Learning and Optimization in Multiagent Decision-Making Systems
Notes 2: Duality, Subgradient Method, and Single Agent Dynamic Optimization
Instructor: Rasoul Etesami

6 Duality

We define the dual function $q:\mathbb{R}^r\to\mathbb{R}$ associated with the primal problem $\min\{f(x):g(x)\le 0, x\in X\}$ as $q(\mu)=\inf\{L(x,\mu):x\in X\}$, where $L(x,\lambda)=f(x)+\mu' g(x)$. The function $q(\mu)$ can be interpreted as the highest point of interception with the vertical axis over all hyperplanes with normal vector $(\mu,1)$, which contains the set $S=\{(g(x),f(x)):x\in X\}$ in their positive halfspace. The dual problem is given by $q^*=\max\{q(\mu):\mu\ge 0\}$ and corresponds to finding the maximum point of interception over all hyperplanes with normal vector $(\mu,1)$ where $\mu\ge 0$. In fact, as we have seen before, the domain $D=\{\mu:q(\mu)>-\infty\}$ of the dual function q is convex and q is concave over D.

Theorem 26 (Weak Duality). *For any primal feasible solution x and any dual feasible solution $\mu\ge 0$, we have $q(\mu)\le f(x)$. In particular, we have $q^*\le f^*$.*

Proof: $\forall x\in X$ with $g(x)\le 0$ and $\forall \mu\ge 0$, we have $q(\mu)=\inf_{z\in X}L(z,\mu)\le f(x)+\mu' g(x)\le f(x)$. □

Remark 8. *If there is no duality gap (i.e., $f^*=q^*$), the set of geometric multipliers is equal to the set of optimal dual solutions. Moreover, if there is a duality gap, the set of geometric multipliers is empty. The reason is that a vector $\mu^*\ge 0$ is a geometric multiplier $\iff f^*=q(\mu^*)\le q^*\le f^*$, which by weak duality holds if and only if $f^*=q^*$ and μ^* is a dual optimal solution.*

6.1 Strong Duality

Theorem 27. *Consider* $\min\{f(x):g(x)\le 0, x\in X\}$ *with $f^*<\infty$. Assume X is a convex set and f,g are convex over X. If there exists a vector $\bar{x}\in X$ that satisfies the Slater condition $g(\bar{x})<0$, then there is no duality gap and there exist at least one geometric multiplier.*

Proof: Consider the epigraph of the joint constraint-objective set $S=\{(g(x),f(x)):\ x\in X\}$ defined by $A=\{(z,w): x\in X, \text{such that } g(x)\le z,\ f(x)\le w\}$. Using the convexity of f and g, one can easily show that A is a convex set. Moreover, $(0,f^*)$ is not an interior point of A, otherwise there exists $\epsilon>0$ such that $(0,f^*-\epsilon)\in A$, contradicting the definition of f^* as the optimal primal value. Therefore there exists a separating hyperplane with normal $(\mu,\beta)\neq 0$ passing through $(0,f^*)$ and containing A on one side of it, i.e., $\beta f^*\le \beta w+\mu' z\ \forall (z,w)\in A$. This relation implies $\beta\ge 0,\mu\ge 0$, because for each $(z,w)\in A$ and any $\epsilon>0$, we also have $(z,\epsilon+w)\in A$ and $(z+\epsilon\mathbf{1},w)\in A$. Furthermore, $\beta>0$, otherwise, if $\beta=0$, we would have $0\le \mu' z\ \forall (z,w)\in A$. Since $(g(\bar{x}),f(\bar{x}))\in A \implies 0\le \mu' g(\bar{x})$, which cannot hold unless $\mu=0$, contradicting the fact that $(\mu,\beta)\neq 0$.

Now by normalizing β, we may assume $\beta=1$. Thus, since $(g(x),f(x))\in A$, we get $f^*\le f(x)+\mu' g(x)\ \forall x\in X$. Taking infimum over $x\in X$ and because $\mu\ge 0$, we get $f^*\le \inf_{x\in X}\{f(x)+\mu' g(x)\}=q(\mu)\le q^*$. This, in view of weak duality, implies $f^*=q^*$ and that $\mu\ge 0$ is a geometric multiplier. □

