

A Least Square Kernel Machine with Box Constraints

Jayanta Basak

IBM India Research Lab, New Delhi, India

bjayanta@in.ibm.com

Abstract

In this paper, we present a least square kernel machine with box constraints (LSKMBC). The existing least square machines assume Gaussian hyperpriors and subsequently express the optima of the regularized squared loss as a set of linear equations. The generalized LASSO framework deviates from the assumption of Gaussian hyperpriors and employs a more general Huber loss function. In our approach, we consider uniform priors and obtain the loss functional for a given margin considered to be a model selection parameter. The framework not only differs from the existing least square kernel machines, but also it does not require Mercer condition satisfiability. Experimentally we validate the performance of the classifier and show that it is able to outperform SVM and LSSVM on certain real-life datasets.

1 Introduction

In support vector machines (SVM) [11], the margin between two classes is maximized in a higher dimensional space $\phi(\cdot)$ under the constraint of inequality type $t_i(w^T \phi(x_i) + b) \geq 1$ where w is the separating hyperplane. The problem is transformed into an unconstrained problem by the method of Lagrange undetermined multipliers such that the functional

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i (t_i (w^T \phi(x_i) + b) - 1) \quad (1)$$

is minimized with respect to w , b , and maximized with respect to $\alpha_i \geq 0$. Finally introducing slack variables for non-separable cases, and taking into account of the inner product in the Hilbert space, the functional $W_{svm}(\alpha)$ in the dual space is expressed as

$$W_{svm}(\alpha) = \sum_i \alpha_i - \frac{1}{2C} \sum_{i,j} \alpha_i \alpha_j t_i t_j K(x_i, x_j) \quad (2)$$

subject to $0 \leq \alpha_i \leq 1$ and $\sum_i \alpha_i t_i = 0$, where C is judiciously chosen constant and $K(x_i, x_j) = \phi(x_i) \phi(x_j)$ is a symmetric kernel expressible as an inner product in the higher dimensional space subject to Mercer condition [11]. Once the functional $W_{svm}(\alpha)$ is maximized with respect to α_i s (support vectors), classlabel of a test sample is obtained as $sign(\sum_i \alpha_i K(x, x_i) t_i + b)$. Support vector machines have been generalized to multiclass classification, and also used in one-against-all classification.

In the least square kernel machines, the quadratic optimization functional is replaced by linear functional where priors over the Lagrangians are subjected to spherical Gaussian distribution. For example, in ridge regression, a functional with a Lagrangian a , $L = \|t - w^T x\|^2 + a \|w\|^2$ is optimized. In adaptive ridge regression [7], automatic relevance determination is performed in such a way that each variable is penalized by the method of automatic balancing while keeping the average penalty constant. In relevance vector machines [9] also, an automatic relevance determination over the priors is introduced by having prior variance for each expansion coefficient (a Bayesian framework). In the regularized least square classification (RLSC) [5], the objective functional is derived as a simple square-loss function in terms of reproducing Hilbert kernels, and regularized using the Tikhonov regularization with a quadratic term in α vectors given as

$$L = \frac{1}{N} (t - K\alpha)^T (t - K\alpha) + \frac{\lambda}{2} \alpha^T K \alpha \quad (3)$$

where N is the number of samples, λ is a regularization parameter. Equating the first order derivatives to zero with an assumption of Gaussian hyperpriors, the classifier is expressed as a set of linear equations

$$(K + \lambda NI)\alpha = t \quad (4)$$

In the least-square support vector machine (LSSVM) [10] also, a similar squared error term is considered in addition to a regularization term $w^T w$ where w represents the separating hyperplane. In LASSO (least absolute shrinkage and selection operator) [3], a similar

quadratic function is minimized which is

$$L = \|t - K\alpha\|_2^2 + a\|\alpha\|_1 \quad (5)$$

considering the exponential hyperpriors, and a is a Lagrangian parameter. In generalized LASSO [6], the assumption of Gaussian priors was relaxed by taking into account the more general Huber loss [8] which is quadratic for smaller deviation, and linear for larger deviation, and subsequently iteratively reweighted least square (IRLS) technique has been used to optimize the functional.

In this paper, we propose a least square kernel machine with box constraints. The proposed variant of the least square kernel machine does not consider Gaussian hyperpriors. Rather, we formulate the machine with uniform hyperpriors. Experimentally, we show that for certain datasets, the proposed kernel classifier is able to outperform the SVM as well as LSSVM.

