ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 183-188

Support Vector Classifier with Asymmetric Kernel Functions

Koji Tsuda

Electrotechnical Laboratory
1-1-4, Umezono, Tsukuba-shi, Ibaraki-ken, Japan, 305-8568.
E-Mail: tsudak@etl.go.jp

Abstract. In support vector classifier, asymmetric kernel functions are not used so far, although they are frequently used in other kernel classifiers. The applicable kernels are limited to symmetric semipositive definite ones because of Mercer's theorem. In this paper, SVM is extended to be applicable to asymmetric kernel functions. It is proven that, when a positive definite kernel is given, the extended SVM is identical with the conventional SVM. In the 3D object recognition experiment, the extended SVM with asymmetric kernels performed better than the conventional SVM.

1. Introduction

Kernel functions have been used for constructing classifiers including Parzen windows, RBF networks and support vector machines (SVM)[1]. In SVM, only symmetric kernel functions are used so far. It is because the kernel function is regarded as the dot product in a high dimensional feature space. The dot product must be symmetric by axiom, so the kernel must be symmetric.

But, in other kernel methods such as Parzen Windows and RBF Networks, asymmetric kernels are often used[2]. A typical example is the *variable* kernel function, whose parameters change with regard to the position of the kernel:

$$K(\mathbf{x}, \mathbf{y}; \sigma(\mathbf{y})). \tag{1}$$

It is known that appropriate adjustment of parameters makes the performance better. It is expected that, if asymmetric kernels are used, the performance of SVM might be improved as in other classifiers.

In this paper, we propose an extension of SVM that can be applied to asymmetric kernel functions. The extended SVM is formulated without the "kernel trick". A sample $\mathbf{x}$ is represented as an n-dimensional vector whose i-th element is $K(\mathbf{x}, \mathbf{s}_i)$, where $\mathbf{s}_1, \cdots, \mathbf{s}_n$ denote the training samples. The extended SVM is formulated as a linear discriminant classifier in this n-dimensional space. It is proven that, when a symmetric and positive definite kernel is given, the extended SVM becomes completely identical with SVM.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 183-188

To validate the effectiveness of the extended SVM, 3D object recognition experiments are performed. As a result, the extended SVM with asymmetric kernels outperformed SVM with symmetric kernels.

The paper is organized as follows: In Sec. 2, the conventional SVM is briefly reviewed. In Sec. 3, the extended SVM is proposed. In Sec. 4, the connection of the extended SVM to SVM is shown. In Sec. 5, the effectiveness of the extended SVM is validated through 3D object recognition experiments. Sec. 6 is the conclusion.

2. Support Vector Machine

SVM consists of the feature extraction using the eigenfunctions of a kernel, and the optimal hyperplane classifier (OHC). Since very high dimensional features are extracted by the eigenfunctions, the linear separability of classes is improved, which makes the classification accuracy so high. In this section, we explain the feature extraction only and OHC is omitted because it has nothing to do with our extension.

Let the input space be $\Re^p$. Define a positive semidefinite kernel function $K(\mathbf{x}, \mathbf{y})$ on $\Re^p \times \Re^p$. Let the positive eigenvalues of K be $\eta_i (i = 1, \cdots, q)$ and the eigenfunctions be φ_i. According to the Mercer's theorem[1], the following equation holds:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{q} \eta_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{y}). \tag{2}$$

We define the nonlinear mapping $\mathcal{G} : \Re^p \rightarrow \Re^q$ as follows:

$$\mathcal{G}\mathbf{x} = (\sqrt{\eta_1}\varphi_1(\mathbf{x}), \cdots, \sqrt{\eta_q}\varphi_q(\mathbf{x}))^T. \tag{3}$$

The image space of $\mathcal{G}$ is the feature space where OHC is applied. By Eq. 2, the dot product of the images of two points $\mathbf{x}, \mathbf{y}$ is equal to $K(\mathbf{x}, \mathbf{y})$:

$$(\mathcal{G}\mathbf{x})^T(\mathcal{G}\mathbf{y}) = K(\mathbf{x}, \mathbf{y}). \tag{4}$$

In general, q is much larger than p[1]. So, the feature space is a very high dimensional space.

