EECS 189 Introduction to Machine Learning
Fall 2020 DIS0

In this discussion section we will practice using the remote collaboration tools we want you to use during discussions this semester. We will walk you through some setup, then practice by working through the first problem from Homework 0 together.

Once it is time to open the Jupyter notebook, you can get to it from the course website, or follow the direct link to open the file in the datahub directly. One group member should screenshare the notebook via zoom once the group has reached that part of the problem.

We expect you to take the first 15 minutes of discussion to get set up and practice using Jamboard. The rest of the time will be used to work through the discussion problem and play with the provided Jupyter notebook. If you have questions, we will answer them via the queue at oh.eecs189.org.

1 Jamboard Introduction

This semester we will be using Google Jamboard as a remote collaborative whiteboard for groups during discussions. This question will walk your group through the process of creating, sharing, and using a jamboard.

(a) **Using your Berkeley accounts**, all members of the group should navigate to `jamboard.google.com` on the device running Zoom **and** install Jamboard on their tablet to use for input. This will facilitate sharing links.

If you do not have a tablet, there are several options:

- Use your mouse/trackpad to participate on the collaborative whiteboard (Medium)
- Download Jamboard on your phone and use your phone as a tablet (Easy). You should buy a capacitive touch stylus since they are quite cheap, and better than finger painting.
- Download Jamboard on your phone and quickly paste pictures of your work onto the whiteboard (Easy)
- Communicate your ideas to a student who has a tablet (Hard)
- Display your written work through your phone camera (using Zoom on your phone) or webcam (Easy).

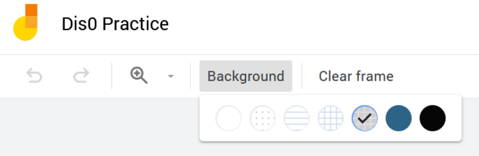
As your group members are opening jamboard, one member should create a new Jam and share it with the rest of the group. Be aware that sharing by email address can take several minutes to reach the recipients. You will also need a link to share with visiting GSIs/Readers so they can view your work.

(b) This semester we will use `oh.eecs189.org` to manage queues for both discussions and homework parties. Your discussion group should create a new group in the queue using the following settings:

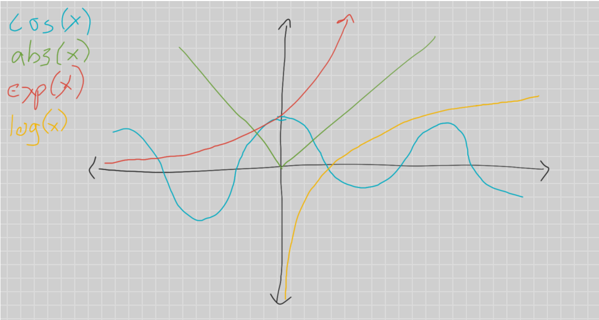
- *Assignment*: Dis 0
- *Question*: Breakout $< N >$ where $< N >$ is the number of the breakout room your group is assigned to
- *Video Call Link*: put any dummy URL here, it won't be used for discussion
- *Shared Document Link*: put an editable link to the group's Jam here
- *Location*: Online

Click `Create` to create the group, and have all members join the group. This will be how the instructors find your Jams and how you request help. If you run into technical issues this is also the best way to request help from one of the instructors.

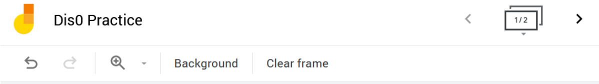
(c) Once all members have access to the Jam, one member should change the background to graph paper, as shown below.



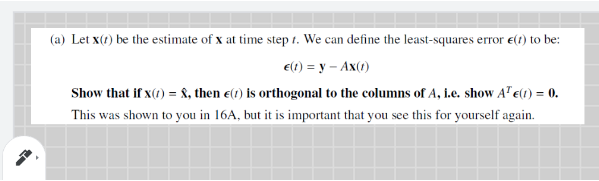
(d) Now the group members should take turns drawing functions on **shared** axes in unique-per-person colors to ensure everyone can use the interface and feels comfortable interacting with others' work. Try erasing and redrawing parts of your graph to practice.



(e) Finally, each group member should create a new page using the frame bar shown in the upper right of the figure below.



Screengrab one subpart of the next discussion problem and insert it into your page. Try resizing and rotating it. These screengrabs will provide context for work shown on the page.