2 Least Square Kernel Machine With Box Constraint (LSKMBC)

We formulate the classifier for two-class classification task and then generalize it for multi-class classification using one-against-all strategy. For a two-class classifier with known sample labels $t \in \{-1, 1\}$, we formulate the loss function as

$$L = \frac{1}{2} \sum_i \left(\sum_j \alpha_j K(x_i, x_j) t_j - \lambda t_i \right)^2 \quad (6)$$

where $\lambda \in (0, 1]$ is a given margin acting as a model selection parameter. For a given λ , the minimization of L is equivalent to maximization of the functional

$$W(\alpha) = \lambda t^T K D(t) \alpha - \frac{1}{2} \alpha^T D(t) K^T K D(t) \alpha \quad (7)$$

subject to $0 \leq \alpha_i \leq 1$ for all i . $D(t)$ is a diagonal matrix with the diagonal equal to t . The loss can be reformulated as

$$L = \frac{\lambda^2}{2} \sum_i \left(\sum_j \frac{\alpha_j}{\lambda} K(x_i, x_j) t_j - t_i \right)^2 \quad (8)$$

which is equivalent to minimizing a functional

$$L = \frac{1}{2} \sum_i \left(\sum_j K(x_i, x_j) \hat{\alpha}_j t_j - t_i \right)^2 \quad (9)$$

subject to the constraints $0 \leq \hat{\alpha}_i \leq \frac{1}{\lambda}$. From the Equation (9) and the respective constraints, we observe that we employ a uniform prior over the coefficients α . The

uniform prior can be better observed if we replace $\hat{\alpha}_i t_i$ by β_i such that

$$L = \frac{1}{2} \sum_i \left(\sum_j K(x_i, x_j) \beta_j - t_i \right)^2 \quad (10)$$

subject to the constraint that $0 \leq |\beta_i| \leq \frac{1}{\lambda}$, and $\text{sign}(\beta_i) = t_i$ where t_i is a binary observation in $\{-1, 1\}$. Thus in a Bayesian framework the model selection parameter is governed by λ , and the coefficients have uniform prior over $[0, \pm 1]$ depending on the class-label of the observed sample. In other words, the kernel machine operates on a quadratic optimization functional similar to the ‘‘least square’’ kernel machines except that it employs box constraint on the parameter values (priors).

We observe the similarity with RLSC and LSSVM in the formulation except that both RLSC and LSSVM assume Gaussian hyperpriors and employ Tikhonov regularization. On the other hand, LSKMBC do not employ Tikhonov regularization, however, deviates from the assumption of Gaussian hyperpriors and employ uniform hyperpriors with box constraint derived from a given margin λ . Also LSKMBC does not require the Mercer condition satisfiability of the kernels which is required in RLSC and SVM. Generalized LASSO also deviates from the Gaussian hyperprior model and develops on a more robust Huber loss measure. LSKMBC also has a similarity with ν -SVM [2], where the optimization functional is given as

$$L = -\frac{1}{2} \alpha^T D(t) K D(t) \alpha \quad (11)$$

subject to $0 \leq \alpha_i \leq \frac{1}{\lambda}$, λ being a regularization parameter $\lambda \in [0, 1]$, and $\alpha^T t = 0$, $\sum \alpha_i \geq \nu$. However, ν -SVM does not contain any linear term and it is not derived based on uniform hyperpriors with box constraints.

In SVM optimization functional is derived from the principle of margin maximization with respect to the separating hyperplane in some higher dimensional space $\phi(\cdot)$ such that K is expressible as an inner product, $K(x, x_i) = \phi(x) \phi(x_i)$. In our case, we do not explicitly consider the margin maximization with respect to separating hyperplane in the space of $\phi(\cdot)$, rather we minimize the superfluity in the outcome for a given margin λ with respect to the difference in the class posteriors. This leads to a difference in the quadratic term $K^T K$ in our formulation in Equation (7) from that in the SVM. Apart from the quadratic term, we also observe the difference with SVM in the linear term in the coefficients.

Once we obtain the pattern vectors with non-zero α values by maximizing the functional $W(\alpha)$ with respect

to α_i s (Equation 7), we classify any new test sample x , i.e., assign a classlabel t to x as

$$t = \begin{cases} 1 & \text{if } \sum_i \alpha_i K(x, x_i) t_i \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (12)$$

Multi-class Classification: In the case of multi-class classification, we consider the one-against-all strategy [4], i.e., classify the patterns of each class against all other classes and obtain the coefficients for each class separately. Formally let there be l classlabels $L = \{1, 2, \dots, l\}$. Each time we consider one classlabel at a time. Let the classlabel of concern be c . In that case $t_i \in L$ is transformed into a vector $\{t_{i1}, t_{i2}, \dots, t_{il}\}$ such that

$$t_{ic} = \begin{cases} 1 & \text{if } t_i = c \\ -1 & \text{otherwise} \end{cases} \quad (13)$$

For each classlabel c , we compute the α_{ic} s separately, and the classlabel t to a new sample x is assigned as

$$t = \operatorname{argmax}_{c \in L} \left\{ \sum_i \alpha_{ic} K(x, x_i) t_{ic} \right\} \quad (14)$$

where α_{ic} s are the non-zero coefficients.

