CS229 Note Lecture 07

The hypothesis represented as

$$h_{w,b}(x) = g(w^T x + b)$$
$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$
$$y \in \{-1, 1\}$$

Functional margin

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

Geometric margin

$$\gamma^{(i)} = y^{(i)}\left(\frac{w^T}{||w||}x + \frac{b}{||w||}\right)$$

We want

$$\gamma = \min_i \gamma^{(i)}$$
$$\hat{\gamma} = \min_i \hat{\gamma}^{(i)}$$

Our learning algorithm would choose parameters w and b so as to maximize the geometric margin. So our goal is to find the separating hyperplane that separates the positive and negative examples with as large a distance as possible between hyperplane and the positive and negative examples.

Changing w and b won't change the value of geometric margin. One interpretation is that if looking at hyperplane, the line which separating positive and negative examples, if we scale w and b, that doesn't change the position of this hyperplane.

That means we can choose whatever scaling for w and b for convenient. e.g. $||w|| = 1$, $|w_1| - 1$, $w^2 + |w_1| = 1$.

Optimal Margin Classifier

Problem 1

$$\begin{aligned}\max_{\gamma,w,b}\ & \gamma \\ \text{s.t. } & y^{(i)}(w^T x^{(i)} + b) \geq \gamma \quad i = 1, \dots, m \\ & \|w\| = 1\end{aligned}$$

But this is not a very nice optimization problem because $\|w\|$ is a nasty, **non-convex constraints**. So we need to change the optimization problem as #2.

Problem 2

$$\begin{aligned}\max_{\hat{\gamma},w,b}\ & \frac{\hat{\gamma}}{\|w\|} \\ \text{s.t. } & y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}\end{aligned}$$

In this problem, the object $\frac{\hat{\gamma}}{\|w\|}$ we want to maximize is a non-convex function in parameter w. So here is the scaling we are going to choose to add

$$\begin{aligned}\hat{\gamma} &= 1 \\ \min_i y^{(i)}(w^T x^{(i)} + b) &= 1\end{aligned}$$

Put this into Problem #2, we get

Problem 3

$$\begin{aligned}\min_{w,b}\ & \|w\|^2 \sim \max \frac{1}{\|w\|} \\ \text{s.t. } & y^{(i)}(w^T x^{(i)} + b) \geq 1\end{aligned}$$

This is actually final formulation of the optimal margin classifier problem

Primal/Dual Optimization

Lagrange Multipliers

Suppose we have a function $f(w)$ to minimize

$$\begin{aligned}\min_w\ & f(w) \\ \text{s.t. } & h_i(w) = 0 \quad i = 1, \dots, l\end{aligned}$$

write those in vector form

$$h(w) = \begin{bmatrix} h_1(w) \\ h_2(w) \\ \vdots \\ h_i(w) \end{bmatrix} = \vec{0}$$

Construct **Lagrangian**

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

$\beta_i h_i(w)$ is called **Lagrange Multipliers**.

The way to solve the optimization problem is

$$\frac{\partial \mathcal{L}}{\partial w} \overset{set}{=} 0, \ \frac{\partial \mathcal{L}}{\partial \beta} \overset{set}{=} 0$$

For some value w^* to be a solution, it is necessary that

$$\exists \beta^* \text{ s.t. } \frac{\partial \mathcal{L}(w^* \beta^*)}{\partial w} = 0, \ \frac{\partial \mathcal{L}(w^* \beta^*)}{\partial \beta} = 0$$

Primal Problem

Suppose we want to minimize $f(w)$

$$\begin{aligned} &\min \ f(w) \\ &\quad \text{s.t. } g_i(w) \leq 0 \quad i = 1, \dots, k \\ &\quad \text{s.t. } h_i(w) = 0 \quad i = 1, \dots, l \end{aligned}$$

using vector notation

$$g(w) \leq \vec{0}, \ h(w) = \vec{0}$$

Lagrangian

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

Define

$$\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta \, : \, \alpha \geq 0} \mathcal{L}(w, \alpha, \beta)$$

Consider

$$p^* = \min_w \max_{\alpha,\beta\,:\,\alpha\geq 0} \mathcal{L}(w,\alpha,\beta) = \min_w \theta_{\mathcal{P}}(w)$$

$\mathcal{P}$ stands for **Primal**.

