Support Vector Mixture for Classification and Regression Problems

James Tin-Yau Kwok
Department of Computing Studies
Hong Kong Baptist University
Kowloon Tong
Hong Kong
jamesk@comp.hkbu.edu.hk

Abstract

In this paper, we study the incorporation of the support vector machine (SVM) into the (hierarchical) mixture of experts model to form a support vector mixture. We show that, in both classification and regression problems, the use of a support vector mixture leads to quadratic programming (QP) problems that are very similar to those for a SVM, with no increase in the dimensionality of the QP problems. Moreover, a support vector mixture, besides allowing for the use of different experts in different regions of the input space, also supports easy combination of different architectures such as polynomial networks and radial basis function networks.

1. Introduction

In recent years, there has been a lot of interest in studying support vector machine (SVM) for classification and regression problems [1, 2, 6, 7, 8]. SVM is based on the idea of *structural risk minimization* [8], which shows that the generalization error is bounded by the sum of the training set error and a term which depends on the Vapnik-Chervonenkis dimension of the classifier. By minimizing the sum of both quantities, high generalization performance can be achieved. Moreover, unlike other machine learning methods, the number of free parameters in the SVM does not depend explicitly on the input dimensionality of the problem, which suggests that SVM can be especially useful in problems with a large number of inputs. Experimentally, SVM has also achieved superior performance. For example, it outperforms radial basis networks on recognizing the US postal service database of handwritten digits [7] and improves the best known result on a time series benchmark in the Santa Fe Competition by a factor of 37% [6].

Another powerful technique in solving classification and regression problems is the use of a (hierarchical) mixture of experts [3, 4, 5]. This divides a complex learning problem into a set of sub-problems, each of which may be simpler to solve than the original problem. The architecture has a number of *expert networks* specialized in different regions of the input space, with their outputs selectively combined by a *gating network*. This use of the divide-and-conquer principle has led to many simple, elegant and efficient solutions in practice. It is thus promising to extend SVM to such a mixture model.

Moreover, as noted in [5], divide-and-conquer algorithms generally tend to be variance-increasing. This may lead to poor generalization performance, especially in problems with high input dimensionality where the data become exceedingly sparse. To mitigate this problem, Jordan and Jacobs [5] used "soft" splits of data and piecewise linear functions in the expert and gating networks. However, training algorithms in use (typically, either gradient descent algorithms [3] or the EM algorithm [5]) are based on maximum likelihood estimation (a form of *empirical risk minimization* [8]). As mentioned earlier, SVM, being based on structural risk minimization, should be a more viable alternative. Ridge regression has also been suggested as a replacement for maximum likelihood estimation [4]. But, in comparison, SVM allows the functional forms of the expert networks to be easily changed[1], can readily be applied to classification problems, and is also devoid of local optima in the optimization process.

The rest of this paper is organized as follows. A brief summary of the SVM is given in Section 2. Section 3 introduces the support vector mixture and Sections 4 and 5 provide further extensions to multiplicative and hierarchical mixtures. The last section gives some concluding remarks.

[1] As we will see in later sections, this is done by changing the kernel function of the SVM.

2. Support Vector Machine

We first review the use of SVM in classification problems. Let the training set be $\{(x_i, y_i)\}_{i=1}^N$, with each input $x_i \in \Re^m$ and $y_i \in \{\pm 1\}$. The SVM first performs a mapping $\phi : \Re^m \rightarrow \Re^n$. Usually, n is much greater than m so that the input vector is mapped to a high dimensional vector. When the data are linearly separable, SVM constructs a hyperplane in $\Re^n$ ($w'\phi(x) + b$) for which the margin between the positive and negative examples is maximized. It can be shown that the w for this "optimal" hyperplane can be written as a linear combination of $\phi(x)$'s, i.e., $w = \sum_{i=1}^N \alpha_i y_i \phi(x_i)$. These α_i's can be found by solving the following quadratic programming (QP) problem:

$$\max \quad W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \phi(x_i)' \phi(x_j) \qquad (1)$$

with respect to the α_i's, under the constraints $\sum_{i=1}^N \alpha_i y_i = 0$ and $\alpha_i \geq 0$ for $i = 1, \ldots, N$. To obtain $\phi(x_i)'\phi(x_j)$ in (1), one does not need to compute $\phi(x_i)$ and $\phi(x_j)$ explicitly. Instead, for some ϕ, one can find a *kernel* $K(\cdot, \cdot)$ such that $K(x_i, x_j) = \phi(x_i)'\phi(x_j)$. For example, the kernel for a polynomial classifier of degree d is $K(x_i, x_j) = (1 + x_i' x_j)^d$. (1) then becomes:

$$\max \quad W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j). \qquad (2)$$

For the training examples along the decision boundary, the corresponding α_i's are greater than zero by the Kuhn-Tucker theorem, and these examples are called the *support vectors*. Experimentally, their number is usually small, and Vapnik [8] showed that it is proportional to the generalization error of the classifier. During testing, a test vector $x \in \Re^m$ is assigned a class label according to $\text{sign}(w'\phi(x) + b) = \text{sign}(\sum_{i=1}^N \alpha_i y_i K(x, x_i) + b)$.

