Chapter 2
Support Vector Machines

Abstract In this chapter, we study support vector machines (SVM). We will see that optimization methodology plays an important role in building and training of SVM.

2.1 Basic SVM

One fundamental function of machine learning is to make classification from a set of labeled training data. Suppose these data samples are denoted as $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, L = L^+ + L^-\}$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ are feature vectors and $y_i \in \{-1, +1\}$ are the labels. If these two kinds of examples formulate two disjoint convex hulls in $\mathbb{R}^n$, we can find a hyperplane $\{\boldsymbol{x} \mid \boldsymbol{w}^T\boldsymbol{x} + b = 0\}$ to separate them, because of the strong separation theorem. This indeed gives a classification function

$$H(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b \tag{2.1}$$

Any point $\boldsymbol{x}$ giving $H(\boldsymbol{x}) > 0$ will be recognized as Class I and any point $\boldsymbol{x}$ giving $H(\boldsymbol{x}) < 0$ will be recognized as Class II.

There might be infinite such hyperplanes that can separate these two convex sets. Here, we would like to find the separating hyperplane which has the largest distance to two convex sets. This will lead to the following optimization problem

$$\max_{\boldsymbol{w},b} \quad \min \quad \left\{|\boldsymbol{x} - \boldsymbol{x}_i|_2^2 \mid \boldsymbol{w}^T\boldsymbol{x} + b = 0,\ i = 1, \ldots, L\right\} \tag{2.2}$$

$$\text{s.t.} \quad \boldsymbol{w}^T\boldsymbol{x}_j + b \geq 1,\ j = 1, \ldots, L^+ \tag{2.3}$$

$$\boldsymbol{w}^T\boldsymbol{x}_k + b \leq -1,\ k = 1, \ldots, L^- \tag{2.4}$$

However, this optimization problem is not easy to solve. Therefore, we should consider its equivalent form instead.

Notice that the parameters $\boldsymbol{w}$ and b can be rescaled in such a way that the points closest to the hyperplane $\{\boldsymbol{x} \mid \boldsymbol{w}^T\boldsymbol{x} + b = 0\}$ must lie on either the hyperplane $\{\boldsymbol{x} \mid \boldsymbol{w}^T\boldsymbol{x} + b = +1\}$ or the hyperplane $\{\boldsymbol{x} \mid \boldsymbol{w}^T\boldsymbol{x} + b = -1\}$. Meanwhile, the distance between these two parallel hyperplanes can be gotten as $\frac{2}{|\boldsymbol{w}|_2^2}$; see Fig. 2.1.

L. Li, *Selected Applications of Convex Optimization*, Springer Optimization and Its Applications 103, DOI 10.1007/978-3-662-46356-7_2

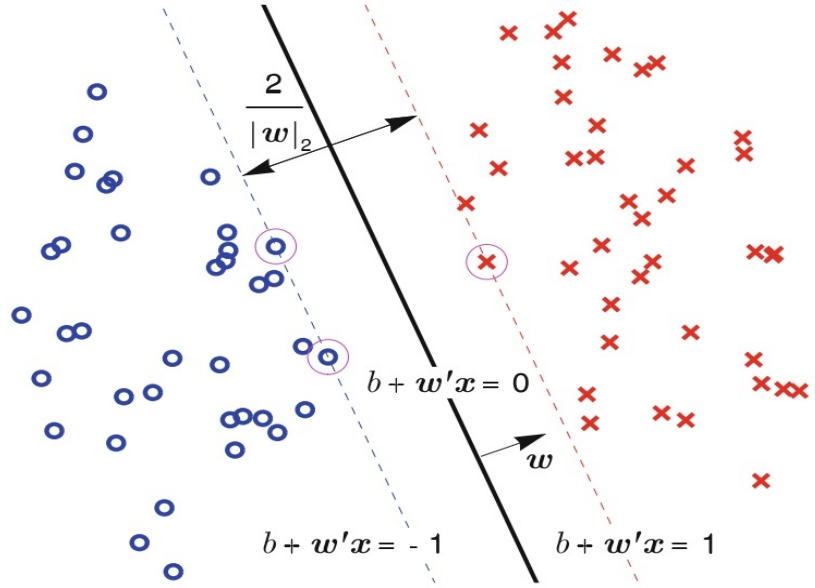

Fig. 2.1 An illustration of support vector machines for binary classification

So, we can reach an optimization problem equivalent to (2.2)–(2.4) as

$$\max_{w,b} \quad \frac{2}{|w|_2^2} \tag{2.5}$$

$$\text{s.t.} \quad w^T x_j + b \geq 1, \; j = 1, \ldots, L^+ \tag{2.6}$$

$$w^T x_k + b \leq -1, \; k = 1, \ldots, L^- \tag{2.7}$$

This is still not a convex optimization problem, so we consider its corresponding minimization problem

$$\min_{w,b} \quad \frac{1}{2} |w|_2^2 \tag{2.8}$$

$$\text{s.t.} \quad w^T x_j + b \geq 1, \; j = 1, \ldots, L^+ \tag{2.9}$$

$$w^T x_k + b \leq -1, \; k = 1, \ldots, L^- \tag{2.10}$$

In other words, we turn a classification problem into a convex optimization problem. This optimization problem can be viewed as the primal form of the basic *Supporting Vector Machines* (SVM) [1–7].

