Applied and Computational Mathematics
2017; 6(4-1): 1-15
http://www.sciencepublishinggroup.com/j/acm
doi: 10.11648/j.acm.s.2017060401.11
ISSN: 2328-5605 (Print); ISSN: 2328-5613 (Online)

Tutorial on Support Vector Machine

Loc Nguyen

Sunflower Soft Company, Ho Chi Minh City, Vietnam

Email address:
ng_phloc@yahoo.com

To cite this article:
Loc Nguyen. Tutorial on Support Vector Machine. *Applied and Computational Mathematics*. Special Issue: Some Novel Algorithms for Global Optimization and Relevant Subjects. Vol. 6, No. 4-1, 2017, pp. 1-15. doi: 10.11648/j.acm.s.2017060401.11

Received: September 7, 2015; **Accepted:** September 8, 2015; **Published:** June 17, 2016

Abstract: Support vector machine is a powerful machine learning method in data classification. Using it for applied researches is easy but comprehending it for further development requires a lot of efforts. This report is a tutorial on support vector machine with full of mathematical proofs and example, which help researchers to understand it by the fastest way from theory to practice. The report focuses on theory of optimization which is the base of support vector machine.

Keywords: Support Vector Machine, Optimization, Separating Hyperplane, Sequential Minimal Optimization

1. Support Vector Machine

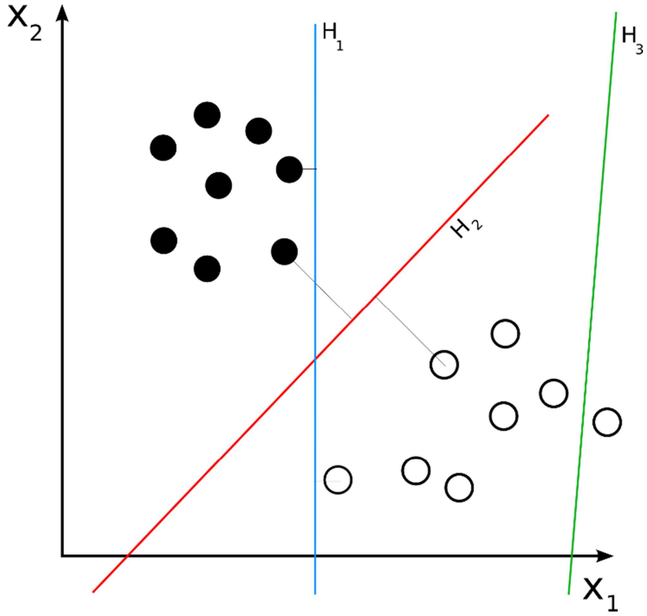

Figure 1. *Separating hyperplanes.*

Support vector machine (SVM) [1] is a supervised learning algorithm for classification and regression. Given a set of *p*-dimensional vectors in vector space, SVM finds the *separating hyperplane* that splits vector space into sub-set of vectors; each separated sub-set (so-called data set) is assigned by one class. There is the condition for this separating hyperplane: "it must maximize the margin between two sub-sets". Fig. 1 [2] shows separating hyperplanes H_1, H_2, and H_3 in which only H_2 gets maximum margin according to this condition.

Suppose we have some *p*-dimensional vectors; each of them belongs to one of two classes. We can find many *p*–1 dimensional hyperplanes that classify such vectors but there is only one hyperplane that maximizes the margin between two classes. In other words, the nearest between one side of this hyperplane and other side of this hyperplane is maximized. Such hyperplane is called *maximum-margin hyperplane* and it is considered as the SVM *classifier*.

Let $\{X_1, X_2,\ldots, X_n\}$ be the training set of *n* vectors X_i (s) and let $y_i = \{+1, -1\}$ be the class label of vector X_i. Each X_i is also called a data point with attention that *vectors can be identified with data points* and data point can be called *point*, in brief. It is necessary to determine the maximum-margin hyperplane that separates data points belonging to $y_i=+1$ from data points belonging to $y_i=-1$ as clear as possible.

According to theory of geometry, arbitrary hyperplane is represented as a set of points satisfying *hyperplane equation* specified by (1).

$$W \circ X_i - b = 0 \tag{1}$$

Where the sign "∘" denotes the dot product or scalar product and *W* is *weight vector* perpendicular to hyperplane and *b* is the *bias*. Vector *W* is also called perpendicular vector or normal vector and it is used to specify hyperplane. Suppose $W=(w_1, w_2,\ldots, w_p)$ and $X_i=(x_{i1}, x_{i2},\ldots, x_{ip})$, the scalar product

$W \circ X_i$ is:

$$W \circ X_i = X_i \circ W = w_1x_{i1} + w_2x_{i2} + \cdots + w_nx_{ip} = \sum_{j=1}^{p} w_jx_{ij}$$

Given scalar value *w*, the multiplication of *w* and vector X_i denoted wX_i is a vector as follows:

$$wX_i = (wx_{i1}, wx_{i2}, \ldots, wx_{ip})$$

Please distinguish scalar product $W \circ X_i$ and multiplication wX_i.

The essence of SVM method is to find out weight vector *W* and bias *b* so that the hyperplane equation specified by (1) expresses the maximum-margin hyperplane that maximizes the margin between two classes of training set.

The value *b*/|*W*| is the offset of the (maximum-margin) hyperplane from the origin along the weight vector *W* where |*W*| or ||*W*|| denotes length or module of vector *W*.

$$|W| = ||W|| = \sqrt{W \circ W} = \sqrt{w_1^2 + w_2^2 + \cdots + w_p^2}$$
$$= \sqrt{\sum_{i=1}^{p} w_i^2}$$

Note that we use two notations |.| and ||.|| for denoting the length of vector.

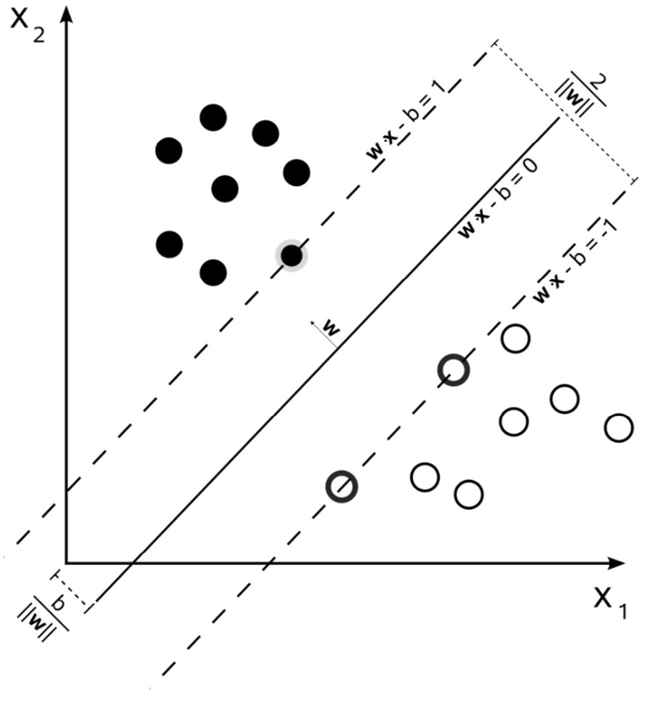

Figure 2. *Maximum-margin hyperplane, parallel hyperplanes and weight vector W.*

Additionally, the value 2/|*W*| is the width of the margin as seen in fig. 2. To determine the margin, two parallel hyperplanes are constructed, one on each side of the maximum-margin hyperplane. Such two parallel hyperplanes are represented by two hyperplane equations, as shown in (2) as follows.

$$\left.\begin{matrix} W \circ X_i - b = 1 \\ W \circ X_i - b = -1 \end{matrix}\right\} \quad (2)$$

Fig. 2 [2] illustrates maximum-margin hyperplane, weight vector *W* and two parallel hyperplanes. As seen in the fig. 2, the margin is limited by such two parallel hyperplanes. Exactly, there are two margins (each one for a parallel hyperplane) but it is convenient for referring both margins as the unified single margin as usual. You can imagine such margin as a road and SVM method aims to maximize the width of such road. Data points lying on (or are very near to) two parallel hyperplanes are called support vectors because they construct mainly the maximum-margin hyperplane in the middle. This is the reason that the classification method is called support vector machine (SVM).

To prevent vectors from falling into the margin, all vectors belonging to two classes y_i=1 and y_i=–1 have two following constraints, respectively:

$$\begin{cases} W \circ X_i - b \geq 1 \text{ (for } X_i \text{ belonging to class } y_i = +1) \\ W \circ X_i - b \leq -1 \text{ (for } X_i \text{ belonging to class } y_i = -1) \end{cases}$$

As seen in fig. 2, vectors (data points) belonging to classes y_i=+1 and y_i=–1 are depicted as black circles and white circles, respectively. Such two constraints are unified into the so-called classification constraint specified by (3) as follows:

$$y_i(W \circ X_i - b) \geq 1 \Leftrightarrow 1 - y_i(W \circ X_i - b) \leq 0 \quad (3)$$

As known, y_i=+1 and y_i=–1 represent two classes of data points. It is easy to infer that maximum-margin hyperplane which is the result of SVM method is the classifier that aims to determined which class (+1 or –1) a given data point *X* belongs to. Your attention please, each data point X_i in training set was assigned by a class y_i before and maximum-margin hyperplane constructed from the training set is used to classify any different data point *X*.