When the training set cannot be separated without error in $\Re^n$, the SVM algorithm tries to separate with a minimum number of errors, by minimizing $\frac{1}{2}|w|^2 + C \sum_{i=1}^N \xi_i$, where $\xi_i \geq 0$ measures the difference between y_i and the output from SVM, and C is a constant balancing contributions from the first and second terms. Again, this optimization problem can be transformed into a QP problem: maximize (2) under the constraints $\sum_{i=1}^N \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, N$.

The ideas in classification carry over to regression: map the data x to a high dimensional space via ϕ and then perform linear regression in that space. A robust cost function, called the *ϵ-insensitive cost function*, given by

$$V(e) = \begin{cases} 0 & \text{if } |e| < \epsilon, \\ |e| - \epsilon & \text{otherwise,} \end{cases}$$

is used. The parameter ϵ represents, in a sense, the resolution at which we want to look at the data, and errors which are smaller than ϵ are assigned zero cost.

There are two approaches to the regression problem. The first places a constraint $|w|^2 < U$ (for some $U > 0$) in the optimization process, leading to the QP problem:

$$\max \quad W(\alpha, \alpha^*, C) = -\epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) + \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i)$$
$$- \frac{1}{2C} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j) - \frac{CU}{2}, \quad (3)$$

subject to $C \geq 0$, $\sum_{i=1}^N \alpha_i = \sum_{i=1}^N \alpha_i^*$, and $0 \leq \alpha_i, \alpha_i^* \leq 1$ for $i = 1, \ldots, N$. The second formulation minimizes a weighted combination of $|w|^2$ and the error term, controlled by a constant C. The resultant QP problem is:

$$\max \quad W(\alpha, \alpha^*) = -\epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) + \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i)$$
$$- \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j), \qquad (4)$$

subject to the constraints $\sum_{i=1}^N \alpha_i^* = \sum_{i=1}^N \alpha_i$, and $0 \leq \alpha_i, \alpha_i^* \leq C$ for $i = 1, \ldots, N$.

3. Support Vector Mixture

While SVM uses only one linear expert (either a hyperplane classifier or a linear regression surface) in the high dimensional space, we consider using a combination of n_e linear experts as in the mixture of experts model [3, 5]. The weighted output is given by:

$$y = f(x) = \sum_{k=1}^{n_e} \pi_k(x)(w_k' \phi(x) + b_k), \qquad (5)$$

where w_k, b_k are the weight and bias of the kth expert respectively, and $\pi_k(x)$ is the kth output of the gating network[2]. Note that it is necessary for π_k to be dependent on x. Otherwise, (5) can always be replaced by one single expert. As in [3, 5], (5) yields a smoothed, piecewise linear classification/regression surface (in the transformed high-dimensional space governed by ϕ). In the following sections, we formulate the training of this mixture in (5) as QP problems similar to that for SVM.

3.1. Classification with Separable Data

We want to minimize[3]

$$\Psi = \frac{1}{2} \sum_k |w_k|^2$$

[2] In the sequel, we use $\pi_{ki} \equiv \pi_k(x_i)$ to simplify the notation.

[3] Using the regularization theory framework in [2], it can be shown that this optimization problem is the same as minimizing the smoothness functional

$$\Phi[f] = \sum_k \sum_n \frac{c_{kn}^2}{\lambda_n},$$

subject to the constraints

$$y_i \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) \geq 1, \quad i = 1, \ldots, N. \qquad (6)$$

First, we form the Lagrangian:

$$L = \frac{1}{2}\sum_k |w_k|^2 - \sum_i \alpha_i [y_i \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) - 1], \quad (7)$$

where α_i's are the Lagrangian multipliers. It is known that the solution to this optimization problem is determined by the saddle point of L, where the minimum should be taken with respect to w_k, b_k, and the maximum be taken with respect to the α_i's. Upon simplifying, we obtain the following QP problem used for the determination of α_i's:

$$\max \ W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{ijk} \alpha_i \alpha_j y_i y_j \pi_{ki}\pi_{kj} K(x_i, x_j), \quad (8)$$

subject to the constraints $\sum_i \alpha_i y_i \pi_{ki} = 0$ for $k = 1, \ldots, n_e$, and $\alpha_i \geq 0$ for $i = 1, \ldots, N$. Comparing this with the corresponding QP problem for SVM in (2), one can see that the problem is basically unchanged except for the extra term $\sum_k \pi_{ki}\pi_{kj}$ in $W(\alpha)$ and the slight changes in the set of linear constraints. Moreover, note that this extra term leaves the positive semi-definiteness of the original QP problem unchanged, which means that the α_i's obtained from (8) always correspond to a global optimum.

