Lec27

Sunday, March 26, 2023 2:49 PM

Delayed feedback

Friday, March 20, 2009 3:31 PM

- In the above discussions, we have assumed a slotted model where the feedback is assumed to be instantaneous after each time slot.

$$x_s(t) = \arg\max_{x_s} U_s(x_s) - x_s \cdot \sum_l q_l(t) H_s^l$$

$$q_l(t+1) = \left[q_l(t) + \gamma \left(\sum_s H_s^l x_s(t) - R_l \right) \right]^+$$

- In practice, the feedback delay can be larger than the time it takes for sources/links to update their control decisions

- Also, the feedback delay can vary significantly.

- It then leads to a new set of equations

$$x_s(t) = \arg\max_{x_s} U_s(x_s) - x_s \sum_l H_s^l \, q_l(t - D_s^l(t))$$

$$q_l(t+1) = \left[q_l(t) + \gamma \left(\sum_s H_s^l x_s(t - D_l^s(t)) - R_l \right) \right]^+$$

- In typical control systems, such delay tends to lead to oscillation (instability)

$\Rightarrow$ Needs to reduce the "gain" to ensure stability.

$\Rightarrow$ Needs the stepsize to be small

- Let us try to understand why small stepsize leeds to convergence (even with delay t_0).

  - When γ is small, the value of q changes slowly

    $\Rightarrow$ the estimation $\hat{q}$ will not be too different from q

    $\Rightarrow$ the update direction will be closer to the gradient

(10)

S. H. Low and D. E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 861-874, December 1999.

Stepsize and delay - simplified

Tuesday, November 7, 2023 9:09 AM

- Suppose that we wish to minimize $f(x)$ using a gradient algorithm:

$$x(t+1) = x(t) - \gamma \nabla f(x(t))$$

- Due to delay, however, at time t we can only access $\nabla f(x(t-D))$, i.e.

$$x(t+1) = x(t) - \gamma \nabla f(x(t-D))$$

- We can still analysis the change of norm as before

$$\|x(t+1) - x^*\|^2 = \|x(t) - x^*\|^2 - 2\gamma \langle \nabla f(x(t-D)), x(t) - x^* \rangle + \gamma^2 \|\nabla f(x(t-D))\|^2$$

$$= \|x(t) - x^*\|^2 - 2\gamma \langle \nabla f(x(t-D)), x(t-D) - x^* \rangle + \gamma^2 \|\nabla f(x(t-D))\|^2$$

$$- 2\gamma \langle \nabla f(x(t-D)), x(t) - x(t-D) \rangle$$

- When γ is small, this should be $O(\gamma \cdot D \cdot \nabla f)$
- Together, the additional term should be $O(\gamma^2)$

should be negative when γ is sufficiently small & f is smooth

$\approx O(\gamma \|\nabla f\|^2)$

- Overall, we can expect that $\|x(t) - x^*\|$ will still decrease when γ is small
  - Need $\gamma \sim \frac{1}{D}$.

Stepsize and delay - skip

Friday, March 20, 2009 3:45 PM

Why small stepsize improves stability?

- Consider a delayed version of the dual controller.

$$p_l(t+1) = \left[p_l(t) + \gamma \left(\sum_s H_s^l \hat{x}_{ls}(t) - R^l \right) \right]^+$$

where $\hat{x}_{ls}(t)$ is a delayed estimate of the source rates

$$\hat{x}_{ls}(t) = \sum_{t'=t-t_0}^{t} a_{ls}(t', t)\, x_s(t')$$

$$\sum_{t'=t-t_0}^{t} a_{ls}(t', t) = 1 \quad \forall t$$

For example:

① constant delay d: $a_{ls}(t', t) = \begin{cases} 1 & t' = t-d \\ 0 & \text{o/w} \end{cases}$

② use past average: $a_{ls}(t', t) = \frac{1}{t_0+1}$

Similarly.

$$x_s(t) = \arg\max\ U_s(x_s) - x_s \cdot \sum_l H_s^l \hat{p}_{ls}(t)$$

where $\hat{p}_{ls}(t) = \sum_{t'=t-t_0}^{t} b_{ls}(t', t) \cdot p_l(t')$

$$\sum_{t'=t-t_0}^{t} b_{ls}(t', t) = 1 \quad \forall t$$

- Let us try to understand why small stepsize leads to convergence (even with delay too).

