bayesian Classification: An Approach for Silhouette Based Vehicle Classification

Article · November 2015

CITATIONS
0

READS
36

3 authors, including:

Ali Mirzaei
Amirkabir University of Technology
6 PUBLICATIONS **1** CITATION
SEE PROFILE

Hamid Sheikhzadeh
Amirkabir University of Technology
97 PUBLICATIONS **760** CITATIONS
SEE PROFILE

Adaptive Gaussian Kernel Learning for Sparse Bayesian Classification: An Approach for Silhouette Based Vehicle Classification

Ali Mirzaei
Electrical Engineering Department
Amirkabir University of Technology
Email: ali_mirzaei@aut.ac.ir

Yalda Mohsenzadeh
Center for Vision Research
York University, Toronto, ON, Canada
Email: myalda@yorku.ca

Hamid Sheikhzadeh
Electrical Engineering Department
Amirkabir University of Technology
Email: hsheikh@aut.ac.ir

***Abstract*—Kernel based approaches are one of the most well-known methods in regression and classification tasks. Type of kernel function and also its parameters have a considerable effect on the classifier performance. Usually kernel parameters are obtained by cross-validation or validation dataset. In this paper we propose a classification learning approach which learn the parameter (kernel width) of Gaussian kernel function during learning stage. The proposed method is an extension of RVM which is a Bayesian counter-part of well-known SVM classifier. The evaluation results on both synthetic and real datasets show better performance and also model sparsity compared to competing algorithms. Particularly the proposed algorithm outperforms other existing methods on vehicle classification based on their silhouettes.**

***Index Terms*—Bayesian Inference, Sparse Bayesian Learning Methods, Kernel Learning Methods, Adaptive kernel, Vehicle Classification.**

I. Introduction

In recent years, kernel-based supervised learning methods have been used frequently in a wide rang of computer vision applications such as image classification [1],[2],[3],[4]. Support vector machine (SVM) [5] is a well-known representative of these family which has been used in many applications for about 20 years [6]. Although SVM has a good performance on many applications but it suffers from the following disadvantages [7]:

1) The number of support vectors increase linearly by the number of training samples.
2) It provides point estimations for the test samples since SVM model is not a probabilistic model.
3) Its performance depends on the trade of parameter C and before training this value must be set.
4) The kernel function must satisfy the Mercer's condition.

Relevance Vector Machine (RVM) [7] is a Bayesian learning method which tackles the mentioned problems. In RVM model there is no constraint on kernel function and It has a statistical view on training data therefore it is more robust against noise and outliers.

The learned model using RVM is highly dependent to the type and parameters of kernel function. Typically Gaussian kernel function is used for real applications and the variance (kernel width) of this function must be set before training the model. There are several paper that they tried to eliminate this problem [8],[9], [10], [11]. Almost all these papers emphasize on regression problems, except [10] which considers both regression and classification problems. The authors of [10] presented a kernel width learning for incremental RVM [12] which is a sub-optimum version of RVM.

In this paper, we propose an adaptive approach for kernel width learning of the standard RVM in the classification task. We present Adaptive Gaussian RVM (AGRVM) for classification problems and introduce a multi-step algorithm which performs the classification. In this method each Relevance Vector (RV) can have a different kernel width and the experiments show the learned model using this method is sparse.

The rest of this paper is organized as follows: part II reviews the details of RVM, part III describes multistage optimization of AGRVM, part IV explains the evaluation of the presented algorithm on both synthetic and real world datasets and part V concludes the paper.

II. Relevance Vector Machine

In supervised learning we are given a set of samples $\{X_n\}_{n=1}^N$ and their corresponding target values $\{t_n\}_{n=1}^N$. RVM considers the following model:

$$y(X;\boldsymbol{w}) = \sum_{n=1}^{N} w_i K(x, x_i) + w_0, \tag{1}$$

where $K(.)$ is known as the kernel function. In the above equation the output is written as a weighted sum of kernel functions which their centers are learning samples. $\boldsymbol{w} = [w_0, w_1, \ldots, w_N]$ are the coefficients of these functions which should be learned using given training samples. The observation model for the regression case is defined as:

