2006 Autumn semester Pattern Information Processing
Topic 4. Pattern recognition

Session 12. (3) Support vector machine and kernel method

Support vector machine is a method of obtaining the optimal boundary of two sets in a vector space independently on the probabilistic distributions of training vectors in the sets. Its fundamental idea is very simple; locating the boundary that is most distant from the vectors nearest to the boundary in both of the sets. This simple idea is a traditional one, however, recently has attracted much attention again. This is because of the introduction of *kernel method*, which is equivalent to a transformation of the vector space for locating a nonlinear boundary.

Basic support vector machine

We assume at first a linearly separable problem, as shown in Fig. 1. Our aim is finding the "optimal" boundary hyperplane which exactly separates one set from the other. Note that our "optimal" boundary hyperplane should classify not only the training vectors, but also unknown vectors in each set. In the first session of this topic, the classification method by estimating probabilistic distributions of the vectors was explained. However, an accurate estimation is difficult since the dimension of vectors is often much higher than the number of training vectors. It was refered as "curse of dimensionality" also in that session.

Now we try another simple approach without any estimation of distribution. In this approach, the "optimal" boundary is defined as the most distant hyperplane from both sets. In other words, this boundary passes the "midpoint" between these sets. Although the distribution of each set is unknown, this boundary is expected to be the optimal classification of the sets, since this boundary is the most isolated one from both of the sets. The training vectors closest to the boundary are called *support vectors.*

Such boundary is defined to be passing through the midpoint of the shortest line segment between the convex hulls of the sets and is orthogonal to the line segment.

Let $\boldsymbol{x}$ be a vector in a vector space. A boundary hyperplane is expressed as one of the hyperplanes

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0, \tag{1}$$

where $\boldsymbol{w}$ is a weight coefficient vector and b is a bias term. The distance between a training vector $\boldsymbol{x}_i$ and the boundary, called *margin,* is expressed as follows:

$$\frac{|\boldsymbol{w}^T\boldsymbol{x}_i + b|}{\|\boldsymbol{w}\|}. \tag{2}$$

Since the hyperplanes expressed by Eq. (1) where $\boldsymbol{w}$ and b are multiplied by a common constant are identical, we introduce a restriction to this expression, as follows:

$$\min_i |\boldsymbol{w}^T\boldsymbol{x}_i + b| = 1. \tag{3}$$

The optimal boundary maximizes the minimum of Eq. (2). By the restriction of Eq. (3), this is reduced to the maximization of $1/\|\boldsymbol{w}\|^2 = 1/\boldsymbol{w}^T\boldsymbol{w}$. Consequently, the optimization is formalized as

$$\begin{array}{ll} \textit{minimize} & \boldsymbol{w}^T\boldsymbol{w} \\ \textit{subject to} & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \end{array} \tag{4}$$

where y_i is 1 if $\boldsymbol{x}_i$ belongs to one set and -1 if $\boldsymbol{x}_i$ belongs to the other set. If the boundary classifies the vectors correctly, $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)$ and it is identical to the margin.

This conditional optimization is achieved by Lagrange's method of indeterminate coefficient. Let us define a function

$$L(\boldsymbol{w}, b, \alpha_i) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_i \alpha_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1], \tag{5}$$

where $\alpha_i \geq 0$ are the indeterminate coefficients. If $\boldsymbol{w}$ and b take the optimal value, the partial derivatives

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_i \alpha_i y_i \boldsymbol{x}_i \\ \frac{\partial L}{\partial b} &= -\sum_i \alpha_i y_i \end{aligned} \tag{6}$$

are zero. Setting the derivatives of Eq. (6) to zero, we get

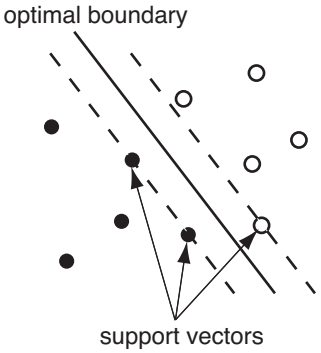

Fig. 1: Optimal boundary by support vector machine.

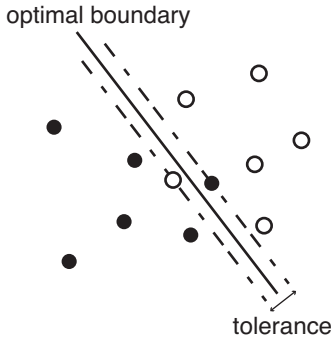

Fig. 2: Linearly nonseparable case.

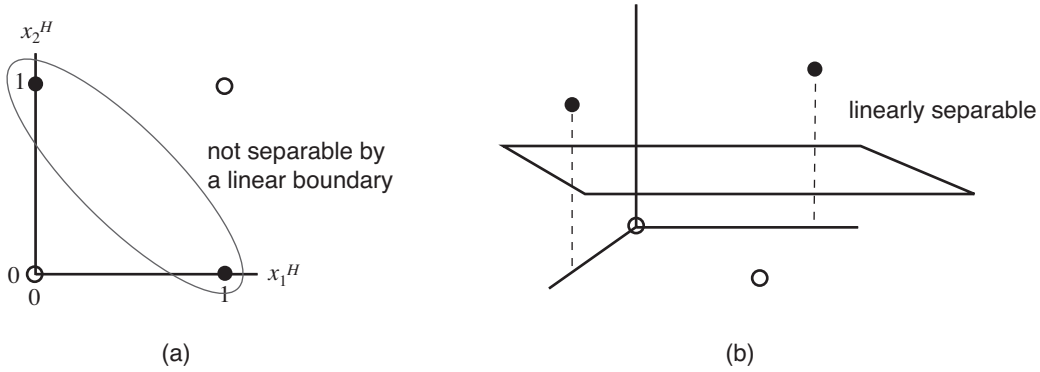

Fig. 3: Transformation to higher dimensional space.