Introducing the sign variable y_j and y_k as

$$y_j = 1 \text{ for } \; w^T x_j + b \geq 1, \quad j = 1, \ldots, L^+ \tag{2.11}$$

$$y_k = -1 \text{ for } w^T x_k + b \leq -1, \; k = 1, \ldots, L^- \tag{2.12}$$

we can further rewrite (2.11)–(2.12) into a uniform constraint

$$y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) \geq 1, \; i = 1, \ldots, L \tag{2.13}$$

We can directly calculate the best parameters $\boldsymbol{w}^*$ and b^* of this SVM by solving this primal problem. The sample Matlab code snippet of SVM in primal form is given below.

```
function [w, b] = svm_prim_sep(data, labels)
% INPUT
%  data:   num-by-dim matrix. num is the number of data points,
%          dim is the dimension of a point
%  labels: num-by-1 vector, specifying the class that each point
%          belongs to.
%          either be +1 or be -1
% OUTPUT
%  w:      dim-by-1 vector, the normal direction of hyperplane
%  b:      a scalar, the bias
    [num, dim] = size(data);

    cvx_begin
        variables w(dim) b;
        minimize(norm(w));
        subject to
            labels .* (data * w + b) >= 1;
    cvx_end
end
```

Since it is a convex optimization problem, we can also attack its dual problem instead; see discussions in Sect. 1.3.2. To get its dual problem, let us first write down the generalized Lagrangian function as

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} |\boldsymbol{w}|_2^2 - \sum_{i=1}^{L} \alpha_i \left[y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) - 1\right] \tag{2.14}$$

where $\boldsymbol{\alpha} \in \mathbb{R}_+^L$ are the associated Lagrange multipliers (dual variables).

Considering the partial derivatives of Lagrangian function with respect to $\boldsymbol{w}$ and b as zero, based on KKT conditions for differentiable convex problems, we have

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i \tag{2.15}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \implies \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.16}$$

Further eliminating the primal decision variables $\boldsymbol{w}$ and b, we have the objective of the Lagrange dual problem as

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{w}^T \left[\sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i\right] - \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{w}^T \boldsymbol{x}_i \\ &\quad - \sum_{i=1}^{L} \alpha_i y_i b + \sum_{i=1}^{L} \alpha_i \qquad (2.17) \\ &= -\frac{1}{2}\boldsymbol{w}^T \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i - \sum_{i=1}^{L} \alpha_i y_i b + \sum_{i=1}^{L} \alpha_i \\ &= -\frac{1}{2}\left[\sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i\right]^T \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i - b\left[\sum_{i=1}^{L} \alpha_i y_i\right] + \sum_{i=1}^{L} \alpha_i \qquad (2.18) \\ &= \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T \boldsymbol{x}_j \qquad (2.19) \end{aligned}$$

where we substitute (2.15) in (2.17) and (2.18).

Therefore, the integrate dual problem can be written as

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T \boldsymbol{x}_j \qquad (2.20)$$

$$\text{s.t.} \quad \alpha_i \geq 0 \qquad (2.21)$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \qquad (2.22)$$

where (2.22) directly inherits from (2.16).

Suppose the solutions to the primal/dual optimization problem are denoted as $\boldsymbol{w}^*, b^*, \boldsymbol{\alpha}^*$ The nonzero dual variables $\alpha_i^* > 0$ are called support vectors. The complementary slackness condition in KKT conditions implies that

$$\alpha_i^* \left[y_i \left(\boldsymbol{x}_i^T \boldsymbol{w}^* + b^*\right) - 1\right] = 0 \qquad (2.23)$$

This indicates that the constraints (2.9)–(2.10) are active with equality for all the support vectors. In other words, the support vectors are lying on the hyperplanes $\{\boldsymbol{x} \mid \boldsymbol{w}^{*T}\boldsymbol{x} + b = \pm 1\}$.

The parameter $\boldsymbol{w}^*$ is then recovered from the solution α^* of the dual optimization problem.

$$\boldsymbol{w}^* = \sum_i \alpha_i^* y_i \boldsymbol{x}_i \tag{2.24}$$

Moreover, for the few $\alpha_i > 0$, the corresponding $\boldsymbol{x}_i$ satisfies $y_i \left[(\boldsymbol{w}^*)^T \boldsymbol{x}_i + b^* \right] = 1$. This means

$$(\boldsymbol{w}^*)^T \boldsymbol{x}_i + b^* = 1/y_i = y_i \implies b^* = y_i - (\boldsymbol{w}^*)^T \boldsymbol{x}_i \tag{2.25}$$

In practice, it is more robust to average over all support vectors and calculate b^* as

$$b^* = \frac{1}{|S|} \sum_{i \in S} \left[y_i - (\boldsymbol{w}^*)^T \boldsymbol{x}_i \right] \tag{2.26}$$

where S denotes the set of the indices of all support vectors and $|S|$ is the cardinality of S.