6.2 Linear Constraints and Duality

Consider the problem

$$\min f(\mathbf{x}) \tag{5}$$
$$\text{s.t. } e_i'\mathbf{x} \le d_i, \quad i = 1, \ldots, m \tag{6}$$
$$a_j'\mathbf{x} \le b_j, \quad j = 1, \ldots, r, \quad \mathbf{x} \in X \tag{7}$$

where e_i, a_j and d_i, b_j are given vectors and scalars, respectively, $f : \mathbb{R}^n \to \mathbb{R}$ is a convex continuously differentiable function, and X is a polyhedral set. We refer to 5, 6, 7 as the primal problem. The dual function associated with this program is given by

$\mathrm{q}(\lambda, \mu) = \inf_{\mathbf{x} \in X} L(\mathbf{x}, \lambda, \mu)$, where $L(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i (e_i'\mathbf{x} - d_i) + \sum_{j=1}^r \mu_j (a_j'\mathbf{x} - b_j)$ is the Lagrangian function. The dual problem is defined by

$$\max q(\lambda, \mu) \tag{8}$$
$$\text{s.t. } \lambda \in \mathbb{R}^m, \mu \ge 0 \tag{9}$$

In fact, it is easy to see that if X is bounded, the dual function always takes finite values. However, in general $q(\lambda, \mu)$ can be $-\infty$.

Theorem 28. *1. If 5, 6, 7 has an optimal solution, then 8, 9, also has an optimal solution and the corresponding optimal values are equal.*

2. In order for $\mathbf{x}^*$ to be an optimal primal solution and (λ^*, μ^*) to be an optimal dual solution, it is necessary and sufficient that $\mathbf{x}^*$ is primal feasible, $\mu_j^* \ge 0$, $\mu_j^* = 0 \ \ \forall j \notin A(x^*)$, and

$$\mathbf{x}^* \in \arg\min_{\mathbf{x} \in X} L(\mathbf{x}, \lambda^*, \mu^*).$$

Example (Dual of linear program): Consider the problem

$$\begin{aligned} &\min \ c'\mathbf{x} \\ &\text{s.t. } e_i'\mathbf{x} \le d_i, i = 1, \ldots, m \quad (P) \\ &\mathbf{x} \ge 0 \end{aligned}$$

$q(\lambda) = \inf_{x \ge 0} \{\sum_{j=1}^n (c_j - \sum_{i=1}^m \lambda_i e_{ij}) x_j + \sum_{i=1}^m \lambda_i d_i\}$. Now if $c_j - \sum_{i=1}^m \lambda_i e_{ij} \ge 0 \ \ \forall j$, the infinum is attained for $x = 0$, and we have $q(\lambda) = \sum_{i=1}^m \lambda_i d_i$. On the other hand, if $c_j - \sum_{i=1}^m \lambda_i \varepsilon_{ij} < 0$ for some j, we can make the expression inside braces arbitrarily small, so in this case $q(\lambda) = -\infty$. Therefore, the dual problem is given by

$$\max \sum_{i=1}^m \lambda_i d_i : \quad \sum_{i=1}^m \lambda_i e_{ij} \le c_j \ \ \forall j$$

Now if x^* and (λ^*, μ^*) are optimal primal and dual solutions, using part b of the previous proposition,

$$x \in \arg\min_{x \ge 0} \{\sum_{j=1}^n (c_j - \sum_{i=1}^m \lambda_i^* e_{ij}) x_j + \sum_{i=1}^m \lambda_i^* d_i\} \implies c \ge \sum_{i=1}^m \lambda_i^* e_i, \quad x^* \ge 0$$

$$x_j^* > 0 \implies c_J = \sum_{i=1}^{m} \lambda_i^* e_{ij} \quad \forall j$$

$$\sum_{i=1}^{m} \lambda_i^* e_{ij} < c_j \implies x_J^* = 0 \quad \forall j$$

Example (Dual of a quadratic program)
Consider the quadratic program:

$$\begin{aligned} &\min \frac{1}{2}\mathbf{x}'Q\mathbf{x} + c'\mathbf{x} \\ &\text{st. } A\mathbf{x} \leq b \\ &Q > 0, A \in \mathbb{R}^{rxn} \end{aligned}$$

The dual function is given by

$$q(\mu) = \inf_{\mathbf{x} \in \mathbb{R}^n} \left\{ \frac{1}{2}\mathbf{x}'Q\mathbf{x} + c'\mathbf{x} + \mu'(b + AQ^{-1}c) \right\}$$

The infinum is attained for $x = -Q^{-1}(c + A'\mu)$ and hence,

$$q(\mu) = -\frac{1}{2}\mu' AQ^{-1}A'\mu - \mu'(b + AQ^{-1}c) - \frac{1}{2}c'Q^{-1}c$$

After dropping the constant term $\frac{1}{2}c'Q^{-1}c$ and changing the minus sign to convert max to min, we get

$$\begin{aligned} &\min \frac{1}{2}\mu' P\mu + t'\mu \\ &\text{s.t. } \mu \geq 0 \end{aligned}$$

where $P = AQ^{-1}A'$, $t = b + AQ^{-1}c$.