3. Extension of SVM

In this section, we propose an extension of SVM which can be applied to asymmetric kernels. Here, the kernel function $K(\mathbf{x}, \mathbf{y})$ can be any real function defined on $\Re^p \times \Sigma$, where Σ denotes the finite set of the training samples:

$$\Sigma = \{\mathbf{s}_1, \cdots, \mathbf{s}_n\}. \tag{5}$$

The extended SVM also consists of feature extraction and OHC. Since the classifier used in the feature space is the same as SVM, the difference lies only in the feature extraction part.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 183-188

The mapping of feature extraction is described as $W\mathcal{H}$, which is the combination of a nonlinear mapping $\mathcal{H} : \Re^p \rightarrow \Re^n$ and a linear mapping $W : \Re^n \rightarrow \Re^m$. $\mathcal{H}$ is described as follows:

$$\mathcal{H}\mathbf{x} = (K(\mathbf{x}, \mathbf{s}_1), \cdots, K(\mathbf{x}, \mathbf{s}_n))^T. \tag{6}$$

This vector can be obtained by substituting $\mathbf{x}$ into the kernel functions centered on the training samples.

The image of training sample $\mathbf{s}_i$ is denoted as $\mathcal{H}\mathbf{s}_i$. Let S denote an $n \times n$ matrix whose i-th column vector is $\mathcal{H}\mathbf{s}_i$. Let $m = \mathrm{Rank}(S)$, then the singular value decomposition of S is described as

$$S = ULV^T, \tag{7}$$

where L is an $m \times m$ diagonal matrix whose diagonal elements are the positive singular values $\lambda_1, \cdots, \lambda_m$, U is an $n \times m$ matrix whose column vectors are left singular vectors, and V is an $n \times m$ matrix whose column vectors are right singular vectors. Let us assume $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m > 0$.

The linear mapping W is described as follows:

$$W = L^{-1/2}U^T. \tag{8}$$

This mapping has a whitening effect[3]. The distribution of the training samples becomes "nearly" spherical by this mapping. The correlation matrix Q of the training samples $\mathcal{H}\mathbf{s}_i$ is described as

$$Q = \frac{1}{n}SS^T. \tag{9}$$

By substituting Eq. 7 into Eq. 9, we have

$$Q = \frac{1}{n}UL^2U^T. \tag{10}$$

This equation shows the eigendecomposition of Q. Let the eigenvalues of Q be $a_1, \cdots, a_m$ in a descending order. By Eq. 10, $a_i = \lambda_i^2/n$. The correlation matrix Q_t of the mapped training sample $W\mathcal{H}\mathbf{s}_i$ is described as

$$Q_t = \frac{1}{n}WSS^TW^T = \frac{1}{n}L. \tag{11}$$

Then, the i-th eigenvalue of Q_t is $\lambda_i/n = \sqrt{a_i/n}$. Since

$$\frac{\sqrt{a_1/n}}{\sqrt{a_n/n}} \leq \frac{a_1}{a_n},$$

the ratio of the largest eigenvalue to the smallest eigenvalue is decreased by the mapping W. Thus, the distribution of the training samples is sphered in a certain degree. The mapping $\mathcal{W}$ increases the *effective* dimensionality of the feature space, because it makes the training samples spread over all

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 183-188

dimensions. In usual whitening, the distribution becomes completely spherical ($Q_t = 1/nI$, where I is a unit matrix). Since the mapping W does not whiten the distribution completely, we call this mapping "half whitening".

We apply OHC in the m-dimensional feature space obtained by $W\mathcal{H}$. Although this feature space seems completely different from the one described in Sec. 2, the two feature spaces become identical in terms of dot product, when K is positive definite.

4. Connection to SVM

In this section, we prove that, when K is symmetric and strictly positive definite, the extended SVM is completely identical with SVM.

Both methods use OHC in the feature space. The discriminant function of OHC depends only on the dot products between the unlabeled sample and the training samples. Also, the optimization problem to train OHC depends only on the dot products between the training samples[1]. So, if these dot products are equal in both feature spaces, the two classifiers are identical.

Let an unlabeled sample be denoted as $\mathbf{z} \in \Re^p$. In the feature space of SVM, the dot product between the unlabeled sample and the training sample is described as

$$(\mathcal{G}\mathbf{z})^T(\mathcal{G}\mathbf{s}_i) = K(\mathbf{z}, \mathbf{s}_i). \tag{12}$$

On the other hand, in the feature space of the extended SVM, the dot product is described as

$$(W\mathcal{H}\mathbf{z})^T(W\mathcal{H}\mathbf{s}_i) = (\mathcal{H}\mathbf{z})^T U L^{-1} U^T \mathcal{H}\mathbf{s}_i. \tag{13}$$

When the kernel K is symmetric and strictly positive definite, the matrix S is symmetric, full rank ($m = n$) and $U = V$. Accordingly, $S = ULU^T$ and thus

$$S^{-1} = UL^{-1}U^T. \tag{14}$$

Substituting this into Eq. 13,

$$(\mathcal{H}\mathbf{z})^T S^{-1}(\mathcal{H}\mathbf{s}_i) = (\mathcal{H}\mathbf{z})^T \mathbf{e}_i = K(\mathbf{z}, \mathbf{s}_i), \tag{15}$$

where $\mathbf{e}_i$ is an n-dimensional vector whose i-th element is 1 and all the other elements are 0.