Because maximum-margin hyperplane is defined by weight vector *W*, it is easy to recognize that the essence of constructing maximum-margin hyperplane is to solve the constrained optimization problem as follows:

$$\underset{W,b}{\text{minimize}} \frac{1}{2}|W|^2 \text{ subject to } y_i(W \circ X_i - b) \geq 1, \forall i = \overline{1,n}$$

Where |*W*| is the length of weight vector *W* and $y_i(W \circ X_i - b) \geq 1$ is the classification constraint specified by (3). The reason of minimizing $\frac{1}{2}|W|^2$ is that distance between two parallel hyperplanes is 2/|*W*| and we need to maximize such distance in order to maximize the margin for maximum-margin hyperplane. Then maximizing 2/|*W*| is to minimize $\frac{1}{2}|W|$. Because it is complex to compute the length |*W*|, we substitute $\frac{1}{2}|W|^2$ for $\frac{1}{2}|W|$ when $|W|^2$ is equal to the scalar product $W \circ W$ as follows:

$$|W|^2 = \|W\|^2 = W \circ W = w_1^2 + w_2^2 + \cdots + w_p^2$$

The constrained optimization problem is re-written, shown in (4) as below:

$$\left.\begin{array}{c}\text{minimize}_{W,b} f(W) = \text{minimize}_{W,b} \frac{1}{2}|W|^2 \\ \text{subject to:} \\ g_i(W,b) = 1 - y_i(W \circ X_i - b) \leq 0, \forall i = \overline{1,n}\end{array}\right\} \quad (4)$$

Where $f(W) = \frac{1}{2}|W|^2$ is called target function with regard to variable *W*. Function $g_i(W,b) = 1 - y_i(W \circ X_i - b)$ is called constraint function with regard to two variables *W*, *b* and it is derived from the classification constraint specified by (3). There are *n* constraints $g_i(W,b) \leq 0$ because training set $\{X_1, X_2,\ldots, X_n\}$ has *n* data points X_i (s). Constraints $g_i(W,b) \leq 0$ inside (3) implicate the perfect separation in which there is no data point falling into the margin (between two parallel hyperplanes, see fig. 2). On the other hand, the imperfect separation allows some data points to fall into the margin, which means that each constraint function $g_i(W,b)$ is subtracted by an error $\xi_i \geq 0$. The constraints become [3, p. 5]:

$$g_i(W,b) = 1 - y_i(W \circ X_i - b) - \xi_i \leq 0, \forall i = \overline{1,n}$$

We have a *n*-component error vector $\xi=(\xi_1, \xi_2,\ldots, \xi_n)$ for *n* constraints. The penalty $C \geq 0$ is added to the target function in order to penalize data points falling into the margin. The penalty *C* is a pre-defined constant. Thus, the target function *f*(*W*) becomes:

$$f(W) = \frac{1}{2}|W|^2 + C\sum_{i=1}^{n} \xi_i$$

If the positive penalty is infinity, $C = +\infty$ then, target function *f*(*W*) may get maximal when all errors ξ_i must be 0, which leads to the perfect separation specified by (4).

Equation (5) specifies the general form of constrained optimization originated from (4).

$$\left.\begin{array}{c}\underset{W,b,\xi}{\text{minimize}} \frac{1}{2}|W|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to:} \\ 1 - y_i(W \circ X_i - b) - \xi_i \leq 0, \forall i = \overline{1,n} - \xi_i \leq 0, \forall i = \overline{1,n}\end{array}\right\} \quad (5)$$

Where $C \geq 0$ is the penalty.

The *Lagrangian function* [4, p. 215] is constructed from constrained optimization problem specified by (5). Let *L*(*W*, *b*, ξ, λ, μ) be Lagrangian function where $\lambda=(\lambda_1, \lambda_2,\ldots, \lambda_n)$ and $\mu=(\mu_1, \mu_2,\ldots, \mu_n)$ are *n*-component vectors, $\lambda_i \geq 0$ and $\mu_i \geq 0$, $\forall i = \overline{1,n}$. We have:

$$L(W,b,\xi,\lambda,\mu) = f(W) + \sum_{i=1}^{n} \lambda_i g_i(W,b) - \mu_i\xi_i = \frac{1}{2}|W|^2 + C\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \lambda_i(1 - y_i(W \circ X_i - b) - \xi_i) - \sum_{i=1}^{n} \mu_i\xi_i$$

$$= \frac{1}{2}|W|^2 + C\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n} \lambda_i y_i(W \circ X_i) + \sum_{i=1}^{n} b\lambda_i y_i - \sum_{i=1}^{n} \lambda_i\xi_i - \sum_{i=1}^{n} \mu_i\xi_i$$

$$= \frac{1}{2}|W|^2 - W \circ \left(\sum_{i=1}^{n} \lambda_i y_i X_i\right) + \sum_{i=1}^{n} \lambda_i + b\sum_{i=1}^{n} \lambda_i y_i + \sum_{i=1}^{n} (C - \lambda_i - \mu_i)\xi_i$$

In general, (6) represents Lagrangian function as follows:

$$\left\{\begin{array}{c}L(W,b,\xi,\lambda,\mu) = \frac{1}{2}|W|^2 - W \circ \left(\sum_{i=1}^{n} \lambda_i y_i X_i\right) + \sum_{i=1}^{n} \lambda_i + b\sum_{i=1}^{n} \lambda_i y_i + \sum_{i=1}^{n} (C + \lambda_i - \mu_i)\xi_i \\ \text{Where } \xi_i \geq 0, \lambda_i \geq 0, \mu_i \geq 0, \forall i = \overline{1,n}\end{array}\right. \quad (6)$$

Note that $\lambda=(\lambda_1, \lambda_2,\ldots, \lambda_n)$ and $\mu=(\mu_1, \mu_2,\ldots, \mu_n)$ are called Lagrange multipliers or Karush-Kuhn-Tucker multipliers [5] or dual variables. The sign "∘" denotes scalar product and every training data point X_i was assigned by a class y_i before.

Suppose (W^*, b^*) is solution of constrained optimization problem specified by (5) then, the pair (W^*, b^*) is minimum point of target function *f*(*W*) or target function *f*(*W*) gets minimum at (W^*, b^*) with all constraints $g_i(W,b) = 1 - y_i(W \circ X_i - b) + \xi_i \leq 0, \forall i = \overline{1,n}$. Note that W^* is called *optimal weight vector* and b^* is called *optimal bias*. It is easy to infer that the pair (W^*, b^*) represents the maximum-margin hyperplane and it is possible to identify (W^*, b^*) with the maximum-margin hyperplane. The ultimate goal of SVM method is to find out W^* and b^*. According to Lagrangian duality theorem [4, p. 216] [6, p. 8], the pair (W^*, b^*) is the extreme point of Lagrangian function as follows:

$$\left.\begin{array}{c}(W^*, b^*) = \text{argmin}_{W,b} L(W,b,\xi,\lambda,\mu) \\ \lambda^* = \text{argmax}_{\lambda \geq 0}\left(\min_{W,b} L(W,b,\lambda,\mu)\right)\end{array}\right\} \quad (7)$$

Where Lagrangian function *L*(*W*, *b*, ξ, λ, μ) is specified by (6).

Now it is necessary to solve the Lagrangian duality problem represented by (7) to find out W^*. Thus, the Lagrangian function *L*(*W*, *b*, ξ, λ, μ) is minimized with respect to the primal variables *W*, *b* and maximized with respect to the dual variables *λ*=(*λ*₁, *λ*₂,…, *λₙ*) and *μ*=(*μ*₁, *μ*₂,…, *μₙ*), in turn. If gradient of *L*(*W*, *b*, *ξ*, *λ*, *μ*) is equal to zero then, *L*(*W*, *b*, *ξ*, *λ*, *μ*) will gets minimum value with note that gradient of a multi-variable function is the vector whose components are first-order partial derivative of such function. Thus, setting the gradient of *L*(*W*, *b*, *ξ*, *λ*, *μ*) with respect to *W*, *b*, and *ξ* to zero, we have:

$$\begin{cases}\dfrac{\partial L(W,b,\xi,\lambda,\mu)}{\partial W}=0\\ \dfrac{\partial L(W,b,\xi,\lambda,\mu)}{\partial b}=0\\ \dfrac{\partial L(W,b,\xi,\lambda,\mu)}{\partial \xi_i}=0,\forall i=\overline{1,n}\end{cases}\Leftrightarrow\begin{cases}W-\displaystyle\sum_{i=1}^{n}\lambda_i y_i X_i=0\\ \displaystyle\sum_{i=1}^{n}\lambda_i y_i=0\\ C-\lambda_i-\mu_i=0,\forall i=\overline{1,n}\end{cases}\Rightarrow\begin{cases}W=\displaystyle\sum_{i=1}^{n}\lambda_i y_i X_i\\ \displaystyle\sum_{i=1}^{n}\lambda_i y_i=0\\ \lambda_i=C-\mu_i,\forall i=\overline{1,n}\end{cases}$$

In general, *W** is determined by (8) as follows:

$$\begin{cases}W^*=\sum_{i=1}^{n}\lambda_i y_i X_i\\ \sum_{i=1}^{n}\lambda_i y_i=0\\ \lambda_i=C-\mu_i,\lambda_i\geq 0,\mu_i\geq 0,\forall i=\overline{1,n}\end{cases}\quad(8)$$

It is required to determine Lagrange multipliers *λ*=(*λ*₁, *λ*₂,…, *λₙ*) in order to evaluate *W**. Substituting (8) into Lagrangian function *L*(*W*, *b*, *ξ*, *λ*, *μ*) specified by (6), we have:

$$l(\lambda)=\min_{W,b}L(W,b,\xi,\lambda,\mu)=\min_{W,b}\left(\frac{1}{2}|W|^2-W\circ\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)+\sum_{i=1}^{n}\lambda_i+b\sum_{i=1}^{n}\lambda_i y_i+\sum_{i=1}^{n}(C+\lambda_i-\mu_i)\xi_i\right)$$
$$=\frac{1}{2}\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)^2-\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)\circ\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)+\sum_{i=1}^{n}\lambda_i$$

(According to (8), *L*(*W*,*b*,*ξ*,*λ*,*μ*) gets minimum at $W=\sum_{i=1}^{n}\lambda_i y_i X_i$ and $\sum_{i=1}^{n}\lambda_i y_i=0$ and $\lambda_i=C-\mu_i$)

$$=\frac{1}{2}\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)\circ\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)-\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)\circ\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)+\sum_{i=1}^{n}\lambda_i=-\frac{1}{2}\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)\circ\left(\sum_{i=1}^{n}\lambda_i y_i X_i\right)+\sum_{i=1}^{n}\lambda_i$$
$$=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\left(X_i\circ X_j\right)+\sum_{i=1}^{n}\lambda_i$$

Where *l*(*λ*) is called *dual function* represented by (9).

$$l(\lambda)=\min_{W,b}L(W,b,\xi,\lambda,\mu)=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\left(X_i\circ X_j\right)+\sum_{i=1}^{n}\lambda_i\quad(9)$$

According to Lagrangian duality problem represented by (7), *λ*=(*λ*₁, *λ*₂,…, *λₙ*) is calculated as the maximum point *λ**=(*λ*₁*, *λ*₂*,…, *λₙ**) of dual function *l*(*λ*). In conclusion, maximizing *l*(*λ*) is the main task of SVM method because the optimal weight vector *W** is calculated based on the optimal point *λ** of dual function *l*(*λ*) according to (8).

$$W^*=\sum_{i=1}^{n}\lambda_i y_i X_i=\sum_{i=1}^{n}\lambda_i^* y_i X_i$$

Maximizing *l*(*λ*) is quadratic programming (QP) problem, specified by (10).

$$\left.\begin{array}{c}\underset{\lambda}{\text{maximize}}-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\left(X_i\circ X_j\right)+\sum_{i=1}^{n}\lambda_i\\ \text{subject to:}\\ \sum_{i=1}^{n}\lambda_i y_i=0\\ 0\leq\lambda_i\leq C,\forall i=\overline{1,n}\end{array}\right\}\quad(10)$$

The constraints $0\leq\lambda_i\leq C,\forall i=\overline{1,n}$ are implied from the equations $\lambda_i=C-\mu_i,\forall i=\overline{1,n}$ when $\mu_i\geq 0,\forall i=\overline{1,n}$. The QP problem specified by (10) is also known as Wolfe problem [3, p. 42].