7 The Subgradient Method

Here we consider algorithms for solving the dual problem $\max\{q(\mu) : \mu \in M\}$, where $M = \{\mu \geq 0 : q(\mu) > -\infty\}$, which are based on the use of *subgradients*.

Definition 29. *Given a concave function $q(\mu)$, a vector g is called a subgradient of q at μ if we have $q(\bar{\mu}) \leq q(\mu) + (\bar{\mu} - \mu)'g$, $\forall \bar{\mu} \in \mathbb{R}^r$. Note that if $q(\cdot)$ is differentiable, one can take $g = \nabla q(\mu)$, which is the unique subgradient at μ.*

Lemma 30. *For a given μ, suppose that $x_\mu \in \arg\min_{x \in X} L(x, \mu) = \arg\min_{x \in X}\{f(x) + \mu' g(x)\}$. Then, $g(x_\mu)$ serves as a subgradient of the dual function $q(\cdot)$ at μ.*

Proof: For all $\bar{\mu} \in \mathbb{R}^n$, we can write

$$\begin{aligned} q(\bar{\mu}) &= \inf_{x \in X}\{f(x) + \bar{\mu}' g(x)\} \leq f(x_\mu) + \bar{\mu}' g(x_\mu) \\ &= f(x_\mu) + \mu' g(x_\mu) + (\bar{\mu} - \mu)' g(x_\mu) \\ &= q(\mu) + (\bar{\mu} - \mu)' g(x_\mu), \end{aligned} \tag{10}$$

which shows that $g(x_\mu)$ is a subgradient of $q(\cdot)$ at the point μ. □

Let us assume for every $\mu \in M$, we can calculate some vector $x_\mu \in \arg\min_{x\in X} L(x,\mu)$, yielding a subgradient $g(x_\mu)$ of $q(\cdot)$ at μ. The subgradient method generates a sequence of dual feasible points according to the iteration

$$\mu^{k+1} = [\mu^k + s^k g^k]^+,$$

where $g^k := g(x_{\mu^k})$ is the subgradient, $[\cdot]^+$ denotes projection on the set M, and $s^k > 0$ is a stepsize. While the iterate looks like a projected gradient method, however, unlike that method, the subgradient method is not always guaranteed to improve the dual objective value, i.e., for some k we might have $q([\mu^k + sq^k]^+) < q(\mu^k)$. However, if μ^k is not optimal, then for every optimal dual solution μ^*, we have $\|\mu^{k+1} - \mu^*\| < \|\mu^k - \mu^*\|$, for sufficiently small stepsizes s^k.

7.1 Convergence Proof for the Subgradient Method

In the remainder of this section, we describe the subgradient method for optimizing a general convex function $f(\cdot)$ over a convex set X, i.e.,

$$\min\{f(x) : \; x \in X\},$$

which can be easily adapted for the special case of maximizing the dual function. Note that since we consider minimizing a convex function, in the definition of the subgradient (Definition 29), the direction of the inequality must be flipped. As we mentioned earlier, subgradient method is not a descent method, it is common to keep track of the best point found so far, i.e., the one with smallest function value. At each step, we set

$$f^k_{\text{best}} = \min\left\{f^{k-1}_{\text{best}}, f(x^k)\right\}.$$

Theorem 31. *Consider* $\min\{f(x) : \; x \in X\}$, *where* $f(\cdot)$ *is a convex function and* X *is a closed convex set. Let* x^* *be any minimizer of* f *and* $f^* = f(x^*)$. *Consider the subgradient method*

$$x^{k+1} = [x^k - \alpha_k g^k]^+_X,$$

where $[\cdot]^+_X$ *is the projection on the convex set* X, g^k *is a subgradient of* $f(\cdot)$ *at the point* x^k, *and* $\{\alpha_k\}$ *is a nonnegative stepsize sequence. Assume that the norm of the subgradients is bounded, i.e., there is a* G *such that* $\|g^k\|_2 \le G$ *for all* k.[2] *Then, we have*

$$0 \le f^k_{best} - f^* \le \frac{\|x^1 - x^*\|^2 + G^2 \sum_{i=1}^k \alpha_i^2}{\sum_{i=1}^k \alpha_i}.$$

