Reading: Murhpy Ch 14 (14.1, 14.2–14.2.4 kernels, 14.4 and equations (14.28, 14.29) kernel trick, 14.5.1–3 Support Vector Machines) Additional Reading: C. Burges - “A tutorial on SVM for pattern recognition”

These notes: Section 2, 3 (convex optimization) are optional.

**Notation reminder and a VC bound**

The data set: inputs $x^i \in \mathbb{R}^n$, $i = 1, \ldots, N$, labels $y^i \in \{-1, +1\}$

Assumption: $(x, y) \sim P_{XY}$, i.i.d

Classifier: $y = f(x, \theta)$ for new points $x$; $\theta$ = the parameters

The classifier family: $\mathcal{F} = \{f(\cdot, \theta)\}$

Empirical loss $\hat{L}_{01}(\theta) = \frac{1}{2N} \sum_i |y^i - \text{sgn} f(x^i, \theta)|$

Average loss $L_{01}(\theta) = \frac{1}{2} \int |y - \text{sgn} f(x, \theta)|dP(x, y)$

VC bound: $L_{01}(\theta) \leq \hat{L}_{01}(\theta) + \sqrt{\frac{h[1+\log(2N/h)]+\log(4/\delta)}{N}}$ w.p. $> 1 - \delta$, where $h = \text{VCdim } \mathcal{F}$ and $\delta < 1$ the confidence
A linear classifier is denoted as \( f(x; w, b) = w^T x + b \), where \( x \) takes label equal to \( \text{sgn}(f(x; w, b)) \). The margin of \( f \) on data point \( x^i \) is as usual equal to \( y^i f(x^i; w, b) \).

1 Linear SVM’s

1.1 The margin and the expected classification error

The following two theorems suggest that large margin is a predictor of good generalization error.

**Theorem** Let \( \mathcal{F}_R \) be the class of hyperplanes \( f(x) = w^T x, x, w \in \mathbb{R}^n \), that are \( R \) away from any data point\(^1\) in the training set \( \mathcal{D} \). Then,

\[
\text{VCdim } \mathcal{F}_R \leq 1 + \min \left( n, \frac{R_D^2}{R^2} \right)
\]

where \( R_D \) is the radius of the smallest ball that encloses the dataset.

**Theorem** Let \( \mathcal{F} = \{ \text{sgn}(w^T x), ||w|| \leq \Lambda, ||x|| \leq R \} \) and let \( \rho > 0 \) be any “margin”. Then for any \( f \in \mathcal{F} \), w.p \( 1 - \delta \) over training sets

\[
L_{01}(f) \leq \nu + \sqrt{\frac{c}{N} \left( \frac{R^2 \Lambda^2}{\rho^2} \ln N^2 + \ln \frac{1}{\delta} \right)}
\]

where \( \nu \) is the fraction of the training examples for which \( y^i w^T x_i < \rho \) and \( c \) is a universal constant.

1.2 Maximum Margin Linear classifiers

We assume for the moment that the data are linearly separable. Then, there are an infinity of linear classifiers that perform equally well on the training set. Which one to choose? The First idea leading to Support Vector Machines is to select the classifier that has maximum margin on the training set.

\(^1\)In other words, a set \( \mathcal{D} \) is shattered only if all the linear classifiers pass at least \( R \) away from its points.
Let \( \min_{i=1:N} y^i f(x^i) \) be the margin of classifier \( f \) on the training set. Hence, for a linear classifier \( f(x) = w^T x + b \), a first idea is to choose \( w, b \) so that \( \min_{i=1:N} y^i(w^T x^i + b) \) is maximized over \( w \in \mathbb{R}^n, b \in \mathbb{R} \). However, note that (a) if the data is linearly separable, the margins \( y^i(w^T x^i + b) \) are > 0 for all candidate classifiers, and (b) one can arbitrarily increase the margin of such a classifier by multiplying \( w \) and \( b \) by a positive constant. Hence, we need to “normalize” the set of candidate classifier by requiring instead

\[
\max \min_{i=1:N} d(x, H_{w,b}), \text{ s.t. } y^i(w^T x^i + b) \geq 1 \text{ for } i = 1 : N, \tag{3}
\]

where \( d() \) denotes the Euclidean distance and \( H_{w,b} = \{ x \mid w^T x + b = 0 \} \) is the decision boundary of the linear classifier.