There are some methods to solve this QP problem but this report introduces a so-called Sequential Minimal Optimization (SMO) developed by author [7]. The SMO algorithm is very effective method to find out the optimal (maximum) point *λ** of dual function *l*(*λ*).

$$l(\lambda)=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\left(X_i\circ X_j\right)+\sum_{i=1}^{n}\lambda_i$$

Moreover SMO algorithm also finds out the optimal bias *b**, which means that SVM classifier (*W**, *b**) is totally determined by SMO algorithm. The next section described SMO algorithm in detail.

2. Sequential Minimal Optimization

The ideology of SMO algorithm is to divide the whole QP problem into many smallest optimization problems. Each smallest problem relates to only two Lagrange multipliers. For solving each smallest optimization problem, SMO algorithm includes two nested loops as shown in table 1 [7, pp. 8-9]:

Table 1. *Ideology of SMO algorithm.*

SMO algorithm solves each smallest optimization problem via two nested loops:

1. The outer loop finds out the first Lagrange multiplier λ_i whose associated data point X_i violates KKT condition [5]. Violating KKT condition is known as the first choice heuristic.
2. The inner loop finds out the second Lagrange multiplier λ_j according to the second choice heuristic. The second choice heuristic that maximizes optimization step will be described later.
3. Two Lagrange multipliers λ_i and λ_j are optimized jointly according to QP problem specified by (10).

SMO algorithm continues to solve another smallest optimization problem. SMO algorithm stops when there is convergence in which no data point violating KKT condition is found; consequently, all Lagrange multipliers λ_1, λ_2,…, λ_n are optimized.

Before describing SMO algorithm in detailed, the KKT condition with subject to SVM is mentioned firstly because violating KKT condition is known as the first choice heuristic of SMO algorithm. KKT condition indicates both partial derivatives of Lagrangian function and complementary slackness are zero [5]. Referring (8) and (4), the KKT function of SVM is summarized as (11):

$$\begin{cases} W = \sum_{i=1}^{n} \lambda_i y_i X_i \\ \sum_{i=1}^{n} \lambda_i y_i = 0 \\ \lambda_i = C - \mu_i, \lambda_i \geq 0, \mu_i \geq 0, \forall i \\ \lambda_i\left(1 - y_i(W \circ X_i - b) - \xi_i\right) = 0, \forall i \\ -\mu_i \xi_i = 0, \forall i \end{cases} \quad (11)$$

When we understand deeply convex optimization, the KKT condition is the same to the QP problem specified by (10) if target function and constraint sets are convex. Thus, the solution (W^*, λ^*) is saddle point of Lagrangian function.

KKT condition is analyzed into three following cases [3, p. 7]:

1. If λ_i=0 then, μ_i = C – λ_i = C. It implies ξ_i=0 from equation $\mu_i\xi_i = 0$. Then, from equation $\lambda_i(1 - y_i(W \circ X_i - b) - \xi_i) = 0$ we have:
$$1 - y_i(W \circ X_i - b) \leq 0$$
2. If 0 < λ_i < C then, we have $1 - y_i(W \circ X_i - b) - \xi_i = 0$. Due to μ_i = C – λ_i > 0, it implies ξ_i=0 from equation $\mu_i\xi_i = 0$. It is easy to infer that:
$$1 - y_i(W \circ X_i - b) = 0$$
3. If λ_i=C then, we have μ_i = C – λ_i = 0 and $1 - y_i(W \circ X_i - b) - \xi_i = 0$. Due to μ_i = 0, it implies $\xi_i \geq 0$ from equation $\mu_i\xi_i = 0$. Given $\xi_i \geq 0$ the equation $1 - y_i(W \circ X_i - b) - \xi_i = 0$ leads to:
$$1 - y_i(W \circ X_i - b) \geq 0$$

Let $E_i = y_i - (W \circ X_i - b)$ be prediction error, we have:

$$y_i E_i = (y_i)^2 - y_i(W \circ X_i - b) = 1 - y_i(W \circ X_i - b)$$

The KKT condition implies:

$$\left.\begin{array}{c} \lambda_i = 0 \Longrightarrow y_i E_i \leq 0 \\ 0 < \lambda_i < C \Longrightarrow y_i E_i = 0 \\ \lambda_i = C \Longrightarrow y_i E_i \geq 0 \\ \text{Where } E_i \text{ is prediction error:} \\ E_i = y_i - (W \circ X_i - b) \end{array}\right\} \quad (12)$$

Equation (12) expresses directed corollaries from KKT condition. It is commented on (12) that if E_i=0, the KKT condition is always satisfied. Data points X_i satisfying equation y_iE_i=0 lie on the margin (lie on the two parallel hyperplanes). These points are called *support vectors*. According to KKT corollary, support vectors are always associated with non-zero Lagrange multipliers such that 0<λ_i<C. Note, such Lagrange multipliers 0<λ_i<C are also called non-boundary multipliers because they are not bounds such as 0 and C. So, support vectors are also known as *non-boundary* data points. It easy to infer from (8)

$$W^* = \sum_{i=1}^{n} \lambda_i y_i X_i$$

that support vectors along with their non-zero Lagrange multipliers form mainly the optimal weight vector W^* representing the maximum-margin hyperplane – the SVM classifier. This is the reason that this classification approach is called support vector machine (SVM). Fig. 3 [8, p. 5] illustrates an example of support vectors.

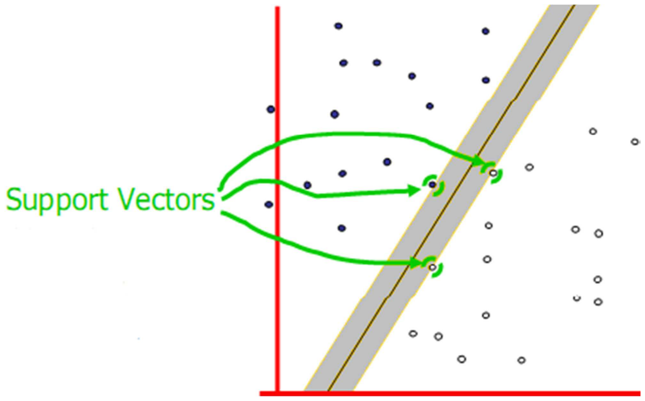

Figure 3. *Support vectors.*

Violating KKT condition is the first choice heuristic of SMO algorithm. By negating three corollaries specified by (12), KKT condition is violated in three following cases:

$$\begin{array}{lll} \lambda_i = 0 & \text{and} & y_i E_i > 0 \\ 0 < \lambda_i < C & \text{and} & y_i E_i \neq 0 \\ \lambda_i = C & \text{and} & y_i E_i < 0 \end{array}$$

By logic induction, these cases are reduced into two cases specified by (13).

$$\left.\begin{array}{c}\lambda_i < C \text{ and } y_iE_i > 0\\ \lambda_i > 0 \text{ and } y_iE_i < 0\\ \text{Where } E_i \text{ is prediction error:}\\ E_i = y_i(W \circ X_i - b)\end{array}\right\} \quad (13)$$

Equation (13) is used to check whether given data point X_i violates KKT condition.

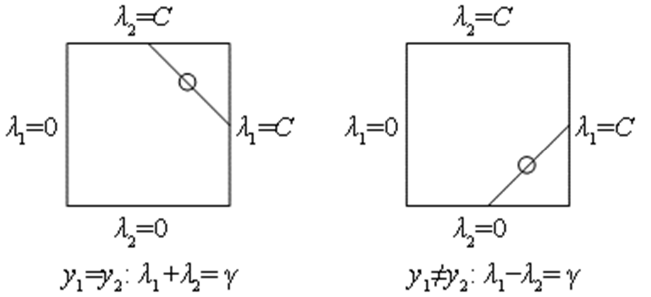

Figure 4. *Linear constraint of two Lagrange multipliers.*

The main task of SMO algorithm (see table 1) is to optimize jointly two Lagrange multipliers in order to solve each smallest optimization problem, which maximizes the dual function $l(\lambda)$.

$$l(\lambda) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_jy_iy_j(X_i \circ X_j) + \sum_{i=1}^{n}\lambda_i$$

Where,

$$\sum_{i=1}^{n}\lambda_iy_i = 0$$
$$0 \le \lambda_i \le C, \forall i = \overline{1,n}$$

Without loss of generality, two Lagrange multipliers λ_i and λ_j that will be optimized are λ_1 and λ_2 while all other multipliers λ_3, λ_4,…, λ_n are fixed. Old values of λ_1 and λ_2 are denoted λ_1^{old} and λ_2^{old}. Your attention please, old values are known as current values. Thus, λ_1 and λ_2 are optimized based on the set: λ_1^{old}, λ_2^{old}, λ_2, λ_3,…, λ_n. The old values λ_1^{old} and λ_2^{old} are initialized by zero [3, p. 9]. From the condition $\sum_{i=1}^{n}\lambda_iy_i = 0$, we have:

$$\lambda_1^{\text{old}}y_1 + \lambda_2^{\text{old}}y_2 + \lambda_3y_3 + \lambda_4y_4 + \cdots + \lambda_ny_n = 0$$

and

$$\lambda_1y_1 + \lambda_2y_2 + \lambda_3y_3 + \lambda_4y_4 + \cdots + \lambda_ny_n = 0$$

It implies following equation of line with regard to two variables λ_1 and λ_2:

$$\lambda_1y_1 + \lambda_2y_2 = \lambda_1^{\text{old}}y_1 + \lambda_2^{\text{old}}y_2 \quad (14)$$

Equation (14) specifies the linear constraint of two Lagrange multipliers λ_1 and λ_2. This constraint is drawn as diagonal lines in fig. 4 [3, p. 9].