For any new sample $\boldsymbol{z}$, the decision of classification can be given as

$$\text{sign}\left[(\boldsymbol{w}^*)^T \boldsymbol{z} + b^* \right] = \text{sign}\left(\sum_{i=1}^{L} \alpha_i^* y_i \boldsymbol{x}_i^T \boldsymbol{z} + b^* \right) \tag{2.27}$$

The sample Matlab code snippet of SVM in dual form is given below.

```
function [w, b, alpha] = svm_dual_sep(data, labels)
% INPUT
%  data:   num-by-dim matrix. num is the number of data points,
%          dim is the dimension of a point
%  labels: num-by-1 vector, specifying the class that each point
%          belongs to.
%          either be +1 or be -1
% OUTPUT
%  w:      dim-by-1 vector, the normal direction of hyperplane
%  b:      a scalar, the bias
%  alpha:  num-by-1 vector, dual variables
    [num, ~] = size(data);
    H = (data * data') .* (labels * labels');

    cvx_begin
        variable alpha(num);
        maximize(sum(alpha) - alpha' * H * alpha / 2);
        subject to
            alpha >= 0;
            labels' * alpha == 0
    cvx_end

    sv_ind = alpha > 1e-4;
    w = data' * (alpha .* labels);
```

```
    b = mean(labels(sv_ind) - data(sv_ind, :) * w);
end
```

2.2 Soft Margin SVM

If the data samples are not linearly separable, we could still build a linear classifier likewise. To make the classification errors as small as possible, we usually introduce a loss function (penalty function) on the classification errors.

The basic loss function is a linear function of loss on the violation. Using it, we can then formulate a soft margin SVM as follows:

$$\min_{\boldsymbol{w},b,\xi_i} \quad \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i \tag{2.28}$$

$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1 - \xi_i,\ i = 1,\ldots,L \tag{2.29}$$

$$\xi_i \geq 0,\ i = 1,\ldots,L \tag{2.30}$$

where $C \in \mathbb{R}^+$ is the penalty coefficient and $\xi_i \in \mathbb{R}^+, i = 1,\ldots,L$ are the degree of violation for each data sample; see Fig. 2.2.

Clearly, this is still a convex optimization problem, and we can form the generalized Lagrangian function as

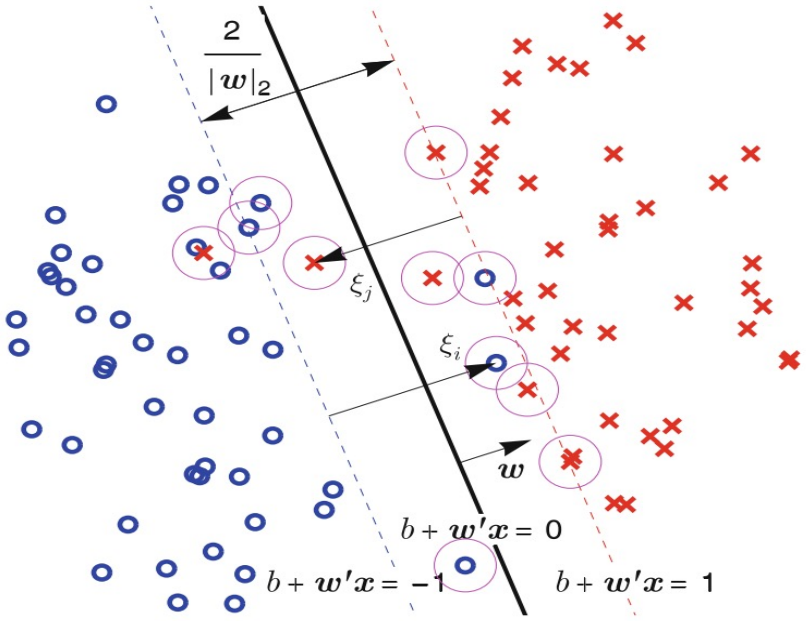

Fig. 2.2 An illustration of soft margin support vector machines

$$L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{L}\beta_i\xi_i \tag{2.31}$$

where $\boldsymbol{\alpha} \in \mathbb{R}_+^L$ and $\boldsymbol{\beta} \in \mathbb{R}_+^L$ are the associated Lagrange multipliers.

Letting its partial derivatives with respect to $\boldsymbol{w}$, b, and ξ_i be zero, we have

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_{i=1}^{L}\alpha_i y_i \boldsymbol{x}_i \tag{2.32}$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = 0 \implies \sum_{i=1}^{L}\alpha_i y_i = 0 \tag{2.33}$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = 0 \implies C - \alpha_i - \beta_i = 0 \tag{2.34}$$

Eliminating the primal decision variables $\boldsymbol{w}$, b, and ξ_i, we have the objective of the Lagrange dual problem as

$$\begin{aligned} L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{L}\beta_i\xi_i \\ &= \frac{1}{2}|\boldsymbol{w}|_2^2 - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1\right] + \sum_{i=1}^{L}\xi_i\left[C - \alpha_i - \beta_i\right] \\ &= \frac{1}{2}|\boldsymbol{w}|_2^2 - \sum_{i=1}^{L}\left[\alpha_i y_i \boldsymbol{w}^T\boldsymbol{x}_i - \alpha_i\right] - \left(\sum_{i=1}^{L}\alpha_i y_i\right) b \\ &= \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T\boldsymbol{x}_j \end{aligned} \tag{2.35}$$

where we substitute (2.32) in the last step.

The whole dual problem can be written as

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T\boldsymbol{x}_j \tag{2.36}$$

$$\text{s.t.} \quad \alpha_i \geq 0,\ \beta_i \geq 0,\ \alpha_i + \beta_i = C \tag{2.37}$$

$$\sum_{i=1}^{L}\alpha_i y_i = 0 \tag{2.38}$$

Since the objective function of the Lagrange dual problem does not include $\boldsymbol{\beta}$, we have

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{2.39}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C \tag{2.40}$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.41}$$

We can see that the dual problem remains almost the same as the dual problem (2.20)–(2.22), except we have a new upper bound for the dual variable $\boldsymbol{\alpha}$. That is, the key advantage of a linear penalty function is that the slack variables vanish from the dual problem, with the constant C appearing only as an additional constraint on the Lagrange multipliers. Cortes and Vapnik received the 2008 ACM Paris Kanellakis Award for the above formulation and its huge impact in practice [8].