For $\theta_{\mathcal{P}}(w)$, notice that

$$\begin{aligned} &\text{If } g_i(w) > 0 && \text{then } \theta_{\mathcal{P}}(w) = \infty \\ &\text{If } h_i(w) \neq 0 && \text{then } \theta_{\mathcal{P}}(w) = \infty \\ &\text{Otherwise} && \theta_{\mathcal{P}}(w) = f(w) \end{aligned}$$

Thus

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constrains} \\ \infty & \text{otherwise} \end{cases}$$

So

$$\min_w \theta_{\mathcal{P}}(w) = \text{original problem}$$

Dual Problem

Define

$$\theta_{\mathcal{D}}(w,\beta) = \min_w \mathcal{L}(w,\alpha,\beta)$$

Dual problem is

$$d^* = \max_{\alpha\geq 0,\,\beta} \min_w \mathcal{L}(w,\alpha,\beta) = \max_{\alpha\geq 0,\,\beta} \theta_{\mathcal{D}}(\alpha,\beta)$$

Actually it turns out that

$$d^* \leq p^* \Leftrightarrow \max\min\ (\ldots) \leq \min\max\ (\ldots)$$

e.g

$$\max_{y\in\{0,1\}} \min_{x\in\{0,1\}} 1\{x=y\} \leq \min_{x\in\{0,1\}} \max_{y\in\{0,1\}} 1\{x=y\}$$

Sometimes under certain conditions, primal and dual optimization problem have the same value.

Problem KKT(Karush-Kuhn-Tucker Conditions)

Let f be convex(Hessian $H \geq 0$)

Suppose h_i is **Affine Function**[linear with intercept term, e.g. $h_i(w) = a_i^T w + b$] and g_i's are **Strictly Feasible**[$\exists w$, s.t. $\forall_i g_i(w) < 0$].

Then $\exists w^* \alpha^* \beta^*$ s.t. w^* solves the primal problem, α^*, β^* solve the dual problem and $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$.

Futher

$$\begin{aligned} \frac{\partial}{\partial w}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0 \\ \frac{\partial}{\partial \beta}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0 \\ \alpha_i^* g_i(w) &= 0 \\ g_i(w^*) &\leq 0 \\ \alpha_i^* &\geq 0 \end{aligned}$$

these five equations called **KKT Complementary Condition**.

Considering $\alpha_i^* g_i(w) = 0$, the product of two numbers is equals to 0, means that at least one of these things must be equal to 0. So KKT complementary condition implies that

$$\alpha_i > 0 \Rightarrow g_i(w) = 0$$

Usually,

$$\alpha_i^* \neq 0 \Leftrightarrow g_i(w^*) = 0$$

when this holds true, $g_i(w)$ is a "**active**" constraint

SVM Dual

Lagrange Multipliers: $\alpha_i, \beta_i \ \rightarrow \alpha_i$

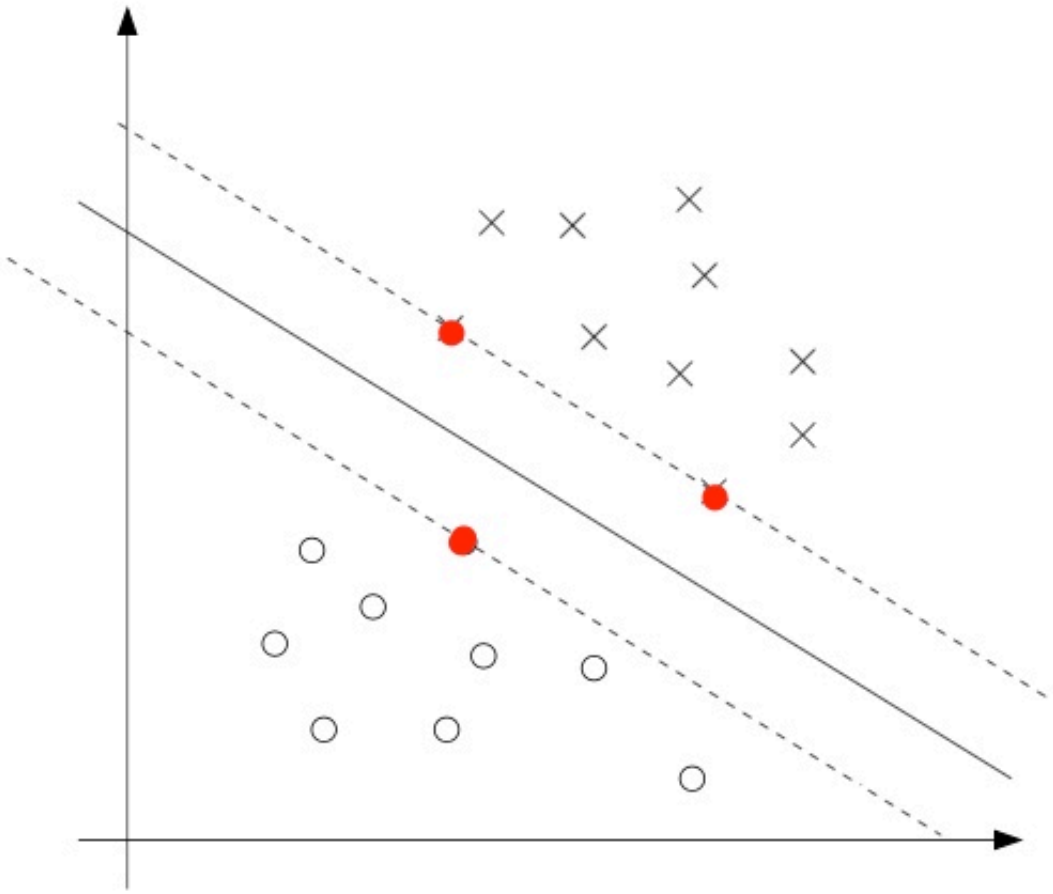
Parameters: $w \rightarrow w, b$

$$\begin{aligned} \min \ & \frac{1}{2}||w||^2 \\ \text{s.t. } & y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad i = 1, \dots, m \end{aligned}$$