From Eq. 12 and 15, it is shown that the dot product between the unlabeled sample and the training sample is equal in both feature spaces. With regard to the dot product between the training samples, the proof can be done in the same way. Therefore, the extend SVM and SVM is completely identical when the kernel is symmetric and positive definite.

When the kernel is positive semidefinite, the connection cannot be proven for any n, because S may not be full rank. But, when the nonlinearity of the kernel is high ($n \ll q$), the possibility that S is full rank is very high. In such cases, there exists the connection.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 183-188

Figure 1: 3D shapes used in the experiment

5. 3D Object Recognition Experiment

In this section, the performance of the extended SVM is compared with SVM in 3D object recognition experiments. Eight kinds of polygon shape models shown in Fig. 1 were used. Each shape was rotated to a random direction around the center of gravity, and 80 images were obtained for each shape. Among them, 40 were used for training and 40 were used for testing. The choice of samples was performed randomly. We performed the experiment 10 times with different choice of samples. Every error rate shown below is the average over 10 times.

In extracting features from the image, we used the elongated Gaussian receptive fields[4], which resemble the receptive fields in human retina. By this feature extraction, the directions of edges can be taken into account. The i-th feature value is obtained by the convolution of an ellipsoidal Gaussian filter which has random position and direction. In this experiment, the length ratio of the two axis of the ellipsoid was set to $3:1$. The number of filters was set to 200, so we had the 200-dimensional input space ($p = 200$).

We used a Gaussian kernel function as follows:

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\| \mathbf{x} - \mathbf{y} \|^2}{\sigma(\mathbf{y})^2}). \tag{16}$$

For SVM, the width function $\sigma(\mathbf{y})$ is fixed to a constant value σ. Then, K is a symmetric and strictly positive definite function defined on $\Re^p \times \Re^p$.

For the extended SVM, the width function is defined in Σ, then K is a function defined in $\Re^p \times \Sigma$, which is asymmetric in $\Sigma \times \Sigma$. The width on a training sample is determined by the following heuristics[2]: Find the distance to the nearest training sample which belongs to a different class and assign this value multiplied by β to the width.

Since SVM is a binary classifier, there are several ways to apply it to multiclass problems. In this experiment, a SVM is associated to a class. The SVM of a class is trained by the training samples of the class and the ones of all the other classes. In classification of an unlabeled sample, the output values of SVMs are compared and the unlabeled sample is classified to the class with the largest value.

We obtained the error rates of SVM and the extended SVM with various parameter values. The results are shown in Tab. 1 and 2. The smallest error rate of SVM was 8.72% and that of the extended SVM was 7.78%. This result confirms the effectiveness of the extended SVM.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 183-188

Table 1: Error rates of SVM against width parameter σ

σ ($\times 10^6$)	0.5	0.75	1.0	1.25	1.5	1.75
Error Rate (%)	12.1	9.09	8.72	8.90	9.25	9.47

Table 2: Error rates of the extended SVM against width ratio β

β	0.8	1.0	1.2	1.4	1.6	1.8
Error Rate (%)	9.15	8.18	7.78	7.90	8.06	8.28

6. Conclusion

In this paper, we extended SVM to be applicable to asymmetric kernels. It is proven that, when the kernel is positive definite, the extended SVM becomes identical with SVM. In the 3D object recognition experiment, the effectiveness of this method is validated.

So far, SVM is formulated using so-called "kernel trick". Here, the dimensionality of the feature space is very high, sometimes infinite. But, since the number of training samples is n, only n-dimensional subspace works for classification. This formulation seems elegant, but it is not a good formulation which captures the nature of the classifier, because it contains many unnecessary dimensions.

The extended SVM gives another formulation of SVM. Here, SVM consists of the mapping to n-dimensional space, half whitening and OHC. This formulation is by far simpler than the conventional one, and it does not contain unnecessary dimensions. Thus, this formulation should be preferred, especially by non-mathematicians.

References

[1] V. N. Vapnik: "The Nature of Statistical Learning Theory", Springer-Verlag (1995).

[2] Y.-S. Hwang and S.-Y. Bang: "An efficient method to construct a radial basis function neural network classifier", Neural Networks, **10**, 8, pp. 1495–1503 (1997).

[3] E. Oja: "Pca, ica and nonlinear hebbian learning", ICANN'95, pp. 89–94 (1995).

[4] Y. Weiss and S. Edelman: "Representation with receptive fields: Gearing up for recognition", CS-TR 93-09, Weitzmann Institute of Science (1993).