Suppose the solutions to the primal/dual optimization problem are denoted as $\boldsymbol{w}^*, b^*, \boldsymbol{\alpha}^*$. The nonzero dual variables $\alpha_i^* > 0$ are called support vectors. The complementary slackness condition in KKT conditions implies that

$$\alpha_i^* \left[y_i \left(\boldsymbol{w}^* \right)^T \boldsymbol{x}_i + y_i b^* - 1 + \xi_i \right] = 0 \tag{2.42}$$

This indicates that the support vectors correspond to data samples that are wrongly classified or lie right on the hyperplanes $\{\boldsymbol{x} \mid \boldsymbol{w}^{*T} \boldsymbol{x} + b = \pm 1\}$.

Moreover, for the few $\alpha_i > 0$, the corresponding $\boldsymbol{x}_i$ satisfies $y_i \left[(\boldsymbol{w}^*)^T \boldsymbol{x}_i + b^* \right] = 1 - \xi_i$. This means

$$\begin{aligned} \left(\boldsymbol{w}^* \right)^T \boldsymbol{x}_i + b^* &= (1 - \xi_i)/y_i = y_i - y_i \xi_i \\ &\Longrightarrow b^* = y_i - y_i \xi_i - \left(\boldsymbol{w}^* \right)^T \boldsymbol{x}_i \end{aligned} \tag{2.43}$$

Since we do not know ξ_i in the dual problem, it is more robust to average over all support vectors with $\xi_i = 0$ and $i \in S$, and calculate b^* as

$$b^* = \frac{1}{|S|} \sum_{i \in S} \left[y_i - \left(\boldsymbol{w}^* \right)^T \boldsymbol{x}_i \right] \tag{2.44}$$

where S denotes the set of the indices of all support vectors with $\xi_i = 0, i \in S$, $|S|$ is the cardinality of S.

For any new sample $\boldsymbol{z}$, the decision of classification is still given as (2.27).

We can use different penalty functions to construct different soft margin SVMs. Usually, we require the penalty function to be a convex function; otherwise, we

cannot use convex optimization techniques to solve it. Moreover, we require the value of penalty function to be 0 when the classification result is correct.

For example, we can consider a quadratic form of penalty function

$$\min_{\boldsymbol{w},b,\xi_i} \quad \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i^2 \tag{2.45}$$

$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1 - \xi_i,\ i = 1,\ldots,L \tag{2.46}$$

$$\xi_i \geq 0,\ i = 1,\ldots,L \tag{2.47}$$

where $C \in \mathbb{R}^+$ is the penalty coefficient and $\xi_i \in \mathbb{R}^+, i = 1,\ldots,L$ are the degree of violation for each data sample.

The sample Matlab code snippet of soft margin SVM in primal form (2.45)–(2.47) is given below.

```
function [w, b] = svm_prim_nonsep2(data, labels, C)
% INPUT
%  data:   num-by-dim matrix. num is the number of data points,
%          dim is the dimension of a point
%  labels: num-by-1 vector, specifying the class that each point
%          belongs to.
%          either be +1 or be -1
%  C:      the tuning parameter
% OUTPUT
%  w:      dim-by-1 vector, the normal direction of hyperplane
%  b:      a scalar, the bias
    [num, dim] = size(data);

    cvx_begin
    variables w(dim) b xi(num);
        minimize(sum(w.^2) / 2 + C * sum(xi.^2));
        subject to
            labels .* (data * w + b) >= 1 - xi;
            xi >= 0;
    cvx_end
end
```

Clearly, this is still a convex optimization problem, and we can form the generalized Lagrangian function as

$$L(\boldsymbol{w},b,\xi_i,\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i^2 - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{L}\beta_i\xi_i \tag{2.48}$$

where $\boldsymbol{\alpha} \in \mathbb{R}_+^L$ and $\boldsymbol{\beta} \in \mathbb{R}_+^L$ are the associated Lagrange multipliers.

Letting its partial derivatives with respect to $\boldsymbol{w}$, b, and ξ_i be zero, we have

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i \tag{2.49}$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = 0 \implies \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.50}$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = 0 \implies 2C\xi_i - \alpha_i - \beta_i = 0 \tag{2.51}$$

Eliminating the primal decision variables $\boldsymbol{w}$, b, and ξ_i, we have the objective of the Lagrange dual problem as

$$\begin{aligned} L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i^2 - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{L}\beta_i\xi_i \\ &= \frac{1}{2}|\boldsymbol{w}|_2^2 - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1\right] + \sum_{i=1}^{L}\xi_i\left[C\xi_i - \alpha_i - \beta_i\right] \\ &= \frac{1}{2}|\boldsymbol{w}|_2^2 - \sum_{i=1}^{L}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1\right] - C\sum_{i=1}^{L}\xi_i^2 \\ &= \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \left(\boldsymbol{x}_i\right)^T \boldsymbol{x}_j - \frac{1}{4C}\sum_{i=1}^{L}(\alpha_i + \beta_i)^2 \end{aligned} \tag{2.52}$$

The whole dual problem can be written as

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T \boldsymbol{x}_j - \frac{1}{4C}\sum_{i=1}^{L}(\alpha_i + \beta_i)^2 \tag{2.53}$$

$$\text{s.t.} \quad \alpha_i \geq 0,\ \beta_i \geq 0 \tag{2.54}$$

$$\sum_{i=1}^{L}\alpha_i y_i = 0 \tag{2.55}$$

Clearly, the maximum value is reached when $\beta_i = 0$. So, the whole dual problem can be rewritten as

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^T \boldsymbol{x}_j - \frac{1}{4C}\sum_{i=1}^{L}\alpha_i^2 \tag{2.56}$$

$$\text{s.t.} \quad \alpha_i \geq 0 \tag{2.57}$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.58}$$

b^* is still calculated as (2.44). For any new sample z, the decision of classification is still given as (2.27).