$$t_n = y(x_n;\boldsymbol{w}) + \epsilon_n, \tag{2}$$

where ϵ_n is a sample drawn from a zero-mean Gaussian noise. So the distribution of target values can be written as:

$$p(\boldsymbol{t}|\boldsymbol{w},\sigma^2) = (2\pi\sigma^2)^{-1} exp(-\frac{||t-\Phi\boldsymbol{w}||^2}{2\sigma^2}), \tag{3}$$

where σ^2 is the variance of noise and Φ is a $N\times(N+1)$ matrix (known as the kernel matrix) which each row of this matrix is defined as $\Phi(X_n) = [1, K(X_n, X_1), \ldots, K(X_n, X_N)]$. Now if the probability function of Equation (3) gets maximized with respect to $\boldsymbol{w}$, the achieved model (corresponding to $\boldsymbol{w}$) will be overfited to the training data. To avoid overfiting, a zero mean Gaussian prior probability is assumed on all coefficients $\boldsymbol{w}$ as:

$$p(\boldsymbol{w}|\alpha) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \alpha_i), \tag{4}$$

where $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_N]$ is a vector including the inverse of variances of these Gaussian distributions and they are called hyper-parameters of the model. The final stage is to maximize MAP measure to obtain the optimum values for ($\boldsymbol{w}$,α and σ). The maximum a posterior is defined as :

$$p(\boldsymbol{w}, \alpha, \sigma|\boldsymbol{t}) = p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \sigma)p(\alpha, \sigma|\boldsymbol{t}), \tag{5}$$

where the first term is known as the posterior probability over coefficients and the second term called the Marginal Likelihood. Both terms could be calculated analytically. For the posterior probability we have :

$$p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \sigma) = (2\pi)^{-\frac{N+1}{2}}|\Sigma|^{-1}exp(-\frac{1}{2}(\boldsymbol{w}-\mu)\Sigma^{-1}(\boldsymbol{w}-\mu)^T), \tag{6}$$

where μ and Σ are :

$$\Sigma = (\sigma^{-2}\Phi^T\Phi + A)^{-1}, \mu = \sigma^{-2}\Sigma\Phi^T\boldsymbol{t}, \tag{7}$$

and $A = diag(\alpha_0, \alpha_1, \ldots, \alpha_N)$. and for Marginal Likelihood we have:

$$\begin{aligned} p(\alpha, \sigma|\boldsymbol{t}) = (2\pi)^{-N/2}|\sigma^2 I + \Phi A^{-1}\Phi^T|^{-1/2} \\ exp(-\frac{1}{2}\boldsymbol{t}^T(\sigma^2 I + \Phi A^{-1}\Phi^T)^{-1}\boldsymbol{t}) \end{aligned} \tag{8}$$

For optimization and obtaining parameters and hyper-parameter the Expectation-Maximization (EM) algorithm is employed. In EM the optimization is performed in two stages: In the first stage hyper-parameters are considered as known and using the values of the model parameters would be obtained and in the second stage the obtained parameters are used to estimate the hyper-parameters values. After some iterations algorithm converges to a local optimum for both parameters and hyper-parameters (Please note that EM can only gaurantee convergence to a local optimum). In the optimization process the samples which their corresponding hyper-parameters become bigger than a predefine large value will be pruned and in this way the sparse model will be obtained.

III. Adaptive RVM for Classification

In this section we explain adaptive RVM for binary classification. This assumption is not restrictive since we can extend this method using *one-vs-one* and *one-vs-all* algorithms for multi-class cases. In the classification task, in contrast to the regression case there is no Gaussian noise on the target values (labels) and the distribution of observations is no longer Gaussian. In the binary classification case if we assume the labels 0 and 1, the observations distribution is Bernoulli as:

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} p_n^{t_n}(1-p_n)^{1-t_n}, \tag{9}$$

where p_n is called membership probability of the class which its label is 1 and defined as:

$$p_n = \sigma(y(X_n; \boldsymbol{w})), \tag{10}$$

where σ is the sigmoid function $\sigma(y) = 1/(1+e^{-y})$.