Proof: Recall that for the standard gradient descent method, the convergence proof is based on the function value decreasing at each step. In the subgradient method, the key quantity is not the function value (which often increases); it is the Euclidean distance to the optimal set. Let x^* be an arbitrary optimal point. Using the nonexpansive property of the projection, we can write

$$\begin{aligned}\|x^{k+1} - x^*\|_2^2 &= \left\|[x^k - \alpha_k g^k]^+_X - [x^*]^+_X\right\|^2 \le \|x^k - \alpha_k g^k - x^*\|^2 \\ &= \|x^k - x^*\|_2^2 - 2\alpha_k (g^k)'(x^k - x^*) + \alpha_k^2 \|g^k\|_2^2 \\ &\le \|x^k - x^*\|_2^2 - 2\alpha_k (f(x^k) - f^*) + \alpha_k^2 \|g^k\|_2^2,\end{aligned}$$

[2] This will be the case if, for example, f satisfies the Lipschitz condition, i.e., $|f(x) - f(x)| \le G\|x - x\|_2$ for all x, y.

where $f^* = f(x^*)$. The last line follows from the definition of subgradient, which gives

$$f(x^*) \geq f(x^k) + (g^k)'(x^* - x^k).$$

Applying the inequality above recursively, we have

$$\|x^{k+1} - x^*\|_2^2 \leq \|x^1 - x^*\|_2^2 - 2\sum_{i=1}^{k} \alpha_i (f(x^i) - f^*) + \sum_{i=1}^{k} \alpha_i^2 \|g^i\|_2^2.$$

Using $\|x^{k+1} - x^*\|_2^2 \geq 0$, we have

$$2\sum_{i=1}^{k} \alpha_i (f(x^i) - f^*) \leq \|x^1 - x^*\|_2^2 + \sum_{i=1}^{k} \alpha_i^2 \|g^i\|_2^2.$$

Combining this with

$$\sum_{i=1}^{k} \alpha_i (f(x^i) - f^*) \geq \left(\sum_{i=1}^{k} \alpha_i\right) \min_{i=1,\ldots,k} (f(x^i) - f^*),$$

we have the inequality

$$f_{\text{best}}^k - f^* = \min_{i=1,\ldots,k} (f(x^i) - f^*) \leq \frac{1}{2\sum_{i=1}^k \alpha_i} \left(\|x^1 - x^*\|_2^2 + \sum_{i=1}^{k} \alpha_i^2 \|g^i\|_2^2 \right).$$

Finally, using the assumption $\|g^i\|_2 \leq G$, we obtain

$$f_{\text{best}}^k - f^* = \min_{i=1,\ldots,k} (f(x^i) - f^*) \leq \frac{\|x^1 - x^*\|_2^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2\sum_{i=1}^k \alpha_i}.$$

□

Remark 9. *Since in the above theorem x^* is any minimizer of f, we can state that*

$$f_{best}^k - f^* \leq \frac{dist(x^1, X^*)^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2\sum_{i=1}^k \alpha_i},$$

where X^* denotes the optimal set, and $dist(x^1, X^*)$ is the (Euclidean) distance of x^1 to X^*.

From the result of the above theorem, we can read off various convergence results. For instance, if the stepsize sequence satisfies $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, $\sum_{k=1}^{\infty} \alpha_k = \infty$, then we have $f_{\text{best}}^k \to f^*$ as $k \to \infty$. One can see that by choosing $\alpha_k = \frac{1}{k}$, the convergence rate is sublinear of the order $O(\frac{1}{\log k})$. Similarly, by choosing $\alpha_k = \frac{1}{k^{0.5+\beta}}$ for some small $\beta > 0$, the convergence rate is sublinear of the order $O(\frac{1}{k^{0.5-\beta}})$, which is assymptotically $O(\frac{1}{\sqrt{k}})$ as $\beta \to 0$.

