EE 381V: Large Scale Optimization **Fall 2014**

Lecture 12 — October 7

Lecturer: Sanghavi *Scribe: Ger Yang & David Liau & Che-Chun Lin*

12.1 Recap

Last time we showed that for any optimization problem

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{Subject to} \quad & f_i(x) \leq 0, \quad i = 1, ..., m \\ & h_j(x) = 0, \quad j = 1, ..., p \end{aligned} \tag{12.1}$$

where the objective and constraint functions are differentiable. If strong duality holds, then any pair of primal and dual optimal points must satisfy the following KKT conditions

$$\begin{aligned} \nabla_x L(x^*, \lambda^*, v^*) &= 0 \\ \nabla_\lambda L(x^*, \lambda^*, v^*) &\preceq 0 \\ \nabla_v L(x^*, \lambda^*, v^*) &= 0 \\ \lambda_i^* &\geq 0, \quad i = 1, ..., m \\ \lambda_i^* f_i(v^*) &= 0, \quad i = 1, ..., m \end{aligned} \tag{12.2}$$

where x^* is primal optimal, (λ^*, v^*) is dual optimal , and $L(x, \lambda, v) = f_o(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} v_j h_j(x)$. Furthermore, if the primal problem is convex, the KKT conditions are also sufficient for the points to be primal and dual optimal. Today we want to show the application of KKT conditions.

12.2 Application 1: Waterfilling

In communication theory, there is a well-known algorithm called *Waterfilling*. [1] Consider there are K independent additive Gaussian channels with a common power constraint $\bar{P}$. The objective is to allocate the power to each channel such that the capacity is maximized.

For each channel k, we can write down the received signal y_k as the sum of the transmitted signal x_k and the Gaussian noise n_k, that is,

$$y_k = x_k + n_k. \tag{12.3}$$

[1] For more information of Waterfilling and the derivation of channel capacity, please refer to Elements of information theory, by Cover, Thomas M., and Joy A. Thomas. John Wiley & Sons, 2012.

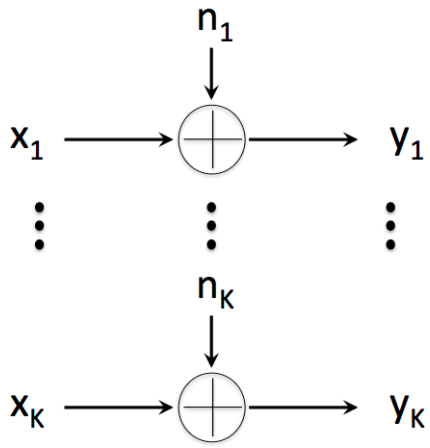

Figure 12.1. Parallel Gaussian Channels

Since all channels share a common power constraint, it can be written as

$$\sum_{k=1}^{K} E[x_k^2] \leq \bar{P}. \tag{12.4}$$

According to Shannon's information theory, the channel capacity for each individual channel k is $C_k = log(1 + \frac{P_k}{N_k})$, where $P_k = E[x_k^2]$ and $N_k = Var(n_k)$ are the power of the input signal and the Gaussian noise of channel k, respectively. Now we consider a parallel channel as in Figure **??**. The total capacity C of k independent channels will be

$$C = \sum_{l=1}^{K} log(1 + \frac{P_l}{N_l}) \tag{12.5}$$

In order to find the power allotment that maximizes the total capacity subject to the power constraint, we can write it as a standard convex optimization problem:

$$\begin{aligned} \min_{P} \quad & -\sum_{l=1}^{K} log(1 + \frac{P_l}{N_l}) \\ \text{Subject to} \quad & \sum_{l=1}^{K} P_i \leq \bar{P} \\ & P_i \geq 0 \end{aligned} \tag{12.6}$$

To find its optimal solution, we first calculate its Lagrangian,

$$L(P, \lambda) = -\sum_{l=1}^{K} log(1 + \frac{P_l}{N_l}) + \lambda(\bar{P} - \sum_{l=1}^{K} P_l). \tag{12.7}$$