3 Experimental Results

We illustrate the behavior of the classifier in Figures 1 and 2 with Gaussian kernel for a Gaussian parity problem. In Figure 1, we vary the kernel width, and in Figure 2, we vary the margin λ . We observe that with the increase in the kernel width, the coefficients concentrate near the class boundaries whereas with the increase in λ , the number of non-zero coefficients increases. Note that, it is not necessarily true that the non-zero vectors are always concentrated near the class boundary as in the case of SVM. This is due to the fact that we do not necessarily maximize the margin between the classes, rather we minimize the error with respect to some given margin.

We demonstrate the effectiveness of our classifier on certain real-life data sets as available in the UCI machine learning repository [1]. In addition to Gaussian kernels, we select two other kernel functions which are centrally peaked with longer tails in nature, such that the distant non-zero vectors have greater interaction between them. Based on the Cauchy distribution, we define the first kernel as

$$K_1(x, \mu) = \frac{1}{1 + \left(\frac{\|x - \mu\|}{\sigma} \right)^2} \quad (15)$$

In the second kernel, we use Gaussian kernel near the center of the kernel (the logarithm of Gaussian is essentially squared deviation) and exponential decay away

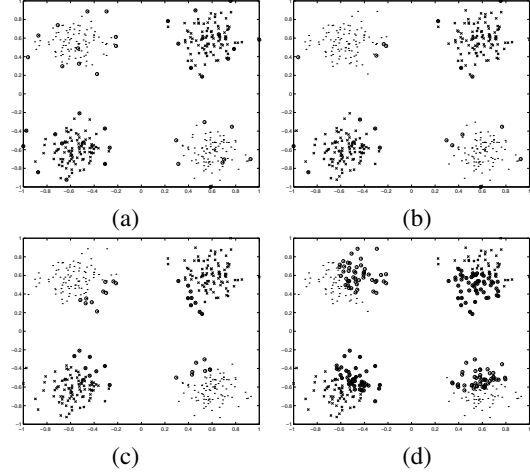


Figure 1. Behavior of the classifier (non-zero vectors) for a 2-D Gaussian parity problem with Gaussian kernel widths (a) $\sigma = 0.2$, (b) $\sigma = 0.5$, (c), $\sigma = 1.0$, and (d) $\sigma = 2.0$ for $\lambda = 1$.

from the center (the logarithm of exponential represents linear deviation) such that

$$K_2(x, \mu) = \begin{cases} \exp\left(-\frac{\|x - \mu\|^2}{2\sigma^2}\right) & \text{for } \|x - \mu\| \leq \sqrt{2}\sigma \\ \exp\left(-\frac{\|x - \mu\|}{\sqrt{2}\sigma}\right) & \text{otherwise} \end{cases} \quad (16)$$

In Table 1, we report the 10-fold cross-validation results with random sampling (averaged over 10 different trials each with 10 folds) for LSKMBC, and compare it with SVM (using Gaussian and polynomial kernels) and LSSVM using Gaussian kernel. In comparing the performance, we use the same training set and the test set for each trial and each fold throughout for all these classifiers. We report the best results for every classifier searching over different sets of parameter values. Note that, we do not use Cauchy kernel and Gaussian+Exponential (G+E) kernel for the SVM and LSSVM because these kernels do not satisfy the Mercer condition. Corresponding to each dataset, we report the parameter values of each classifier for which the best validation score is obtained. As a statistical significance, we also report the standard deviation of the scores for each dataset and each model.

We observe that the LSKMBC outperforms the SVM on several datasets but not all that we have used. For the real-life datasets like ‘Pima’, ‘Bupa’, ‘Wpbc’, ‘Iris’ and ‘Ecoli’, LSKMBC outperforms the SVM and LSSVM, particularly when we observe the performance of the LSKMBC with Cauchy kernel. For the ‘Wdbc’ dataset,

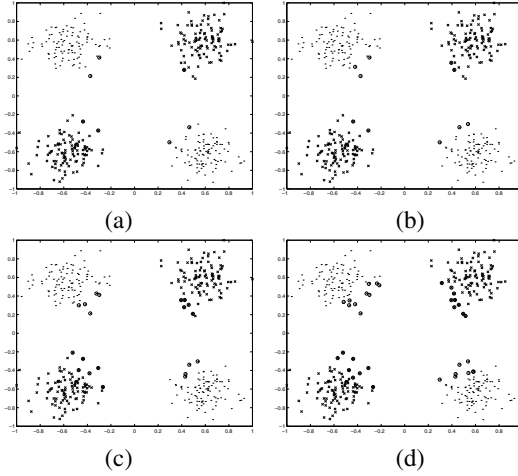


Figure 2. Behavior of the classifier for $\sigma = 1$ and (a) $\lambda = 0.1$, (b) $\lambda = 0.2$, (c) $\lambda = 0.5$, and (d) $\lambda = 1.0$ respectively.

the best performance of the SVM is marginally better than that of the LSKMBC although the LSSVM performs much worse than the other two. Interestingly, we observe that the performance of the LSKMBC often improves significantly with long-tailed kernels such as the Cauchy kernel.