(a) Let $\mathbf{x}(t)$ be the estimate of $\mathbf{x}$ at time step t. We can define the least-squares error $\boldsymbol{\epsilon}(t)$ to be:

$$\boldsymbol{\epsilon}(t) = \mathbf{y} - A\mathbf{x}(t)$$

Show that if $\mathbf{x}(t) = \hat{\mathbf{x}}$, then $\boldsymbol{\epsilon}(t)$ is orthogonal to the columns of A, i.e. show $A^T\boldsymbol{\epsilon}(t) = \mathbf{0}$.

This was shown to you in 16A, but it is important that you see this for yourself again.

(f) After your group members are comfortable using Jamboard, use the queue to `Ask for Help`. An instructor will join your breakout and make sure we can work with your Jam. At this point you can move on to the main discussion problem. We recommend that you ensure all group members can access the Jupyter notebook on the datahub, and have one group member screenshare their notebook for the group to view as a reference.

2 Stability for information processing: solving least-squares via gradient descent with a constant step size

Although ideas of control were originally developed to understand how to control physical and electronic systems, they can be used to understand purely informational systems as well. Most of modern machine learning is built on top of fundamental ideas from control theory. This is a problem designed to give you some of this flavor.

In this problem, we will derive a dynamical system approach for solving a least-squares problem which finds the $\mathbf{x}$ that minimizes $||A\mathbf{x} - \mathbf{y}||^2$. We consider A to be tall and full rank — i.e. it has linearly independent columns.

As covered in EE16A, this has a closed-form solution:

$$\hat{\mathbf{x}} = (A^TA)^{-1}A^T\mathbf{y}.$$

Direct computation requires the "inversion" of A^TA, which has a complexity of $O(N^3)$ where $(A^TA) \in \mathbb{R}^{N\times N}$. This may be okay for small problems with a few parameters, but can easily become unfeasible if there are lots of parameters that we want to fit. Instead, we will solve the problem iteratively using something called "gradient descent" which turns out to fit into our perspective of state-space dynamic equations. Again, this problem is just trying to give you a flavor for this and connect to stability, "gradient descent" itself is not yet in scope for 16B.

(a) Let $\mathbf{x}(t)$ be the estimate of $\mathbf{x}$ at time step t. We can define the least-squares error $\boldsymbol{\epsilon}(t)$ to be:

$$\boldsymbol{\epsilon}(t) = \mathbf{y} - A\mathbf{x}(t)$$

Show that if $\mathbf{x}(t) = \hat{\mathbf{x}}$, then $\boldsymbol{\epsilon}(t)$ is orthogonal to the columns of A, i.e. show $A^T\boldsymbol{\epsilon}(t) = \mathbf{0}$.

This was shown to you in 16A, but it is important that you see this for yourself again.

Solution: Plugging in $\hat{\mathbf{x}} = (A^TA)^{-1}A^T\mathbf{y}$ for $\mathbf{x}$:

$$\boldsymbol{\epsilon}(t) = \mathbf{y} - A\mathbf{x}(t)$$

$$\begin{aligned}
\boldsymbol{\epsilon}(t) &= \mathbf{y} - A(A^T A)^{-1} A^T \mathbf{y} \\
A^T \boldsymbol{\epsilon}(t) &= A^T \left(\mathbf{y} - A(A^T A)^{-1} A^T \mathbf{y}\right) \\
&= A^T \mathbf{y} - (A^T A)(A^T A)^{-1} A^T \mathbf{y} \\
&= A^T \mathbf{y} - I A^T \mathbf{y} \\
&= A^T \mathbf{y} - A^T \mathbf{y} \\
&= \mathbf{0}
\end{aligned}$$

(b) We would like to develop a "fictional" state space equation for which the state $\mathbf{x}(t)$ will converge to $\mathbf{x}(t) \to \hat{\mathbf{x}}$, the true least squares solution. The evolution of these states reflects what is happening computationally.

Here $A\mathbf{x}(t)$ represents our current reconstruction of the output $\mathbf{y}$. The difference $(\mathbf{y} - A\mathbf{x}(t))$ represents the current residual.

We define the following update:

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \alpha A^T (\mathbf{y} - A\mathbf{x}(t)) \tag{1}$$

that gives us an updated estimate from the previous one. Here α is the step-size that we get to choose. For us in 16B, it doesn't matter where this iteration comes from. But if you want, this can be interpreted as a tentative sloppy projection. If A had orthonormal columns, then $A^T(\mathbf{y} - A\mathbf{x}(t))$ would take us exactly to where we need to be. It would update the parameters perfectly. But A doesn't have orthonormal columns, so we just move our estimate a little bit in that direction where α controls how much we move. You can see that if we ever reach $\mathbf{x}(t) = \hat{\mathbf{x}}$, the system reaches equilibrium — it stops moving. At that point, the residual is perfectly orthogonal to the columns of A. In a way, this is a dynamical system that was chosen based on where its equilibrium point is.

