ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 177-182

Integrating the Evidence Framework and the Support Vector Machine

James Tin-Yau Kwok
Department of Computer Science
Hong Kong Baptist University
Hong Kong
jamesk@comp.hkbu.edu.hk

Abstract. In this paper, we show that training of the support vector machine (SVM) can be interpreted as performing the level 1 inference of MacKay's evidence framework. We further on show that levels 2 and 3 can also be applied to SVM. This allows automatic adjustment of the regularization parameter and the kernel parameter. More importantly, it opens up a wealth of Bayesian tools for use with SVM. Performance is evaluated on both synthetic and real-world data sets.

1. Introduction

Recently, there has been a lot of interest in studying the support vector machine (SVM) [1, 4, 5]. SVM is based on the idea of structural risk minimization, which shows that the generalization error is bounded by the sum of the training set error and a term depending on the Vapnik-Chervonenkis dimension of the learner. By minimizing this bound, high generalization performance can be achieved. Moreover, unlike other machine learning methods, SVM's generalization error is not related to the problem's input dimensionality. This explains why SVM can have good performance even in high-dimensional problems.

However, some parameters in the SVM still have to be tuned. Two important ones are the regularization parameter and the kernel parameter. Sometimes, these are just hand-picked. A more disciplined way is to use a validation set or by cross-validation, but that can also be very computationally expensive.

In this paper, we apply a well-known Bayesian method, MacKay's evidence framework [2], to SVM, with focus on classification problems. The evidence framework has been applied successfully to feedforward neural networks. Compared with the traditional approach, it provides a rigorous framework for the automatic adjustment of the regularization parameters to their near-optimal values, without the need to set data aside in a validation set. Moreover, it allows objective comparison among solutions using different architectures. Among others, the evidence framework can also assign error bars to network predictions and avoid making over-confident predictions in regions of sparse data.

The rest of this paper is organized as follows. Sections 2 and 3 briefly review SVM and the evidence framework. Section 4 discusses how they can

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 177-182

be integrated and then be used to determine the regularization and kernel parameters. Simulation results are presented in Section 5, and the last section gives some concluding remarks.

2. Support Vector Machine for Classification

Let the training set D be $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with input $\mathbf{x}_i$ and output $y_i \in \{\pm 1\}$. The SVM first maps $\mathbf{x}$ to $\mathbf{z} = \phi(\mathbf{x}) \in \mathcal{F}$. When the data is linearly separable in $\mathcal{F}$, the SVM constructs a hyperplane $\mathbf{w}^T\mathbf{z} + b$ for which separation between the positive and negative examples is maximized. It can be shown that $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{z}_i$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ can be found by solving the following quadratic programming (QP) problem: maximize $W(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}$, subject to $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{\alpha}^T\mathbf{y} = 0$. Here, $\mathbf{y} = (y_1, \ldots, y_N)^T$ and $\mathbf{Q}$ has entries $y_i y_j \mathbf{z}_i^T \mathbf{z}_j = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, where $K(\cdot, \cdot)$ is called a *kernel*. Notice that $\mathbf{Q}$ is positive semi-definite and so there is no local optima in the optimization.

When the training set is not separable in $\mathcal{F}$, the SVM algorithm introduces non-negative slack variables $\xi_i \geq 0$. The resultant problem becomes

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^N \xi_i, \tag{1}$$

subject to $y_i(\mathbf{w}^T\mathbf{z}_i + b) \geq 1 - \xi_i$. C is a regularization parameter controlling the tradeoff between model complexity and training error. Again, (1) can be transformed to a QP problem: maximize $W(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}$ subject to $\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C1}$ and $\boldsymbol{\alpha}^T\mathbf{y} = 0$.

3. The Evidence Framework

A model $\mathcal{H}$, with a k-dimensional parameter vector $\mathbf{w}$, consists of its functional form f, the distribution $p(D|\mathbf{w}, \mathcal{H})$ that the model makes about the data D, and a prior parameter distribution $p(\mathbf{w}|\mathcal{H}, \lambda)$, which is usually of the form: $p(\mathbf{w}|\mathcal{H}, \lambda) = \exp(-\lambda E_W(\mathbf{w}|\mathcal{H}))/Z_W(\lambda)$, where λ is a regularization parameter.