8 Dynamic Optimization: Optimal Control Problem

8.1 Optimal Control Problem

Given a dynamical system $\dot{x}(t) = f(x(t), u(t), t)$, $x(t_0) = x_0$, where $\dot{x}(t) = \frac{d}{dt}x(t)$, we want to find a controller $u[t_0, t_1]$ to minimize the objective cost $V(u) = \int_{t_0}^{t_1} l(x(t), u(t), t)dt + m(x(t_1))$, where t_1 is the final time, $l(\cdot)$ is a scalar valued loss function, and $m(\cdot)$ is a function of state, which is referred to as terminal cost. Here, we assume x_0, t_0, t_1, $l(\cdot)$ and $m(\cdot)$ are known and fixed, and $x(t_1)$ is free to choose.

8.2 Hamilton-Jacobi-Bellman (HJB) Equation

Definition: The value function $V^o(x,t)$ is defined to be the optimal value of $V(u)$ over all controls u assuming that the initial time and state are t and x, respectively, i.e.,

$$V^o(x,t) = \min_{u[t,t_1]} \left\{ \int_t^{t_1} l(x(\tau),u(\tau),\tau)d\tau + m(x(t_1)) \right\}.$$

Theorem 32. *The value function satisfies the HJB equation:*

$$-\frac{\partial V^o(x,t)}{\partial t} = \min_u \left\{ l(x,u,t) + \left(\nabla_x V^o(x,t)\right)' f(x,u,t) \right\},$$

for every time t and state x, with the boundary condition $V(x(t_1),t_1) = m(x(t_1))$.[3]

Proof: Let x and t be an arbitrary initial state and initial time, and let $t < t_m < t_1$ be an intermediate time. Assuming that $x(\tau)$, $\tau \in [t,t_1]$ is a solution to state equation with initial condition $x(t) = x$ and control $u[t,t_1]$, we must have:

$$\begin{aligned}
V^o(x,t) &= \min_{u[t,t_1]} \left\{ \int_t^{t_m} l(x(\tau),u(\tau),\tau)d\tau + \int_{t_m}^{t_1} l(x(\tau),u(\tau),\tau)d\tau + m(x(t_1)) \right\} \\
&= \min_{u[t,t_m]} \left\{ \int_t^{t_m} l(x(\tau),u(\tau),\tau)d\tau + \min_{u[t_m,t_1]} \left(\int_{t_m}^{t_1} l(x(\tau),u(\tau),\tau)d\tau + m(x(t_1)) \right) \right\} \\
&= \min_{u[t,t_m]} \left(\int_t^{t_m} l(x(\tau),u(\tau),\tau)d\tau + V^o(x(t_m),t_m) \right)
\end{aligned}$$

Note that the second equality holds by principle of optimality: If the trajectory generated by the optimal control u^o passes through state $x(t_m)$ at time t_m, then the control $u^o[t_m,t_1]$ must be optimal for the system starting at $x(t_m)$ at time t_m. This is because if a better controller exists on the interval $[t_m,t_1]$, we would have chosen it. Now by letting $\Delta t := t_m - t$ approach 0, we can derive a partial differential equation for the value function V^o.

Specifically, let $x(t_m) = x(t+\Delta t) = x(t) + \Delta x = x + \Delta x$, and $u(t) = u$. Assuming that the value function V^o is sufficiently smooth, by using Taylor expansion on the last equality we obtain:

$$\begin{aligned}
&V^o(x,t) = V^o(x(t),t) \\
&= \min_{u[t,t_m]} \left\{ l(x(t),u(t),t)\Delta t + V^o(x(t),t) + \left(\nabla_x V^o(x(t),t)\right)' \Delta x + \frac{\partial V^o(x(t),t)}{\partial t}\Delta t + o(\Delta t) \right\} \\
&= \min_{u[t,t_m]} \left\{ l(x,u,t)\Delta t + V^o(x,t) + \left(\nabla_x V^o(x,t)\right)' \Delta x + \frac{\partial V^o(x,t)}{\partial t}\Delta t + o(\Delta t) \right\}.
\end{aligned}$$

Dividing through by Δt, letting $\Delta t \to 0$, and noting that the ratio $\frac{\Delta x}{\Delta t}$ can be replaced by the derivative $\dot{x}(t)$, we obtain:

$$0 = \min_u \left\{ l(x,u,t) + \left(\nabla_x V^o(x,t)\right)' \dot{x}(t) \right\} + \frac{\partial V^o(x,t)}{\partial t},$$

where we note that as $\Delta t \to 0$, $u(t,t+\Delta t)$ becomes $u := u(t)$, which is a single variable (rather than a function), and $\frac{\partial V^o}{\partial t}(x,t)$ comes out of minimum as it is independent of u. Since $\dot{x}(t) = f(x(t),u(t),t) = f(x,u,t)$ we obtain the desired HJB equation. □

[3]Note that HJB equation is a pointwise equation meaning that u in the minimization refers to a variable (rather than a function).