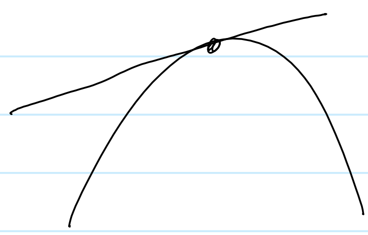
- When γ is small, the value of q changes slowly
  - $\Rightarrow$ the estimation $\hat{q}$ will not be too different from q
  - $\Rightarrow$ the update direction will be closer to the gradient

---

- Easier to work with the dual objective function

Let $\lambda(t) = \vec{q}(t+1) - \vec{q}(t)$. By growth lemma

$$g(\vec{q}(t+1)) \geq g(\vec{q}(t)) + \nabla g(\vec{q}(t)) \cdot \lambda(t) - \frac{1}{2}\beta \|\lambda(t)\|^2$$



- When there is no delayed feedback, $\lambda(t)$ will be around the direction of $\nabla g(\vec{q}(t))$.

Let $\lambda(t) = \gamma \nabla g(\vec{q}(t))$, then

$$\nabla g(\vec{q}(t)) \cdot \lambda(t) = \frac{1}{\gamma} \|\lambda(t)\|^2$$

When γ is small $\Rightarrow g \uparrow$.

- When there is delay, in general $\tilde{\lambda}(t)$ is NOT in the same direction as $\nabla g(\vec{q}(t))$.

Let $\lambda(t) = \sum_s H_s^l \hat{X}_{ls}(t) - R^l$. Then $\tilde{\lambda}(t)$ is along the direction $\lambda(t)$.

Assume $\tilde{\lambda}(t) = \gamma \lambda(t)$, then

$$\lambda(t) \cdot \tilde{\lambda}(t) = \frac{1}{\gamma} \| \tilde{\lambda}(t) \|^2.$$

- Hence

$$g(\vec{q}(t+1)) \geq g(\vec{q}(t)) + \lambda(t) \cdot \tilde{\lambda}(t) + \left[\nabla \vec{g}(\vec{q}(t)) - \lambda(t) \right] \cdot \tilde{\lambda}(t) - \frac{\beta}{2} \| \tilde{\lambda}(t) \|^2$$

$$= g(\vec{q}(t)) + \left(\frac{1}{\gamma} - \frac{\beta}{2} \right) \| \tilde{\lambda}(t) \|^2 + \left[\nabla \vec{g}(\vec{q}(t)) - \lambda(t) \right] \cdot \tilde{\lambda}(t).$$

- Note that the last-term represents the "error" due to the difference btw the update direction & the gradient

- We now show that the last-term is on the order of $\sum_{t''=t-2t_0}^{t} \| \tilde{\lambda}(t'') \|^2$ (Hence, the "error" is comparatively small when $\gamma \downarrow$)

- The difference between $\nabla \vec{g}(\vec{q}(t)) - \lambda(t)$ is

$$\left| \sum_i H_s^l X_s(t) - \sum_s H_s^l \hat{X}_{ls}(t) \right|$$

$$= \left| \sum_s H_{s}^{l} \cdot \sum_{t'=t-t_0}^{t} a_{ls}(t', t) \cdot (x_s(t) - x_s(t')) \right|$$

$$\leq \sum_s H_{s}^{l} \cdot \max_{t'} |x_s(t) - x_s(t')|$$

- The difference between $|x_s(t) - x_s(t')|$ is bounded by

$$\left| U_s'^{-1}\left(\sum_l H_{s}^{l}\, q_l(t)\right) - U_s'^{-1}\left(\sum_l H_{s}^{l}\, \hat{q}_{ls}(t')\right) \right|$$

- By the bounded curvature property of $U_s'^{-1}(\cdot)$,

$$|x_s(t) - x_s(t')| \sim \sum_l H_{s}^{l}\, |q_l(t) - \hat{q}_{ls}(t')|$$

- The difference between $q_l(t) - \hat{q}_{ls}(t')$ is

$$\left| q_l(t) - \sum_{t''=t'-t_0}^{t'} b_l(t'', t) \cdot q_l(t'') \right|$$

$$\leq \max_{t-2t_0 \leq t'' \leq t} |q_l(t) - q_l(t'')|$$

$$\leq \sum_{t''=t-2t_0}^{t} \|z_l(t'')\|$$

Hence, by choosing small γ, the positive term will dominate

$\Rightarrow$ The algorithm will still converge

The larger the delay, the larger the error term

$\Rightarrow$ The smaller the stepsize needs to be

See handout (window.pdf).