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i, \tag{7}$$

$$\sum_i \alpha_i y_i = 0. \tag{8}$$

Rewriting Eq. (5), we get

$$L(\boldsymbol{w}, b, \alpha_i) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_i \alpha_i y_i \boldsymbol{w}^T \boldsymbol{x}_i - b\sum_i \alpha_i y_i \sum_i \alpha_i. \tag{9}$$

Substituting Eqs. (7) and (8) to Eq. (5), we get

$$\begin{aligned} L(\boldsymbol{w}, b, \alpha_i) &= \frac{1}{2}\left(\sum_i \alpha_i y_i \boldsymbol{x}_i\right)^T \left(\sum_j \alpha_j y_j \boldsymbol{x}_j\right) \\ &\quad - \sum_i \alpha_i y_i \left(\sum_j \alpha_j y_j \boldsymbol{x}_j\right)^T \boldsymbol{x}_i + \sum_i \alpha_i \\ &= -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_i \alpha_i \end{aligned} \tag{10}$$

The contribution of the second term of Eq. (5) should be minimum, and L should be maximized subject to α. Consequently, the optimization is reduced to a quadratic programming problem as follows:

$$\begin{array}{ll} \textit{maximize} & -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_i \alpha_i \\ \textit{subject to} & \sum_i \alpha_i y_i = 0, \alpha_i \geq 0 \end{array} \tag{11}$$

Many software packages for solving the quadratic programming problem are commercially available.

Soft margin

The above discussion is applicable to the case of linearly separable sets only. If the sets are not linearly separable, a hyperplane exactly classifying the sets does not exist, as explained in the previous session.

The method called *soft margin* is a solution to such case. This method replaces the restriction in Eq. (4)

with the following:

$$subject\ to \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i. \tag{12}$$

where ξ_i , called *slack variables,* are positive variables that indicate tolerances of misclassification. This replacement indicates that a training vector is allowed to exist in a limited region in the erroneous side along the boundary, as shown in Fig. 2. Several optimization functions are proposed for this case, for example

$$minimize \quad \boldsymbol{w}^T\boldsymbol{w} + C\sum_i \xi_i. \tag{13}$$

The second term of the above expression is a penalty term for misclassification, and the constant C determines the degree of contribution of the second term.

Kernel method

The soft margin method is an extension of the support vector machine within the linear framework. The *kernel method* explained here is a method of finding truly nonlinear boundaries.

The fundamental concept of kernel method is a deformation of the vector space itself to a higher dimensional space. We consider the linearly nonseparable example presented in the previous session, as shown in Fig. 3(a). If the two-dimensional space is transformed to the threedimensional one as shown in Fig. 3(b), "black" vectors and "white" vectors are linearly separable.

Let Φ be a transformation to a higher dimensional space. The transformed space should satisfy that the distance is defined in the transformed space and the distance has a relationship to the distance in the original space. The *kernel function* $K(\boldsymbol{x}, \boldsymbol{x}')$ is introduced for satisfying the above conditions. The kernel function satisfies

$$K(\boldsymbol{x}, \boldsymbol{x}') = \Phi(\boldsymbol{x})^T\Phi(\boldsymbol{x}'). \tag{14}$$

The above equation indicates that the kernel function is equivalent to the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ measured in the higher dimensional space transformed by Φ. If we measure the margin by the kernel function and perform the optimization, a nonlinear boundary is obtained. Note that the boundary in the transformed space is obtained as

$$\boldsymbol{w}^T\Phi(\boldsymbol{x}) + b = 0. \tag{15}$$

Substituting Eq. (7) into the above equation with replacing $\boldsymbol{x}$ with $\Phi(\boldsymbol{x})$, we get

$$\sum_i \alpha_i y_i \Phi(\boldsymbol{x}_i^T)\Phi(\boldsymbol{x}) + b = \sum_i \alpha_i y_i K(\boldsymbol{x}_i^T\boldsymbol{x}) + b = 0. \tag{16}$$

The optimization function of Eq. (11) in the transformed space is also obtained by substituting $\boldsymbol{x}_i^T\boldsymbol{x}_j$ with $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. These results mean that all the calculation can be achieved by using $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ only, and we do not need to know what Φ or the transformed space actually is.

A sufficient condition for satisfying Eq. (14) is that K is positive definite. Several example of such kernel functions are known, as follows:

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T\boldsymbol{x}' + 1)^p \quad \text{(polynomial kernel)}, \tag{17}$$

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{\sigma^2}) \quad \text{(Gaussian kernel)}. \tag{18}$$

Empirical risk and expected risk

The term *empirical risk* means the misclassification rate for *known* training vectors. It is not what we want to minimize; Our objective is minimizing the misclassification rate for *all* vectors in each set, including *unknown* vectors. This misclassification rate is called *expected risk.*

In case of linearly separable problems, there exists a boundary hyperplane that makes the empirical risk zero. The concept of support vector machine to find the boundary with the largest margin is equivalent to selecting a hyperplane minimizing the expected risk, from the set of hyperplanes that makes the empirical risk zero. This is formally explained in the framework of *structural risk minimization* with the concept of *Vapnik- Chervenenkis (VC) dimensionality.*

Reference

K. K. Chin, "Support Vector Machines applied to Speech Pattern Classification," MPhil. thesis, http://svr-www.eng.cam.ac.uk/~kkc21/thesis_main/thesis_main.html

津田宏治, "サポートベクターマシンとは何か," 信学誌, 83, 6, 460 - 466 (2000).