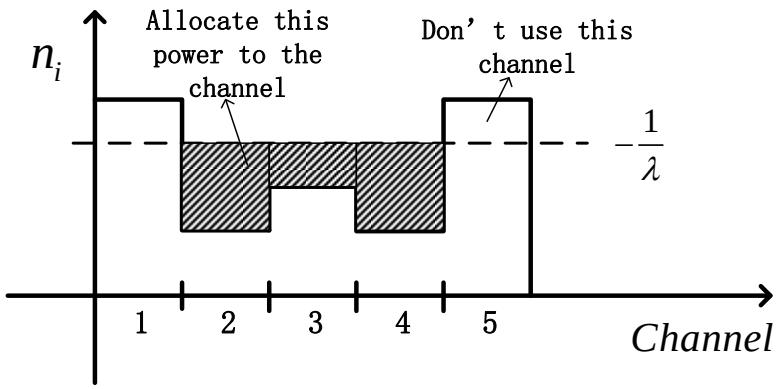

Figure 12.2. Illustration of Waterfilling algorithm

Then, we obtain the KKT conditions

$$\nabla_{P_l} = -\frac{1}{1+\frac{P_l}{N_l}}(\frac{1}{N_l}) - \lambda \begin{cases} \leq 0 & \text{if } P_l = 0 \\ = 0 & \text{if } P_l > 0 \end{cases} \tag{12.8}$$

$$\lambda \leq 0 \tag{12.9}$$

and gives us the following

$$N_l + P_l = \frac{-1}{\lambda}, \forall l \text{ s.t. } P_l > 0 \tag{12.10}$$

rearrange the inequality we have the solution

$$P_l = (\frac{-1}{\lambda} - N_l)^+ \tag{12.11}$$

where λ is chosen such that $\sum_{l=1}^{K} P_l = \bar{P}$, and $(x)^+ = \max\{0, x\}$ denotes the positive part of x.

The solution is illustrated in Figure **??**. For the channel of noise power greater than $\frac{-1}{\lambda}$, we will not allocate any power to the signal using that channel. On the other hand, for the rest of the channels, we *fill* the channel with P_l to let the total power $N_l + P_l$ of that channel equals to $\frac{-1}{\lambda}$. As a result, this gives the name of *Waterfilling* of such a problem.

12.3 Application 2: Classification

Classification is a fundamental problem in machine learning and statistics. Suppose we are given a finite set of points $\{x_1 \dots x_N\}$ in $\mathbb{R}^n$, and for each x_i there is a label $y_i \in \{-1, +1\}$. Our goal is to find out the labeling rule, i.e. a mapping function $f : \mathbb{R}^n \to \mathbb{R}$ that separates the points with different labels. More specifically, the separating function f satisfies the following inequalities

$$f(x_i) > 0, \forall x_i \in \mathcal{X}_+, i = 1, \cdots, N \qquad f(x_i) < 0, \forall x_i \in \mathcal{X}_-, i = 1, \cdots, N \tag{12.12}$$

where $\mathcal{X}_+ \triangleq \{x_i : y_i > 0\}$ and $\mathcal{X}_- \triangleq \{x_i : y_i < 0\}$ are the sets of points labelled with -1 and 1 respectively. If indeed these inequalities hold, we say that the function f *separates*, *classifies*, or *discriminates* the set of training points. We sometimes also consider weak separation, in which the weak versions of the inequalities hold, i.e., $y_i \cdot f(x_i) \geq 0$ for all i.

12.3.1 Linear Separation

For simplicity, we consider the case that the set of points is *linear-separable*, that is, inequality **??** holds for all $x \in \{x_1 \cdots x_N\}$. In linear separation, we seek an affine function $f(x) = a^T x - b$ that separates the two point sets, as shown in Figure **??**. We have the following formulation: Find a hyperplane, parameterized by a normal vector w and offset b, such that

$$y_i \cdot (\langle w, x_i \rangle + b) > 0, \quad i = 1, \cdots, N. \tag{12.13}$$

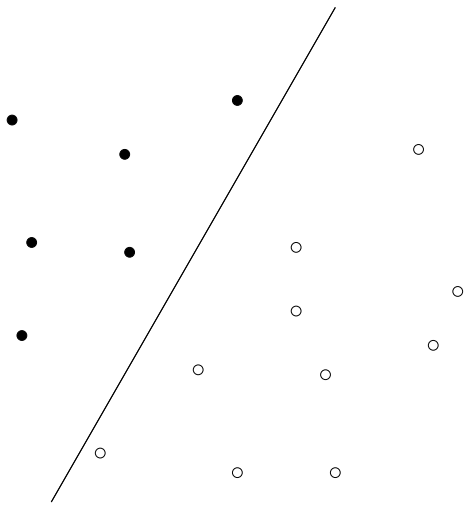

Figure 12.3. The points $\mathcal{X}_+ \triangleq \{x_i : y_i > 0\}$ and $\mathcal{X}_- \triangleq \{x_i : y_i < 0\}$ are shown as open and filled circles, respectively. These two sets are classified by an affine function f, whose 0-level set appears to be a separate line.