**Fact** The distance of a point \( x \) to a hyperplane \( H_{w,b} \) is given by

\[
d(x, H_{w,b}) = \frac{|w^T x + b|}{||w||} \tag{4}
\]

Intuition: denote

\[
\tilde{w} = \frac{w}{||w||}, \quad \tilde{b} = \frac{b}{||w||} \tag{5}
\]

\[
x' = \tilde{w}^T x. \tag{6}
\]

Obviously \( H_{w,b} = H_{\tilde{w},\tilde{b}} \), and \( x' \) is the length of the projection of point \( x \) on the direction of \( w \). The distance is measured along the normal through \( x \) to \( H \); note that if \( x' = -\tilde{b} \) then \( x \in H_{w,b} \) and \( d(x, H_{w,b}) = 0 \); in general, the distance along this line will be \( |x' - (-\tilde{b})| \).

Under the special conditions of (3), because there are points for which \( |w^T x + b| = 1 \), maximizing \( d(x, H_{w,b}) \) over \( w, b \) for such a point is the same as minimizing \( ||w|| \). The **Second idea** leading to SVM is to formulate this as a quadratic problem. Thus, the **Linear SVM (primal) optimization problem** is

\[
\min_{w,b} \frac{1}{2} ||w||^2 \text{ s.t. } y^i(w^T x^i + b) - 1 \geq 0 \text{ for all } i = 1 : N \tag{7}
\]

**Optimization with Lagrange multipliers**

\[
L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum \alpha_i[y^i(w^T x^i + b) - 1]. \tag{8}
\]

\(^2\)The derivation of the dual problem is found in section 3.
KKT conditions imply:

\[ w = \sum_i \alpha_i y_i x_i \]  

(9)

**Support vector** is a data point \( x_i \) such that \( \alpha_i > 0 \). According to (9), the final decision boundary is determined by the support vectors (i.e. does not depend explicitly on any data point that is not a support vector).

The classifier \( w = \sum_{i, \alpha_i > 0} \alpha_i y_i x_i \), \( b = y_i - w^T x_i \) for some \( x_i \) support vector.

**Dual SVM optimization problem** Any convex optimization problem has a dual problem. In SVM, it is both illuminating and practical to solve the dual problem. The dual to problem (8) is

\[
\text{maximize } g(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j x_i^T x_j \text{ s.t. } \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_i \alpha_i y_i = 0.
\]

(10)

This is a quadratic problem with \( N \) variables on a convex domain. At the dual optimum, \( \alpha_i > 0 \) for constraints that are satisfied with equality, \( \alpha_i = 0 \) otherwise.

### 1.3 Non-linearly separable problems and their duals

The C-SVM

\[
\text{minimize}_{w,b,\xi} \quad \frac{1}{2} ||w||^2 + C \sum_i \xi_i \\
\text{subject to } \quad y_i(w^T x_i + b) \geq 1 - \xi_i \\
\xi_i \geq 0
\]

(11)

In the above, \( \xi_i \) are the slack variables. Lagrangean \( L(w, b, \xi, \alpha, \mu) = \frac{1}{2} ||w||^2 + C \sum_i \xi_i - \sum_i \alpha_i [y_i(w^T x_i + b) - 1 + \xi_i] - \sum_i \mu_i \xi_i \) with \( \alpha_i \geq 0 \), \( \xi_i \geq 0 \), \( \mu_i \geq 0 \).
Dual:

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j x_i^T x_j$$ (12)

s.t. \( C \geq \alpha_i \geq 0 \) for all \( i \)
\[ \sum_i \alpha_i y_i = 0 \]

\[ \Rightarrow \text{two types of SV} \]

- \( \alpha_i < C \) data point \( x^i \) is “on the margin” \( \Leftrightarrow y^i(w^T x^i + b) = 1 \) (original SV)
- \( \alpha_i = C \) data point \( x^i \) cannot be classified with margin 1 (margin error) \( \Leftrightarrow y^i(w^T x^i + b) < 1 \)

