Week 9

Nicolo Michelusi

I. Stochastic Optimization

- Up to this point, we have assumed that parameters of a convex problem are fixed and known (e.g., probability distribution of a fading channel). We thus only need to optimize the unknown control variables (power, rate, etc.)
- In addition, we have assumed that the value of the control variables in the previous iteration is known precisely.
- In reality, however, randomness and uncertainty in the system may prevent us from knowing the precise value.
  - For example, in a wireless channel, the fading distribution is unknown and need to be estimated online in order to perform optimal power allocation.
  - As another example, in congestion control, the rate of the sources needs to be known in order to optimize the control variables, but the rate value is not readily available and needs to be estimated first.
- One simple approach is to first observe the system for some amount of time, to estimate the quantities of interest; when these estimates become available, solve the optimization problem. However, this approach is problematic:
  - we need a long observation time in order to acquire accurate estimates; until that time, the control variables are not being optimally allocated, resulting in poor performance
  - the system parameters may change over time, hence the estimates might become obsolete (for example, a wireless user might move, resulting in changes in the fading distribution)

  In order to overcome these limitations, we need schemes that allow us to do estimation and control on the same time scale!
- Let us motivate the proposed algorithm through the simples estimation problem of estimating the mean of a sequence of i.i.d. random variables $\{X_t, t \geq 0\}$,

$$\mu = \mathbb{E}[X],$$

with variance

$$\sigma^2 = \mathbb{E}[(X-\mu)^2].$$

Note that this problem is equivalent to the convex optimization problem

$$\mu = \arg\min_u \frac{1}{2}\mathbb{E}[(X-u)^2]$$

In fact, the objective function can be expressed as

$$\frac{1}{2}\mathbb{E}[(X-u)^2] = \frac{1}{2}\mathbb{E}[X^2] + \frac{1}{2}u^2 - u\mathbb{E}[X],$$

which is minimized at $u = \mathbb{E}[X]$.

Since the optimization problem is convex, we can attempt to solve it using the gradient descent algorithm:

$$u_{k+1} = u_k + \gamma(\mathbb{E}[X] - u_k)$$

Note that the estimation error $e_k = u_k - \mathbb{E}[X]$ satisfies

$$e_{k+1} = (1-\gamma)e_k = (1-\gamma)^{k+1}e_0,$$

which converges to zero with linear rate if $0 < \gamma < 1$.

- In practice, however, we do not have access to the expectation $\mathbb{E}[X]$, hence we cannot compute the gradient $(u_k - \mathbb{E}[X])$, but only to sample realizations of the process, $\{X_t, t \geq 0\}$. In this case, after observing the first k samples of the process, we can estimate the expectation using the sample mean

$$u_k = \frac{1}{k}\sum_{t=0}^{k-1} X_t.$$

Note that u_k is an unbiased estimator:

$$\mathbb{E}[u_k] = \mu,$$

and its mean squared error is given by

$$\mathbb{E}[e_k^2] = \mathbb{E}[(u_k-\mu)^2] = \frac{\sigma^2}{k},$$

i.e., it decays as $O(1/k)$ with the sample size (logarithmic convergence).

We can execute the sample mean online as the following algorithm: $u_0 = 0$ and, for $k \geq 0$,

$$u_{k+1} = \frac{1}{k+1}\sum_{t=0}^{k} X_t = \left(1 - \frac{1}{k+1}\right)\frac{1}{k}\sum_{t=0}^{k-1} X_t + \frac{1}{k+1}X_k = u_k + \gamma_k(X_k - u_k),$$

where we have defined the step-size $\gamma_k = \frac{1}{k+1}$.

- Note the similarities and differences with the gradient descent algorithm:

1) we replace the negative gradient $u_k - \mathbb{E}[X]$ with the sample realization $u_k - X_k$: this sample realization is an unbiased estimate of the gradient, since

$$\mathbb{E}[u_k - X_k | u_k] = u_k - \mathbb{E}[X].$$

2) We replace the constant step-size γ with a decreasing step-size $\gamma_k = \frac{1}{k+1}$; this is necessary to average out the randomness in the process $\{X_t, t \geq 0\}$.

- This idea forms the basis of stochastic approximation algorithms: suppose that we want to solve

$$\min f(x), \text{ s.t. } x \in \mathcal{F}$$

The gradient projection algorithm is given by

$$x_{k+1} = [x_k - \gamma \nabla f_0(x_k)]^+.$$

In a stochastic approximation framework, we do not have access to $\nabla f_0(x_k)$, but only to random realizations:

$$y_k = \nabla f(x_k) + w_k,$$

where w_k is zero-mean i.i.d. noise, with variance σ^2. Then, in stochastic gradient descent we replace $\nabla f(x_k)$ with y_k and introduce a decreasing step-size:

$$x_{k+1} = [x_k - \gamma_k y_k]^+.$$

Under some assumptions on the step-size, we can prove the following theorem (here only for strongly convex functions with Lipschitz continuous gradient, but can be generalized; a similar result under slightly different conditions is provided in "Understanding Machine Learning," Ch.14):

Theorem 1. *Let f be a strongly convex function with Lipschitz continuous gradient,*

$$L\|x-y\|_2^2 \geq [\nabla f(x) - \nabla f(y)]^T(x-y) \geq \rho\|x-y\|_2^2,\ \forall x, y \in \mathbb{R}^n,$$

where $L \geq \rho > 0$, and let $x^* = \arg\min_{x\in\mathcal{F}} f(x)$ (unique since f is strongly convex). If

$$\sum_k \gamma_k = \infty,\ \sum_k \gamma_k^2 < \infty,$$

then x_k converges to x^* almost surely, and $\mathbb{E}[\|x_k - x^*\|_2^2] = O(1/k)$ (i.e., the mean-squared error converges to zero logarithmically).

Note that the step-size $\gamma_k = 1/(k+1)$ satisfies both conditions. However, the conditions in the theorem are more general. Note also that a constant step-size does NOT satisfy the above conditions.

- Example: water-filling in fading channels

- Benefits:

  - no need to estimate the channel distribution before hand

  - only need to measure the current channel state

  - if the channel distribution changes slowly over time, the algorithm automatically adapts to the changes; however, in this case we need a constant (but small) step-size (otherwise, if we let $\gamma_k \to 0$, the algorithm will stop adapting to changes after some time)