In fig. 4, the box is bounded by the interval [0, C] of Lagrange multipliers, $0 \le \lambda_i \le C$. SMO algorithm moves λ_1 and λ_2 along diagonal lines so as to maximize the dual function $l(\lambda)$. Multiplying two sides of equation

$$\lambda_1y_1 + \lambda_2y_2 = \lambda_1^{\text{old}}y_1 + \lambda_2^{\text{old}}y_2$$

by y_1, we have:

$$\lambda_1y_1y_1 + \lambda_2y_1y_2 = \lambda_1^{\text{old}}y_1y_1 + \lambda_2^{\text{old}}y_1y_2 \Longrightarrow \lambda_1 + s\lambda_2 = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}}$$

Where s=y_1y_2. Let,

$$\gamma = \lambda_1 + s\lambda_2 = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}}$$

We have (15) as a variant of the linear constraint of two Lagrange multipliers λ_1 and λ_2 [3, p. 9]:

$$\begin{array}{c}\lambda_1 = \gamma - s\lambda_2\\ \text{Where,}\\ s = y_1y_2\\ \gamma = \lambda_1 + s\lambda_2 = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}}\end{array} \quad (15)$$

By fixing multipliers λ_3, λ_4,…, λ_n, all arithmetic combinations of λ_1^{old}, λ_2^{old}, λ_3, λ_4,…, λ_n are constants denoted by term "*const*". The dual function $l(\lambda)$ is re-written [3, pp. 9-11]:

$$l(\lambda) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_jy_iy_j(X_i \circ X_j) + \sum_{i=1}^{n}\lambda_i$$
$$= -\frac{1}{2}\left((\lambda_1)^2y_1y_1(X_1 \circ X_1) + (\lambda_2)^2y_2y_2(X_2 \circ X_2) + 2\lambda_1\lambda_2y_1y_2(X_1 \circ X_2) + 2\left(\sum_{i=3}^{n}\lambda_i\lambda_1y_iy_1(X_i \circ X_1)\right)\right.$$
$$\left. + 2\left(\sum_{i=3}^{n}\lambda_i\lambda_2y_iy_2(X_i \circ X_2)\right) + const\right) + \lambda_1 + \lambda_2 + const$$

$$= -\frac{1}{2}\left((X_1 \circ X_1)(\lambda_1)^2 + (X_2 \circ X_2)(\lambda_2)^2 + 2s(X_1 \circ X_2)\lambda_1\lambda_2 + 2\left(\sum_{i=3}^{n} \lambda_i\lambda_1 y_i y_1 (X_i \circ X_1)\right) + 2\left(\sum_{i=3}^{n} \lambda_i\lambda_2 y_i y_2 (X_i \circ X_2)\right)\right)$$
$$+ \lambda_1 + \lambda_2 + const$$

Let,

$$K_{11} = X_1 \circ X_1$$

$$K_{12} = X_1 \circ X_2$$

$$K_{22} = X_2 \circ X_2$$

Let W^{old} be the optimal weight vector $W = \sum_{i=1}^{n} \lambda_i y_i X_i$ based on old values of two aforementioned Lagrange multipliers. Following linear constraint of two Lagrange multipliers specified by (14), we have:

$$W^{\text{old}} = \lambda_1^{\text{old}} y_1 X_1 + \lambda_2^{\text{old}} y_2 X_2 + \sum_{i=3}^{n} \lambda_i y_i X_i = \sum_{i=1}^{n} \lambda_i y_i X_i = W$$

Let,

$$v_j = \sum_{i=3}^{n} \lambda_i y_i \left(X_i \circ X_j\right) = \sum_{i=3}^{n} (\lambda_i y_i X_i) \circ X_j = \left(\sum_{i=3}^{n} \lambda_i y_i X_i\right) \circ X_j = \left(W^{\text{old}} - \lambda_1^{\text{old}} y_1 X_1 - \lambda_2^{\text{old}} y_2 X_2\right) \circ X_j = W^{\text{old}} \circ X_j -$$
$$\lambda_1^{\text{old}} y_1 X_1 \circ X_j - \lambda_2^{\text{old}} y_2 X_2 \circ X_j$$

We have [3, p. 10]:

$$l(\lambda) = -\frac{1}{2}\left(K_{11}(\lambda_1)^2 + K_{22}(\lambda_2)^2 + 2sK_{12}\lambda_1\lambda_2 + 2y_1v_1\lambda_1 + 2y_2v_2\lambda_2\right) + \lambda_1 + \lambda_2 + const$$

$$= -\frac{1}{2}\left(K_{11}(\gamma - s\lambda_2)^2 + K_{22}(\lambda_2)^2 + 2sK_{12}(\gamma - s\lambda_2)\lambda_2 + 2y_1v_1(\gamma - s\lambda_2) + 2y_2v_2\lambda_2\right) + (\gamma - s\lambda_2) + \lambda_2 + const$$

$$= -\frac{1}{2}\left(K_{11}\gamma^2 - 2sK_{11}\gamma\lambda_2 + K_{11}(\lambda_2)^2 + K_{22}(\lambda_2)^2 + 2sK_{12}\gamma\lambda_2 - 2K_{12}(\lambda_2)^2 + 2y_1v_1\gamma - 2sy_1v_1\lambda_2 + 2y_2v_2\lambda_2\right) + (1-s)\lambda_2$$
$$+ \gamma + const$$

$$= -\frac{1}{2}(K_{11} + K_{22} - 2K_{12})(\lambda_2)^2 + sK_{11}\gamma\lambda_2 - sK_{12}\gamma\lambda_2 + sy_1v_1\lambda_2 - y_2v_2\lambda_2 + (1-s)\lambda_2 - \frac{1}{2}K_{11}\gamma^2 - y_1v_1\gamma + \gamma + const$$

$$= -\frac{1}{2}(K_{11} + K_{22} - 2K_{12})(\lambda_2)^2 + sK_{11}\gamma\lambda_2 - sK_{12}\gamma\lambda_2 + sy_1v_1\lambda_2 - y_2v_2\lambda_2 + (1-s)\lambda_2 + const$$

$$\left(\text{Because } -\tfrac{1}{2}K_{11}\gamma^2 - y_1v_1\gamma + \gamma \text{ is also constant}\right)$$

$$= -\tfrac{1}{2}(K_{11} + K_{22} - 2K_{12})(\lambda_2)^2 + (1 - s + sK_{11}\gamma - sK_{12}\gamma + sy_1v_1 - y_2v_2)\lambda_2 + const$$

$$= -\tfrac{1}{2}(K_{11} + K_{22} - 2K_{12})(\lambda_2)^2 + (1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2v_1 - y_2v_2)\lambda_2 + const$$

Let $\eta = K_{11} - 2K_{12} + K_{22}$ and assessing the coefficient of λ_2, we have [3, p. 11]:

$$1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2v_1 - y_2v_2 = 1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2(v_1 - v_2)$$

$$= 1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2\left(W^{old} \circ X_1 - \lambda_1^{\text{old}} y_1 X_1 \circ X_1 - \lambda_2^{\text{old}} y_2 X_2 \circ X_1 - W^{old} \circ X_2 + \lambda_1^{\text{old}} y_1 X_1 \circ X_2 + \lambda_2^{\text{old}} y_2 X_2 \circ X_2\right)$$

$$= 1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2(W^{old} \circ X_1 - W^{old} \circ X_2) - \lambda_1^{\text{old}} y_1 y_2 X_1 \circ X_1 - \lambda_2^{\text{old}} y_2 y_2 X_2 \circ X_1 + \lambda_1^{\text{old}} y_1 y_2 X_1 \circ X_2$$
$$+ \lambda_2^{\text{old}} y_2 y_2 X_2 \circ X_2$$

$$= 1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2(W^{old} \circ X_1 - W^{old} \circ X_2) - \lambda_1^{\text{old}} y_1 y_2 X_1 \circ X_1 - \lambda_2^{\text{old}} X_2 \circ X_1 + \lambda_1^{\text{old}} y_1 y_2 X_1 \circ X_2 + \lambda_2^{\text{old}} X_2 \circ X_2$$

$$= 1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2(W^{old} \circ X_1 - W^{old} \circ X_2) - sK_{11}\lambda_1^{\text{old}} - K_{12}\lambda_2^{\text{old}} + sK_{12}\lambda_1^{\text{old}} + K_{22}\lambda_2^{\text{old}}$$

$$= 1 - s + sK_{11}\left(\lambda_1^{\text{old}} + s\lambda_2^{\text{old}}\right) - sK_{12}\left(\lambda_1^{\text{old}} + s\lambda_2^{\text{old}}\right) + y_2(W^{old} \circ X_1 - W^{old} \circ X_2) - sK_{11}\lambda_1^{\text{old}} - K_{12}\lambda_2^{\text{old}} + sK_{12}\lambda_1^{\text{old}}$$
$$+ K_{22}\lambda_2^{\text{old}}$$

$$= 1 - s + sK_{11}\lambda_1^{\text{old}} + K_{11}\lambda_2^{\text{old}} - sK_{12}\lambda_1^{\text{old}} - K_{12}\lambda_2^{\text{old}} + y_2(W^{old} \circ X_1 - W^{old} \circ X_2) - sK_{11}\lambda_1^{\text{old}} - K_{12}\lambda_2^{\text{old}} + sK_{12}\lambda_1^{\text{old}} + K_{22}\lambda_2^{\text{old}}$$

$$= 1 - s + (sK_{11} - sK_{12} - sK_{11} + sK_{12})\lambda_1^{\text{old}} + (K_{11} - K_{12} - K_{12} + K_{22})\lambda_2^{\text{old}} + y_2\left(W^{\text{old}} \circ X_1 - W^{\text{old}} \circ X_2\right)$$

$$= 1 - s + (K_{11} - 2K_{12} + K_{22})\lambda_2^{\text{old}} + y_2\left(W^{\text{old}} \circ X_1 - W^{\text{old}} \circ X_2\right)$$

$$= 1 - s + \eta\lambda_2^{\text{old}} + y_2\left(W^{\text{old}} \circ X_1 - W^{\text{old}} \circ X_2\right)$$

(due to $\eta = K_{11} - 2K_{12} + K_{22}$)

$$= 1 - s + \eta\lambda_2^{\text{old}} + y_2\left(\left(y_2 - \left(W^{\text{old}} \circ X_2 - b^{\text{old}}\right)\right) - \left(y_1 - \left(W^{\text{old}} \circ X_1 - b^{\text{old}}\right)\right)\right) - y_2y_2 + y_1y_2$$

(Where b^{old} is the old value of the bias *b*)

$$= 1 - s + \eta\lambda_2^{\text{old}} + y_2\left(\left(y_2 - \left(W^{\text{old}} \circ X_2 - b^{\text{old}}\right)\right) - \left(y_1 - \left(W^{\text{old}} \circ X_1 - b^{\text{old}}\right)\right)\right) - 1 + s$$

$$= \eta\lambda_2^{\text{old}} + y_2\left(\left(y_2 - \left(W^{\text{old}} \circ X_2 - b^{\text{old}}\right)\right) - \left(y_1 - \left(W^{\text{old}} \circ X_1 - b^{\text{old}}\right)\right)\right) = \eta\lambda_2^{\text{old}} + y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)$$

According to (13), E_2^{old} and E_1^{old} are old prediction errors on X_2 and X_1, respectively:

$$E_j^{\text{old}} = y_j - \left(W^{old} \circ X_j - b^{\text{old}}\right)$$

Recall that we had:

$$l(\lambda) = -\frac{1}{2}(K_{11} + K_{22} - 2K_{12})(\lambda_2)^2 + (1 - s + sK_{11}\gamma - sK_{12}\gamma + y_2v_1 - y_2v_2)\lambda_2 + const$$

