Machine Learning **Spring 2021**

Lecture 5: Lagrange Duality

Lecturer: Liwei Wang *Scribe: Zhuming Shi, Yuhan Song, Le'an wang, Ying Wang, Yushuo Xiao*

5.1 Linear Separability

Last week we consider linear classifiers as

$$\mathcal{F} = \{sgn(\langle \mathbf{w}, \cdot \rangle + b) | \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\} \tag{5.1}$$

then we formulate the optimization of the problem into a linear programming

$$\begin{aligned} \min_{w,b} \quad & \tfrac{1}{2} \| w \|^2 \\ \text{subject to} \quad & y_i(w^\mathsf{T} x_i + b) \geq 1 \end{aligned} \tag{5.2}$$

5.2 Lagrange Duality

We consider an optimization problem:

$$\begin{aligned} (P) \quad \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, 2, ..., m \\ & h_j(x) = 0, j = 1, 2, ..., n \end{aligned} \tag{5.3}$$

where $f(x)$ and $g_i(x)$ are convex functions, and $h_i(x)$ are linear.

Let

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{n} \mu_j h_j(x). \tag{5.4}$$

Then the problem is equivalent to

$$\min_x \max_{\lambda \geq 0, \mu} L(x, \lambda, \mu). \tag{5.5}$$

For fixed x, $L(x, \lambda, \mu)$ is linear with respect to λ and μ, thus a concave function. For fixed $\lambda \geq 0$ and μ, since $f(x)$ and $g_i(x)$ are convex functions and $h_i(x)$ are linear, $L(x, \lambda, \mu)$ is a convex funtion with respect to x. Then by Sion's Minimax Theorem we know that

$$\min_x \max_{\lambda \geq 0, \mu} L(x, \lambda, \mu) = \max_{\lambda \geq 0, \mu} \min_x L(x, \lambda, \mu). \tag{5.6}$$

Then, given λ and μ, we need to find x to minimize $L(x, \lambda, \mu)$. Such x should satisfy

$$\frac{\partial L}{\partial x} = 0, \tag{5.7}$$

from which we can express x as a function of λ and μ:

$$x = \varphi(\lambda, \mu). \tag{5.8}$$

Then the primal problem P becomes another optimization problem D:

$$\begin{aligned} (D) \quad & \max_{\lambda,\mu} && L(\varphi(\lambda,\mu),\lambda,\mu) \\ & \text{subject to} && \lambda_i \geq 0, \qquad i = 1, 2, ..., m \end{aligned} \tag{5.9}$$

We call D the dual of P.

5.3 Apply to Linear Classifier

Let's return to problem 5.2. Now we have the optimization problem D, using 5.7, we have

$$L(w, b, \lambda) := \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \lambda_i \left(1 - y_i \left(w^\top x_i + b\right)\right) \tag{5.10}$$

Applying the Lagrange optimization to 5.10 we get

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{1}{2} \times 2\mathbf{w} - \sum_{i=1}^{n} \lambda_i y_i x_i = 0 \tag{5.11}$$

and

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \lambda_i y_i = 0 \tag{5.12}$$

so we get

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y_i x_i \tag{5.13}$$

and

$$\sum_{i=1}^{n} \lambda_i y_i = 0 \tag{5.14}$$

Now we use these result to 5.10, finally we have

$$\min_{\lambda} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j x_i^\mathsf{T} x_j - \sum_{i=1}^{n} \lambda_i \tag{5.15}$$

$$\text{subject to } \sum_{i=1}^{n} \lambda_i y_i = 0, 0 \leq \lambda_i \leq C \tag{5.16}$$

5.4 KKT Condition

Given general problem:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \dots m \\ & h_j(x) = 0, \quad j = 1, \dots n \end{aligned} \tag{5.3}$$

We have the Lagrangian function:

$$L(x, \lambda, \mu) := f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{n} \mu_j h_j(x) \tag{5.4}$$

and the Lagrange dual function:

$$g(\lambda, \mu) := \min_x L(x, \lambda, \mu) \tag{5.17}$$

The subsequent dual problem is:

$$\begin{aligned} \max_{\lambda,\mu} \quad & g(\lambda, \mu) \\ \text{subject to} \quad & \lambda \geq 0 \end{aligned} \tag{5.18}$$

The Karush-Kuhn-Tucker conditions or KKT conditions for (x^*, λ^*, μ^*) are:

(1) Stationarity: $\bigtriangledown_x L|_{x^*,\lambda^*,\mu^*} = 0$.