3.1. Level 1 Inference

For a given value of λ, the first level infers the posterior distribution of $\mathbf{w}$ by the Bayes rule: $p(\mathbf{w}|D, \lambda, \mathcal{H}) = p(D|\mathbf{w}, \mathcal{H})p(\mathbf{w}|\lambda, \mathcal{H})/p(D|\lambda, \mathcal{H})$. Substituting in $p(\mathbf{w}|\mathcal{H}, \lambda)$ above, it can be shown that finding the *maximum a posteriori* (MAP) estimate $\mathbf{w}_{MP}$ of $\mathbf{w}$ is the same as minimizing

$$M(\mathbf{w}) \equiv \lambda E_W(\mathbf{w}) - \log p(D|\mathbf{w}, \mathcal{H}). \tag{2}$$

3.2. Level 2 Inference

The second level of inference determines the value of λ by maximizing $p(\lambda|D, \mathcal{H})$. When $p(\lambda|\mathcal{H})$ is flat, the evidence $p(D|\lambda, \mathcal{H})$ can be used to assign a preference to alternative values of λ. By approximating the posterior distribution of $\mathbf{w}$ by a single Gaussian at $\mathbf{w}_{MP}$, it can be shown that

$$\log p(D|\lambda,\mathcal{H}) = -\lambda E_W^{MP} + G^{MP} - \frac{1}{2}\log\det\mathbf{A} + \frac{k}{2}\log\lambda, \tag{3}$$

where $\mathbf{A} = \nabla^2 M, G(\mathbf{w}) \equiv \log p(D|\mathbf{w},\mathcal{H})$, and E_W^{MP} and G^{MP} are the values of E_W and G evaluated at $\mathbf{w}_{MP}$.

3.3. Level 3 Inference

The third level of inference ranks different models by examining their posterior probabilities $p(\mathcal{H}|D)$. Assuming a flat $p(\mathcal{H})$ for all models, different models can then be rated by their evidence $p(D|\mathcal{H})$. Again, assuming that the evidence maximum can be well approximated by a Gaussian, then $p(D|\mathcal{H}) \propto p(D|\lambda_{MP},\mathcal{H})/\sqrt{\gamma}$, where γ is so-called *effective number of parameters*.

4. Applying the Evidence Framework to SVM

Recently, Smola *et al.* [3] proposed a Bayesian interpretation of SVM based on a function-space view. This, however, cannot be readily incorporated into the evidence framework as its prior is based on the weights. Williams [6] did show a prior based on the weight-space view, but only if the squared error loss function were used.

4.1. Another Bayesian Interpretation of SVM

First, consider the case when the training set is not separable in $\mathcal{F}$. In order to determine $\mathbf{w}$, we minimize (1), which, for a fixed C, is the same as minimizing

$$\frac{\|\mathbf{w}\|^2}{2C} + \sum_{i=1}^{N} \xi_i. \tag{4}$$

If we adopt the probability model that (1) the prior over $\mathbf{w}$ is the Gaussian prior $p(\mathbf{w}|C) \propto \exp(-\frac{\|\mathbf{w}\|^2}{2C})$, and (2) the probability[1] of generating pattern i is $\exp(-\xi_i)$, then the log data likelihood $\log p(D|\mathbf{w})$ is $-\sum_{i=1}^N \xi_i$ by assuming that the patterns are i.i.d. Putting $\lambda = 1/C$ and comparing (4) with (2), we see that minimizing (1) during SVM training can be interpreted as performing the level 1 inference under this probability model.

When the training set is separable in $\mathcal{F}$, let α_{max} be the largest Lagrangian multiplier in the set of support vectors. We can view the training process as minimizing (1) with $C = \alpha_{max}$. With a larger $C > \alpha_{max}$, both $\mathbf{w}$ and ξ_i will remain unchanged. Hence, effectively, we can take $C = \alpha_{max}$ in (4). Similarly, back in the non-separable case, we may supply a C to (1) which turns out to

[1] Recall that $\xi_i \geq 0$, hence the probability so defined is normalized ($\int_0^\infty \exp(-\xi_i)\, d\xi_i = 1$).

be larger than all the Lagrangian multipliers in the solution. In this case, the effective C we are using will just be the largest Lagrangian multiplier.