So let us just take this constraint, and rewrite it as a constraint

$$g_i(w, b) = y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0$$

We know $\alpha_i > 0 \Rightarrow g_i(w, b) = 0$ (active constraint). It actually turns out $g_i(w, b) = 0 \Leftrightarrow (x^{(i)}, y^{(i)})$ has functional margin 1.



The functional margin between these three points and the hyperplane are 1. These three training examples is called **Support Vector**(usually $\alpha_i \neq 0$), and other training examples which $a_i = 0$ is called non-support vector.

So we have a maximize margin optimization problem, first we go and write down the margin, and because we only have inequality constraints g_i^*, so Lagrangian will be

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i(y^{(i)}(w^T x^{(i)} + b) - 1)$$

And so let's work out what the dual problem is

$$\theta_D(\alpha) = \min_{w,b} \mathcal{L}(w, b, \alpha)$$

take the derivative of w and b

$$\nabla_w \mathcal{L}(w, b, \alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \overset{set}{=} 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\frac{\partial}{\partial b} \mathcal{L}(w, b, \alpha) = - \sum_{i=1}^{m} y^{(i)} x^{(i)} \overset{set}{=} 0$$

Take w's value and plug back in Lagrangian

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i (y^{(i)} (w^T x^{(i)} + b) - 1) \\
&= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} <x^{(i)}, x^{(j)}> - \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} <x^{(i)}, x^{(j)}> + \sum_{i=1}^{m} \alpha_i \\
&= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} <x^{(i)}, x^{(j)}>
\end{aligned}$$

$<>$ denote inner product. We call the this equation $W(\alpha)$

Our dual problem is

$$\begin{aligned}
\max \ & W(\alpha) \\
\text{s.t. } & \alpha_i \geq 0 \\
& \sum_i y_i \alpha_i = 0
\end{aligned}$$

The last constraint is got from $\frac{\partial}{\partial b} \mathcal{L}(w, b, \alpha)$. The interpretation of this constraint is that if $\sum_i y_i \alpha_i \neq 0$ then $\theta_D(\alpha) = -\infty$. If our goal is $\max_{\alpha \geq 0} \theta_D(\alpha)$, then we've gotta choose α for $\sum_i y_i \alpha_i \neq 0$. And when $\sum_i y_i \alpha_i = 0$, then $\theta_D(\alpha) = W(\alpha)$. That's why we endup deciding to maximize $W(\alpha)$ subject to that $\sum_i y_i \alpha_i = 0$.

We'll solve along this dual optimization problem for the parameters α^*.

$$w = \sum_{i=1}^{M} \alpha_i y^{(i)} x^{(i)}$$

once we solve α, we can then go back and quickly derive w. And moreover, once we solve α and w, it's easy to solve for b

$$b = - \frac{\max_{i:y^{(i)}=-1} w^{*T} x^{(i)} + \min_{i:y^{(i)}=1} w^{*T} x^{(i)}}{2}$$

Kernels

We can express the entire algorithm in terms of inner product. The parameters w is the sum of input examples, and we need to make a prediction

$$\begin{aligned} h_{w,b}(x) &= g(w^T x + b) \\ &= g\Big(\sum_{i=1}^{m} \alpha_i y^{(i)} <x^{(i)}, x> + b \Big) \end{aligned}$$

And it turns out that in the source of feature spaces where used to SVM, sometimes our training examples may be very high-dimensional, it may even be the case that the features that you want to use are infinite-dimensional feature vector. But despite it, there'll be an interesting representation that you can use that will allow you compute inner products like these efficiently. And this holds true only for certain feature spaces. But we talk about the idea of kernels we'll see examples where even though you have extremely high-dimensional feature vectors, you may never want to represent $x^{(i)}$, but you will nonetheless be able to computer inner products between different input feature vectors very efficiently.

And this pointed also, the other reason we derive the dual was because $W(\alpha)$ actually are the same property since $W(\alpha)$ also written as inner product.And if we actually look at the dual optimization problem we'll find that we actually do everything we want without ever needing to represent $x^{(i)}$ directly.

One last property of this algorithm is the α_i's are non-0 only for the support vectors that functional margin 1. It means that if you represent w this way, then w when represented as a fairly small fraction of training examples because mostly $\alpha_i = 0$, so when we summing up the sum $\sum_{i=1}^{m} \alpha_i y^{(i)} <x^{(i)}, x>$, we need to compute inner products only if the support vectors, which is usually a small fraction of our training set.