By the way, it is no coincidence that the gradient of $||A\mathbf{x} - \mathbf{y}||^2$ with respect to $\mathbf{x}$ is

$$\nabla ||A\mathbf{x} - \mathbf{y}||^2 = 2A^T (A\mathbf{x} - \mathbf{y})$$

This can be derived directly by using vector derivatives (outside of 16B's class scope) or by carefully using partial derivatives as we will do for linearization, later in 16B. So, the heuristic update (1) is actually just taking a step along the negative gradient direction. This insight is what lets us adapt this heuristic for a kind of "linearization" applied to other optimization problems that aren't least-squares. (But all this is currently out-of-scope for 16B at this point, and is something discussed further in 127 and 189. Here, at this point in 16B, (1) is just some discrete-time linear system that we have been given.)

To show that $\mathbf{x}(t) \to \hat{\mathbf{x}}$, we define a new state variable $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}$.

Derive the discrete-time state evolution equation for $\Delta\mathbf{x}(t)$, and show that it takes the form:

$$\Delta\mathbf{x}(t+1) = (I - \alpha G)\Delta\mathbf{x}(t). \tag{2}$$

Solution:

$$\begin{aligned}
\Delta\mathbf{x}(t+1) &= \mathbf{x}(t+1) - \hat{\mathbf{x}} \\
&= \mathbf{x}(t) - \alpha A^T(A\mathbf{x}(t) - \mathbf{y}) - \hat{\mathbf{x}} \\
&= (\mathbf{x}(t) - \hat{\mathbf{x}}) - \alpha A^T(A\mathbf{x}(t) - \mathbf{y}) \\
&= \Delta\mathbf{x}(t) - \alpha A^T A\mathbf{x}(t) - \alpha A^T\mathbf{y} \\
&= \Delta\mathbf{x}(t) - \alpha A^T A(\mathbf{x}(t) - (A^T A)^{-1}A^T\mathbf{y}) \\
&= \Delta\mathbf{x}(t) - \alpha A^T A(\mathbf{x}(t) - \hat{\mathbf{x}}) \\
&= \Delta\mathbf{x}(t) - \alpha A^T A(\Delta\mathbf{x}(t)) \\
&= (I - \alpha A^T A)\Delta\mathbf{x}(t)
\end{aligned}$$

So $G = A^T A$.

(c) We would like to make the system such that $\Delta\mathbf{x}(t)$ converges to 0. As a first step, we just want to make sure that we have a stable system. To do this, we need to understand the eigenvalues of $I - \alpha G$. **Show that the eigenvalues of matrix $I - \alpha G$ are $1 - \alpha\lambda_{i\{G\}}$, where $\lambda_{i\{G\}}$ are the eigenvalues of G.**

Solution:

We actually know that G is symmetric and of the form $A^T A$. This means that it has all non-negative eigenvalues and orthonormal eigenvectors. But this is not important for this part.

Here all we need to notice that if $\lambda_{i\{G\}}$, $\mathbf{v}$ is an eigenvalue eigenvector pair for G, then $(I - \alpha G)\mathbf{v} = \mathbf{v} - \alpha\lambda_{i\{G\}}\mathbf{v} = (1 - \alpha\lambda_{i\{G\}})\mathbf{v}$.

Hence, the eigenvalues of $I - \alpha G$ are $1 - \alpha\lambda_{i\{G\}}$.

(d) To be stable, we need all these eigenvalues to have magnitudes that are smaller than 1 (since this is a discrete-time system). Since the matrix G above has a special form, all of the eigenvalues of G are non-negative and real. **For what α would the eigenvalue $1 - \alpha\lambda_{max\{G\}} = 0$ where $\lambda_{max\{G\}}$ is the largest eigenvalue of G. At this α, what would be the largest magnitude eigenvalue of $I - \alpha G$? Is the system stable?**

(Hint: Think about the smallest eigenvalue of G. What happens to it? Feel free to assume that this smallest eigenvalue $\lambda_{min\{G\}}$ is strictly greater than 0.)

Solution: Firstly, we have that $\lambda_{i\{G\}} \geq 0$. For the given condition, we want $\alpha = \frac{1}{\lambda_{max\{G\}}}$.

To find the largest magnitude eigenvalue of $(I - \alpha G)$, we need to maximize $1 - \alpha\lambda_{i\{G\}}$. We know that for the chosen α, the minimum value here is 0. To find the maximum value, the minus sign tells us that we must look at the minimum eigenvalue of G. So the maximum eigenvalue of $(I - \alpha G)$ is $1 - \alpha\lambda_{min\{G\}} = 1 - \frac{\lambda_{min\{G\}}}{\lambda_{max\{G\}}}$.