Definition 33. *The term inside the minimum of HJB equation is known as Hamiltonian, and is denoted by*

$$H(x,p,u,t) := l(x,u,t) + p' f(x,u,t)$$

where $p = \nabla_x V^o(x,t) = \left(\frac{\partial V^o}{\partial x_1}(x,t), \dots, \frac{\partial V^o}{\partial x_n}(x,t)\right)'$.

Theorem 34. *Suppose V^o is a function with continuous partial derivatives. Then V^o is the optimal value function with the optimal control input u^o and corresponding optimal state trajectory x^o, if and only if V^o satisfies the HJB equation subject to $V^o(x(t_1),t_1) = m(x(t_1))$ and $u^o(t) = \arg\min_u H(x^o(t), \nabla_x V^o(x^o(t),t), u, t)\ \forall t \in [t, t_1]$.*

Remark 10. *Since the optimal controller is obtained by*

$$u^o(t) = \arg\min_u H(x^o(t), \nabla_x V^o(x^o(t),t), u, t) := \bar{u}(x^o(t), t),$$

this means that the optimal control can be written in a state feedback form $u^o(t) = \bar{u}(x^o(t),t)$. Of course, to find this feedback form, we first need to find value function V^o, which can be hard.

Example: $V(x,u) = \int_{t_0}^{t_1} (x^4(t) + u^2(t))dt$, $\dot{x} = f(x,u,t) = u$, $l(x,u,t) = (x^4 + u^2)$, $m(\cdot) = 0$. $H(x,p,u,t) = u^2 + x^4 + pu$. The HJB equation becomes:

$$-\frac{\partial V^o}{\partial t}(x,t) = \min_u \left\{\nabla_x V^o(x,t) \cdot u + u^2 + x^4\right\}.$$

Minimizing the right hand side with respect to u we obtain $u^o = -\frac{1}{2}\frac{\partial V^o}{\partial x}(x^o,t)$ which is of the form of state feedback. Therefore, the optimal state has the form $\dot{x}^o(t) = -\frac{1}{2}\frac{\partial V^o}{\partial x}(x^o,t)$, i.e., the control forces the state to move in the direction in which the "cost to go" V^o decreases. In particular, the HJB equation becomes $-\frac{\partial V}{\partial t} = -\frac{1}{4}\left(\frac{\partial V^o}{\partial x}(x,t)\right)^2 + x^4$, with boundary condition $V^o(x,t_1) = 0$. This is as far as we can go analytically and one must solve this equation numerically.

One important special case is when the system is time invariant, i.e., $\dot{x} = f(x,u)$ and the time horizon is infinite, i.e., $t_1 = \infty$ (and $m = 0$) and cost function does not depend on time, i.e., $l = l(x,u)$. In that case, the optimal value function is independent of time and we have: $\frac{\partial V^o(x)}{\partial t} = 0$ and the above HJB equation simplifies to

$$\min_u \left\{ l(x,u) + \left(\nabla_x V^o(x)\right)' f(x,u)\right\} = 0.$$

Thus, in this example we have $x^4 = \frac{1}{4}(\nabla_x V^o(x))^2 \implies \nabla_x V^o(x) = 2x^2 \implies V^o(x) = \frac{2}{3}x^3 \implies u^o(t) = -\left(x^o(t)\right)^2$.

Example (LQR Problem): A special case of the optimal control problem is the Linear Quadratic Regulator (LQR) problem given by: $\dot{x} = A(t)x + B(t)u$, $x(t_0) = x_0$, $V(u) = \int_{t_0}^{t_1} \left(x^T Q(t) x + u^T R(t) u\right) dt + x^T(t_1) M x(t_1)$, where $M(t), Q(t)$, and $R(t)$ are positive semi-definite matrix-valued functions of time, and $A(t), B(t), Q(t), R(t)$ are piecewise continuous functions in t.