The geometric meaning is straightforward, as illustrated in Figure **??**, we seek a hyperplane that separates these two sets of points. Since the inequalities **??** are homogeneous in w and b, they are feasible if and only if the following non-strict linear inequalities hold:

$$\langle w, x \rangle + b \geq 1, \ \forall x_i \in \mathcal{X}_+, \qquad \langle w, x \rangle + b \leq -1, \ \forall x_i \in \mathcal{X}_-. \tag{12.14}$$

Take Figure **??** for example. We can assume the offset on b is zero, while targeting on examining w. Then we could have the following inequalities:

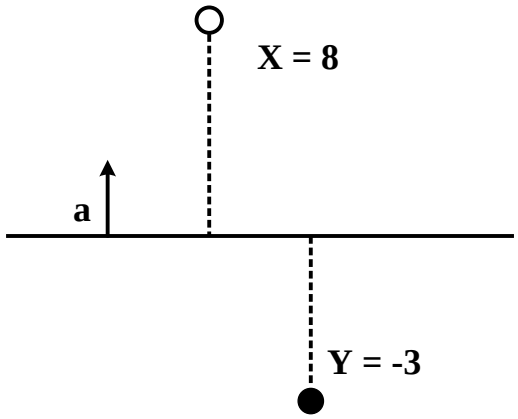

Figure 12.4. Example of solving inequalities (**??**).

$$\begin{aligned} & \langle w, x\rangle \geq 1, \quad \langle w, y\rangle \leq -1 \\ \Rightarrow \quad & 8w \geq 1, \quad -3w \leq -1 \\ \Rightarrow \quad & w \geq 1/8, \quad w \geq 1/3 \\ \Rightarrow \quad & w \geq 1/3 \end{aligned} \tag{12.15}$$

From inequalities **??**, it is desirable to maximize $1/\|w\|$. Indeed, we show below that this is proportional to the margin. Maximizing $1/\|w\|$ is the same as minimizing $\|w\|$, and therefore, we can formulate the entire linear classification problem into a convex optimization as follows:

$$\begin{aligned} \min_{w,b} \quad & \|w\|_2 \\ \text{Subject to} \quad & \langle w, x^{(+)}\rangle + b \geq 1, \quad \forall x^{(+)} \in \mathcal{X}_+ \\ & \langle w, x^{(-)}\rangle + b \leq -1, \quad \forall x^{(-)} \in \mathcal{X}_- \end{aligned} \tag{12.16}$$

12.3.2 Maximum Gap of Separation

We make the above assertion about margin maximization more precise here. Consider the setting depicted in Figure **??**. There are infinitely many hyperplanes that separate the labelled points. One sensible choice is the one that *maximizes the margin*, that is, that maximizes the distance to both the closest positively and negatively labelled point. Rearrange the constraints in eq. **??**, we have

$$\langle w, x^{(+)} - x^{(-)}\rangle \geq 2 \tag{12.17}$$

Then we have

$$\|x^{(+)} - x^{(-)}\| \geq \frac{2}{\|w\|}, \quad \forall x^{(+)} \in \mathcal{X}_+,\ x^{(-)} \in \mathcal{X}_- \tag{12.18}$$

Hence we can write the max-margin problem as

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|_2 \\ \text{Subject to} \quad & y_i\langle w, x_i\rangle + b \geq 1, \quad \forall i = 1 \cdots N \end{aligned} \tag{12.19}$$

12.3.3 Non-Linear Separation

We have examined the linear separation technique. But what happens if the data does not happen to be linearly separable? Figure **??** illustrates an example where linear separation is not applicable. As the figure suggests, there seems to exist another straightforward classification rule. We can easily find that if we lift the data into a higher dimensional space with a non-linear transformation (for example, with the distance from the origin), then there will exist a hyperplane that separates the data well, i.e. the data becomes linear-separable. This idea is depicted in Figure **??**. This non-linear mapping, denoted by $\Phi(x)$ in Figure **??**, is called the *kernel map*.