Ref:

S. H. Low and D. E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 861-874, December 1999.

30

Global stability - skip

Friday, March 20, 2009 4:26 PM

We can show that

$$g(\vec{q}(t+1)) - g(\vec{q}(t))$$

$$\geq \left(\frac{1}{\gamma} - \frac{\beta}{2}\right) \|\lambda(t)\|^2$$

$$- A_2 \sum_{t'=t-2t_0}^{t} \|\lambda(t')\| \, \|\lambda(t)\|$$

$$\geq \left(\frac{1}{\gamma} - \frac{\beta}{2} - A_2 t_0\right) \cdot \|\lambda(t)\|^2$$

$$- \frac{A_2}{2} \cdot \sum_{t'=t-2t_0}^{t} \|\lambda(t')\|^2$$

Summing over $k=1, \cdots, t$

$$g(\vec{q}(t+1)) \geq g(\vec{q}(0))$$

$$+ \left(\frac{1}{\gamma} - \frac{\beta}{2} - A_2 t_0\right) \cdot \sum_{k=1}^{t} \|\lambda(k)\|^2$$

$$- \frac{A_2}{2}(2t_0+1) \cdot \sum_{k=1}^{t} \|\lambda(k)\|^2$$

If $\frac{1}{\gamma} - \frac{\beta}{2} - A_2\left(2t_0 + \frac{1}{2}\right) > 0$ done

Note: $\beta = \bar{\alpha}\bar{S}\bar{L}$

$$A_L = \sqrt{L}\,\bar{\alpha}\,\bar{s}\,\bar{L}$$

$\bar{\alpha}\,\bar{s}\,\bar{L} \uparrow \qquad \gamma \downarrow$

$\tau_0 \uparrow \qquad \sigma \downarrow$

S. H. Low and D. E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 861-874, December 1999.

Why MDP (Markov Decision Process)?

Thursday, November 2, 2023 10:21 AM

- Let us consider the following set-up
  - Customers arrive to a system at the rate of $\lambda(p)$, which depends on the price p
  - For example

$$\lambda(p) = 1 - p \quad , \quad 0 \le p \le 1$$

    - The higher the price, the lower the interests.
  - Suppose that the system controller wishes to set the price p so that the maximum revenue is attained.
  - This is a simple unconstrained optimization problem

$$\max \; p \cdot \lambda(p) = p(1-p)$$

    - The maximum solution is

$$p_1^* = \frac{1}{2}$$

$$\lambda(p_1^*) = \frac{1}{2}$$

    - The revenue is

$$\frac{1}{2} \times \frac{1}{2} = 0.25$$

- But suppose that the system has capacity constraints
  - It can serve on average 0.2 customers per unit time

- We should then model this as a constrained optimization problem

$$\max \quad p\lambda(p) = p(1-p)$$

$$\text{sub to} \quad \lambda(p) = (1-p) \le 0.2$$

- The solution is

$$p_2^* = 0.8.$$

$$\lambda(p_2^*) = 0.2$$

The revenue is

$$0.8 \times 0.2 = 0.16$$

- But let us consider more stringent performance requirements.
  - If the arrivals and services of the customers are <u>random</u>, then typically you will see some queue build-up
    - There is a new arrival in each time slot with prob. $\lambda(p)$, there is no arrival otherwise
    - The system can serve one customer in each time-slot with prob. 0.2, and serve no customer otherwise.
  - Can we constrain our problem, so that the average queue length is bounded, say, by 9?
  - It turns out that, if we use $\lambda(p_2^*) = 0.2$, the average queue length will actually be $+\infty$!
  - For certain queueing system (e.g. M/M/1

- For certain queueing system (e.g. M/M/1 queue), the average queue length is

$$E[Q] = \frac{\lambda/\mu}{1-\lambda/\mu}$$

  - where λ is the arrival rate & μ is the service rate
  - Indeed, if $\lambda=\mu$, then $E[Q]=+\infty$!

- Let us use the above expression as an approximation of the queue length in our system

- We can then still formulate an opt. problem

$$\max \; p\lambda(p) = p(1-p)$$

$$\text{sub to} \quad \frac{\frac{\lambda(p)}{0.2}}{1-\frac{\lambda(p)}{0.2}} \le 9$$

  - The constraint is equivalent to

$$\frac{\lambda(p)}{0.2} = \frac{1-p}{0.2} \le 0.9$$

  - The solution is then

$$p_3^* = 0.82$$

$$\lambda(p_3^*) = 0.18$$

The revenue is

$0.82 \times 0.18 = 0.148$

- As we have seen, as we impose more & more stringent constraints, our rooms for optimization becomes smaller & smaller
- As a result, the solution/decision becomes more & more conservative
- But so far we are still able to solve a convex optimization problem.