The \( \nu \)-SVM

$$\text{minimize}_{w, b, \xi, \rho} \quad \frac{1}{2} ||w||^2 - \nu \rho + \frac{1}{N} \sum_i \xi_i$$ (13)

s.t. \( y^i(w^T x^i + b) \geq \rho - \xi_i \) (14)
\[ \xi_i \geq 0 \] (15)
\[ \rho \geq 0 \] (16)

where \( \nu \in [0, 1] \) is a parameter. Lagrangean \( L(w, b, \xi, \rho, \alpha, \mu, \delta) = \frac{1}{2} ||w||^2 - \nu \rho + \frac{1}{N} \sum_i \xi_i - \alpha_i[y^i(w^T x^i + b) - \rho + \xi_i] - \sum_i \mu_i \xi_i - \delta \rho \) with \( \alpha_i \geq 0, \ \delta \geq 0, \ \mu_i \geq 0 \)

Dual:

$$\text{maximize}_\alpha \quad -\frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j x_i^T x_j$$ (17)

s.t. \( \frac{1}{N} \geq \alpha_i \geq 0 \) for all \( i \) (18)
\[ \sum_i \alpha_i y^i = 0 \] (19)
\[ \sum_i \alpha_i \geq \nu \] (20)

**Properties** If \( \rho > 0 \) then:
• $\nu$ is an upper bound on $\#\text{margin errors}/N$ (if $\sum_i \alpha_i = \nu$)
• $\nu$ is a lower bound on $\#(\text{original support vectors} + \text{margin errors})/N$
• $\nu$-SVM leads to the same $w, b$ as C-SVM with $C = 1/\nu$

A simple error bound

$$L_{01}(f_N) \leq E\left[\frac{\#\text{support vectors of } f_{N+1}}{N+1}\right]$$

(21)

where $f_N$ denotes the SVM trained on a sample of size $N$.

2 Convex optimization in a nutshell

A set $D \subseteq \mathbb{R}^n$ is convex iff for every two points $x^1, x^2 \in D$ the line segment defined by $x = tx^1 + (1-t)x^2, t \in [0, 1]$ is also in $D$. A function $f: D \to \mathbb{R}$ is convex iff, for any $x^1, x^2 \in D$ and for any $t \in [0, 1]$ for which $tx^1 + (1-t)x^2 \in D$ the following inequality holds

$$f(tx^1 + (1-t)x^2) \leq tf(x^1) + (1-t)f(x^2)$$

(22)

If $f$ is convex, then the set $\{ x \mid f(x) \leq c \}$ is convex for any value of $c$. Convex functions defined on convex sets have very interesting properties which have engendered the field called convex optimization.

The optimization problem

$$\min_x f_0(x)$$

s.t. $f_i(x) \leq 0$ for $i = 1, \ldots m$

(23)

is a convex optimization problem if all the functions $f, f_i$ are convex. Note that in this case the feasible domain $A = \cap_i \{ x \mid f_i(x) \leq 0 \}$ is a convex set.

It is known that if $A$ has a non empty interior then the convex optimization problem has at most one optimum $x^*$. If $A$ is also bounded, $x^*$ always exists.
Assuming that $x^*$ exists, there are two possible cases: (1) The **unconstrained minimum** of $f_0$ lies in $A$. In this case, the optimum can be found by solving the equations $\frac{\partial f_0}{\partial x} = 0$. (2) The unconstrained minimum of $f_0$ lies outside $A$. Figure 1 depicts what happens at the optimum $x^*$ in this case.

Assume there is only one constraint $f_1$. The domain $A$ is the inside of the curve $f_1(x) = 0$. The optimum $x^*$ is the point where a level curve $f_0(x) = c$ is tangent to $f_1 = 0$ from the outside. In this point, the gradients of two curves lie along the same line, pointing in opposite directions. Therefore, we can write $\frac{\partial f_0}{\partial x} = -\alpha \frac{\partial f_1}{\partial x}$. Equivalently, we have that at $x^*$, $\frac{\partial f_0}{\partial x} + \alpha \frac{\partial f_1}{\partial x} = 0$.

Note that this is a necessary but not a sufficient condition. The above set of equations represents the **Karush-Kuhn-Tucker optimality conditions (KKT)**.