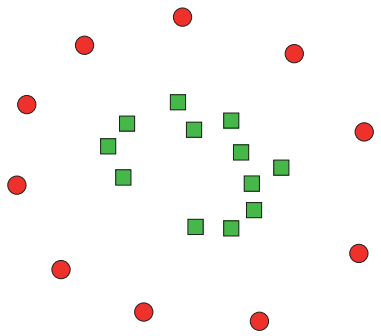

Figure 12.5. An example of non-linear data distribution in a two-dimensional (2D) plane.

In many practical applications, one often uses a mapping to a much higher dimensional – possibly even infinite dimensional – space. The immediate question is to understand how this impacts the optimization problem. Finding a hyperplane in n dimensions is an optimization problem in $(n+1)$ variables (one must find w and b, or only w if we impose a normalization and set $b = 1$). Then, if we lift to $N \gg n$ dimensions, does computation time increase accordingly?

It turns out that this is not the case, as long as we use a special kind of non-linear mapping that has a special property. The property required is that it is "easy" to compute inner products in the high-dimensional space. That is, there is a kernel function K such that

$$\langle \Phi(x_i), \Phi(x_j)\rangle = K(x_i, x_j), \tag{12.20}$$

and $K(x_i, x_j)$ is easy to compute, where "easy" means that it is about as hard as computing $\langle x_i, x_j\rangle$ in the first place.

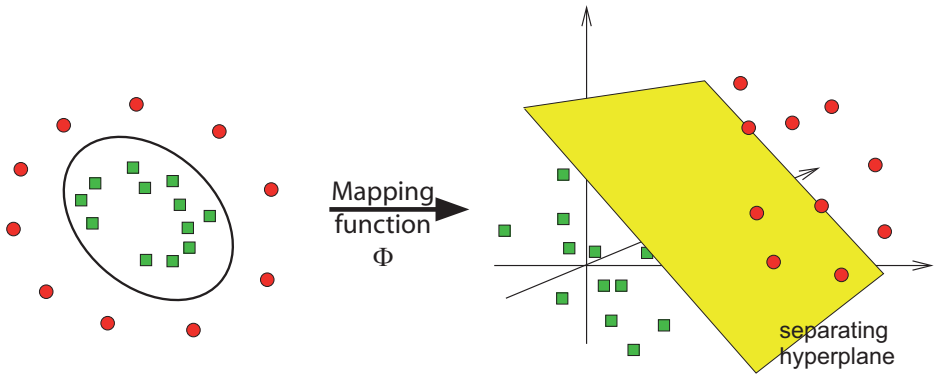

Figure 12.6. Mapping of the data to higher-dimensional space where a linear separation is possible.

To see why this property of Φ is so important, and why when it holds it implies that we can quickly solve problems and find separating hyperplanes in much higher dimension without working any harder, we need to use the dual. Recall that the primal problem is as follows:

$$\begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 \\ \text{Subject to} \quad & \langle w, x\rangle + b \geq 1, \quad \forall x \in \mathcal{X}_+ \\ & \langle w, x\rangle + b \leq -1, \quad \forall x \in \mathcal{X}_-. \end{aligned} \tag{12.21}$$

By writing the constraints above more compactly as

$$y_i(\langle w, x_i\rangle + b) - 1 \geq 0, \quad \forall i = 1, \cdots, N \tag{12.22}$$

we can write the Lagrangian of this as:

$$L(w, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i y_i(\langle w, x_i\rangle + b) + \sum_{i=1}^{n} \lambda_i. \tag{12.23}$$

Note that the objective function and the constraints are convex, and in particular, Slater's condition is satisfied. This means that strong duality holds. Recall that as a fundamental consequence of strong duality, we know that we have:

$$q(\lambda^*) = \max_{\lambda} \min_{w,b} L(w, b, \lambda) = \min_{w,b} \max_{\lambda} L(w, b, \lambda) = p(w^*, b^*), \tag{12.24}$$

where $p(\cdot, \cdot)$ denotes the primal function, and $q(\cdot)$ denotes the dual function. In particular, this implies, by convexity, that at (w^*, b^*, λ^*), the first order necessary conditions are satisfied:

$$\nabla_{w,b} L(w^*, b^*, \lambda^*) = 0, \quad \nabla_{\lambda} L(w^*, b^*, \lambda^*) = 0. \tag{12.25}$$