Theorem 35. *For the LQR problem with $R(t) > 0\ \forall t$, the optimal control is given by $u^o(t) = -R(t)^{-1}B(t)^T P(t) x^o(t)$, where $P(t)$ is the solution to the Riccati Differential Equation (RDE): $-\dot{P} = Q + PA + A^T P - PBR^{-1}B^T P$. In particular, the optimal value function has the quadratic form $V^o(x,t) = x^T P(t) x$.*

9 Dynamic Optimization: Online Convex Optimization

In online convex optimization (OCO), an online player iteratively makes decisions. At the time of each decision the outcome associated with it is unknown to the player. After committing to a decision, the decision maker suffers a loss. These losses are unknown to the decision maker beforehand. The losses can be adversarially chosen, and even depend on the action taken by the player.

9.1 Model and Assumptions

Assumptions: The losses determined by an adversary should be bounded, and the decision set must be bounded and "structured" (e.g. a bounded convex set).

Model: Decision set $X \subset \mathbb{R}^n$, where X is a bounded convex set.

At iteration $t = 1, 2, ...$, the online player chooses $x_t \in X$. After the player has committed to this choice, a convex cost function $f_t \in F : X \to \mathbb{R}$ is revealed. Here, F is the bounded family of cost functions available to the adversary. The cost incurred by the online player is $f_t(x_t)$, the value of the cost function for the choice x_t. Let T denote the total number of game iterations.

Let $\{x_t\}_{t=1}^T$ be the decisions by an OCO algorithm where x_t can depend on the game history up to time t, i.e., $x_t : \{f_1, ..., f_{t-1}, x_1, ..., x_{t-1}\} \to X$. We define the regret of an algorithm after T iterations as

$$R(T) = \sup_{\{f_1,...,f_T\} \in F} \left\{ \sum_{t=1}^{T} f_t(x_t) - \min_{x \in X} \sum_{t=1}^{T} f_t(x) \right\}$$

Examples:

Prediction with Expert Advice:

The decision maker has to choose among the advice of n given experts. After making her choice, a loss between 0 and 1 is incurred. This scenario is repeated iteratively, and at each iteration, the costs of the various experts are arbitrary. The goal of the decision maker is to do as well as the best expert in hindsight.

$$X = \Delta_n = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \ \forall i\}$$

Let $c_t^{(i)}$ be the cost of the i-th expert at iteration t, and let $c_t = \left(c_t^{(1)}, \ldots, c_t^{(n)}\right)^T$ at time t. Then the expected cost of choosing an expert according to distribution x is given by $f_t(x) = c_t' x$.

Online Shortest Paths:

The decision maker is given a directed graph $G = (V, E)$ and a source-sink pair $u, v \in V$. At each iteration t, the decision maker chooses a path $p_t \in P_{u,v}$, where $P_{u,v}$ is the set of all $u - v$ paths in the graph. The adversary independently chooses weights (lengths) on the edges of the graph denoted by $\omega_t : E \to \mathbb{R}$, which can be represented as a vector $\omega_t \in \mathbb{R}^m$, where $|E| = m$. The decision maker suffers and observes a loss, which is the weighted length of the chosen path

$\sum_{e \in p_t} \omega_t(e)$.

$$X = \left\{\text{set of all distributions over } P_{u,v} \text{ (flows)}\right\}$$
$$= \left\{x \in [0,1]^m : \sum_{e:=i,j)\in E} x(e) = \sum_{e=(j,i)\in E} x(e)\ \forall j \in V \backslash \{u,v\}, \sum_{e=(u,j),j\in E} x(e) = \sum_{e:(j,v),j\in E} x(e) = 1\right\}$$
.

The expected cost of a given flow $x \in X$ (distribution over paths) is then a linear function, given by $f_t(x) = \omega_t' x$.

9.2 Online Gradient Descent Algorithm:

Input: Convex set X, T, $x_1 \in X$, stepsize $\{\eta_t\}$
For $t = 1, 2, \ldots, T$
play x_t and observe cost $f_t(\cdot)$
update $x_{t+1} = [x_t - \eta_t \nabla f_t(x_t)]_X^+$ (Euclidean projection onto the set X)

Theorem 36. *Online gradient descent with step sizes $\eta_t = \Theta\left(\frac{1}{\sqrt{t}}\right)$ guarantees the following*

$$R(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x\in X} \sum_{t=1}^{T} f_t(x) \leq \frac{3GD\sqrt{T}}{2}$$

where $D = \max_{x,y\in X} \|x - y\|$ is the diameter of the convex set X and G is an upper bound for the gradient of the convex functions $f_t \in F$.