With more than one constraint, the KKT conditions are equivalent to requiring that the gradient of $f_0$ lies in the subspace spanned by the gradients of the constraints.

$$\frac{\partial f_0}{\partial x} = -\sum_i \alpha_i \frac{\partial f_i}{\partial x} \text{ with } \alpha_i \geq 0 \text{ for all } i$$

(24)

Note that if a certain constraint $f_i$ does not participate in the boundary of $D$ at $x^*$, i.e if the constraint is not **active**, the coefficient $\alpha_i$ should be 0.
Equation (24) can be rewritten as
\[
\frac{\partial}{\partial x} \left[ f_0(x) + \sum_i \alpha_i f_i(x) \right] = 0 \quad \text{for some } \alpha_i \geq 0 \text{ for } i = 1, \ldots, m \tag{25}
\]

The optimum \(x^*\) has to satisfy the equation above. The new function \(L(x, \alpha)\) is the Lagrangean of the problem and the variables \(\alpha_i\) are called Lagrange multipliers. The Lagrangean is convex in \(x\) and affine (i.e linear + constant) in \(\alpha\).

The dual problem

Define the function
\[
g(\alpha) = \inf_x L(x, \alpha) \quad \alpha = (\alpha_i), \quad \alpha_i \geq 0 \tag{26}
\]

In the above, the infimum is over all the values of \(x\) for which \(f_0, f_i\) are defined, not just \(A\) (but everything still holds if the infimum is only taken over \(A\)). Two facts are important about \(g\)

- \(g(\alpha) \leq L(x, \alpha) \leq f(x)\) for any \(x \in A, \alpha \geq 0\), i.e \(g\) is a lower bound for \(f_0\), and implicitly for the optimal value \(f_0(x^*)\), for any value of \(\alpha \geq 0\).
- \(g(\alpha)\) is concave (i.e \(-g(\alpha)\) is convex).

We also can derive from (25) that if \(x^*\) exists then for an appropriate value \(\alpha^*\) we have
\[
g(\alpha^*) = L(x^*, \alpha^*) = f_0(x^*) + 0 \tag{27}
\]
and therefore \(g(\alpha^*)\) must be the unique maximum of \(g(\alpha)\). The second term in \(L\) above is zero because \(x^*\) is on the boundary of \(A\); hence for the active constraints \(f_i(x^*) = 0\) and for the inactive constraints \(\alpha_i^* = 0\). This surprising relationship shows that by solving the dual problem
\[
\max g(\alpha) \quad \text{s.t } \alpha \geq 0 \tag{28}
\]
we can obtain the values \(\alpha^*\) that plugged into (24) will allow us to find the solution \(x^*\) to our original (primal) problem. The constraints of the dual are simpler than the constraints of the primal. In practice, it is surprisingly often possible to compute the function \(g(\alpha)\) explicitly. Below we give a simple example thereof. This is also the case of the SVM optimization problem, which will be discussed in section 3.
2.1 A simple optimization example

Take as an example the convex optimization problem

$$
\min \frac{1}{2} x^2 \quad \text{s.t.} \quad x + 1 \leq 0
$$

(29)

By inspection the solution is $x^* = -1$.

Let us now apply to it the convex optimization machinery. We have

$$
L(x, \alpha) = \frac{1}{2} x^2 + \alpha (x + 1)
$$

(30)

defined for $x \in \mathbb{R}$ and $\alpha \geq 0$.

$$
g(\alpha) = \inf_x \left[ \frac{1}{2} x^2 + \alpha (x + 1) \right]
$$

(31)

$$
= \inf_x \left[ \frac{1}{2} (x + \alpha)^2 - \frac{1}{2} \alpha^2 + \alpha \right]
$$

(32)

$$
= -\frac{1}{2} \alpha^2 + \alpha
$$

(33)

$$
= \frac{1}{2} \alpha (2 - \alpha) \quad \text{attained for} \quad x = -\alpha
$$

(34)

The dual problem is

$$
\max \frac{1}{2} \alpha (2 - \alpha) \quad \text{s.t} \quad \alpha \geq 0
$$

(35)

and its solution is $\alpha = 1$ which, using equation (34) leads to $x = -1$.

