67577 – Intro. to Machine Learning **Fall semester, 2003/4**

Lecture 4: Support Vector Machines and Kernel Functions I

Lecturer: Amnon Shashua *Scribe: Amnon Shashua* [1]

Consider the 2-class hyperplane separation problem introduced in the context of the Perceptron algorithm in the previous lecture. We are given a training set of instances $\mathbf{x}_i \in R^n$, $i = 1, ..., l$, and class labels $y_i = \pm 1$ (i.e., the training set is made up of "positive" and "negative" examples). We wish to find a hyperplane direction $\mathbf{w} \in R^n$ and an offset scalar b such that $\mathbf{w} \cdot \mathbf{x}_i - b > 0$ for positive examples and $\mathbf{w} \cdot \mathbf{x}_i - b < 0$ for negative examples — which together means that the margins $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) > 0$ are positive.

Assuming that such a hyperplane exists, clearly it is not unique. We therefore need to introduce another constraint so that we could find the most "sensible" solution among all (infinitley many) possible hyperplanes which separate the training data. Another issue is that the framework is very limited in the sense that for most real-world classification problems it is somewhat unlikely that there would exist a linear separating function to begin with. We therefore need to find a way to extend the framework to include non-linear decision boundaries at a reasonable cost. These two issues will be the focus of this lecture.

Regarding the first issue, since there is more than one separating hyperplane (assuming the training data is linearly separable) then the question we need to ask ourselves is among all those solutions which of them has the best "generalization" properties? In other words, our goal in constructing a learning machine is not necessarily to do very well (or perfect) on the training data, because the training data is merely a sample of the instance space, and not necessarily a "representative" sample — it is simply a sample. Therefore, doing well on the sample (the training data) does not necessarily guarantee (or even imply) that we will do well on the entire instance space (i.e., on instances we haven't seen — the test data). The goal of constructing a learning machine is to maximize the performance on the test data (the instances we haven't seen), which in turn means that we wish to generalize "good" classification performance on the training set onto the entire instance space.

The formal definition of "generalizing" and the resulting bounds we can hope to achieve will be studied in details in class 6 so for now we will simply "hand waive" it by arguing that a decision boundary which lies close to some of the training instances is less likely to generalize well because the learning machine will be susceptible to small perturbations of those instance vectors. For example, you have seen in the previous class that the number of mistakes made by the perceptron algorithm is inversely proportional to the margin defined by the separating hyperplane, i.e., the minimal distance from any instance to the separating hyperplane. Therefore, by means of introspection we may conclude that the most sensible decision surface is one that is farthest away from the training data. In other words, if we define the distance between the boundary of the two sets (positive and negative) as the "margin induced by the hyperplane", then we would like to find a separating hyperplane which *maximizes the margin* between the positive and negative instances of the training set.

Another issue which we would like to address is the learning strategy when the training set is *not linearly separable*. In the previous lecture, in the context of Perceptron on-line learning, we

[1]

discussed the error bound as a function of the number of margin errors and their magnitude given an arbitrary hyperplane. The Perceptron algorithm is not guaranteed to minimize the error bound over all hyperplanes, however the one we will discuss today does.

4.1 Large Margin Classifier as a Quadratic Linear Programming

We would first like to set up the linear separating hyperplane as an optimization problem which is both consistent with the training data and maximizes the margin induce by the separating hyperplane over all possible consistent hyperplanes.

Formally speaking, the distance between a point $\mathbf{x}$ and the hyperplane is defined by

$$\frac{\mid \mathbf{w}\cdot\mathbf{x}-b \mid}{\sqrt{\mathbf{w}\cdot\mathbf{w}}}.$$