Proof: Let $x^* \in \arg\min_{x\in X} \sum_{t=1}^{T} f_t(x)$. Because each f_t is a convex function, we have

$$f_t(x_t) - f_t(x^*) \leq \nabla f_t(x_t)'(x_t - x^*), \quad \forall t \tag{11}$$

Now we can write:

$$\begin{aligned}
\|x_{t+1} - x^*\|^2 &= \|\,[x_t - \eta_t \nabla f_t(x_t)]_X^+ - [x^*]_X^+\|^2 \\
&\leq \|x_t - \eta_t \nabla f_t(x_t) - x^*\|^2 \\
&= \|x_t - x^*\|^2 + \eta_t^2 \|\nabla f_t(x_t)\|^2 - 2\eta_t \nabla f_t(x_t)'(x_t - x^*)
\end{aligned}$$

$$\Rightarrow 2\nabla f_t(x_t)'(x_t - x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + \eta_t G^2 \tag{12}$$

Summing 11 and 12 from $t = 1$ to T, and setting $\eta_t = \frac{D}{G\sqrt{T}}$ (assuming $\frac{1}{\eta_0} = 0$), we get:

$$\begin{aligned}
2\sum_{t=1}^{T}[f_t(x_t) - f_t(x^*)] &\leq \sum_{t=1}^{T} \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \|x_t - x^*\|^2 + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq D^2 \sum_{t=1}^{T} \left(\frac{1}{\eta_T} - \frac{1}{\eta_0}\right) + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq D^2 \frac{1}{\eta_T} + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq 3DG\sqrt{T}
\end{aligned}$$

$$\Rightarrow R(T) = \sum_{t=1}^{T}[f_t(x_t) - f_t(x^*)] \leq \frac{3DG\sqrt{T}}{2}.$$

In fact, one can show that the above upper bound is tight up to a constant factor. □

Theorem 37. *Any algorithm for online convex optimization incurs $\Omega(DG\sqrt{T})$ regret in the worst case.*

Theorem 38. *For the class of α-strongly convex loss functions, online gradient descent with stepsize $\eta_t = \frac{1}{\alpha t}$ achieves a regret bound of*

$$R(T) \leq \frac{G^2}{2\alpha}(1 + \log T).$$

Proof: The proof is identical to the previous theorem once we replace 11 with the stronger inequality

$$f_t(x_t) - f_t(x^*) \leq \nabla f_t(x_t)'(x_t - x^*) - \frac{\alpha}{2}\|x_t - x^*\|^2$$

due to strong convexity assumption and the choice of stepsize $\eta_t = \frac{1}{\alpha t}$.

10 Follow the Regularized Leader (FTRL)

As we saw in the previous proof, the regret of convex cost functions can be bounded by a linear function via inequality $f_t(x_t) - f_t(x^*) \leq \nabla f_t(x_t)'(x_t - x^*)$. Thus the overall regret can be bounded by

$$\sum_t [f_t(x_t) - f_t(x^*)] \leq \sum_t \nabla f_t'(x_t)(x_t - x^*).$$

10.1 FTRL Algorithm

Input: $\eta > 0$, regularization function $\mathscr{R}$, and a bounded closed convex set X.

Let $x_1 = \arg\min_{x \in X} \mathscr{R}(x)$

for $t = 1, \ldots, T$

play x_t and observe cost $f_t(x_t)$

update $x_{t+1} = \arg\min_{x \in X} \left\{\eta \left(\sum_{\tau=1}^{t} \nabla f_\tau(x_\tau)\right)' x + \mathscr{R}(x)\right\}$

Theorem 39. *FTRL algorithm attains for every $y \in X$ the following bound on the regret.*

$$R(T) \leq 2\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2 + \frac{\mathscr{R}(y) - \mathscr{R}(x_1)}{\eta}.$$

In particular, if $\|\nabla f_t(x_t)\| \leq G, \quad \forall t$, by optimizing over the parameter $\eta = \frac{1}{\sqrt{T}}$, we have $R(T) = O(\sqrt{T})$.

Exercise: What happens if we take $\mathscr{R}(x) = \sum_{i=1}^{n} x_i \log x_i$? Can you find a closed form for the iterates of FTRL?