Thus, equation (16) specifies dual function with subject to the second Lagrange multiplier λ_2 that is optimized in conjunction with the first one λ_1 by SMO algorithm.

$$\begin{gathered}
l(\lambda_2) = -\tfrac{1}{2}\eta(\lambda_2)^2 + \left(\eta\lambda_2^{\text{old}} + y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)\right)\lambda_2 + const \\
\text{Where} \\
E_j^{\text{old}} = y_j - \left(W^{old} \circ X_j - b^{\text{old}}\right) \\
\eta = K_{11} - 2K_{12} + K_{22} = X_1 \circ X_1 - 2X_1 \circ X_2 + X_2 \circ X_2 \\
W^{old} = \lambda_1^{\text{old}}y_1X_1 + \lambda_2^{\text{old}}y_2X_2 + \textstyle\sum_{i=3}^{n}\lambda_iy_iX_i \\
\lambda_1 = \gamma - s\lambda_2 \\
\gamma = \lambda_1 + s\lambda_2 = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}} \\
s = y_1y_2
\end{gathered} \tag{16}$$

The first and second derivatives of dual function $l(\lambda_2)$ with regard to λ_2 are:

$$\frac{\mathrm{d}l(\lambda_2)}{\mathrm{d}\lambda_2} = -\eta\lambda_2 + \eta\lambda_2^{\text{old}} + y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)$$

$$\frac{\mathrm{d}^2l(\lambda_2)}{\mathrm{d}(\lambda_2)^2} = -\eta$$

The quantity η is always non-negative due to:

$$\eta = X_1 \circ X_1 - 2X_1 \circ X_2 + X_2 \circ X_2 = (X_1 - X_2) \circ (X_1 - X_2) = |X_1 - X_2|^2 \geq 0$$

Recall that the goal of QP problem is to maximize the dual function $l(\lambda_2)$ so as to find out the optimal multiplier (maximum point) λ_2^*. The second derivative of $l(\lambda_2)$ is always non-negative and so, $l(\lambda_2)$ is concave function and there always exists the maximum point λ_2^*. The function $l(\lambda_2)$ gets maximal if its first derivative is equal to zero:

$$\frac{\mathrm{d}l(\lambda_2)}{\mathrm{d}\lambda_2} = 0 \Longrightarrow -\eta\lambda_2 + \eta\lambda_2^{\text{old}} + y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right) = 0 \Longrightarrow \lambda_2 = \lambda_2^{\text{old}} + \frac{y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)}{\eta}$$

Therefore, the new values of λ_1 and λ_2 that are solutions of the smallest optimization problem of SMO algorithm are:

$$\lambda_2^* = \lambda_2^{\text{new}} = \lambda_2^{\text{old}} + \frac{y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)}{\eta}$$

$$\lambda_1^* = \lambda_1^{\text{new}} = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}} - s\lambda_2^{\text{new}} = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}} - s\lambda_2^{\text{old}} - s\frac{y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)}{\eta} = \lambda_1^{\text{old}} - s\frac{y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)}{\eta}$$

Obviously, λ_1^{new} is totally determined in accordance with λ_2^{new}, thus we should focus on λ_2^{new}. Because multipliers λ_i are bounded, $0 \le \lambda_i \le C$, it is required to find out the range of λ_2^{new}. Let L and U be lower bound and upper bound of λ_2^{new}, respectively. We have [3, pp. 11-13]:

1. If $s = 1$, then $\lambda_1 + \lambda_2 = \gamma$. There are two sub-cases (see fig. 5 [3, p. 12]) as follows [3, p. 11]:
   If $\gamma \ge C$ then $L = \gamma - C$ and $U = C$.
   If $\gamma < C$ then $L = 0$ and $U = \gamma$.
2. If $s = -1$, then $\lambda_1 - \lambda_2 = \gamma$. There are two sub-cases (see fig. 6 [3, p. 13]) as follows [3, pp. 11-12]:
   If $\gamma \ge 0$ then $L = 0$ and $U = C - \gamma$.
   If $\gamma < 0$ then $L = -\gamma$ and $U = C$.

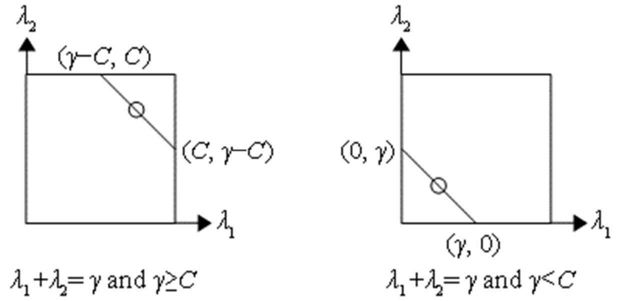

Figure 5. *Lower bound and upper bound of two new multipliers in case s = 1.*

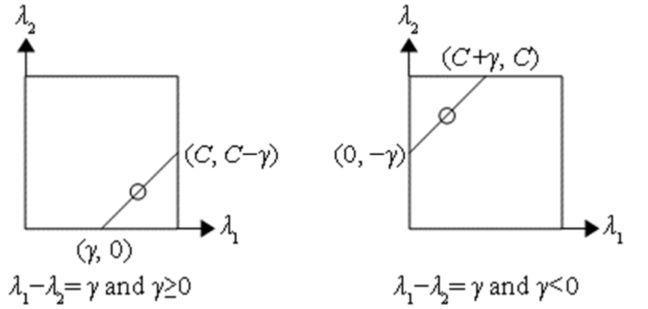

Figure 6. *Lower bound and upper bound of two new multipliers in case s = –1.*

Table 2. *SMO algorithm optimizes jointly two Lagrange multipliers.*

If $\eta > 0$:

$$\lambda_2^{\text{new}} = \lambda_2^{\text{old}} + \frac{y_2\left(E_2^{\text{old}} - E_1^{\text{old}}\right)}{\eta}$$

$$\lambda_2^* = \lambda_2^{\text{new,clipped}} = \begin{cases} L & \text{if } \lambda_2^{\text{new}} < L \\ \lambda_2^{\text{new}} & \text{if } L \le \lambda_2^{\text{new}} \le U \\ U & \text{if } U < \lambda_2^{\text{new}} \end{cases}$$

If $\eta = 0$:

$$\lambda_2^* = \lambda_2^{\text{new,clipped}} = \underset{\lambda_2}{\text{argmax}}\{l(\lambda_2 = L), l(\lambda_2 = U)\}$$

Where prediction errors E_j^{old} and dual function $l(\lambda_2)$ are specified by (16).
Lower bound L and upper bound U are described as follows:

1. If s=1 and $\gamma > C$ then $L = \gamma - C$ and $U = C$.
2. If s=1 and $\gamma < C$ then $L = 0$ and $U = \gamma$.
3. If s=–1 and $\gamma > 0$ then $L = 0$ and $U = C - \gamma$.
4. If s=–1 and $\gamma < 0$ then $L = -\gamma$ and $U = C$.

Where $\gamma = \lambda_1 + s\lambda_2 = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}}$ according to (15).
Let $\Delta\lambda_1$ and $\Delta\lambda_2$ represent the changes in multipliers λ_1 and λ_2, respectively.

$$\Delta\lambda_2 = \lambda_2^* - \lambda_2^{\text{old}}$$
$$\Delta\lambda_1 = -s\Delta\lambda_2$$

The new value of the first multiplier λ_1 is re-written in accordance with the change $\Delta\lambda_1$.

$$\lambda_1^* = \lambda_1^{\text{new}} = \lambda_1^{\text{old}} + \Delta\lambda_1$$

The value λ_2^{new} is clipped as follows [3, p. 12]:

$$\lambda_2^{\text{new,clipped}} = \begin{cases} L & \text{if } \lambda_2^{\text{new}} < L \\ \lambda_2^{\text{new}} & \text{if } L \le \lambda_2^{\text{new}} \le U \\ U & \text{if } U < \lambda_2^{\text{new}} \end{cases}$$

In the case η=0 that λ_2^{new} is undetermined, $\lambda_2^{\text{new,clipped}}$ is assigned by which bound (L or U) maximizes the dual function $l(\lambda_2)$.

$$\lambda_2^{\text{new,clipped}} = \underset{\lambda_2}{\text{argmax}}\{l(\lambda_2 = L), l(\lambda_2 = U)\} \text{ if } \eta = 0$$

In general, table 2 summarizes how SMO algorithm optimizes jointly two Lagrange multipliers.

Basic tasks of SMO algorithm to optimize jointly two Lagrange multipliers are now described in detailed. The ultimate goal of SVM method is to determine the classifier (W^*, b^*). Thus, SMO algorithm updates optimal weight W^* and optimal bias b^* based on the new values λ_1^{new} and λ_2^{new} at each optimization step.

Table 3. *SMO algorithm.*

All multipliers λ_i (s), weight vector W, and bias b are initialized by zero. SMO algorithm divides the whole QP problem into many smallest optimization problems. Each smallest optimization problem focuses on optimizing two joint multipliers. SMO algorithm solves each smallest optimization problem via two nested loops:

1. The outer loop alternates one sweep through all data points and as many sweeps as possible through non-boundary data points (support vectors) so as to find out the data point X_i that violates KKT condition according to (13). The Lagrange multiplier λ_i associated with such Xi is selected as the first multiplier aforementioned as λ_1. Violating KKT condition is known as the first choice heuristic of SMO algorithm.
2. The inner loop browses all data points at the first sweep and non-boundary ones at later sweeps so as to find out the data point X_j that maximizes the deviation $\left|E_j^{\text{old}} - E_i^{\text{old}}\right|$ where E_j^{old} and E_i^{old} are prediction errors on X_i and X_j, respectively, as seen in (16). The Lagrange multiplier λ_j associated with such X_j is selected as the second multiplier aforementioned as λ_2. Maximizing the deviation $\left|E_2^{\text{old}} - E_1^{\text{old}}\right|$ is known as the second choice heuristic of SMO algorithm.
   a. Two Lagrange multipliers λ_1 and λ_2 are optimized jointly, which results optimal multipliers λ_1^{new} and λ_2^{new}, as seen in table 2.
   b. SMO algorithm updates optimal weight W^* and optimal bias b^* based on the new values λ_1^{new} and λ_2^{new} according to (17).

SMO algorithm continues to solve another smallest optimization problem. SMO algorithm stops when there is convergence in which no data point violating KKT condition is found. Consequently, all Lagrange multipliers λ_1, λ_2,…, λ_n are optimized and the optimal SVM classifier (W^*, b^*) is totally determined.