In the proposed method, in contrast to the classical RVM in addition to learning the parameters($\boldsymbol{w}$) and the hyper-parameters(α), the kernel parameter (l) should be also learned in the training stage:

$$K(x, y) = exp(-(x-y)^2/l) \tag{11}$$

As described in classical RVM [7] the classification problem can be mapped to a regression one with a Laplace approximation. In this kind of approximation the Bernoulli distribution of labels (Equation 9) is approximated with a Gaussian distribution. Therefore the posterior probability over parameters $p(\boldsymbol{w}|\boldsymbol{t}, \alpha)$ can be approximated with a Gaussian distribution with the following parameters :

$$\Sigma^{-1} = \Phi^T B\Phi + A, \tag{12}$$

$$\mu = \Sigma\Phi^T B\hat{t}, \tag{13}$$

where $B = diag(\beta_1, \beta_2, \ldots, \beta_N)$ is a diagonal matrix and $\beta_n = \sigma(y(x_n))(1-\sigma(y(x_n)))$ and the regression targets $\hat{\boldsymbol{t}} = (\hat{t_1}, \hat{t_2}, \ldots, \hat{t_n})$ are

$$\hat{t} = \Phi\boldsymbol{w} + B^{-1}(\boldsymbol{t} - \boldsymbol{y}). \tag{14}$$

Evidently regression targets in this case are dependent on kernel matrix Φ and output values $\boldsymbol{y}$ and so it depends on kernel widths. Therefore we can not derive the Marginal Likelihood (equation 8) with respect to kernel width. Because of this reason we can not follow the same procedure as regression case [8] for optimizing the kernel widths. In proposed method (AGRVM) we optimize the posterior probability over model parameters $p(\boldsymbol{w}|\boldsymbol{t}, \alpha)$ and obtain the optimized values for kernel widths. To avoid overfitting the following considerations are taken:

1) When the posterior probability over $\boldsymbol{w}$ is optimized only kernel width are updated and parameters and hyper-parameters are assumed fix and known. Parameters $\boldsymbol{w}$ and hyper-parameters α are updated the same way of classic RVM.
2) The optimization of kernel width are not done in every iteration and every H (for example H=5) iterations they are updated. In other words we update H times parameters and hyper-parameters and one time kernel width.

3) The optimization of kernel width dose not continue to converge and we stop the optimization after S (for example S=5) steps

For posterior probability we have :

$$p(\boldsymbol{w}|\alpha, l, t) \propto p(t|\boldsymbol{w}, l)p(\boldsymbol{w}|\alpha) \tag{15}$$

In the above equation the first term is a Bernoulli distribution and the second one is a Gaussian distribution and its log can be calculated as:

$$\mathcal{L} = \sum_{n=1}^{N}(t_n log(y_n) + (1-t_n)log(1-y_n)) - \frac{1}{2}\boldsymbol{w}A\boldsymbol{w}^T, \tag{16}$$

where $A = diag(\alpha)$.

For maximizing the posterior probability it is required to calculate the derivative of Equation (16) with respect to l. To this end, at first we change l_m to γ_m as $\gamma_m = 1/l_m$ and calculate the derivative with respect to γ:

$$\frac{\partial \mathcal{L}}{\gamma_m} = \frac{\partial \mathcal{L}}{\partial \Phi_{nm}} \frac{\partial \Phi_{nm}}{\partial \gamma_m} \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \Phi_{nm}} = (t_n \sigma(z_n) e^{-z_n} - (1-t_n)\sigma(z_n))\boldsymbol{w}_n, \tag{18}$$

where

$$z_j = \sum_{i=1}^{n} \Phi_{ij}\boldsymbol{w}_i, \tag{19}$$

and

$$\frac{\partial \Phi_{nm}}{\partial \gamma_m} = -\sum_{d=1}^{L}(x_m^{(d)} - x_n^{(d)})^2 \Phi_{nm}. \tag{20}$$

Using the above derivative and gradient decent algorithm the optimum values for all kernel widths would be gained. Algorithm 1 illustrates the AGRVM algorithm.