After obtaining the α_i's, we can find the b_k's by using the fact that the product of the Lagrangian multiplier with its corresponding constraint vanishes, i.e., $\alpha_i[y_i \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) - 1] = 0$. Thus, from the set of support vectors (for which their α_i's are positive), we can obtain a linear system from which the b_k's can be obtained by computing the pseudo-inverse.

3.2. Classification with Non-separable Data

When the data are not separable, we minimize

$$\Psi = \frac{1}{2}\sum_k |w_k|^2 + C\sum_i \xi_i,$$

where $C > 0$ is a constant balancing the error term and the complexity term, subject to the constraints

$$y_i \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N.$$

subject to the constraints (6). Here, c_{kn}'s are the coefficients of $\phi_n(x)$ in the expansion of (5):

$$f(x) = \sum_k \pi_k(x)[\sum_n c_{kn}\phi_n(x) + b],$$

and λ_n, ϕ_n are the eigenvalues and eigenfunctions of the kernel K. The ϕ_n's are also related to ϕ by $\phi(x) = [\phi_1(x), \phi_2(x), \ldots]'$.

In this case, the Lagrangian is:

$$L = \frac{1}{2}\sum_k |w_k|^2 + C\sum_i \xi_i - \sum_i \alpha_i[y_i \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) - 1 + \xi_i] - \sum_i r_i \xi_i.$$

Minimizing L with respect to w_k, b_k, ξ_i and maximizing with respect to the Lagrange multipliers α_i, r_i, we obtain the following QP problem:

$$\max \quad W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{ijk} \alpha_i \alpha_j y_i y_j \pi_{ki}\pi_{kj} K(x_i, x_j),$$

subject to $\sum_i \alpha_i y_i \pi_{ki} = 0$ for $k = 1, \ldots, n_e$, and $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, N$. Again, note the close resemblance to the corresponding QP problem for SVM.

3.3. Regression

As mentioned in Section 2, there are two approaches to the regression problem. Firstly, we can minimize $\sum_i(\xi_i + \xi_i^*)$ subject to $\sum_k |w_k|^2 < U$ and

$$\begin{aligned} y_i - \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) &\leq \epsilon + \xi_i^*, \\ \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) - y_i &\leq \epsilon + \xi_i, \end{aligned} \qquad (9)$$

$\xi_i, \xi_i^* \geq 0$ for $i = 1, \ldots, N$. The Lagrangian is:

$$\begin{aligned} L = &\sum_i (\xi_i + \xi_i^*) - \sum_i \alpha_i [y_i - \sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) \\ &+ \epsilon + \xi_i] - \sum_i \alpha_i^* [\sum_k \pi_{ki}(w_k'\phi(x_i) + b_k) - y_i + \epsilon + \xi_i^*] \\ &- \sum_i (\gamma_i \xi_i + \gamma_i^* \xi_i^*) - \frac{C}{2}(U - \sum_k |w_k|^2). \end{aligned}$$

Minimizing L with respect to w_k, b_k, ξ_i, ξ_i^* and maximizing with respect to the Lagrangian multipliers $C, \alpha_i, \alpha_i^*, \gamma_i, \gamma_i^*$, we obtain the QP problem:

$$\begin{aligned} \max \quad W(\alpha, \alpha^*) = &-\epsilon \sum_i (\alpha_i^* + \alpha_i) + \sum_i y_i(\alpha_i^* - \alpha_i) \\ &- \frac{1}{2C}\sum_{ijk} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\pi_{ki}\pi_{kj}K(x_i, x_j) - \frac{CU}{2}, \end{aligned} \qquad (10)$$

subject to $C \geq 0$, $\sum_i \pi_{ki}(\alpha_i - \alpha_i^*) = 0$ for $k = 1, \ldots, n_e$, and $0 \leq \alpha_i, \alpha_i^* \leq 1$ for $i = 1, \ldots, N$.