Let $W^* = W^{\text{new}}$ be the new (optimal) weight vector, according (11) we have:

$$W^{\text{new}} = \sum_{i=1}^{n} \lambda_i y_i X_i = \lambda_1^{\text{new}} y_1 X_1 + \lambda_2^{\text{new}} y_2 X_2 + \sum_{i=3}^{n} \lambda_i y_i X_i$$

Let $W^* = W^{\text{old}}$ be the old weight vector:

$$W^{\text{old}} = \sum_{i=1}^{n} \lambda_i y_i X_i = \lambda_1^{\text{old}} y_1 X_1 + \lambda_2^{\text{old}} y_2 X_2 + \sum_{i=3}^{n} \lambda_i y_i X_i$$

It implies:

$$W^* = W^{\text{new}} = \lambda_1^{\text{new}} y_1 X_1 - \lambda_1^{\text{old}} y_1 X_1 + \lambda_2^{\text{new}} y_2 X_2 - \lambda_2^{\text{old}} y_2 X_2 + W^{\text{old}}$$

Let E_2^{new} be the new prediction error on X_2:

$$E_2^{\text{new}} = y_2 - (W^{\text{new}} \circ X_2 - b^{\text{new}})$$

The new (optimal) bias $b^* = b^{\text{new}}$ is determining by setting $E_2^{\text{new}} = 0$ with reason that the optimal classifier (W^*, b^*) has zero error.

$$E_2^{\text{new}} = 0 \Leftrightarrow y_2 - (W^{\text{new}} \circ X_2 - b^{\text{new}}) = 0 \Leftrightarrow b^{\text{new}} = W^{\text{new}} \circ X_2 - y_2$$

In general, equation (17) specifies the optimal classifier (W^*, b^*) resulted from each optimization step of SMO algorithm.

$$\begin{gathered} W^* = W^{\text{new}} = \left(\lambda_1^{\text{new}} - \lambda_1^{\text{old}}\right) y_1 X_1 \\ +\left(\lambda_2^{\text{new}} - \lambda_2^{\text{old}}\right) y_2 X_2 + W^{\text{old}} \\ \text{Where } W^{\text{old}} \text{ is the old value of weight vector,} \\ \text{of course we have:} \\ W^{\text{old}} = \lambda_1^{\text{old}} y_1 X_1 + \lambda_2^{\text{old}} y_2 X_2 + \sum_{i=3}^{n} \lambda_i y_i X_i \end{gathered} \quad (17)$$

By extending the ideology shown in table 1, SMO algorithm is described particularly in table 3 [7, pp. 8-9] [3, p. 14].

When both optimal weight vector W^* and optimal bias b^* are determined by SMO algorithm or other methods, the maximum-margin hyperplane known as SVM classifier is totally determined. According to (1), the equation of maximum-margin hyperplane is expressed in (18) as follows:

$$W^* \circ X - b^* = 0 \quad (18)$$

For any data point X, classification rule derived from maximum-margin hyperplane (SVM classifier) is used to classify such data point X. Let R be the classification rule, equation (19) specifies the classification rule as the sign function of point X.

$$R \overset{\text{def}}{=} sign(W^* \circ X - b^*) = \begin{cases} +1 \text{ if } W^* \circ X - b^* \geq 0 \\ -1 \text{ if } W^* \circ X - b^* < 0 \end{cases} \quad (19)$$

After evaluating R with regard to X, if $R(X)$ =1 then, X belongs to class +1; otherwise, X belongs to class –1. This is the simple process of data classification.

The next section illustrates how to apply SMO into classifying data points where such data points are documents.

3. An Example of Data Classification by SVM

Given a set of classes C = {*computer science*, *math*}, a set of terms T = {*computer*, *derivative*} and the corpus $\mathcal{D}$ = {*doc*1.*txt*, *doc*2.*txt*, *doc*3.*txt*, *doc*4.*txt*}. The training corpus (training data) is shown in following table 4 in which cell (i, j) indicates the number of times that term j (column j) occurs in document i (row i); in other words, each cell represents a term frequency and each row represents a document. There are four documents and each document belongs to only one class: computer science or math.

Table 4. *Term frequencies of documents (SVM).*

	computer	derivative	class
doc1.txt	20	55	math
doc2.txt	20	20	computer science
doc3.txt	15	30	math
doc4.txt	35	10	computer science

Let X_i be data points representing documents *doc*1.*txt*, *doc*2.*txt*, *doc*3.*txt*, *doc*4.*txt*, *doc*5.*txt*. We have X_1=(20,55), X_2=(20,20), X_3=(15,30), and X_4=(35,10). Let y_i=+1 and y_i=–1 represent classes "*math*" and "*computer science*", respectively. Let x and y represent terms "*computer*" and "*derivative*", respectively and so, for example, it is interpreted that the data point X_1=(20,55) has abscissa x=20 and ordinate y=55. Therefore, term frequencies from table 4 is interpreted as SVM input training corpus shown in table 5.

Table 5. *Training corpus (SVM).*

	x	y	y_i
X_1	20	55	+1
X_2	20	20	–1
X_3	15	30	+1
X_4	35	10	–1

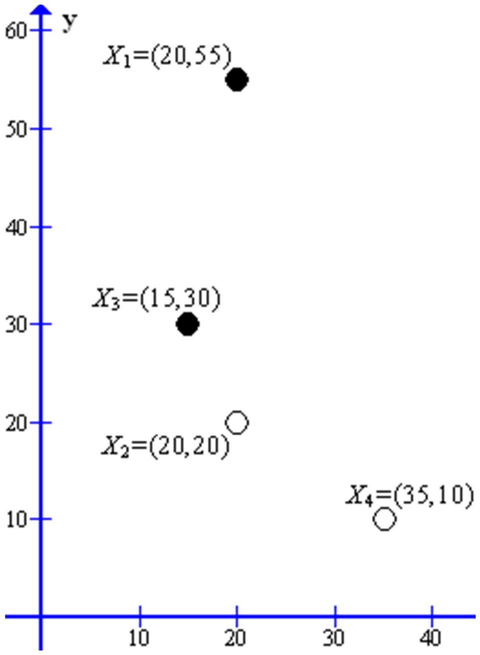

Figure 7. *Data points in training data (SVM).*

Data points X_1, X_2, X_3, and X_4 are depicted in fig. 7 in which classes "*math*" (y_i=+1) and "*computer*" (y_i=–1) are represented by shading and hollow circles, respectively. Note that fig. 7 and fig. 8 in this report are drawn by the software *Graph* http://www.padowan.dk developed by author Ivan Johansen [9].

By applying SMO algorithm described in table 3 into training corpus shown in table 5, it is easy to calculate optimal multiplier λ^*, optimal weight vector W^* and optimal bias b^*. Firstly, all multipliers λ_i (s), weight vector W, and bias b are initialized by zero. This example focuses on perfect separation and so, $C = +\infty$.

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$$

$$W = (0,0)$$

$$b = 0$$

$$C = +\infty$$

At the first sweep:

The outer loop of SMO algorithm searches for a data point X_i that violates KKT condition according to (13) through all data points so as to select two multipliers that will be optimized jointly. We have:

$$E_1 = y_1 - (W \circ X_1 - b) = 1 - \big((0,0) \circ (20,55) - 0\big) = 1$$

Due to λ_1=0 < C=+∞ and y_1E_1=1*1=1 > 0, point X_1 violates KKT condition according to (13). Then, λ_1 is selected as the first multiplier. The inner loop finds out the data point X_j that maximizes the deviation $\left|E_j^{\text{old}} - E_1^{\text{old}}\right|$. We have:

$$E_2 = y_2 - (W \circ X_2 - b) = -1 - \big((0,0) \circ (20,20) - 0\big) = -1$$

$$|E_2 - E_1| = |-1 - 1| = 2$$

$$E_3 = y_3 - (W \circ X_3 - b) = 1 - \big((0,0) \circ (15,30) - 0\big) = 1$$

$$|E_3 - E_1| = |1 - 1| = 0$$

$$E_4 = y_4 - (W \circ X_4 - b) = -1 - \big((0,0) \circ (35,10) - 0\big) = -1$$

$$|E_4 - E_1| = |-1 - 1| = 2$$

Because the deviation $|E_2 - E_1|$ is maximal, the multiplier λ_2 associated with X_2 is selected as the second multiplier. Now λ_1 and λ_2 are optimized jointly according to table 2.

$$\eta = X_1 \circ X_1 - 2X_1 \circ X_2 + X_2 \circ X_2 = |X_1 - X_2|^2 = |(20,55) - (20,20)|^2 = |(0,35)|^2 = 35^2 = 1225$$

$$s = y_1y_2 = 1 * (-1) = -1$$

$$\gamma = \lambda_1^{\text{old}} + s\lambda_2^{\text{old}} = 0 + (-1) * 0 = 0$$

$$\lambda_2^{\text{new}} = \lambda_2^{\text{old}} + \frac{y_2(E_2 - E_1)}{\eta} = 0 + \frac{-1*(-1-1)}{1225} = \frac{2}{1225}$$

$$\Delta\lambda_2 = \lambda_2^{\text{new}} - \lambda_2^{\text{old}} = \frac{2}{1225} - 0 = \frac{2}{1225}$$

$$\lambda_1^{\text{new}} = \lambda_1^{\text{old}} - y_1y_2\Delta\lambda_2 = 0 - 1 * (-1) * \frac{2}{1225} = \frac{2}{1225}$$

Optimal classifier (W^*, b^*) is updated according to (17).

$$W^* = W^{\text{new}} = \left(\lambda_1^{\text{new}} - \lambda_1^{\text{old}}\right)y_1X_1 + \left(\lambda_2^{\text{new}} - \lambda_2^{\text{old}}\right)y_2X_2 + W^{\text{old}} = \left(\frac{2}{1225} - 0\right) * 1 * (20,55) + \left(\frac{2}{1225} - 0\right) * (-1) * (20,20) + (0,0) = \left(0, \frac{2}{35}\right)$$

$$b^* = b^{\text{new}} = W^{\text{new}} \circ X_2 - y_2 = \left(0, \frac{2}{35}\right) \circ (20,20) - (-1) = \frac{15}{7}$$

Now we have:

$$\lambda_1 = \lambda_1^{\text{new}} = \frac{2}{1225}, \lambda_2 = \lambda_2^{\text{new}} = \frac{2}{1225}, \lambda_3 = \lambda_4 = 0$$

$$W = W^* = \left(0, \frac{2}{35}\right)$$

$$b = b^* = \frac{15}{7}$$

The outer loop of SMO algorithm continues to search for another data point X_i that violates KKT condition according to (13) through all data points so as to select two other multipliers that will be optimized jointly. We have:

$$E_2 = y_2 - (W \circ X_2 - b) = -1 - \left(\left(0, \frac{2}{35}\right) \circ (20,20) - \frac{15}{7}\right) = 0$$

$$E_3 = y_3 - (W \circ X_3 - b) = 1 - \left(\left(0, \frac{2}{35}\right) \circ (15,30) - \frac{15}{7}\right) = \frac{10}{7}$$