The sample Matlab code snippet of soft margin SVM in dual form (2.56)–(2.58) is given below.

```
function [w, b, alpha] = svm_dual_nonsep2(data, labels, C)
% INPUT
%  data:   num-by-dim matrix. num is the number of data points,
%          dim is the dimension of a point
%  labels: num-by-1 vector, specifying the class that each point
%          belongs to.
%          either be +1 or be -1
%  C:      the tuning parameter
% OUTPUT
%  w:      dim-by-1 vector, the normal direction of hyperplane
%  b:      a scalar, the bias
%  alpha:  num-by-1 vector, dual variables
    [num, ~] = size(data);
    H = (data * data') .* (labels * labels');

    cvx_begin
        variable alpha(num);
        maximize(sum(alpha) - alpha' * H * alpha
        / 2 - sum(alpha.^2) / (4 * C));
        subject to
            alpha >= 0;
            labels' * alpha == 0
    cvx_end

    sv_ind = alpha > 1e-4;
    w = data' * (alpha .* labels);
    xi = alpha / (2 * C);
    b = mean(labels(sv_ind) .* (1 - xi(sv_ind)) - data
    (sv_ind, :) * w);
end
```

Obviously, when a special penalty function is chosen, we can get a special SVM in both primal and dual formats. Figure 2.3 provides an example to distinguish the difference between soft margin SVM with linear and quadratic penalty functions for classification errors. It is shown that the soft margin SVM with quadratic penalty functions will be more sensitive to outliers.

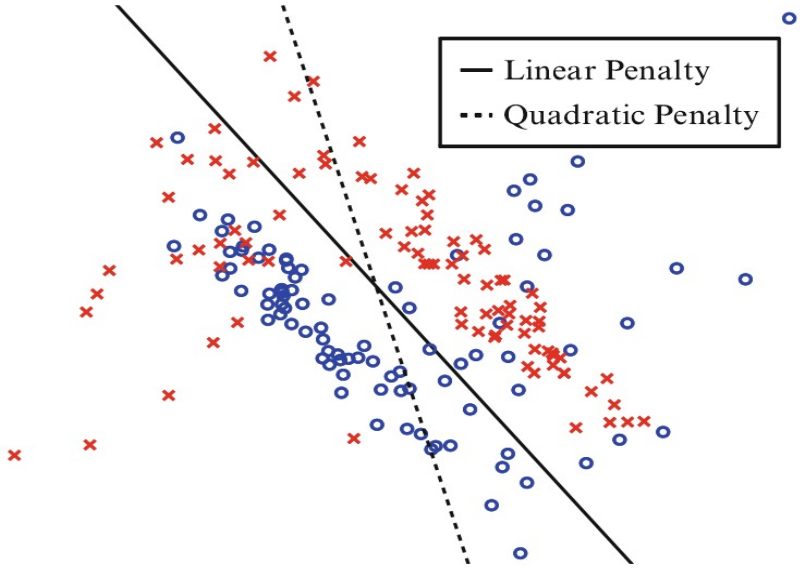

Fig. 2.3 An illustration of different classification hyperplanes found by soft margin SVM with linear and quadratic penalty functions

2.3 Kernel SVM

In many situations, we cannot separate the data with a hyperplane. Instead, we design a nonlinear classification function rather than linear classification function (2.1)

$$H'(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b \tag{2.59}$$

Any point $\boldsymbol{x}$ giving $H'(\boldsymbol{x}) > 0$ will be recognized as Class I, and any point $\boldsymbol{x}$ giving $H'(\boldsymbol{x}) < 0$ will be recognized as Class II.

Suppose we use nonlinear classification functions and get the following convex optimization problem

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2} |\boldsymbol{w}|_2^2 \tag{2.60}$$

$$\text{s.t.} \quad y_i \left[\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b\right] \geq 1, \ i = 1, \ldots, L \tag{2.61}$$

We form the generalized Lagrangian function as

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} |\boldsymbol{w}|_2^2 - \sum_{i=1}^{L} \alpha_i \left[y_i \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b\right) - 1\right] \tag{2.62}$$

Letting its partial derivatives with respect to $\boldsymbol{w}$ and b be zero, we have

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i) \tag{2.63}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \implies \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.64}$$

Further eliminating the primal decision variables $\boldsymbol{w}$ and b, we have the objective of the Lagrange dual problem as

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{w}^T \left[\sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i)\right] - \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) - \sum_{i=1}^{L} \alpha_i y_i b + \sum_{i=1}^{L} \alpha_i \\ &= -\frac{1}{2}\boldsymbol{w}^T \sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i) - \sum_{i=1}^{L} \alpha_i y_i b + \sum_{i=1}^{L} \alpha_i \\ &= -\frac{1}{2}\left[\sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i)\right]^T \sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i) - b\left(\sum_{i=1}^{L} \alpha_i y_i\right) + \sum_{i=1}^{L} \alpha_i \\ &= \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \left[\phi(\boldsymbol{x}_i)\right]^T \phi(\boldsymbol{x}_j) \end{aligned} \tag{2.65}$$