4.2. Computing the Hessian

We next have to determine the hessian $\mathbf{A} = \nabla^2 M = \nabla^2(\lambda E_W + \sum_{i=1}^N \xi_i)$. As an approximation, we assume that ξ_i exactly measures the difference between y_i and $\mathbf{w}^T\mathbf{z}_i + b$ (which in fact is only an upper bound), then

$$\xi_i = \begin{cases} step(1-a_i)(1-a_i) & \text{if } y_i = 1 \\ step(1+a_i)(1+a_i) & \text{if } y_i = -1, \end{cases}$$

where $a_i \equiv \mathbf{w}^T\mathbf{z}_i + b$ and $step(u)$ is the step function. However, $step(u)$ is not differentiable, and we replace it by the sigmoid function $s(u) = 1/(1+e^{-\eta u})$. Noting that $\nabla a_i = \mathbf{z}_i$ and $\nabla^2 a_i = 0$, we obtain $\nabla^2 \xi_i = r(|y_i - a_i|)\mathbf{z}_i\mathbf{z}_i^T \equiv r_i\mathbf{z}_i\mathbf{z}_i^T$, where $r(u) \equiv us''(u) + 2s'(u)$. Thus, $\mathbf{A} = \lambda\mathbf{I} + \mathbf{B}$, where $\mathbf{B} = \sum_{i=1}^N r_i\mathbf{z}_i\mathbf{z}_i^T$.

One can show that the eigenvectors $\{\mathbf{v}_l\}_l$ of $\mathbf{B}$ can be written as $\mathbf{v}_l = \sum_{i=1}^N \mu_{li}\mathbf{z}_i$. For $\mathbf{z}_k = \phi(\mathbf{x}_k), k = 1, \ldots, N$, we have $\rho_l\mathbf{z}_k^T\mathbf{v}_l = \mathbf{z}_k^T\mathbf{B}\mathbf{v}_l$, where ρ_l is an eigenvalue of $\mathbf{B}$. This leads to $\rho_l\mathbf{K}\boldsymbol{\mu}_l = \mathbf{K}\tilde{\mathbf{K}}\boldsymbol{\mu}_l$, where $\boldsymbol{\mu}_l = (\mu_{l1}, \ldots, \mu_{lN})^T$, $\mathbf{K}$ is the $N \times N$ matrix with entries $\mathbf{z}_i^T\mathbf{z}_j = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\tilde{\mathbf{K}}$ is another $N \times N$ matrix with entries $r_i\mathbf{z}_i^T\mathbf{z}_j = r_iK(\mathbf{x}_i, \mathbf{x}_j)$. Assuming that $\mathbf{K}$ is invertible, we have $\rho_l\boldsymbol{\mu} = \tilde{\mathbf{K}}\boldsymbol{\mu}_l$. Solving, we obtain the eigenvalues $\{\hat{\rho}_l\}_l$ of $\mathbf{A}$ as $\hat{\rho}_l = \lambda + \rho_l$.

4.3. Levels 2 and 3 Inference for SVM

Level 2 inference determines the value of λ by maximizing $p(D|\lambda, \mathcal{H})$ in (3). Recall that $p(D|\lambda, \mathcal{H})$ is computed by approximating the posterior distribution of $\mathbf{w}$ by a single k-dimensional Gaussian at $\mathbf{w}_{MP}$. Here, k is usually very large and can even be infinite, depending on the chosen kernel function. In this case, only a (possibly very small) subspace of $\mathbf{w}$ will be affected by the data likelihood, and we argue that the Gaussian approximation is valid only in this subspace. In effect, we replace k in (3) by the number of significant eigenvalues n in $\tilde{\mathbf{K}}$. Moreover, $\det\mathbf{A}$ in (3) can be readily computed given the eigenvalues $\hat{\rho}$ of $\mathbf{A}$, and we get $\log p(D|\lambda, \mathcal{H}) = -\lambda E_W^{MP} + G^{MP} - \frac{1}{2}\log\prod_{i=1}^n \hat{\rho}_i + \frac{n}{2}\log\lambda$.

To obtain the model evidence $p(D|\mathcal{H})$ in level 3 inference, we need to calculate the effective number of parameters γ [2]. This involves $\text{trace}\mathbf{A}^{-1}$, which, again, can be computed readily from the eigenvalues of $\mathbf{A}$. It can be shown that $\gamma = \sum_{i=1}^n \rho_i/(\lambda + \rho_i)$.

5. Simulation

Simulation is performed on two data sets. The first one is a 2-dimensional toy problem, with 500 training patterns and 10,000 test patterns. The second one is the image segmentation data[2] from the UCI machine learning repository.

[2] The original problem is to classify a pattern into one of the seven classes: brickface, sky, foliage, cement, window, path and grass. Here, we only concentrate on the class brickface.

Here, we only report results on the polynomial kernel, though satisfactory performance has also been obtained on the Gaussian kernel.