(2) Primal feasible: $\forall i, g_i(x^*) \leq 0;\ \forall j, h_j(x^*) = 0$.

(3) Dual feasible: $\forall i, \lambda_i \geq 0$.

(4) Complementary slackness: $\forall i, \lambda_i^* g_i(x^*) = 0$.

Then we have the following theorem:

Theorem 5.1 *x^*, λ^* and μ^* are primal and dual solutions if and only if they satisfy the KKT conditions.*

Proof:
The proof for necessity is trivial, we just prove its sufficiency.
Notice that

$$\begin{aligned} g(\lambda^*, \mu^*) &= f(x^*) + \sum_{i=1}^{m} \lambda_i^* g_i(x^*) + \sum_{j=1}^{n} \mu_j^* h_j(x^*) \\ &= f(x^*) \end{aligned}$$

where the first equality holds from stationarity, and the second holds from complementary slackness.
Then, due to the duality (here $\forall \lambda, \mu, x,\ g(\lambda, \mu) \leq f(x)$) between the primary and dual problem, x^*, λ^* and μ^* are primal and dual solutions. ■

5.5 Support Vector Machine

Assume training data is linear separable, we hope SVM can maximize the margin. It means:

$$\begin{aligned} \min_{w,b} \quad & \tfrac{1}{2} \| w \|^2 \\ \text{subject to} \quad & y_i(w^\mathsf{T} x_i + b) \geq 1 \end{aligned} \tag{5.2}$$

Its Lagrange multiplier is:

$$L(w, b, \lambda) := \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \lambda_i \left(1 - y_i \left(w^\top x_i + b\right)\right) \tag{5.10}$$

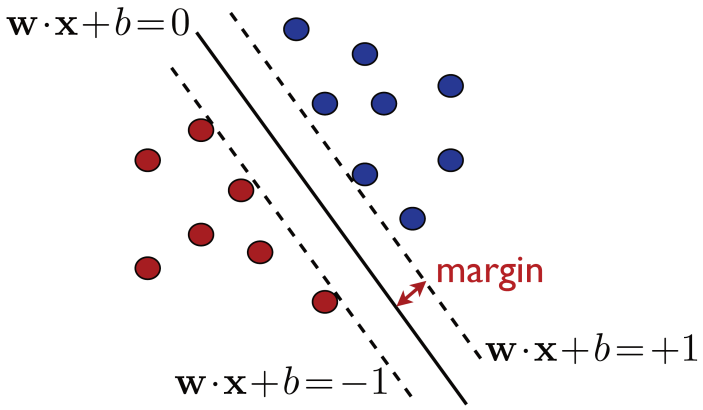
Consider its KKT condition. By stationary and complementary slackness, we have:

$$w^* = \sum_{i=1}^{n} \lambda_i^* y_i x_i \tag{5.19}$$

and

$$\lambda_i^* (y_i \left(w^{*\top} x_i + b^* \right) - 1) = 0 \tag{5.20}$$

So w^* is a linear combination of $y_i x_i$, with weights λ_is. And, it can be observed that only the points closest to the margin can satisfy the condition $y_i \left(w^{*\top} x_i + b^* \right) - 1 = 0$. So only the λ_is corresponding to these points can be positive, while other λ_is are all zero. That is to say, these few points that are closest to the margin decide w^*, and the whole SVM model, and we call these points support vectors (note that SVM is just the abbr. of support vector machine!).
So the dual problem matters. It is not because it helps to calculate the solutions, but that it provides insight into this model.