Algorithm 1 Learning The Parameters, Hyper-Parameters and Kernel Width of Model (AGRVM Algorithm)

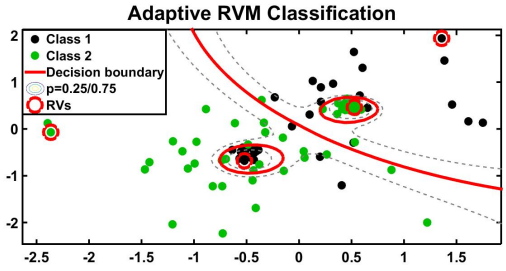
Require: $\mathbf{X} = [X_1, X_2, ...X_N], \mathbf{t}$
  while Counter $<$ Maximum Iterations **do**
    Update Σ and μ using equations 12 and 13
    Update hyper-parameters α using equations presented in [7]
    if Mod(Counter,H)==0 **then**
      Update S times kernel widths using gradient descent algorithm with gradient equation 20
    end if
    if $\alpha_i >$A certain large value **then**
      Prune i^{th} sample from relevant samples
    end if
    if Converged **then**
      Break Loop
    end if
  end while

IV. Evaluation

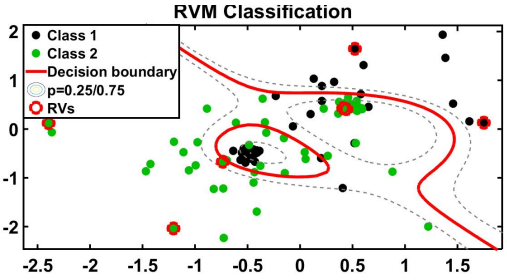
We evaluate the performance of the proposed algorithm on both synthetic and real datasets.

A. Synthetic Dataset

In this section we show the ability of the proposed algorithm to detect variation of variance of training data. To this end we generate two classes and each one has two different variances. Half of samples in class 1 are generated from a Gaussian distribution with mean $\mu = [0.5, 0.5]$ and covariance matrix $\Sigma = [0.5, 0; 0, 0.5]$ and the other half are generated from a Gaussian distribution with mean $\mu = [-0.5, -0.5]$ and covariance matrix $\Sigma = [0.005, 0; 0, 0.005]$. For the second class similarly we generate half of samples from a Gaussian distribution with mean $\mu = [0.5, 0.5]$ and covariance matrix $\Sigma = [0, 005, 0; 0, 0.005]$ and the second half are generated from the same distribution with mean $\mu = [-0.5, -0.5]$ and covariance matrix $\Sigma = [0.5, 0; 0, 0.5]$. We generated 100 samples as learning data and 500 sample as test data.



(a)



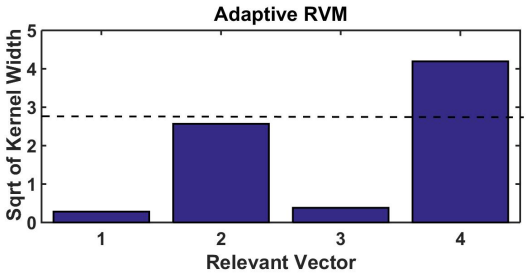
(b)

Fig. 1: Classification Performance of Adaptive RVM (a) and RVM (b)

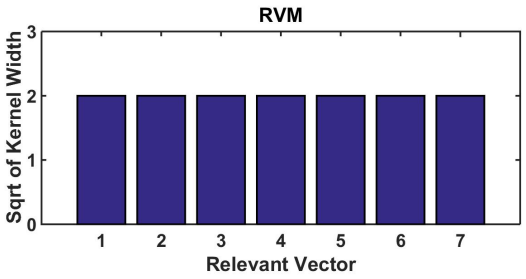
As depicted in Figure 1 the adaptive RVM is able to successfully model different variances of data in each class whereas classical RVM fails to model this kind of behaviour of data because it assumes an equal and constant value for all kernel widths for relevant vectors.The test classification error for AGRVM is 11.2% whereas for RVM this error is 14.3%. Successful modeling of adaptive RVM leads to a sparser learned model as shown in figure 2.