From the KKT condition

$$
\frac{\partial L}{\partial x} = x + \alpha = 0
$$

(36)

we also obtain $x^* = -\alpha^* = -1$.

Figure 2 depicts the function $L$. Note that $L$ is convex in $x$ (a parabola) and that along the $\alpha$ axis the graph of $L$ consists of lines. The areas of $L$ that fall outside the admissible domain $x \leq -1$, $\alpha \geq 0$ are in flat (green) color. The crossection $L(x, \alpha = 0)$ represents the plot of $f$. The constrained minimum of $f$ is at $x = -1$, the unconstrained one is at $x = 0$ outside the admissible
domain. Note that \( g(\alpha) = L(-\alpha, \alpha) \) is concave, and that in the admissible domain it is always below the graph of \( f \). The (red) dot is the optimum \((x^*, \alpha^*)\), which represents a saddle point for \( h \). The line \( L(x = -1, \alpha) \) is horizontal (because \( f_1 = x + 1 = 0 \)) and thus \( L(x^*, \alpha^*) = L(x^*) = f(x^*) \).

3 The SVM solution by convex optimization

The SVM optimization problem

\[
\min_w \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad y^i(w^T x^i + b) \geq 1 \text{ for all } i \quad (37)
\]

is a convex (quadratic) optimization problem where

\[
f_0(w, b) = \frac{1}{2} \|w\|^2 \quad (38)
\]

\[
f_i(w, b) = -y^i w^T x^i + 1 - y^i b \quad (39)
\]
Hence,

\[ L(w, b, \alpha) = \frac{1}{2}||w||^2 + \sum_i \alpha_i [1 - y^i b - y^i x^i w] \]  \hspace{1cm} (40)

Equating the partial derivatives of \( h \) w.r.t \( w, b \) with 0 we get

\[ \frac{\partial L}{\partial w} = w - \sum_i \alpha_i y^i x^i \]  \hspace{1cm} (41)

\[ \frac{\partial L}{\partial b} = \sum_i \alpha_i y^i \]  \hspace{1cm} (42)

or, equivalently

\[ w = \sum_i \alpha_i y^i x^i \hspace{1cm} 0 = \sum_i \alpha_i y^i \]  \hspace{1cm} (43)

Hence, the normal \( w \) to the optimal separating hyperplane is a linear combination of data points. Moreover, we know that only those \( \alpha_i \) corresponding to active constraints will be non-zero. In the case of SVM, these represent points that are classified with \( y^i (w^T x^i + b) = 1 \). We call these points **support points** or **support vectors**. The solution of the SVM problem does not depend on all the data points, it depends only on the support vectors and therefore is **sparse**.

**Computing the solution.** SVM solvers use the dual problem to compute the solution. Below we derive the dual for the SVM problem. \( g(\alpha) \) is computed explicitly by replacing equation (43) in (40). After a simple calculation we obtain

\[ g(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y^i y^j x^i x^j \alpha_i \alpha_j \]  \hspace{1cm} (44)

or, in vector/matrix notation

\[ g(\alpha) = 1^T \alpha - \frac{1}{2} \alpha^T G \alpha \]  \hspace{1cm} (45)

where \( G = [G_{ij}]_{ij} = [y^i y^j x^{iT} x^j]_{ij} \).
3.1 A simple SVM problem

Data: 4 vectors in the plane and their labels

\[ x_1 = (-2, -2) \quad y_1 = +1 \]
\[ x_2 = (-1, 1) \quad y_2 = +1 \]
\[ x_3 = (1, 1) \quad y_3 = -1 \]
\[ x_4 = (2, -2) \quad y_4 = -1 \]

The Gramm matrix \( G = [x_i^T x_j]_{i,j=1:l} \)

\[
G = \begin{bmatrix}
8 & 0 & -4 & 0 \\
0 & 2 & 0 & -4 \\
-4 & 0 & 2 & 0 \\
0 & -4 & 0 & 8
\end{bmatrix}
\]

The dual function to be maximized (subject to \( \alpha_i \geq 0 \)) is

\[
g(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y_i y_j x_i^T x_i
\]

\[
= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - 4\alpha_1^2 - \alpha_2^2 - \alpha_3^2 - 4\alpha_4^2 - 4\alpha_1\alpha_3 - 4\alpha_2\alpha_4
\]

\[
= (2\alpha_1 + \alpha_3) - (2\alpha_1 + \alpha_3)^2 - \alpha_1
\]

\[
+ (\alpha_2 + 2\alpha_4) - (\alpha_2 + 2\alpha_4)^2 - \alpha_4
\]

