IE 539: Convex Optimization **KAIST, Fall 2022**
Lecture #14: OCO II and the connection to stochastic optimization October 18, 2022
Lecturer: Dabeen Lee

1 Outline

In this lecture, we study

- Online binary classification,
- Stochastic optimization through the lens of OCO,
- Stochastic gradient descent.

2 Applications of online convex optimization

2.1 Online binary classification

Let us consider a mathematical model to establish an email spam filtering system. Recall that we used the support vector machine (SVM) for binary classification. Just to remind you what it was, we find a pair of a coefficient vector w and a right-hand side value b to use the hyperplane $w^\top x = b$ to classify data points. Given a feature vector x, we assign it label $\mathrm{sign}(w^\top x - b)$ where $\mathrm{sign}(c)$ has value 1 if $c \geq 0$ and value -1 if $c < 0$. When a training set of multiple data is available, we can find such a classifier (w, b) by solving a convex optimization problem whose objective is to minimize the hinge loss.

However, in some scenarios, data points dynamically arrive so that we gradually accumulate the data. In such cases, we may adjust our model over time, and the learning process continues. To be more specific, let us consider the online binary classification problem described as follows. An email is represented by its feature vector $x \in \mathbb{R}^d$ and label $y \in \{-1, 1\}$. The feature vector can encode words and expressions written in it, while the label indicates whether the email is spam or not. Let's say that $y = 1$ indicates spam and $y = -1$ indicates valid. For each time slot t, we repeat the following procedure.

- The spam filtering system prepares a classifier (w_t, b_t) based on the past emails represented by $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \in \mathbb{R}^d \times \{-1, 1\}$.
- New email with feature vector x_t arrives.
- The spam filter predicts that its label is $\mathrm{sign}(w_t^\top x_t - b)$, while the true label of the email is y_t.
- The spam filter incurs a loss of $\max\{0, 1 - y_t(w_t^\top x_t - b)\}$.

After T emails, the cumulative loss is given by

$$\sum_{t=1}^{T} \max\{0, 1 - y_t(w_t^\top x_t - b)\}.$$

Compared to a best classifier, we incur

$$\sum_{t=1}^{T} \max\{0, 1 - y_t(w_t^\top x_t - b)\} - \min_{(w,b)\in\mathbb{R}^d\times\mathbb{R}} \sum_{t=1}^{T} \max\{0, 1 - y_t(w^\top x_t - b)\}$$

more loss. Denoting the loss function at each time t as

$$f_t(w, b) = \max\{0, 1 - y_t(w^\top x_t - b)\},$$

the excess cumulative loss is rewritten as

$$\sum_{t=1}^{T} f_t(w_t, b_t) - \min_{(w,b)\in\mathbb{R}^d\times\mathbb{R}} \sum_{t=1}^{T} f_t(w, b).$$

Therefore, the online binary classification problem is an instance of online convex optimization where the best fixed decision corresponds to the best spam classifier.

2.2 Stochastic optimization

Stochastic optimization (SO) is an optimization problem of the following form.

$$\underset{x\in C}{\text{minimize}} \quad \mathbb{E}_{\xi\sim\mathbb{P}}\left[h(x,\xi)\right]$$

where

- ξ is a random parameter vector whose underlying distribution is given by $\mathbb{P}$,
- $h(x, \xi)$ is convex with respect to x for any fixed ξ,
- C is the feasible set for the decision vector x.

Then

$$f(x) = \mathbb{E}_{\xi\sim\mathbb{P}}\left[h(x,\xi)\right]$$

is convex. For example, for the linear regression problem, we consider

$$h(\beta, (x, y)) = \frac{1}{2}(y - \beta^\top x)^2,$$

and

$$\text{minimize} \quad \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[h(\beta,(x,y))\right] \quad = \quad \text{minimize} \quad \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[\frac{1}{2}(y-\beta^\top x)^2\right]$$

where x is the feature vector, y is the response variable, and (x, y) follows distribution $\mathbb{P}$.

We can solve the stochastic optimization problem based on online convex optimization. At each iteration t, we sample a random parameter vector ξ_t from $\mathbb{P}$, based on which we update our decision vector. To be more precise, we start with a decision vector $x_1 \in C$. Then we obtain a random vector ξ_1, and we adjust our decision vector to obtain a new decision vector x_2. We repeat this procedure for T time steps. We can relate this process to the online convex optimization framework. For $t = 1, \ldots, T$, we define f_t as

$$f_t(x) = h(x, \xi_t).$$

We decide x_t, after which we observe random vector ξ_t. Hence, we can look at

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x\in C}\sum_{t=1}^{T} f_t(x) = \sum_{t=1}^{T} h(x_t,\xi_t) - \min_{x\in C}\sum_{t=1}^{T} h(x,\xi_t),$$

which is the regret of the corresponding online convex optimization problem. How does it relate to solving the stochastic optimization problem? For stochastic optimization, we consider the average of $x_1,\ldots,x_T$ as a candidate solution and compute the optimality gap given by

$$\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[f\left(\frac{1}{T}\sum_{t=1}^{T} x_t\right)\right] - f(x^*)$$

where the expectation is taken over the randomness in choosing $x_2,\ldots,x_T$ and $x^*\in \operatorname{argmin}_{x\in C} f(x)$.