Since we are allowed to scale the parameters $\mathbf{w}, b$ at will (note that if $\mathbf{w}\cdot\mathbf{x}-b>0$ so is $(\lambda\mathbf{w})\cdot\mathbf{x}-(\lambda b)>0$ for all $\lambda>0$) we can set the distance between the boundary points to the hyperplane to be $1/\sqrt{\mathbf{w}\cdot\mathbf{w}}$ by scaling $\mathbf{w}, b$ such the point(s) with smallest margin (closest to the hyperplane) will be normalized: $\mid \mathbf{w}\cdot\mathbf{x}-b \mid = 1$, therefore the margin is simply $2/\sqrt{\mathbf{w}\cdot\mathbf{w}}$ (see Fig. 4.1). Note that $\text{argmax}_{\mathbf{w}} 2/\sqrt{\mathbf{w}\cdot\mathbf{w}}$ is equivalent to $\text{argmax}_{\mathbf{w}} 2/(\mathbf{w}\cdot\mathbf{w})$ which in turn is equivalent to $\text{argmin}_{\mathbf{w}} \frac{1}{2}\mathbf{w}\cdot\mathbf{w}$. Since all positive points and negative points should be farther away from the boundary points we also have the separability constraints $\mathbf{w}\cdot\mathbf{x}-b\geq 1$ when $\mathbf{x}$ is a positive instance and $\mathbf{w}\cdot\mathbf{x}-b\leq -1$ when $\mathbf{x}$ is a negative instance. Both separability constraints can be combined: $y(\mathbf{w}\cdot\mathbf{x}-b)\geq 1$. Taken together, we have defined the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\mathbf{w}\cdot\mathbf{w} && (4.1)\\
& \textit{subject to} \\
& y_i(\mathbf{w}\cdot\mathbf{x}_i-b)-1\geq 0 \qquad i=1,...,l && (4.2)
\end{aligned}$$

This type of optimization problem has a quadratic criteria function and linear inequalities and is known in the literature as a *Quadratic Linear Programming* (QP) type of problem.

This particular QP, however, requires that the training data are linearly separable — a condition which may be unrealistic. We can relax this condition by introducing the concept of a "soft margin" in which the separability holds approximately with some error:

$$\begin{aligned}
\min_{\mathbf{w},b,\epsilon_i} \quad & \frac{1}{2}\mathbf{w}\cdot\mathbf{w}+\nu\sum_{i=1}^{l}\epsilon_i && (4.3)\\
& \textit{subject to} \\
& y_i(\mathbf{w}\cdot\mathbf{x}_i-b)\geq 1-\epsilon_i \qquad i=1,...,l \\
& \epsilon_i\geq 0
\end{aligned}$$

Where ν is some pre-defined weighting factor. The (non-negative) variables ϵ_i allow data points to be *miss-classified* thereby creating an approximate separation. Specifically, if $\mathbf{x}_i$ is a positive instance ($y_i=1$) then the "soft" constraint becomes:

$$\mathbf{w}\cdot\mathbf{x}_i-b\geq 1-\epsilon_i,$$

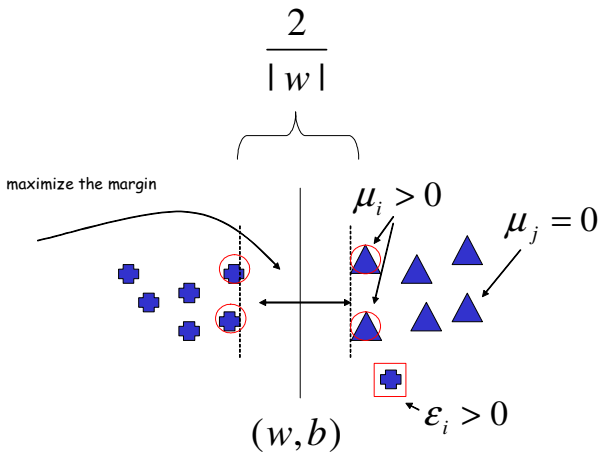

Figure 4.1: Separating hyperplane $\mathbf{w}, b$ with maximal margin. The boundary points are associated with non-vanishing Lagrange multipliers $\mu_i > 0$ and margin errors are associated with $\epsilon_i > 0$ where the criteria function encourages a small number of margin errors.

where if $\epsilon_i = 0$ we are back to the original constraint where $\mathbf{x}_i$ is either a boundary point or laying further away in the half space assigned to positive instances. When $\epsilon_i > 0$ the point $\mathbf{x}_i$ can reside inside the margin or even in the half space assigned to negative instances. Likewise, if $\mathbf{x}_i$ is a negative instance ($y_i = -1$) then the soft constraint becomes:

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 + \epsilon_i.$$

The criterion function penalizes (the L_1-norm) for non-vanishing ϵ_i thus the overall system will seek a solution with few as possible "margin errors" (see Fig. 4.1). Note that the slack variables ϵ_i are the same as the residuals d_i defined in the previous lecture in the context of Perceptron. There we defined D to hold the sum of squares of the residuals (an L_2 norm), whereas here only the sum of the residuals is being considered (and minimized) — i.e. an L_1 norm. Typically, when possible, an L_1 norm is preferable as the L_2 norm overly weighs high magnitude outliers which in some cases can dominate the energy function.