  What if we put even more stringent constraints?

Even more stringent

Thursday, November 2, 2023 10:49 AM

- Let us say we even want to limit the queue length to be ≤ 9 at ALL times.

- This is actually infeasible for a fixed price
  - Due to randomness, there is always the possibility that there are back-to-back 10 customers arriving, and no service at all.
  - So the queue length will have to be larger than 9!

- One possibility, if we still want to stick to a fixed price, is to approximate the constraint:
  - $P[Q \le 9] \ge 1-\varepsilon$, e.g. $\varepsilon = 0.01$
  - Again, assuming M/M/1 queue is a good approximation, we have

$$P[Q = i] = \left(\frac{\lambda}{\mu}\right)^i \left(1-\frac{\lambda}{\mu}\right), \quad i=0,1,\ldots$$

$$\Rightarrow P[Q \le 9] = 1 - \left(\frac{\lambda}{\mu}\right)^{10}$$

  - Our opt. problem becomes

$$\max \; p \cdot \lambda(p) = p(1-p)$$

$$\text{subt to } \left(\frac{\lambda(p)}{0.2}\right)^{10} = \left(\frac{1-p}{0.2}\right)^{10} \le \varepsilon = 0.01$$

- The constraint is equivalent to

$$\frac{\lambda(p)}{0.2} = \frac{1-p}{0.2} \leq 0.01^{1/10} = 0.63$$

- The solution is now

$$p_4^* = 0.874$$

$$\lambda(p_4^*) = 0.126$$

The revenue is

$$0.874 \times 0.126 = 0.11$$

- But this really depends on how we pick ε.
  - If $\varepsilon = 0.001$, then

$$p_5^* = 0.9$$

$$\lambda(p_5^*) = 0.10$$

Revenue

$$0.9 \times 0.1 = 0.09$$

- As $\varepsilon \downarrow$, $\lambda(p^*) \downarrow 0$. Revenue $\downarrow 0$

- Is it really that hard to satisfy this type of constraint?
- No. One possible solution is the following
  - We take the price $p_2^* = 0.8$

$$\lambda(p_2^*) = 0.2$$

  - If the current queue length is < 9, we let new customers in.

- If the current queue length is $= 9$, we do not let new customers in.

- This policy will clearly maintain queue length to be ≤ 9 at all times.

- We will lose some revenue when the customers are rejected.

  - This happens with prob.

$$\approx \frac{\left(\frac{\lambda}{\mu}\right)^9 \left(1 - \frac{\lambda}{\mu}\right)}{1 - \left(\frac{\lambda}{\mu}\right)^{10}}$$

  - At $\frac{\lambda(p_2^*)}{0.2} = 1$, this prob. is $\approx \frac{1}{10}$

- The total revenue is

$$\approx p_2^* \lambda(p_2^*) \cdot \left(1 - \frac{1}{10}\right)$$

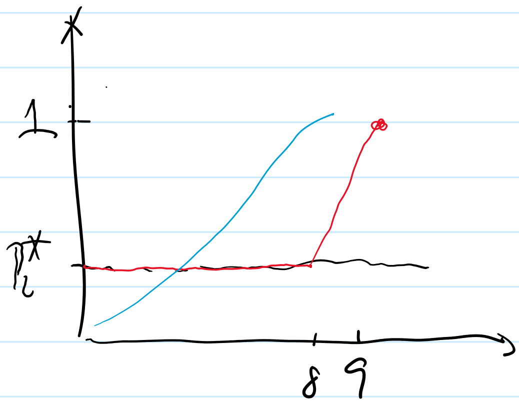
$$\approx 0.8 \times 0.2 \times 0.9 = 0.144$$

which is much better!

- How can we find this type of solutions/policies?
  - One crucial difference from the earlier convex optimization problems is that the decision is now "state" dependent:

  - We use the price p_2^* when queue < 9

use the price 1 when queue = 9



- Conceivably, our particular choice may not even be optimal!
  - We can use another price function of the "state"
- Can we develop a methodology to optimize within such state-dependent policies?

- MDP or Dynamic programming does exactly that
- In summary, MDP is useful when
  - There are stringent performance obj. & constraints
  - We wish to use "state"-dependent decisions.