The whole dual problem can then be written as

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \left[\phi(\boldsymbol{x}_i)\right]^T \phi(\boldsymbol{x}_j) \tag{2.66}$$

$$\text{s.t.} \quad \alpha_i \geq 0, i = 1, \ldots, n \tag{2.67}$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.68}$$

This problem is sometimes easier to solve, because we do not need to know the detailed form of $\phi(\boldsymbol{x})$. Instead, we only need to know the kernel function

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = \left[\phi(\boldsymbol{x})\right]^T \phi(\boldsymbol{y}) \tag{2.69}$$

The dual problem can then be written as

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2.70}$$

$$\text{s.t.} \quad \alpha_i \geq 0, i = 1, \ldots, n \tag{2.71}$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.72}$$

The parameter $\boldsymbol{w}^*$ is then recovered from the solution α^* of the dual optimization problem.

$$\boldsymbol{w}^* = \sum_i \alpha_i^* y_i \boldsymbol{\phi}(\boldsymbol{x}_i) \tag{2.73}$$

Moreover, for the few $\alpha_i > 0$, the corresponding $\boldsymbol{x}_i$ satisfies $y_i \left[(\boldsymbol{w}^*)^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b^*\right] = 1$. This means

$$\left(\boldsymbol{w}^*\right)^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b^* = 1/y_i = y_i \implies b^* = y_i - \left(\boldsymbol{w}^*\right)^T \boldsymbol{\phi}(\boldsymbol{x}_i) \tag{2.74}$$

In practice, it is more robust to average over all support vectors and calculate b^* as

$$b^* = \frac{1}{|S|} \sum_{i \in S} \left[y_i - \left(\boldsymbol{w}^*\right)^T \boldsymbol{\phi}(\boldsymbol{x}_i) \right] \tag{2.75}$$

where S denotes the set of the indices of all support vectors and $|S|$ is the cardinality of S.

For any new sample $\boldsymbol{z}$, the decision of classification can be given as

$$\begin{aligned} \operatorname{sign}\left[\left(\boldsymbol{w}^*\right)^T \boldsymbol{\phi}(\boldsymbol{z}) + b^*\right] &= \operatorname{sign}\left\{ \sum_{i=1}^{L} \alpha_i^* y_i [\boldsymbol{\phi}(\boldsymbol{x}_i)]^T \boldsymbol{\phi}(\boldsymbol{z}) + b^* \right\} \\ &= \operatorname{sign}\left[\sum_{i=1}^{L} \alpha_i^* y_i \Theta(\boldsymbol{x}_i, \boldsymbol{z}) + b^* \right] \end{aligned} \tag{2.76}$$

which can be determined without knowing the detailed form of $\boldsymbol{\phi}(\boldsymbol{x})$.

Applying kernel functions provides us a powerful tool to model possible nonlinear relations within data.

Definition 2.1 A *kernel function* is a function $\Theta : \Omega \times \Omega \mapsto \mathbb{R}$ that for all $\boldsymbol{x}, \boldsymbol{y}$ from a space Ω (which need not be a vector space), it can be expressed as an inner product of vectors $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\phi}(\boldsymbol{y})$

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{y}) \rangle \tag{2.77}$$

where $\boldsymbol{\phi}(\boldsymbol{x}) : \Omega \mapsto \mathbb{H}$ is a mapping from the space Ω to a Hilbert space $\mathbb{H}$ that is usually called the feature space.

For real space, the kernel function Θ can be arbitrarily chosen, when the existence of mapping function ϕ had been guaranteed by *Mercer's condition* [9].

Theorem 2.1 (Mercer's condition) *Let $\Omega \in \mathbb{R}^n$ be a compact set and let $\Theta : \Omega \times \Omega \mapsto \mathbb{R}$ be a continuous and symmetric function. Then, Θ admits a uniformly convergent expansion of the form*

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{\infty} c_i \left[\phi_i(\boldsymbol{x})\right]^T \phi_i(\boldsymbol{y}) \tag{2.78}$$

with $c_i > 0$ if and only if for any square integrable function $g(\boldsymbol{x}) \in L_2(\boldsymbol{x})$, the following condition holds

$$\int \int_{\Omega \times \Omega} [g(\boldsymbol{x})]^T g(\boldsymbol{y}) \Theta(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0 \tag{2.79}$$

Mercer's condition is equivalent to the assumption that the kernel Θ be symmetric positive definite; see also Chap. 5.

Theorem 2.2 *A* kernel function $\Theta : \Omega \times \Omega \mapsto \mathbb{R}$ *is said to be positive definite symmetric if for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \Omega$, the matrix $K = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij} \in \mathbb{R}^{m \times m}$ is symmetric positive semidefinite.*

Apparently, when Θ is a symmetric positive definite kernel function, the above dual problem (2.70)–(2.72) is a convex problem that is easy to solve. More discussions on kernel tricks can be found in [10–13].