The parts depending on \( \alpha_1, \alpha_3 \) and \( \alpha_2, \alpha_4 \) can be maximized separately, and after some short calculations we obtain:

\[
\begin{align*}
\alpha_1 &= 0 \\
\alpha_4 &= 0 \\
\alpha_2 &= \frac{1}{2} \\
\alpha_3 &= \frac{1}{2}
\end{align*}
\]

Hence, the support vectors are \( x_2 \) and \( x_3 \). From these, we obtain

\[
w = \sum_i \alpha_i y_i x_i = \frac{1}{2}(x_2 - x_3) = (-1, 0)
\]

\[
b = y_2 - w^T x_2 = 0
\]
4 Non linearly separable data: the “kernel trick”

We have seen so far how to construct a SVM classifier if the data are linearly separable i.e if there exist \( w, b \) such that the hyperplane \( w^T x + b = 0 \) leaves all the examples labeled +1 (called positive examples) on one side and all the examples labeled −1 (the negative examples) on its other side. If the data are not linearly separable, then no solution to the SVM optimization problem exists. Here we shall see a way of constructing SVM’s that are non linear in the sense that they separate the positive and negative example by a (hyper)surface that is non-linear.

An old trick that allows us to use linear classifier on data that is not linearly separable is the following:

1. Map the data to a higher dimensional space \( x \to z = \phi(x) \in H \), with \( \dim H \gg n \).
2. Construct a linear classifier $w^T z + b$ for the data in $H$

For example, the data $\{(x, y)\}$ below:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

are not linearly separable. We map them to 3 dimensions by $z = \phi(x) = [x_1 \ x_2 \ x_1x_2]$. Now it is easy to see that the classes can be separated by the hyperplane $z_3 = 0$ (which happens to be the maximum margin hyperplane). Hence $w = [001]$ (a vector in $H$) and $b = 0$ and the classification rule is $f(\phi(x)) = w^T \phi(x) + b$. If we express this rule as a function of the original $x$ we get $f(x) = x_1x_2$ which is a quadratic classifier.

In summary, by mapping the data to $H$ by $\phi(x)$ and then using a linear classifier, we are in fact implementing the non-linear classifier $f(x) = w^T \phi(x) + b = w_1\phi_1(x) + w_2\phi_2(x) + \ldots + w_m\phi_m(x) + b$ (46)

Rephrasing the non-linear classification problem in SV language we obtain:

Problem: minimize $||w||^2$ s.t. $y^i(w^T \phi(x^i) + b) - 1 \geq 0$ for all $i$.

Note that the only difference from the linear case is that $x^i$ is now replaced with $\phi(x^i)$. The Lagrangean is similar to the original Lagrangean:

$$L(w, b, \alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \alpha_j y^i y_j \phi(x^i)^T \phi(x_j) \text{ with } \alpha_i \geq 0 \text{ for all } i \quad (47)$$

How much harder has the optimization become now? Surprisingly, the optimization problem is no harder than it was before! Note that the Lagrangean has a linear term that depends only on $\alpha$ and a quadratic term that can be written

$$\bar{\alpha}^T G \bar{\alpha} \quad (48)$$

where $\bar{\alpha} = [\alpha_i y^i]_{i=1:t}$ and $G = [G_{ij}]_{i,j=1}^N$ is the **Gram matrix**

$$G_{ij} = G_{ji} = \phi(x^i)^T \phi(x_j) \quad \text{formerly} \quad G_{ij} = G_{ji} = (x^i)^T x_j \quad (49)$$

A few facts follow from this observation:
1. The $\phi$ vectors enter the SVM optimization problem only through the Gram matrix, thus only as the scalar products $\phi(x_i)^T \phi(x_j)$. We denote by $K(x, x')$ the function

$$K(x, x') = K(x', x) = \phi(x)^T \phi(x')$$  \hspace{1cm} (50)

$K$ is called the kernel function. If $K$ can be computed efficiently, then the Gram matrix $G$ can also be computed efficiently. This is exactly what one does in practice: we choose $\phi$ implicitly by choosing a kernel $K$. Hereby we also ensure that $K$ can be computed efficiently.