Theorem 14.1. *The optimality gap for SO and the regret for OCO satisfy the following relation.*

$$\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[f\left(\frac{1}{T}\sum_{t=1}^{T} x_t\right)\right] - f(x^*) \le \frac{1}{T}\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[\sum_{t=1}^{T} f_t(x_t) - \min_{x\in C}\sum_{t=1}^{T} f_t(x)\right].$$

Proof. First we deduce that

$$\begin{aligned}\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[f\left(\frac{1}{T}\sum_{t=1}^{T} x_t\right)\right] - f(x^*) &\le \mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[\frac{1}{T}\sum_{t=1}^{T} f\left(x_t\right)\right] - f(x^*)\\ &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[f\left(x_t\right)\right] - \frac{1}{T}\sum_{t=1}^{T} f(x^*)\\ &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_{t-1}\sim\mathbb{P}}\left[f\left(x_t\right)\right] - \frac{1}{T}\sum_{t=1}^{T} f(x^*).\end{aligned}$$

Note that

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_{t-1}\sim\mathbb{P}}\left[f\left(x_t\right)\right] - \sum_{t=1}^{T} f(x^*)\\ &= \sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_{t-1}\sim\mathbb{P}}\left[\mathbb{E}_{\xi_t\sim\mathbb{P}}\left[h(x_t,\xi_t)\mid \xi_1,\ldots,\xi_{t-1}\right]\right] - \sum_{t=1}^{T}\mathbb{E}_{\xi_t\in\mathbb{P}}\left[h(x^*,\xi_t)\right]\\ &= \sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_t\sim\mathbb{P}}\left[h(x_t.\xi_t)\right] - \sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_t\sim\mathbb{P}}\left[h(x^*,\xi_t)\right]\\ &= \sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[h(x_t.\xi_t)\right] - \sum_{t=1}^{T}\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[h(x^*,\xi_t)\right]\\ &= \mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[\sum_{t=1}^{T} h(x_t,\xi_t) - \sum_{t=1}^{T} h(x^*,\xi_t)\right]\\ &= \mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*)\right]\\ &\le \mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[\sum_{t=1}^{T} f_t(x_t) - \min_{x\in C}\sum_{t=1}^{T} f_t(x)\right].\end{aligned}$$

Therefore, we obtain

$$\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}x_t\right)\right]-f(x^*)\leq\frac{1}{T}\mathbb{E}_{\xi_1,\ldots,\xi_T\sim\mathbb{P}}\left[\sum_{t=1}^{T}f_t(x_t)-\min_{x\in C}\sum_{t=1}^{T}f_t(x)\right],$$

which provides an upper bound on the optimality gap for stochastic optimization. □

This in turn implies that by designing an online algorithm that minimizes the regret term for the online convex optimization problem, we can solve the stochastic optimization problem. The following is the application of online gradient descent to our stochastic optimization setting.

Algorithm 1 Online gradient descent for stochastic optimization

```
Initialize $x_1 \in C$.
for $t = 1, \ldots, T$ do
    Obtain a random vector $\xi_t \sim \mathbb{P}$ and a subgradient $g(x_t, \xi_t) \in \partial h(x_t, \xi_t)$.
    Obtain $x_{t+1} = \mathrm{Proj}_C\{x_t - \eta_t g_t\}$ for a step size $\eta_t > 0$.
end for
```

In fact, Algorithm 1 is the so-called stochastic gradient descent method. In particular, it is well-known that

$$\mathbb{E}_{\xi_t\sim\mathbb{P}}\left[g(x_t,\xi_t)\right]\in\partial f(x_t),$$

which means that $g(x_t,\xi_t)$ is an unbiased estimator of a subgradient of f at x_t. This is what we need for the convergence of stochastic gradient descent!

3 Stochastic gradient descent

Although stochastic gradient descent (SGD) on its own is a very important subject of study in optimization and machine learning, we present it as an application of online gradient descent. This section will be a gentle introduction to SGD.

Let us get back to the offline convex optimization stated as

$$\min_{x\in C} f(x).$$

If we have an access to its gradient or one of its subgradients, then we can apply gradient descent or the subgradient method. However, depending on situations, it may not be realistic to assume that we have an oracle that provides exact gradients. For example, we have just considered the stochastic optimization setting where f is given by $f(x)=\mathbb{E}_{\xi\sim\mathbb{P}}\left[h(x,\xi)\right]$, the expectation of a random function. Here, $\nabla f(x)=\mathbb{E}_{\xi\sim\mathbb{P}}\left[\nabla h(x,\xi)\right]$, to compute which we need to know the distribution $\mathbb{P}$ in general. Instead of computing the expectation exactly, what we did was to obtain a sample ξ_t so that we may use $\nabla h(x,\xi_t)$ for each iteration t. Here $\nabla h(x,\xi_t)$ is an unbiased estimator of $\nabla f(x)$.