5.6 Soft Margin SVM

In many scenarios, the training data is linearly inseparable. We can solve this problem by allowing the SVM to fail at certain points. Specifically, we introduce Soft Margin SVM.

$$\min_{w,b,\epsilon} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \epsilon_i (\text{C is a fixed constant}) \tag{5.21}$$

$$\text{subject to } y_i (w^{\mathsf{T}} x_i + b) \geq 1 - \epsilon_i \tag{5.22}$$

The Lagrangian duality is as follows:

$$\min_{\lambda} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j x_i^{\mathsf{T}} x_j - \sum_{i=1}^{n} \lambda_i \tag{5.23}$$

$$\text{subject to } \sum_{i=1}^{n} \lambda_i y_i = 0, 0 \leq \lambda_i \leq C \tag{5.24}$$

The detailed derivation process is given below:

Let $g_i = 1 - \epsilon_i - y_i(w^{\mathsf{T}} x_i + b), \tilde{g}_i = -\epsilon_i$.The constraint condition is $g_i \leq 0, \tilde{g}_i \leq 0$.

$$\min_{w,b,\epsilon} \max_{\lambda,\mu} L(w, b, \epsilon, \lambda, \mu) \tag{5.25}$$

$$\text{s.t.} \lambda_i \geq 0, \mu_i \geq 0 \tag{5.26}$$

$$\text{where } L(w, b, \epsilon; \lambda, \mu) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \lambda_i g_i + \sum_{i=1}^{n} \mu_i \tilde{g}_i \tag{5.27}$$

For fixed λ, μ, find w^*, b^*, ϵ^* to minimize L.

Here we have,

$$\frac{\partial L}{\partial w_i} = w_i - \sum_{j=1}^{n} \lambda_j y_j x_{ji} = 0 \tag{5.28}$$

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{5.29}$$

$$\frac{\partial L}{\partial \epsilon_i} = C - \lambda_i - \mu_i = 0 \tag{5.30}$$

then

$$L(w^*(\lambda, \mu), b^*(\lambda, \mu), \epsilon^*(\lambda.\mu); \lambda, \mu) = \frac{1}{2} \| w^* \|^2 + C \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \lambda_i g_i + \sum_{i=1}^{n} \mu_i \tilde{g}_i \tag{5.31}$$

$$= \frac{1}{2} \| w^* \|^2 + C \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \lambda_i [1 - \epsilon_i - y_i({w^*}^{\mathsf{T}} x_i + b)] - \sum_{i=1}^{n} \mu_i \epsilon_i \tag{5.32}$$

$$= \frac{1}{2} \| \sum_{i=1}^{n} \lambda_i y_i x_i \|^2 + C \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \lambda_i [1 - \epsilon_i - y_i (\sum_{j=1}^{n} \lambda_j y_j x_j^{\mathsf{T}} x_i + b)] - \sum_{i=1}^{n} \mu_i \epsilon_i \tag{5.33}$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^{\mathsf{T}} x_j + C \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n} \lambda_i \epsilon_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^{\mathsf{T}} x_j - \sum_{i=1}^{n} \lambda_i y_i b - \sum_{i=1}^{n} \mu_i \epsilon_i \tag{5.34}$$

$$= - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^{\mathsf{T}} x_j + \sum_{i=1}^{n} (C - \lambda_i - \mu_i) \epsilon_i + \sum_{i=1}^{n} \lambda_i - b \sum_{i=1}^{n} \lambda_i y_i \tag{5.35}$$

$$= - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^{\mathsf{T}} x_j + \sum_{i=1}^{n} \lambda_i (\text{where } \sum_{i=1}^{n} \lambda_i y_i = 0,\ 0 \leq \lambda_i = C - \mu_i \leq C) \tag{5.36}$$

So we get the dual form as 5.23.