These give the conditions:

$$w = \sum_i \lambda_i y_i x_i, \tag{12.26}$$

and

$$\sum_i \lambda_i y_i = 0. \tag{12.27}$$

Substituting back in, we have derived the dual optimization problem:

$$\max\ q(\lambda) = \max\ \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle. \tag{12.28}$$

Notice the dependence on dimension: it is only there through the inner product, $\langle x_i, x_j \rangle$. In particular, if we were to use a nonlinear mapping Φ to map the data $\{x_i\}$ to a higher dimensional space, the resulting dual would simply be (check!)

$$\max\ q(\lambda) = \max\ \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \langle \Phi x_i, \Phi x_j \rangle. \tag{12.29}$$

If it happens that we have found an appropriate mapping function Φ that has the special property that $\langle \Phi x_i, \Phi x_j \rangle = K(x_i, x_i)$, for some easily computable function K, then we see that solving this optimization problem can be done essentially with no additional effort than the original one. This is the so-called *kernel trick* and it has been used in regression as well as in classification. One of the popular kernel mapping function is:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|} \tag{12.30}$$

This kernel represents a mapping Φ to infinite dimensions. Without duality, solving for the linear separating hyperplane in infinite dimensions would be computationally hopeless. This method is called *Support vector machine*, and more about this is going to be discussed in the next lecture.

12.4 Congestion Control

Congestion control is an important topic in many network resource allocation problems. The allocation problem can be described as a constrained maximization of some utility function, which can be solved by convex optimization.

As shown in Figure **??**, the flow network is a directed graph $G = \{V, L, C\}$, where V is the node set, L is the edge link, and C is the capacity value associated with each edge. We also have a set of source-sink pairs $source_s \to sink_s$ that can send a maximum amount of flow value x_s (sometimes called source rate). Also, each source-sink pair is associated with an utility U_s, which can be smooth, increasing, and concave. Each link l also has a capacity

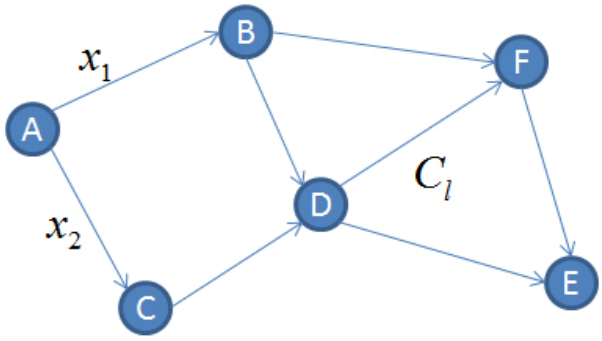

Figure 12.7. Example of a flow network.

C_l. Then given a routing matrix $R_{ls} \in \{1, 0\}$, where the $R_{ls} = 1$ means the flow from source s use the edge link l or $R_{ls} = 0$ otherwise. The objective is try to maximize the flow utility in the network. Thus, the network utility maximization problem can be formulated as follows:

$$\begin{aligned} \max_{x} \quad & \sum_{s} U_s(x_s) \\ \text{Subject to} \quad & RX \preceq C \\ & X \succeq 0 \end{aligned} \tag{12.31}$$

The problem given in (**??**) satisfies the condition of strong duality, i.e., the optimal primal value is equal to the optimal dual value. The Lagrangian dual form can be written as follows:

$$\begin{aligned} L(x, \lambda) &= \sum_{s} U_s(x_s) + \sum_{l} \lambda_l (c_l - \sum_{s} R_{ls} x_s) \\ &= \sum_{s} [U_s(x_s) - (\sum_{l} R_{ls} \lambda_l) x_s] + \sum_{l} c_l \lambda_l \end{aligned} \tag{12.32}$$

Then the dual problem is:

$$\begin{aligned} \min_{\lambda} \quad & \sum_{s} \max_{x_s \geq 0} (U_s(x_s) - \lambda_s x_s) + \sum_{l} c_l \lambda_l \\ \text{Subject to} \quad & \lambda \succeq 0 \end{aligned} \tag{12.33}$$

Additivity of total utility and flow constraints lead to decomposition into individual source terms, a form of dual decomposition. This decomposition is called horizontal across users in network. In practice, suppose each user is associated with a source rate x_s. Then each user can keep his utility private and only needs to know λ_s to solve the problem $\max_{x_s \geq 0}(U_s(x_s) - \lambda_s x_s)$.