B. Real Dataset

The proposed method is evaluated on vehicle silhouettes dataset of UCI. In this dataset there are 946 samples with



(a)



(b)

Fig. 2: Roots square of kernel width for all Relevant Vectors for adaptive RVM (a) and RVM (b); The dashed line in (a) shows the initial values for all kernel widths

TABLE I: Results of 10-fold Cross-validation on Vehicle Dataset

Methods	#RVs	Accuracy	DT size
SVM	161.3 ± 83.5	77.8 ± 12.3	–
RVM	13.5 ± 6.5	81.1 ± 3.2	–
AGRVM	$\mathbf{11.7 \pm 7.6}$	$\mathbf{82.7 \pm 3}$	–
MCMEM	–	71.35	–
C4.5	–	71.4 ± 1.15	181 ± 3.2
C4.5+GA	–	71.1 ± 0.7	$\mathbf{151.9 \pm 8.32}$
LDA+LC	–	78.25	–

18 attributes in 4 separate classes: bus, van, opel and saab. The silhouettes are captured from many different angels and their attributes are extracted according to the compactness, circularity, moments and other statistical features as described in [13]. Since training and test datasets are not separate in this dataset, we evaluate the algorithm by a 10-fold cross validation approach. The *one-vs-one* approach is employed to extend the binary classification to the multi-class case and the highest voted class is assigned to each test sample.

The results are compared to SVM [5] ,RVM [7] , a decision tree (DT) algorithm named C4.5 and its optimized version with genetic algorithm [14], a maximum entropy modelling method (MCMEM [15]) and a feature extraction method known as LDA with linear discriminant classifier [16].

In SVM (implemented with LibSVM [17]), RVM and AGRVM, the initial kernel width (l^2) is set to the number of features (18). The results of other methods are directly reported from their corresponding papers.

As shown in table I AGRVM has a better performance both in accuracy and sparsity compared to other competing algorithms. The major reason for this superiority is the ability of AGRVM to model different variation of samples in each class (something that is much probable for extracted features of vehicle silhouette).

V. CONCLUSION

In this paper we presented an extended version of RVM which adaptively learns the width of Gaussian kernel function during the learning stage. This method obtains different kernel widths for each relevant vector and this property enables it to handle different variation of samples in each class. The results on both synthetic and real dataset show the good performance of AGRVM in accuracy and sparsity of the learned model. Particularly the results on vehicle silhouettes classification were better than the state-of-the-art algorithms.

REFERENCES

[1] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 6, pp. 1351–1362, 2005.

[2] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 51, no. 1, pp. 217–231, 2013.

[3] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: fast feature extraction and svm training," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1689–1696.

[4] Y. Mohsenzadeh, H. Sheikhzadeh, A. M. Reza, N. Bathaee, and M. M. Kalayeh, "The relevance sample-feature machine: A sparse bayesian learning approach to joint feature-sample selection," 2013.

[5] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in neural information processing systems*, pp. 281–287, 1997.

[6] Y. Ma and G. Guo, *Support vector machines applications*. Springer, 2014.

[7] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *The journal of machine learning research*, vol. 1, pp. 211–244, 2001.

[8] J. Yuan, L. Bo, K. Wang, and T. Yu, "Adaptive spherical gaussian kernel in sparse bayesian learning framework for nonlinear regression," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3982–3989, 2009.

[9] Y. Mohsenzadeh and H. Sheikhzadeh, "Gaussian kernel width optimization for sparse bayesian learning," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 709–719, 2015.

[10] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "Sparse bayesian modeling with adaptive kernel learning," *Neural Networks, IEEE Transactions on*, vol. 20, no. 6, pp. 926–937, 2009.

[11] D. Tzikas, A. Likas, and N. Galatsanos, "Incremental relevance vector machine with kernel learning," in *Artificial Intelligence: Theories, Models and Applications*. Springer, 2008, pp. 301–312.

[12] M. E. Tipping, A. C. Faul *et al.*, "Fast marginal likelihood maximisation for sparse bayesian models," in *Proceedings of the ninth international workshop on artificial intelligence and statistics*, vol. 1, no. 3, 2003.

[13] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[14] G. L. Pappa, A. A. Freitas, and C. A. Kaestner, "Attribute selection with a multi-objective genetic algorithm," in *Advances in Artificial Intelligence*. Springer, 2002, pp. 280–290.

[15] A. Popescul, D. M. Pennock, and L. H. Ungar, "Mixtures of conditional maximum entropy models," 2003.

[16] E. K. Tang, P. N. Suganthan, X. Yao, and A. K. Qin, "Linear dimensionality reduction using relevance weighted lda," *Pattern recognition*, vol. 38, no. 4, pp. 485–493, 2005.

[17] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.