5.6.1 Kernel Methods Basics

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using kernel methods, implicitly mapping their inputs into high-dimensional feature spaces. The idea is to define a function $\Phi : \mathcal{X} \to \mathbb{H}$, which maps input x into a Hilbert space[1] $\mathbb{H}$ (called a *feature space*), expecting that $\Phi(x)$ becomes linearly separable in $\mathbb{H}$. However, noticing that x_i, x_j only appear in pairs as an inner product in the dual of the optimization problem of SVM (see equation 5.23), we only need to define a *kernel function*

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle. \tag{5.37}$$

This eliminates the need of explicitly calculating the function $\Phi(x)$, which can be inefficient and sometimes even impossible (if $\dim \mathbb{H} = \infty$).

Kernel methods are often illustrated by the XOR example. The data in Figure 5.1 (a) is clearly not linearly separable. But we can map the data into another space with the function

$$\Phi(x) = (\sqrt{2}x_1x_2, \sqrt{2}x_1). \tag{5.38}$$

The mapped points are shown in Figure 5.1 (b) and become linearly separable.

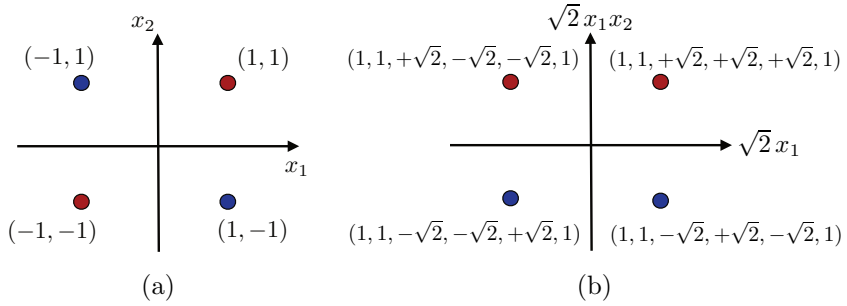


Figure 5.1: Illustration of the XOR classification problem and the use of kernel.

5.7 Boosting

Boosting is a family of learning methods that combines several predictors to create a more accurate one. We specifically focus on AdaBoost (short for *Adaptive Boosting*), which has been shown to be very effective in practice in some scenarios and is based on a rich theoretical analysis.

We first give the algorithm of AdaBoost in Algorithm 1.

The algorithm runs for T rounds. At each round, a new base classifier h_t is selected to minimize training error on S weighted by the distribution D_t. Then the distribution for the next round, D_{t+1}, is calculated. Intuitively, D_{t+1} is defined from D_t by increasing the weight on i if x_i is incorrectly classified, and decreasing it if x_i is correctly classified. After T rounds of boosting, the classifier returned is based on a non-negative linear combination of the base classifiers h_t. More accurate classifiers are assigned a larger weight and less accurate classifiers are assigned a smaller weight.

(*To be continued...*)

[1] A Hilbert space is a vector space equipped with an inner product, and that is complete (all Cauchy sequences are convergent).

Algorithm 1: AdaBoost algorithm

Input: Sample data $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, base learning algorithm $\mathcal{A}$.
Result: A classifier $F(x)$.
Initialize: $D_1(i) = 1/n,\ i \in [n]$.
for $t = 1, 2, \cdots, T$ **do**
  Use $\mathcal{A}$ to learn a base classifier $h_t(x)$ on D_t.
  $\varepsilon_t \leftarrow \sum_{i=1}^{n} D_t(i) I[y_i \neq h_t(x_i)]$
  $\gamma_t \leftarrow 1 - 2\varepsilon_t$
  $\alpha_t \leftarrow \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$
  $Z_t \leftarrow 2[\varepsilon_t(1-\varepsilon_t)]^{\frac{1}{2}}$
  $D_{t+1}(i) = \frac{D_t(i) \cdot \exp(-y_i \alpha_t h_t(x_i))}{Z_t}$
end
Output: $F(x) = \operatorname{sgn}\left[\sum_{t=1}^{T} \alpha_t h_t(x)\right]$.

References

[1] Rostamizadeh.A Mohri.M and TALWALKAR.A. Foundations of Machine Learning. MIT Press, 2012.

[2] https://www.cs.cmu.edu/~ggordon/10725-F12/slides/16-kkt.pdf

[3] https://en.wikipedia.org/wiki/Support-vector_machine