Example 2.1 Let us consider a simple case, where the real space $\mathbb{R}^n$ with the dot product is taken as a special inner product space Ω

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = [\phi(\boldsymbol{x})]^T \phi(\boldsymbol{y}) \tag{2.80}$$

Suppose we apply the mapping function

$$\phi(\boldsymbol{x}) : (x_1, x_2) \mapsto \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right) = (z_1, z_2, z_3) \tag{2.81}$$

This formulates a symmetric positive definite kernel function

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = [\phi(\boldsymbol{x})]^T \phi(\boldsymbol{y}) = x_1^2 y_1^2 + 2 x_1 x_2 y_1 y_2 + x_2^2 y_2^2 = \left(\boldsymbol{x}^T \boldsymbol{y}\right)^2 \geq 0 \tag{2.82}$$

We can map two kinds of data within or outside an ellipsoid in the original x, y space (see also the discussions in Sect. 7.1)

$$\left\{ (x_1, x_2) \;\middle|\; \frac{x_2^2}{a^2} + \frac{x_2^2}{b^2} \leq 1 \right\} \text{ and } \left\{ (x_1, x_2) \;\middle|\; \frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} > 1 \right\} \tag{2.83}$$

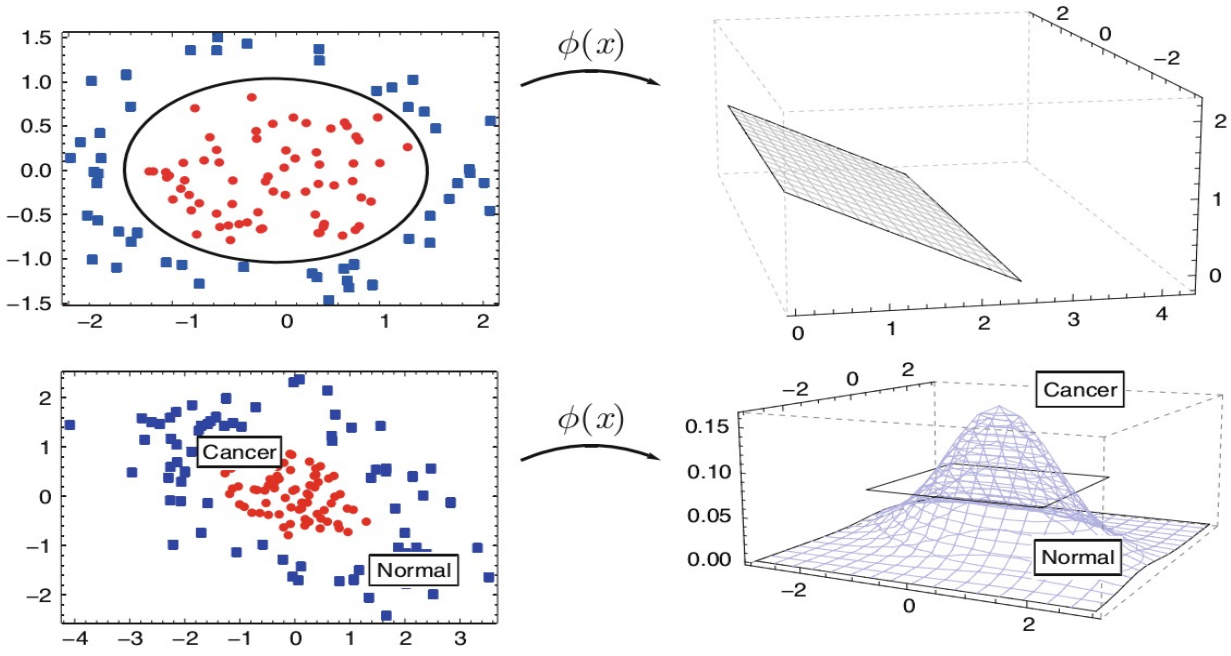

Fig. 2.4 Two illustrations of kernel tricks

into two kinds of data that can be separated by a hyperplane in the feature space z_1, z_2, z_3; see Fig. 2.4 for an illustration.

$$\left\{(z_1, z_2, z_3) \;\middle|\; \frac{z_1}{a^2} + \frac{z_3}{b^2} \leq 1\right\} \text{ and } \left\{(z_1, z_2, z_3) \;\middle|\; \frac{z_1}{a^2} + \frac{z_3}{b^2} > 1\right\} \tag{2.84}$$

We can also tolerate classification errors in kernel SVM. For example, let us formulate a soft margin SVM as follows:

$$\min_{\boldsymbol{w}, b, \xi_i} \quad \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i \tag{2.85}$$

$$\text{s.t.} \quad y_i\left[\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b\right] \geq 1 - \xi_i,\ i = 1, \ldots, L \tag{2.86}$$

$$\xi_i \geq 0,\ i = 1, \ldots, L \tag{2.87}$$

where $C \in \mathbb{R}^+$ is the penalty coefficient and $\xi_i \in \mathbb{R}^+, i = 1, \ldots, L$ are the degree of violation for each data sample.

Clearly, this is still a convex optimization problem, and we can form the generalized Lagrangian function as

$$L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}|\boldsymbol{w}|_2^2 + C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i\left\{y_i\left[\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b\right] - 1 + \xi_i\right\} - \sum_{i=1}^{L}\beta_i\xi_i \tag{2.88}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{L+}$ and $\boldsymbol{\beta} \in \mathbb{R}^{L+}$ are the associated Lagrange multipliers.