So far we have described the problem formulation which *when solved* would provide a solution with "sensible" generalization properties. The area of non-linear programming in general and QLP in particular is rich with techniques and tools for solving these kind of problems starting from classical techniques like "active set methods" to modern techniques of "interior point methods" upto more specialized QLP solvers with names like "semi-definite programming" (SDP) and "second order cone programming" (SOCP). We could have stopped here and refer you to one of the available solvers (commercially or open-source), but the reason to continue further is due to the fact that by "dualizing" (see next) the QLP one obtains another key property of the "large margin" approach which introduces non-linear decision boundaries with very little cost. In the next section we will take a brief tour on the basic principles associated with constrained optimization, the Karush-Kuhn-Tucker (KKT) theorem and the dual form.

4.2 Primer on Constrained Optimization

4.2.1 Equality Constraints and Lagrange Multipliers

Consider first the general optimization with *equality* constraints which gives rise to the notion of *Lagrange multipliers*.

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ & \textit{subject to} \\ & \mathbf{h}(\mathbf{x}) = 0 \end{aligned} \tag{4.4}$$

where $f : R^n \to R$ and $\mathbf{h} : R^n \to R^k$ where $\mathbf{h}$ is a vector function $(h_1, ..., h_k)$ each from R^n to R. We want to derive a necessary and sufficient constraint for a point $\mathbf{x}_o$ to be a local minimum subject to the k equality constraints $\mathbf{h}(\mathbf{x}) = 0$. Assume that $\mathbf{x}_o$ is a *regular* point, meaning that the gradient vectors $\nabla h_j(\mathbf{x})$ are linearly independent. Note that $\nabla \mathbf{h}(\mathbf{x}_o)$ is a $k \times n$ matrix and the null space of this matrix:

$$null(\nabla \mathbf{h}(\mathbf{x}_o)) = \{\mathbf{y} : \nabla \mathbf{h}(\mathbf{x}_o)\mathbf{y} = 0\}$$

defines the *tangent* plane at the point $\mathbf{x}_o$. We have the following fundamental theorem:

$$\nabla f(\mathbf{x}_o) \perp null(\nabla \mathbf{h}(\mathbf{x}_o))$$

in other words, all vectors $\mathbf{y}$ spanning the tangent plane at the point $\mathbf{x}_o$ are also perpendicular to the gradient of f at $\mathbf{x}_o$.

The sketch of the proof is as follows. Let $\mathbf{x}(t)$, $-a \leq t < a$, be a smooth curve on the surface $\mathbf{h}(\mathbf{x}) = 0$, i.e., $\mathbf{h}(\mathbf{x}(t)) = 0$. Let $\mathbf{x}_o = \mathbf{x}(0)$ and $\mathbf{y} = \frac{d}{dt}\mathbf{x}(0)$ the tangent to the curve at $\mathbf{x}_o$. From the definition of tangency, the vector $\mathbf{y}$ lives in $null(\nabla \mathbf{h}(\mathbf{x}_o))$, i.e., $\mathbf{y} \cdot \nabla h_j(\mathbf{x}(0)) = 0$, $j = 1, ..., k$. Since $\mathbf{x}_o = \mathbf{x}(0)$ is a local extremum of $f(\mathbf{x})$, then

$$0 = \frac{d}{dt} f(\mathbf{x}(t))|_{t=0} = \sum \frac{\partial f}{\partial x_i} \frac{dx_i}{dt}|_{t=0} = \nabla f(\mathbf{x}_o) \cdot \mathbf{y}.$$

As a corollary of this basic theorem, the gradient vector $\nabla f(\mathbf{x}_o) \in span\{\nabla h_1(\mathbf{x}_o), ..., \nabla h_k(\mathbf{x}_o)\}$, i.e.,

$$\nabla f(\mathbf{x}_o) + \sum_{i=1}^{k} \lambda_i \nabla h_i(\mathbf{x}_o) = 0,$$

where the coefficients λ_i are called *Lagrange Multipliers* and the expression:

$$f(\mathbf{x}) + \sum_i \lambda_i h_i(\mathbf{x})$$

is called the *Lagrangian* of the optimization problem (4.4).