Due to λ_3=0 < C and y_3E_3=1*(10/7) > 0, point X_3 violates KKT condition according to (13). Then, λ_3 is selected as the first multiplier. The inner loop finds out the data point X_j that maximizes the deviation $\left|E_j^{\text{old}} - E_3^{\text{old}}\right|$. We have:

$$E_1 = y_1 - (W \circ X_1 - b) = 1 - \left(\left(0, \frac{2}{35}\right) \circ (20,55) - \frac{15}{7}\right) = 0$$

$$|E_1 - E_3| = \left|0 - \frac{10}{7}\right| = \frac{10}{7}$$

$$E_2 = y_2 - (W \circ X_2 - b) = -1 - \left(\left(0, \frac{2}{35}\right) \circ (20,20) - \frac{15}{7}\right) = 0$$

$$|E_2 - E_3| = \left|0 - \frac{10}{7}\right| = \frac{10}{7}$$

$$E_4 = y_4 - (W \circ X_4 - b) = -1 - \left(\left(0, \frac{2}{35}\right) \circ (35,10) - \frac{15}{7}\right) = \frac{6}{7}$$

$$|E_4 - E_3| = \left|\frac{6}{7} - \frac{10}{7}\right| = \frac{4}{7}$$

Because both deviations $|E_1 - E_3|$ and $|E_2 - E_3|$ are maximal, the multiplier λ_2 associated with X_2 is selected randomly among $\{\lambda_1, \lambda_2\}$ as the second multiplier. Now λ_3 and λ_2 are optimized jointly according to table 2.

$$\eta = X_3 \circ X_2 - 2X_3 \circ X_2 + X_3 \circ X_2 = |X_3 - X_2|^2 = |(15{,}30) - (20{,}20)|^2 = |(-5{,}10)|^2 = (-5)^2 + 10^2 = 125$$

$$s = y_3 y_2 = 1 * (-1) = -1$$

$$\gamma = \lambda_3^{\mathrm{old}} + s\lambda_2^{\mathrm{old}} = 0 + (-1) * \frac{2}{1225} = -\frac{2}{1225}$$

$$L = -\gamma = \frac{2}{1225}$$

$$U = C = +\infty$$

(L and U are lower bound and upper bound of λ_2^{new})

$$\lambda_2^{\mathrm{new}} = \lambda_2^{\mathrm{old}} + \frac{y_2(E_2 - E_3)}{\eta} = \frac{2}{1225} + \frac{-1*\left(0 - \frac{10}{7}\right)}{125} = \frac{16}{1225}$$

$$\Delta\lambda_2 = \lambda_2^{\mathrm{new}} - \lambda_2^{\mathrm{old}} = \frac{16}{1225} - \frac{2}{1225} = \frac{2}{175}$$

$$\lambda_3^{\mathrm{new}} = \lambda_3^{\mathrm{old}} - y_3 y_2 \Delta\lambda_2 = 0 - 1 * (-1) * \frac{2}{175} = \frac{2}{175}$$

Optimal classifier (W^*, b^*) is updated according to (17).

$$W^* = W^{\mathrm{new}} = \left(\lambda_3^{\mathrm{new}} - \lambda_3^{\mathrm{old}}\right) y_3 X_3 + \left(\lambda_2^{\mathrm{new}} - \lambda_2^{\mathrm{old}}\right) y_2 X_2 + W^{\mathrm{old}} = \left(\frac{2}{175} - 0\right) * 1 * (15{,}30) + \left(\frac{16}{1225} - \frac{2}{1225}\right) * (-1) * (20{,}20) + \left(0, \frac{2}{35}\right) = \left(-\frac{2}{35}, \frac{6}{35}\right)$$

$$b^* = b^{\mathrm{new}} = W^{\mathrm{new}} \circ X_2 - y_2 = \left(-\frac{2}{35}, \frac{6}{35}\right) \circ (20{,}20) - (-1) = \frac{23}{7}$$

Now we have:

$$\lambda_1 = \frac{2}{1225}, \lambda_2 = \lambda_2^{\mathrm{new}} = \frac{16}{1225}, \lambda_3 = \lambda_3^{\mathrm{new}} = \frac{2}{175}, \lambda_4 = 0$$

$$W = W^* = \left(-\frac{2}{35}, \frac{6}{35}\right)$$

$$b = b^* = \frac{23}{7}$$

The outer loop of SMO algorithm continues to search for another data point X_i that violates KKT condition according to (13) through all data points so as to select two other multipliers that will be optimized jointly. We have:

$$E_4 = y_4 - (W \circ X_4 - b) = -1 - \left(\left(-\frac{2}{35}, \frac{6}{35}\right) \circ (35{,}10) - \frac{23}{7}\right) = \frac{18}{7}$$

Due to λ_4=0 < C and y_4E_4=(–1)*(18/7) < 0, point X_4 does not violate KKT condition according to (13). Then, the first sweep of outer loop stops with the results as follows:

$$\lambda_1 = \frac{2}{1225}, \lambda_2 = \lambda_2^{\mathrm{new}} = \frac{16}{1225}, \lambda_3 = \lambda_3^{\mathrm{new}} = \frac{2}{175}, \lambda_4 = 0$$

$$W = W^* = \left(-\frac{2}{35}, \frac{6}{35}\right)$$

$$b = b^* = \frac{23}{7}$$

Note that λ_1 is approximated to 0 because it is very small.

At the second sweep:

The outer loop of SMO algorithm searches for a data point X_i that violates KKT condition according to (13) through non-boundary data points so as to select two multipliers that will be optimized jointly. Recall that non-boundary data points (support vectors) are ones whose associated multipliers are not bounds 0 and C ($0<\lambda_i<C$). At the second sweep, there are three non-boundary data points X_1, X_2 and X_3. We have:

$$E_1 = y_1 - (W \circ X_1 - b) = 1 - \left(\left(-\frac{2}{35}, \frac{6}{35}\right) \circ (20{,}55) - \frac{23}{7}\right) = -4$$

Due to $\lambda_1 = 2/1225 > 0$ and y_1E_1 = 1*(–4) < 0, point X_1 violates KKT condition according to (13). Then, λ_1 is selected as the first multiplier. The inner loop finds out the data point X_j among non-boundary data points ($0<\lambda_i<C$) that maximizes the deviation $\left|E_j^{\mathrm{old}} - E_1^{\mathrm{old}}\right|$. We have:

$$E_2 = y_2 - (W \circ X_2 - b) = -1 - \left(\left(-\frac{2}{35}, \frac{6}{35}\right) \circ (20{,}20) - \frac{23}{7}\right) = 0$$

$$|E_2 - E_1| = |0 - (-4)| = 4$$

$$E_3 = y_3 - (W \circ X_3 - b) = 1 - \left(\left(-\frac{2}{35}, \frac{6}{35}\right) \circ (15{,}30) - \frac{23}{7}\right) = 0$$

$$|E_3 - E_1| = |0 - (-4)| = 4$$

Because the deviation $|E_2 - E_1|$ is maximal, the multiplier λ_2 associated with X_2 is selected as the second multiplier. Now λ_1 and λ_2 are optimized jointly according to table (2).

$$\eta = X_1 \circ X_1 - 2X_1 \circ X_2 + X_2 \circ X_2 = |X_1 - X_2|^2 = |(20{,}55) - (20{,}20)|^2 = |(0{,}35)|^2 = 35^2 = 1225$$

$$s = y_1 y_2 = 1 * (-1) = -1$$

$$\gamma = \lambda_1^{\mathrm{old}} + s\lambda_2^{\mathrm{old}} = \frac{2}{1225} + (-1) * \frac{16}{1225} = -\frac{14}{1225}$$

$$L = -\gamma = \frac{14}{1225} = \frac{2}{175}$$

$$U = C = +\infty$$

(L and U are lower bound and upper bound of λ_2^{new})

$$\lambda_2^{\mathrm{new}} = \lambda_2^{\mathrm{old}} + \frac{y_2(E_2 - E_1)}{\eta} = \frac{16}{1225} + \frac{-1*\left(0 - (-4)\right)}{1225} = \frac{16}{1225} - \frac{4}{1225} = \frac{12}{1225} < L = \frac{2}{175}$$

$$\Longrightarrow \lambda_2^{\mathrm{new}} = L = \frac{2}{175}$$

$$\Delta\lambda_2 = \lambda_2^{\text{new}} - \lambda_2^{\text{old}} = \frac{2}{175} - \frac{16}{1225} = -\frac{2}{1225}$$

$$\lambda_1^{\text{new}} = \lambda_1^{\text{old}} - y_1 y_2 \Delta\lambda_2 = \frac{2}{1225} - 1 * (-1) * \left(-\frac{2}{1225}\right) = 0$$

Optimal classifier (W^*, b^*) is updated according to (17).

$$W^* = W^{\text{new}} = (\lambda_1^{\text{new}} - \lambda_1^{\text{old}}) y_1 X_1 + (\lambda_2^{\text{new}} - \lambda_2^{\text{old}}) y_2 X_2 + W^{\text{old}} = \left(0 - \frac{2}{1225}\right) * 1 * (20{,}55) + \left(\frac{2}{175} - \frac{16}{1225}\right) * (-1) * (20{,}20) + \left(-\frac{2}{35}, \frac{6}{35}\right) = \left(-\frac{2}{35}, \frac{4}{35}\right)$$

$$b^* = b^{\text{new}} = W^{\text{new}} \circ X_2 - y_2 = \left(-\frac{2}{35}, \frac{4}{35}\right) \circ (20{,}20) - (-1) = \frac{15}{7}$$

The second sweep stops with results as follows:

$$\lambda_1 = \lambda_1^{\text{new}} = 0, \lambda_2 = \lambda_2^{\text{new}} = \frac{2}{175}, \lambda_3 = \frac{2}{175}, \lambda_4 = 0$$

$$W = W^* = \left(-\frac{2}{35}, \frac{4}{35}\right)$$

$$b = b^* = \frac{15}{7}$$

At the third sweep:

The outer loop of SMO algorithm searches for a data point X_i that violates KKT condition according to (13) through non-boundary data points so as to select two multipliers that will be optimized jointly. Recall that non-boundary data points (support vectors) are ones whose associated multipliers are not bounds 0 and C ($0<\lambda_i<C$). At the third sweep, there are only two non-boundary data points X_2 and X_3. We have:

$$E_2 = y_2 - (W \circ X_2 - b) = -1 - \left(\left(-\frac{2}{35}, \frac{4}{35}\right) \circ (20{,}20) - \frac{15}{7}\right) = 0$$

$$E_3 = y_3 - (W \circ X_3 - b) = 1 - \left(\left(-\frac{2}{35}, \frac{4}{35}\right) \circ (15{,}30) - \frac{15}{7}\right) = \frac{4}{7}$$