Letting its partial derivatives with respect to $\boldsymbol{w}$, b, and ξ_i be zero, we have

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i) \tag{2.89}$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = 0 \implies \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.90}$$

$$\frac{\partial L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = 0 \implies C - \alpha_i - \beta_i = 0 \tag{2.91}$$

Eliminating the primal decision variables $\boldsymbol{w}$, b, and ξ_i, we have the objective of the Lagrange dual problem as

$$\begin{aligned} L(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2} |\boldsymbol{w}|_2^2 + C \sum_{i=1}^{L} \xi_i - \sum_{i=1}^{L} \alpha_i \left\{ y_i \left[\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b \right] - 1 + \xi_i \right\} \\ &\quad - \sum_{i=1}^{L} \beta_i \xi_i \\ &= \frac{1}{2} |\boldsymbol{w}|_2^2 - \sum_{i=1}^{L} \left[\alpha_i y_i \boldsymbol{w}^T \phi(\boldsymbol{x}_i) - \alpha_i \right] - \left(\sum_{i=1}^{L} \alpha_i y_i \right) b \\ &\quad + \sum_{i=1}^{L} \xi_i \left[C - \alpha_i - \beta_i \right] \\ &= \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \left[\phi(\boldsymbol{x}_i) \right]^T \phi(\boldsymbol{x}_j) \end{aligned} \tag{2.92}$$

The whole dual problem can be written as

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \quad \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \left[\phi(\boldsymbol{x}_i) \right]^T \phi(\boldsymbol{x}_j) \tag{2.93}$$

$$\text{s.t.} \quad \alpha_i \geq 0, \ \beta_i \geq 0, \ \alpha_i + \beta_i = C \tag{2.94}$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.95}$$

Since the objective function of the Lagrange dual problem does not include $\boldsymbol{\beta}$, we have

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} y_i y_j \alpha_i \alpha_j \Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2.96}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C \tag{2.97}$$

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.98}$$

Moreover, for the few $\alpha_i > 0$, the corresponding $\boldsymbol{x}_i$ satisfies $y_i \left[(\boldsymbol{w}^*)^T \phi(\boldsymbol{x}_i) + b^* \right] = 1 - \xi_i$. This means

$$\left(\boldsymbol{w}^*\right)^T \phi(\boldsymbol{x}_i) + b^* = (1 - \xi_i)/y_i = y_i - y_i \xi_i \Longrightarrow b^* = y_i - y_i \xi_i - \left(\boldsymbol{w}^*\right)^T \phi(\boldsymbol{x}_i) \tag{2.99}$$

Since we do not know ξ_i in the dual problem, it is more robust to average over all support vectors with $\xi_i = 0, i \in S$, and calculate b^* as

$$b^* = \frac{1}{|S|} \sum_{i \in S} \left[y_i - \left(\boldsymbol{w}^*\right)^T \phi(\boldsymbol{x}_i) \right] \tag{2.100}$$

where S denotes the set of the indices of all support vectors with $\xi_i = 0, i \in S$, $|S|$ is the cardinality of S.

For any new sample $\boldsymbol{z}$, the decision of classification is still (2.76).

The sample Matlab code snippet of kernel SVM in dual form (2.96)–(2.98) is given below, where the kernel function $\Theta(\boldsymbol{x}, \boldsymbol{y})$ is chosen a Gaussian kernel (2.101). Here, we do not have a close form the mapping function $\phi(\boldsymbol{x})$ so that no code for the kernel SVM in primal form is provided.

```
function [b, alpha] = svm_dual_nonsep_gaussian_kernel(data,
                      labels, C, sigma)
% INPUT
%  data:   num-by-dim matrix. num is the number of data points,
%          dim is the dimension of a point
%  labels: num-by-1 vector, specifying the class that each point
%          belongs to.
%          either be +1 or be -1
%  C:      the tuning parameter
%  sigma:  the parameter of gaussian kernel
% OUTPUT
%  b:      a scalar, the bias
%  alpha:  num-by-1 vector, dual variables
    [num, ~] = size(data);
    K = zeros(num);
    kernel = @(x, y) exp(-norm(x - y)^2 / 2 / sigma^2) /
             sqrt(2 * pi) / sigma;
```

```
    for i = 1:num
        for j = i:num
            K(i, j) = kernel(data(i, :), data(j, :));
            K(j, i) = K(i, j);
        end
    end
    H = (labels * labels') .* K;

    cvx_begin
        variable alpha(num);
        maximize(sum(alpha) - alpha' * H * alpha / 2);
        subject to
            alpha >= 0;
            alpha <= C;
            labels' * alpha == 0;
    cvx_end

    ind = alpha > 1e-4 & alpha < C - 1e-4;
    b = mean(labels(ind) - K(ind, :) * (alpha .* labels));
end
```

Example 2.2 Both the selections of penalty coefficient C and the parameters of kernel function can greatly influence the classification results.

Suppose we choose Gaussian function

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{(2\pi)^{p/2}\sigma} \exp\left\{-\frac{1}{2\sigma^2}|\boldsymbol{x} - \boldsymbol{y}|_2^2\right\} \tag{2.101}$$

as the kernel function for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$.

The choices of σ and C lead to different classification functions; see Fig. 2.5. A relatively large C requires the SVM to try to correctly classify all the known samples. This often results in fractal classification boundaries, when multiple outliers exist. On the other hand, a relatively large σ will make the classification boundaries smooth.

2.4 Multi-kernel SVM

One problem of kernel methods is that the resulting decision function is sometimes hard to interpret and is thus difficult to extract relevant knowledge about the problem. We can solve this problem by considering convex combinations of K kernel functions, each of which has distinct meaning. The resulting multi-kernel SVM can then be given as

$$\Theta(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y}) = \left[\sum_{k=1}^{K} \beta_k \phi_k(\boldsymbol{x}_i)\right]^T \left[\sum_{k=1}^{K} \beta_k \phi_k(\boldsymbol{y}_j)\right]$$