4.2.2 Inequality Constraints and KKT conditions

Consider next the general constrained optimization with inequality constraints (called "non-linear programming"):

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ & \textit{subject to} \\ & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0 \end{aligned} \tag{4.5}$$

where $\mathbf{g} : R^n \rightarrow R^s$. We will assume that the optimal solution $\mathbf{x}_o$ is a regular point which has the following meaning: Let J be the set of indices j such that $g_j(\mathbf{x}_o) = 0$, then $\mathbf{x}_o$ is a regular point if the gradient vectors $\nabla h_i(\mathbf{x}_o), \nabla g_j(\mathbf{x}_o)$, $i = 1, ..., k$ and $j \in J$ are linearly independent. A basic result (we will not prove here) is the Karush-Kuhn-Tucker (KKT) theorem:

Let $\mathbf{x}_o$ be a local minimum of the problem and suppose $\mathbf{x}_o$ is a regular point. Then, there exist $\lambda_1, ..., \lambda_k$ and $\mu_1 \geq 0, ..., \mu_s \geq 0$ such that:

$$\nabla f(\mathbf{x}_o) + \sum_{i=1}^{k} \lambda_i \nabla h_i(\mathbf{x}_o) + \sum_{j=1}^{s} \mu_j \nabla g_j(\mathbf{x}_o) = 0, \tag{4.6}$$

$$\sum_{j=1}^{s} \mu_j g_j(\mathbf{x}_o) = 0. \tag{4.7}$$

Note that the condition $\sum \mu_j g_j(\mathbf{x}_o) = 0$ is equivalent to the condition that $\mu_j g_j(\mathbf{x}_o) = 0$ (since $\mu \geq 0$ and $\mathbf{g}(\mathbf{x}_o) \leq 0$ thus there sum cannot vanish unless each term vanishes) which in turn implies: $\mu_j = 0$ when $g_j(\mathbf{x}_o) < 0$ and $\mu_j > 0$ when $g_j(\mathbf{x}_o) = 0$. The expression

$$L(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{i=1}^{k} \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^{s} \mu_j g_j(\mathbf{x})$$

is the *Lagrangian* of the problem (4.5) and the associated condition $\mu_j g_j(\mathbf{x}_o) = 0$ is called the KKT condition.

The remaining concepts we need are the "duality" and the "Lagrangian Dual" problem.

4.2.3 The Langrangian Dual Problem

The optimization problem (4.5) is called the "Primal" problem. The Lagrangian Dual problem is defined as:

$$\max_{\lambda, \mu} \quad \theta(\lambda, \mu) \tag{4.8}$$

$$\textit{subject to}$$

$$\mu \geq 0 \tag{4.9}$$

where

$$\theta(\lambda, \mu) = \min_{\mathbf{x}} \{ f(\mathbf{x}) + \sum_i \lambda_i h_i(\mathbf{x}) + \sum_j \mu_j g_j(\mathbf{x}) \}.$$

Note that $\theta(\lambda, \mu)$ may assume the value $-\infty$ for some values of λ, μ (thus to be rigorous we should have replaced "min" with "inf"). The first basic result is the *weak duality* theorem:

Let $\mathbf{x}$ be a feasible solution to the primal (i.e., $\mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0$) and let (λ, μ) be a feasible solution to the dual problem (i.e., $\mu \geq 0$), then $f(\mathbf{x}) \geq \theta(\lambda, \mu)$

The proof is immediate:

$$\begin{aligned} \theta(\lambda, \mu) &= \min_{\mathbf{y}} \{ f(\mathbf{y}) + \sum_i \lambda_i h_i(\mathbf{y}) + \sum_j \mu_j g_j(\mathbf{y}) \} \\ &\leq f(\mathbf{x}) + \sum_i \lambda_i h_i(\mathbf{x}) + \sum_j \mu_j g_j(\mathbf{x}) \\ &\leq f(\mathbf{x}) \end{aligned}$$

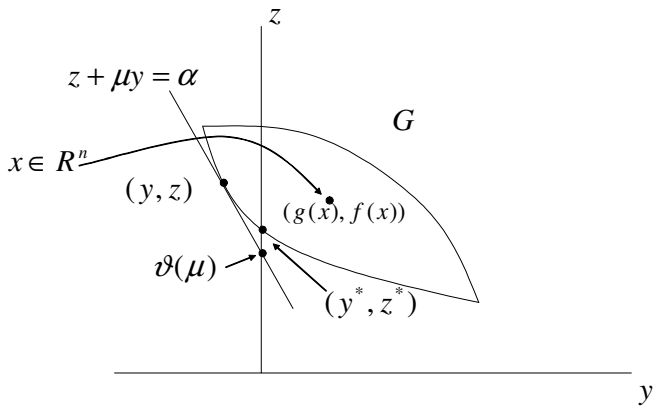

Figure 4.2: Geometric interpreatation of Duality (see text).