Due to $\lambda_3 = 2/175 < C = +\infty$ and y_3E_3 = 1*(4/7) > 0, point X_3 violates KKT condition according to (13). Then, λ_3 is selected as the first multiplier. Because there are only two non-boundary data points X_2 and X_3, the second multiplier is λ_2. Now λ_3 and λ_2 are optimized jointly according to table (2).

$$\eta = X_3 \circ X_2 - 2X_3 \circ X_2 + X_3 \circ X_2 = |X_3 - X_2|^2 = |(15{,}30) - (20{,}20)|^2 = |(-5{,}10)|^2 = (-5)^2 + (10)^2 = 125$$

$$s = y_3 y_2 = 1 * (-1) = -1$$

$$\gamma = \lambda_3^{\text{old}} + s\lambda_2^{\text{old}} = \frac{2}{175} + (-1) * \frac{2}{175} = 0$$

$$L = 0$$

$$U = C - \gamma = +\infty$$

(L and U are lower bound and upper bound of λ_2^{new})

$$\lambda_2^{\text{new}} = \lambda_2^{\text{old}} + \frac{y_2(E_2 - E_3)}{\eta} = \frac{2}{175} + \frac{-1 * \left(0 - \frac{4}{7}\right)}{125} = \frac{2}{125}$$

$$\Delta\lambda_2 = \lambda_2^{\text{new}} - \lambda_2^{\text{old}} = \frac{2}{125} - \frac{2}{175} = \frac{4}{875}$$

$$\lambda_3^{\text{new}} = \lambda_3^{\text{old}} - y_3 y_2 \Delta\lambda_2 = \frac{2}{175} - 1 * (-1) * \frac{4}{875} = \frac{2}{125}$$

Optimal classifier (W^*, b^*) is updated according to (17).

$$W^* = W^{\text{new}} = (\lambda_3^{\text{new}} - \lambda_3^{\text{old}}) y_3 X_3 + (\lambda_2^{\text{new}} - \lambda_2^{\text{old}}) y_2 X_2 + W^{\text{old}} = \left(\frac{2}{125} - \frac{2}{175}\right) * 1 * (15{,}30) + \left(\frac{2}{125} - \frac{2}{175}\right) * 1 * (20{,}20) + \left(-\frac{2}{35}, \frac{4}{35}\right) = \left(\frac{18}{175}, \frac{12}{35}\right)$$

$$b^* = b^{\text{new}} = W^{\text{new}} \circ X_2 - y_2 = \left(\frac{18}{175}, \frac{12}{35}\right) \circ (20{,}20) - (-1) = \frac{347}{35}$$

The third sweep stops with results as follows:

$$\lambda_1 = 0, \lambda_2 = \lambda_2^{\text{new}} = \frac{2}{125}, \lambda_3 = \lambda_3^{\text{new}} = \frac{2}{125}, \lambda_4 = 0$$

$$W = W^* = \left(\frac{18}{175}, \frac{12}{35}\right)$$

$$b = b^* = \frac{347}{35}$$

After the third sweep, two non-boundary multipliers were optimized jointly. You can sweep more times to get more optimal results because data point X_3 still violates KKT condition as follows:

$$E_3 = y_3 - (W \circ X_3 - b) = 1 - \left(\left(\frac{18}{175}, \frac{12}{35}\right) \circ (15{,}30) - \frac{347}{35}\right) = -\frac{32}{35}$$

Due to $\lambda_3 = 2/125 > 0$ and y_3E_3 = 1*(–32/35) < 0, point X_3 violates KKT condition according to (13). But it takes a lot of sweeps so that SMO algorithm reaches absolute convergence (E_3=0 and hence, no KKT violation) because the penalty C is set to be $+\infty$, which implicates the perfect separation. This is the reason that we can stop the SMO algorithm at the third sweep in this example. In general, you can totally stop the SMO algorithm after optimizing two last multipliers which implies that all multipliers were optimized.

As a result, W^* and b^* were determined:

$$W^* = \left(\frac{18}{175}, \frac{12}{35}\right)$$

$$b^* = \frac{347}{35}$$

The maximum-margin hyperplane (SVM classifier) is totally determined as below:

$$W^* \circ X - b^* = 0 \Longrightarrow \left(\frac{18}{175}, \frac{12}{35}\right) \circ (x, y) - \frac{347}{35} = 0$$

$$\Longrightarrow y = -0.3x + 28.9$$

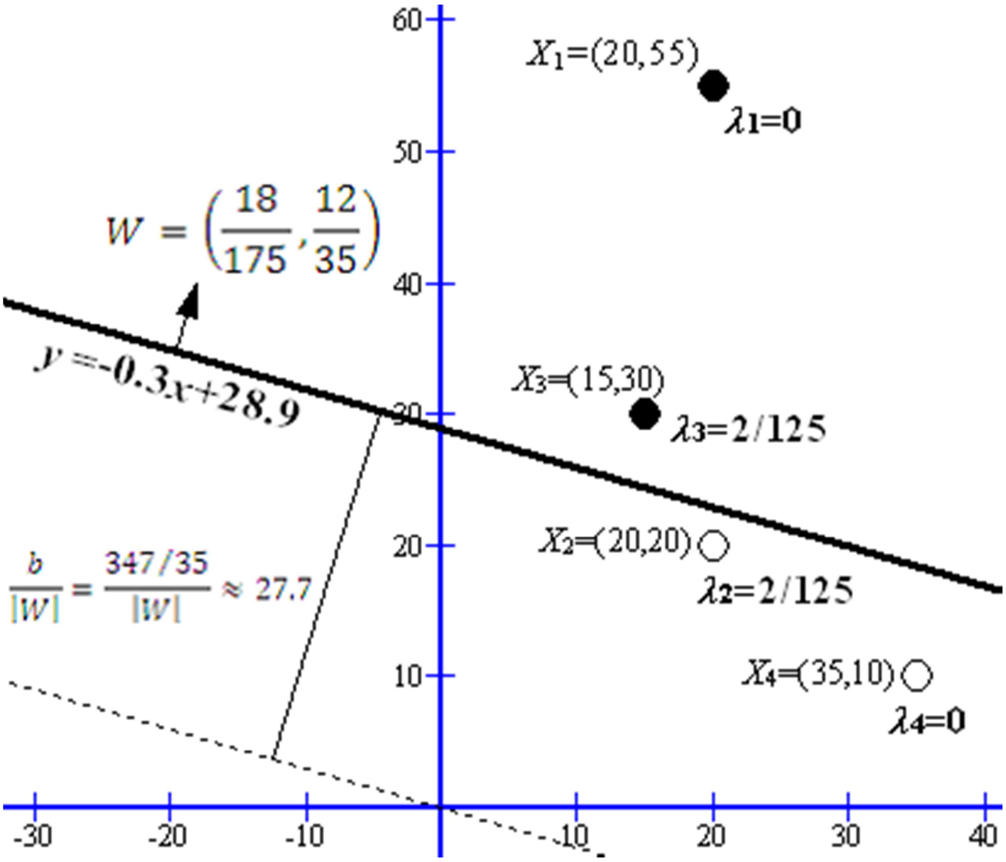

Figure 8. *An example of maximum-margin hyperplane.*

The SVM classifier $y = -0.3x + 28.9$ is depicted in fig. 8. Where the maximum-margin hyperplane is draw as bold line. Data points X_2 and X_3 are support vectors because their associated multipliers λ_2 and λ_3 are non-zero ($0<\lambda_2=2/125<C=+\infty$, $0<\lambda_3=2/125<C=+\infty$). Your attention please, that weight vector W is depicted as an arrow indicates mainly its direction. The scale of weight vector

$$W = (18/175, 12/35)$$

in fig. 8 is very small.

Derived from the above classifier $y = -0.3x + 28.9$, the classification rule is:

$$R \stackrel{\text{def}}{=} sign(0.3x + y - 28.9\,) = \begin{cases} +1 \text{ if } 0.3x + y - 28.9 \ \geq 0 \\ -1 \text{ if } 0.3x + y - 28.9 \ < 0 \end{cases}$$

Now we apply classification rule

$$R \stackrel{\text{def}}{=} sign(0.3x + y - 28.9\,)$$

into document classification. Suppose the numbers of times that terms "*computer*" and "*derivative*" occur in document D are 40 and 10, respectively. We need to determine which class document D=(40, 10) is belongs to. We have:

$$R(D) = sign(0.3 * 40 + 10 - 28.9) = sign(-6.9) = -1$$

Hence, it is easy to infer that document D belongs to class "*computer science*" ($y_i = -1$).

4. Conclusion

In general, the main ideology of SVM is to determine the separating hyperplane that maximizes the margin between two classes of training data. Based on theory of optimization, such optimal hyperplane is specified by the weight vector W^* and the bias b^* which are solutions of constrained optimization problem. It is proved that there always exist these solutions but the main issue of SVM is how to find out them when the constrained optimization problem is transformed into quadratic programming (QP) problem. SMO which is the most effective algorithm divides the whole QP problem into many smallest optimization problems. Each smallest optimization problem focuses on optimizing two joint multipliers. It is possible to state that SMO is the best implementation version of the "architecture" SVM.

SVM is extended by concept of kernel function. The dot product in separating hyperplane equation is the simplest kernel function. Kernel function is useful in case of requirement of data transformation [1, p. 21]. There are many pre-defined kernel functions available for SVM. Readers are recommended to research more about kernel functions [10].

References

[1] M. Law, "A Simple Introduction to Support Vector Machines," 2006.

[2] Wikibooks, "Support Vector Machines," Wikimedia Foundation, 1 January 2008. [Online]. Available: http://en.wikibooks.org/wiki/Support_Vector_Machines. [Accessed 2008].

[3] V. G. Honavar, "Sequential Minimal Optimization for SVM," Vasant Honavar homepage, Ames, Iowa, USA.

[4] S. Boyd and L. Vandenberghe, Convex Optimization, New York, NY: Cambridge University Press, 2009, p. 716.

[5] Wikipedia, "Karush–Kuhn–Tucker conditions," Wikimedia Foundation, 4 August 2014. [Online]. Available: http://en.wikipedia.org/wiki/Karush–Kuhn–Tucker_conditions. [Accessed 16 November 2014].

[6] Y.-B. Jia, "Lagrange Multipliers," 2013.

[7] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Microsoft Research, 1998.

[8] A. W. Moore, "Support Vector Machines," Available at http://www. cs. cmu. edu/~awm/tutorials, 2001.

[9] I. Johansen, Graph software, GNU General Public License, 2012. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (References).

[10] N. Cristianini, "Support Vector and Kernel Machines," in The 28th International Conference on Machine Learning (ICML), Bellevue, Washington, USA, 2001.