Another example is the mean squared error minimization problem for regression.

$$\min_{\beta}\quad f(\beta)=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i-\beta^\top x_i)^2$$

where $(x_1,y_1),\ldots,(x_n,y_n)$ are the given data. In fact, this setting is also a stochastic optimization problem as we can define $\mathbb{P}$ as the empirical distribution over the n samples. To be more specific,

$$\mathbb{P}\left((x,y)=(x_i,y_i)\right)=\frac{1}{n}.$$

Then the gradient of f at β is given by

$$\nabla f(\beta) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[\nabla h(\beta, (x,y))] = -\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^\top x_i)x_i.$$

In this example, we know the precise description of the underlying distribution, from which we can compute the exact gradient. Then, what is the problem? Here, to compute the gradient, we have to go through all data points $(x_1, y_1), \ldots, (x_n, y_n)$, which may not be practical especially when the number of data is large. For this scenario, a strategy is to obtain an estimation of the gradient. We sample a data (x_r, y_r) from the data set uniformly at random and obtain

$$g_r = -(y_r - \beta^\top x_r)x_r.$$

Here r is a random variable following the uniform distribution over $\{1, \ldots, n\}$. Note that

$$\mathbb{E}[g_r] = \sum_{i=1}^{n}\mathbb{P}(r = i)\cdot g_i = \sum_{i=1}^{n}\frac{1}{n}\cdot g_i = -\frac{2}{n}\sum_{i=1}^{n}(y_i - \beta^\top x_i)x_i = \nabla f(\beta).$$

Hence, g_r is an unbiased estimator of g_r.

What we do next is to use g_r to replace $\nabla f(\beta)$ when running gradient descent.

More generally, let $\tilde{g}_x$ be an unbiased estimator of the gradient of f at x or the subgradient for f at x.

Algorithm 2 Stochastic gradient descent (SGD)

```
Initialize $x_1 \in C$.
**for** $t = 1, \ldots, T$ **do**
    Obtain an estimator $\hat{g}_{x_t}$ of some $g \in \partial f(x_t)$.
    Update $x_{t+1} = \mathrm{Proj}_C\{x_t - \eta_t \hat{g}_{x_t}\}$ for a step size $\eta_t > 0$.
**end for**
Return $(1/T)\sum_{t=1}^{T} x_t$.
```

Assume that $\tilde{g}_x$ satisfies

$$\mathbb{E}[\hat{g}_x] = g \text{ for some } g \in \partial f(x), \quad \mathbb{E}\left[\|\hat{g}_x\|^2\right] \leq L^2.$$

Under this assumption, let us analyze the performance of stochastic gradient descent given by Algorithm 2.

Theorem 14.2. *Algorithm 2 with step sizes $\eta_t = R/(L\sqrt{t})$ satisfies*

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}x_t\right)\right] - f(x^*) \leq \frac{3LR}{2\sqrt{T}}$$

where the expectation is taken over the randomness in gradient estimation and $x^* \in \mathit{argmin}_{x\in C} f(x)$.

Proof. Suppose that $\mathbb{E}[\tilde{g}_{x_t}] = g_t \in \partial f(x_t)$ for $t \geq 1$. First, let us observe the following.

$$\begin{aligned}
\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T} x_t\right)\right] - f(x^*) &\leq \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f(x_t)\right] - f(x^*) \\
&= \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}(f(x_t) - f(x^*))\right] \\
&\leq \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} g_t^\top (x_t - x^*)\right] \\
&= \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \tilde{g}_{x_t}^\top (x_t - x^*)\right]
\end{aligned}$$

where the inequalities are due to the convexity of f. Now let us consider functions $f_1, \ldots, f_T$ given by

$$f_t(x) = \tilde{g}_{x_t}^\top x.$$

Then

$$\begin{aligned}
\sum_{t=1}^{T} \tilde{g}_{x_t}^\top (x_t - x^*) &= \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f(x^*) \\
&\leq \sum_{t=1}^{T} f_t(x_t) - \min_{x \in C} \sum_{t=1}^{T} f(x) \\
&\leq \frac{3}{2} LR\sqrt{T}
\end{aligned}$$

where the last inequality is from the convergence result of online gradient descent. Note that this upper bound holds regardless of any realization of $\tilde{g}_{x_t}$'s. Therefore, the result follows. □

References

[Haz16] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3–4):157–325, aug 2016.

[Nes83] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nes04] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer Academic Publishers, Norwell, 2004.