Table 1. 10-fold cross-validation scores with random sampling

Classifier	Pima	Bupa	Wdbc	Wpbc	Iris	Ecoli
SVM (Gauss)	76.88 (4.74)	69.39 (6.53)	98.03 (1.80)	80.78 (6.15)	96.13 (4.87)	88.17 (4.53)
(σ, C)	(1, 2)	(0.3, 1)	(2, 20)	(1, 2)	(1, 1)	(0.5, 1)
SVM (Poly)	75.58 (4.74)	71.72 (6.40)	96.20 (2.26)	74.39 (8.72)	96.13 (5.06)	87.25 (4.88)
(C)	(1)	(5)	(1)	(1)	(5)	(3)
LSSVM (Gauss)	76.28 (5.18)	68.87 (7.21)	93.78 (2.93)	78.13 (3.93)	84.53 (7.90)	77.40 (5.54)
(σ^2, γ)	(5, 0.1)	(1, 2)	(4, 0.3)	(5, 4)	(0.2, 15)	(1, 10)
LSKMBC (Gauss)	77.51 (5.41)	72.42 (6.37)	97.68 (1.65)	81.05 (7.11)	95.87 (5.02)	87.93 (4.79)
(σ, λ)	(5, 0.2)	(2, 0.2)	(1.5, 1)	(5, 0.6)	(1, 0.4)	(1, 0.5)
LSKMBC (G+E)	76.28 (4.78)	73.17 (6.59)	97.61 (1.76)	80.83 (6.35)	97.07 (4.70)	88.77 (4.56)
(σ, λ)	(2, 0.2)	(2, 0.6)	(0.2, 0.2)	(3, 1)	(1, 0.6)	(2, 0.4)
LSKMBC (Cauchy)	77.10 (5.40)	72.77 (6.55)	97.81 (1.65)	82.33 (6.51)	97.60 (4.21)	89.16 (4.37)
(σ, λ)	(5, 0.2)	(3, 0.4)	(2, 0.7)	(3, 0.5)	(2, 0.3)	(3, 0.8)

4 Discussion and Summary

We presented a least square kernel machine classifier with box constraint which employs uniform hyperpriors constrained within a hypercube defined by a given margin. The margin acts as a model selection parameter.

We have shown the relationship of the classifier with the existing least square kernel classifiers such as RLSC and LSSVM. We experimentally demonstrated the effectiveness of the classifier and shown that it is able to outperform the SVM and LSSVM on certain real-life datasets. It may be mentioned here that the LSSVM and RLSC were developed to improve the performance in terms of speed. LSKMBC in that sense is slower than the LSSVM because LSKMBC handles the quadratic optimization task like SVM. However, from the classification performance perspective, LSKMBC can outperform both SVM and LSSVM on several datasets. We also mention that the LSKMBC is not necessarily restricted to Mercer kernels. We used long-tailed kernel functions such as Cauchy kernels and observed that the performance of the LSKMBC significantly improves with long-tailed kernel functions.

References

- [1] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.
- [2] P.-H. Chen, C.-J. Lin, and Schölkopf. A tutorial on nu-support vector machines. *Applied Stochastic Models in Business and Industry*, 21:111–136, 2005.
- [3] M. Osborne, B. Presnell, and B. Turlach. On the LASSO and its dual. *Journal Comput. Graphical Statist.*, 9:319–337, 2000.
- [4] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
- [5] R. Rifkin, G. Yeo, and T. Poggio. Regularized least square classification. In J. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Model and Applications, NATO Science Series III: Computer and System Sciences*, volume 190, pages 131–153. IOS Press, Amsterdam, 2003.
- [6] V. Roth. The generalized LASSO. *IEEE Trans. Neural Networks*, 15(1):16–28, 2004.
- [7] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning (ICML98)*, 1998.
- [8] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC-TR-98-030, NeuroCOLT, Royal Holloway College, University of London, UK, 1998.
- [9] M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal Machine Learning Research*, 1:211–244, 2001.
- [10] T. Van Gestel, J. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and V. J. Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54:5–32, 2004.
- [11] V. Vapnik. *Statistical Learning Theory*. Springer-Verlag, New York, USA, 1998.