5.1. Choosing the Regularization Parameter

Figures 1a and b plot $p(D|\lambda, \mathcal{H})$ and the percentage of correct classifications on the test set at different values of C $(= 1/\lambda)$. In most cases, the evidence for λ follows the testing accuracy closely. The discrepancy can be attributed partly to the Gaussian approximation used for the posterior distribution of $\mathbf{w}$. Another reason is related to the use of $\sum_i \xi_i$ as a measure of the training error in (1). As a result, $p(D|\lambda, \mathcal{H})$ is more accurately related to $1/\sum_i \xi_i$ in the test set. This is confirmed in Figures 1c and d, which show an improved match between the mean of ξ_i on the test set and the evidence for λ at different C.

5.2. Choosing the Kernel Parameter

This section discusses results on using the model evidence $p(D|\mathcal{H})$ to determine the degree d in the polynomial kernel. For a fixed d, the regularization parameter C is estimated in an iterative manner as described in [2]. Figure 2 plots $p(D|\mathcal{H})$ and the percentage of correct classifications on the test set at different values of d. Again, the evidence follows the testing accuracy closely.

6. Conclusion

In this paper, we show that the evidence framework can be applied to the SVM. This integration allows automatic adjustment of the regularization parameter and the kernel parameter to their near-optimal values. Moreover, it opens up a wealth of Bayesian tools for use with SVM, such as the calculation of error bars and moderated outputs.

References

[1] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):955–974, 1998.

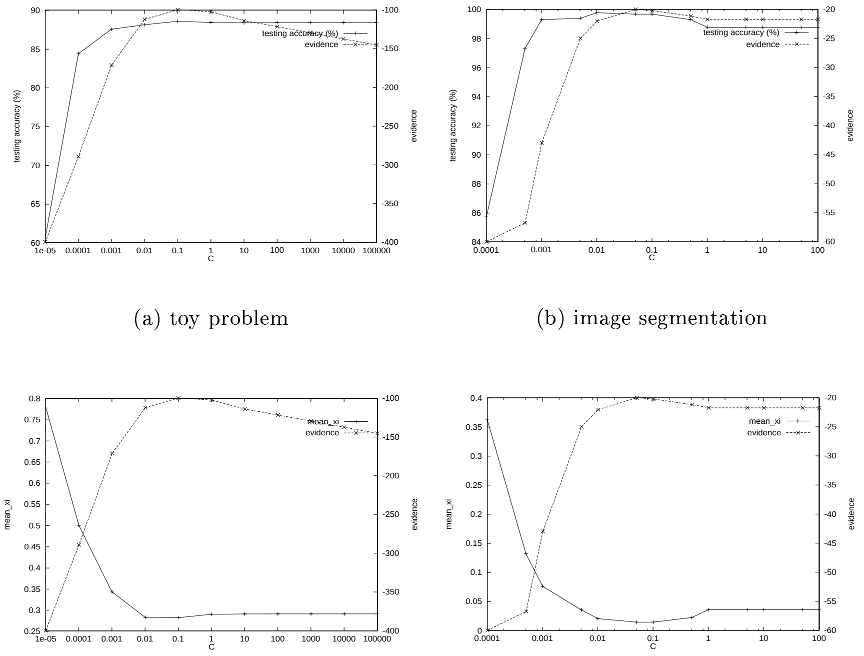
[2] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, May 1992.

[3] A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

[4] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998.

[5] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[6] C.K.I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M.I. Jordan, editor, *Learning and Inference in Graphical Models.* Kluwer, 1998.



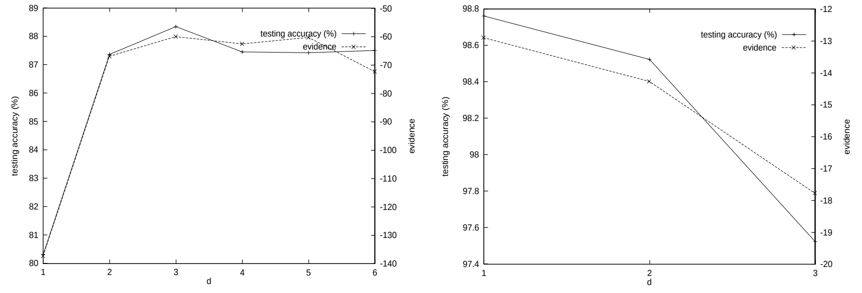
(a) toy problem

(b) image segmentation

(c) toy problem

(d) image segmentation

Figure 1: Results on using different values of C.



(a) toy problem

(b) image segmentation

Figure 2: Results on using different values of d.