Alternatively, one can minimize $\frac{1}{2}\sum_k |w_k|^2 + C\sum_i(\xi_i + \xi_i^*)$ subject to (9). One can show that the resultant QP problem is:

$$\begin{aligned} \max \quad W(\alpha, \alpha^*) = &-\epsilon \sum_i (\alpha_i^* + \alpha_i) + \sum_i y_i(\alpha_i^* - \alpha_i) \\ &- \frac{1}{2}\sum_{ijk} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\pi_{ki}\pi_{kj}K(x_i, x_j), \end{aligned} \qquad (11)$$

subject to the constraints $\sum_i(\alpha_i^* - \alpha_i)\pi_{ki} = 0$ for $k = 1, \ldots, n_e$, and $0 \le \alpha_i, \alpha_i^* \le C$ for $i = 1, \ldots, N$. Comparing (10) and (11) with the corresponding QP problems for SVM ((3) and (4)), one can see that the QP problems are again basically unchanged except for the extra term $\sum_k \pi_{ki}\pi_{kj}$ in $W(\alpha, \alpha^*)$ and the slight changes in the set of linear constraints.

4. Multiplicative Support Vector Mixture

So far, we placed no restrictions on the form of the gating networks. Now, if we assume each output of the gating network to be a linear function, a multiplicative mixture can be obtained[4] (Figure 1). To be more specific, we choose $\pi_k(x) = \bar{w}_k'\bar{\phi}(x) + \bar{b}_k$ in (5). Then, on writing $\bar{\pi}_k(x) \equiv w_k'\phi(x) + b_k$, we obtain

$$y = f(x) = \sum_{k=1}^{n_e} \bar{\pi}_k(x)(\bar{w}_k'\bar{\phi}(x) + \bar{b}_k).$$

With w_k, b_k fixed, training of $\bar{w}_k, \bar{b}_k$'s is thus identical to that of w_k, b_k's. Moreover, note that ϕ and $\bar{\phi}$ need not be the same. Hence, by using different kernels, we can have different components in this multiplicative mixture corresponding to polynomials, radial basis functions, neural networks, splines, Fourier series, etc. [8]. This offers an easy means of combining a number of different architectures into the mixture.

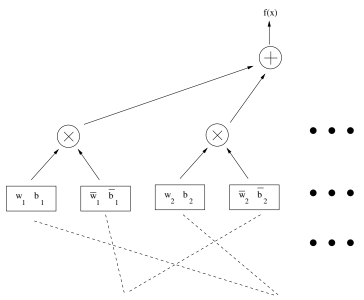

Figure 1. A multiplicative mixture. Each rectangular box implements $w_k\phi(x) + b_k$ or $\bar{w}_k\bar{\phi}(x) + \bar{b}_k$. Those joined together by a dotted line are controlled by one common set of Lagrangian multipliers.

5. Hierarchical Support Vector Mixture

Results in the previous sections can be easily extended to the case of a hierarchical mixture of experts. Here, for simplicity, we just consider a two-level hierarchical mixture:

$$y = f(x) = \sum_m \tilde{\pi}_m(x) \sum_{k \in K(m)} \pi_k(x)(w_k'\phi(x) + b_k),$$

[4] Obviously, this can be generalized to have more than two multiplicative components.

where $\tilde{\pi}_m(x)$ is the mth output of the upper gating network[5], and $K(m)$ contains the indices in the mth branch of the hierarchy. Basically, to obtain the corresponding expressions in the hierarchical mixture, one simply replaces π_{ki} in the previous results by $\tilde{\pi}_{mi}\pi_{ki}$. So, for example, considering the case for separable data in classification, the new QP problem is:

$$\max \quad W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{ijk} \alpha_i\alpha_j y_i y_j \tilde{\pi}_{mi}\tilde{\pi}_{mj}\pi_{ki}\pi_{kj}K(x_i, x_j),$$

subject to the constraints $\sum_i \alpha_i y_i \tilde{\pi}_{mi}\pi_{ki} = 0$ for all $k \in \cup_m K(m)$, and $\alpha_i \ge 0$ for $i = 1, \ldots, N$.

6. Conclusion

In this paper, we studied the incorporation of SVM into the (hierarchical) mixture of experts model. We showed that, in both classification and regression problems, training of the support vector mixture leads to QP problems that are very similar to the ones for SVM. Hence, in order to enjoy the advantages offered by the (hierarchical) mixture models, there is little computational overhead in moving from SVM to support vector mixture. Moreover, the possibility of using different kernel functions for different multiplicative components in the mixture allows easy blending of a number of different architectures.

References

[1] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[2] F. Girosi. An equivalence between sparse approximation and support vector machines. To appear in *Neural Computation*, 1997.

[3] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[4] M. Jordan. A statistical approach to decision tree modeling. In *Proceedings of the Seventh ACM Conference on Computational Learning Theory*, pages 13–20, New Brunswick, NJ, July 1994.

[5] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, Mar. 1994.

[6] K. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In *Proceedings of the International Conference on Artificial Neural Networks*, 1997.

[7] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. A.I. Memo 1599, MIT, Dec. 1996.

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

[5] In the sequel, we use $\tilde{\pi}_{mi} \equiv \tilde{\pi}_m(x_i)$ to simplify the notation.