where the latter inequality follows from $\mathbf{h}(\mathbf{x}) = 0$ and $\sum_j \mu_j g_j(\mathbf{x}) \leq 0$ because $\mu \geq 0$ and $\mathbf{g}(\mathbf{x}) \leq 0$. As a corollary of this theorem we have:

$$\min_{\mathbf{x}}\{f(\mathbf{x}) : \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0\} \geq \max_{\lambda,\mu}\{\theta(\lambda, \mu) : \mu \geq 0\}. \tag{4.10}$$

The next basic result is the *strong duality* theorem which specifies the conditions for when the inequality in (4.10) becomes equality:

Let $f(), \mathbf{g}()$ be convex functions and let $\mathbf{h}()$ be affine, i.e., $\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ where A is a $k \times n$ matrix, then

$$\min_{\mathbf{x}}\{f(\mathbf{x}) : \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0\} = \max_{\lambda,\mu}\{\theta(\lambda, \mu) : \mu \geq 0\}.$$

The strong duality theorem allows one to solve for the primal problem by first dualizing it and solving for the dual problem instead (we will see exactly how to do it when we return to solving the primal problem (4.3)). When the (convexity) conditions above do not hold we obtain

$$\min_{\mathbf{x}}\{f(\mathbf{x}) : \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0\} > \max_{\lambda,\mu}\{\theta(\lambda, \mu) : \mu \geq 0\}$$

which means that the optimal solution to the dual problem provides only a lower bound to the primal problem — this situation is called a *duality gap*.

We will end this section with a geometric interpretation of duality.

4.2.4 Geometric Interpretation of Duality

For clarity we will consider a primal problem with a single inequality constraint: $\min\{f(\mathbf{x}) : g(\mathbf{x}) \leq 0\}$ where $g : R^n \to R$.

Consider the set $G = \{(y, z) : y = g(\mathbf{x}), z = f(\mathbf{x})\}$ in the (y, z) plane. The set G is the image of R^n under the (g, f) map (see Fig. 4.2). The primal problem is to find a point in G that has a $y \leq 0$ with the smallest z value — this is the point (y^*, z^*) in the figure.

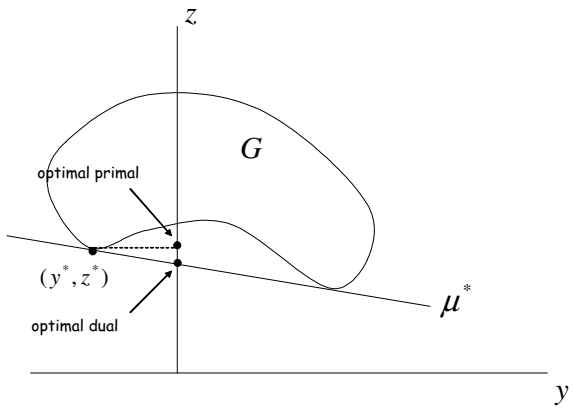

Figure 4.3: An example of duality gap arising from non-convexity (see text).

In this case $\theta(\mu) = \min_{\mathbf{X}}\{f(\mathbf{x}) + \mu g(\mathbf{x})\}$ which is equivalent to minimize $z + \mu y$ over points in G. The equation $z + \mu y = \alpha$ represents a straight line with slope $-\mu$ and intercept (on z axis) α. For a given value μ, to minimize $z + \mu y$ over G we need to move the line $z + \mu y = \alpha$ parallel to itself as far down as possible while it remains in contact with G — in other words G is above the line and touches it. Then, the intercept with the z axis gives $\theta(\mu)$. *The dual problem is therefore equivalent to finding the slope of the supporting hyperplane such that its intercept on the z axis is maximal.*

Consider the non-convex region G in Fig. 4.3 which illustrates a duality gap condition. The optimal primal is the point (y^*, z^*) which is higher than the greatest intercept on the z axis achieved by a line that supports G from below. This is an example of a duality gap caused by the non-convexity of the functions $f(), g()$ (thereby making the set G non-convex).