2. Once $G$ is obtained, the SVM optimization is independent of the dimension of $x$ and of the dimension of $z = \phi(x)$. The complexity of the SVM optimization depends only on $N$ the number of examples. This means that we can choose a very high dimensional $\phi$ without any penalty on the optimization cost.

3. Classifying a new point $x$. As we know, the SVM classification rule is

$$f(x) = w^T \phi(x) + b = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)^T \phi(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x)$$  \hspace{1cm} (51)

Hence, the classification rule is expressed in terms of the support vectors and the kernel only. No operations other than scalar product are performed in the high dimensional space $H$.

The above describes the celebrated kernel trick of the SVM literature.

5 Kernels

The previous section shows why SVMs are often called kernel machines. If we choose a kernel, we have all the benefits of a mapping in high dimensions, without ever carrying on any operations in that high dimensional space. The most usual kernel functions are

- $K(x, x') = (1 + x^T x')^p$ the polynomial kernel of degree $p$
- $K(x, x') = e^{-\frac{||x-x'||^2}{\sigma^2}}$ the Gaussian or radial basis function (RBF) kernel, it’s $\phi$ is $\infty$-dimensional
- $K(x, x') = \tanh(\sigma x^T x' - \beta)$ the “neural network” kernel
How do we verify that a symmetric function $K$ is a valid kernel, i.e. that there is a mapping $\phi$ for which $K$ is the scalar product? This is ensured by the Mercer condition which is a positivity condition

$$\int K(x, x')g(x)g(x')dxdx' \geq 0 \text{ for all } g \text{ such that } \|g(x)\|_{L^2} < \infty$$  \hspace{1cm} (52)

### 6 Extensions to other problems

#### 6.1 Multi-class SVM

For a problem with $K$ possible classes, we construct $K$ separating hyperplanes $w^T_{yr} x + b_r = 0$.

$$\text{minimize} \quad \frac{1}{2} \sum_{r=1}^{K} ||w_r||^2 + \frac{C}{N} \sum_{i,r} \xi_{i,r}$$ \hspace{1cm} (53)

$$\text{s.t.} \quad w^T_{y_i} x^i + b_{y_i} \geq w^T_{r} x^i + b_r + 1 - \xi_{i,r} \text{ for all } i = 1 : N, r \neq y_i$$  \hspace{1cm} (54)

$$\xi_{i,r} \geq 0$$ \hspace{1cm} (55)

#### 6.2 One class SVM

This SVM finds the “support regions” of the data, by separating the data from the origin by a hyperplane. It’s mostly used with the Gaussian kernel, that projects the data on the unit sphere. The formulation below is identical to the $\nu$-SVM where all points have label 1.

$$\text{minimize} \quad \frac{1}{2} ||w||^2 - \nu \rho + \frac{1}{N} \sum \xi_i$$ \hspace{1cm} (56)

$$\text{s.t.} \quad w^T x^i + b \geq \rho - \xi_i$$  \hspace{1cm} (57)

$$\xi_i \geq 0$$ \hspace{1cm} (58)

$$\rho \geq 0$$ \hspace{1cm} (59)
6.3 SV Regression

The idea is to construct a “tolerance interval” of $\pm \epsilon$ around the regressor $f$ and to penalize data points for being outside this tolerance margin. In words, we try to construct the smoothest function that goes within $\epsilon$ of the data points.

\[
\begin{align*}
\text{minimize} \quad & \frac{1}{2} \|w\|^2 + C \sum_i (\xi_i^+ + \xi_i^-) \\
\text{s.t.} \quad & \epsilon + \xi_i^+ \geq w^T x^i + b - y^i \geq -\epsilon - \xi_i^- \\
& \xi_i^\pm \geq 0 \\
& \rho \geq 0
\end{align*}
\]

(60)

(61)

(62)

(63)

The above problem is a linear regression, but with the kernel trick we obtain a kernel regressor of the form $f(x) = \sum_i (\alpha_i^- - \alpha_i^+) K(x^i, x) + b$