Since we are assuming $\lambda_{min\{G\}} > 0$, then the discrete-time system will be stable. Furthermore, all the eigenvalues will be in the range $[0, 1)$.

Only if $\lambda_{min\{G\}} = 0$ could we have a problem. That would happen if $A^T A$ had a nullspace which also means that A would have to have a nullspace. This could happen if we had some redundant columns.

However, this seeming threat of instability is just an illusion. This is because we could just as well eliminate the redundant columns.

The relationship between the $\lambda_\{G\}$ and $\lambda_{I-\alpha G}$ are visually shown on the number lines below.

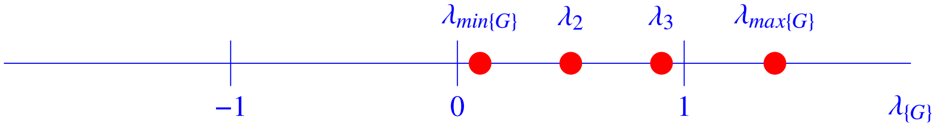


Figure 1: Original eigenvalues of the G matrix. Note all eigenvalues are non-negative. The smallest and largest magnitude eigenvalues are specifically labeled.

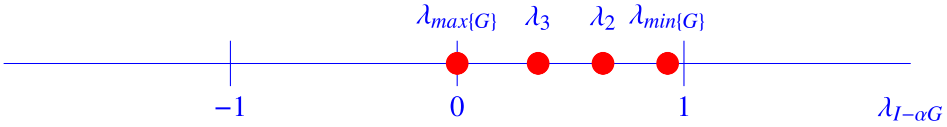


Figure 2: The eigenvalues of $I - \alpha G$ for $\alpha = \frac{1}{\lambda_{max\{G\}}}$. Note that the largest $\lambda_{\{G\}}$ is moved to 0, and the smallest $\lambda_{\{G\}}$ is moved close to 1.

(e) **Above what value of α would the system** (2) **become unstable?** This is what happens if you try to set the learning rate to be too high.

Solution: As we increase the α, the eigenvalues of $(I - \alpha G)$ march to the left. They don't change their order. So once the $\alpha > \frac{2}{\lambda_{max}}$, the set of eigenvalues will cross outside the unit circle because $1 - \alpha\lambda_{max} < -1$. It is the maximum eigenvalue of G that matters here because it is the left-most eigenvalue for $I - \alpha G$.

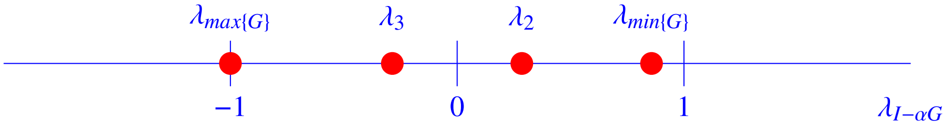


Figure 3: Plot of $\lambda_{I-\alpha G}$ for $\alpha = \frac{2}{\lambda_{max}}$. Note that there is an eigenvalue at -1, so the system is unstable.

(f) Looking back at the part before last (where you moved the largest eigenvalue of G to zero), **if you slightly increased the α, would the convergence become faster or slower?**

(HINT: think about the dominant eigenvalue here. Which is the eigenvalue of $I - \alpha G$ with the largest magnitude?)

Solution: It would converge faster because the dominant eigenvalue would get smaller. This is because the dominant eigenvalue (the one that makes the decay the slowest) is $0 > 1 - \alpha\lambda_{min\{G\}} < 1$. Increasing α slightly makes this closer to zero. Meanwhile, it only moves the maximum eigenvalue a little bit to the left from 0. So the system stays stable.

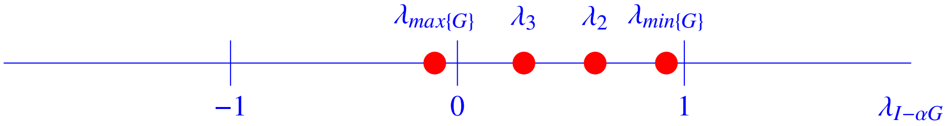


Figure 4: Plot of $\lambda_{I-\alpha G}$ for $\alpha = \frac{1}{\lambda_{max}} + \delta_{small}$. Note that all eigenvalues are still $|\lambda| < 1$, so the system remains stable. However, the one that corresponds to λ_{min} has moved further away from 1 and so this is actually more stable.

(g) **What is the α that would result in the system being stable, and converge fastest to $\Delta\mathbf{x} = 0$?** *(HINT: When would growing α stop helping shrink the biggest magnitude eigenvalue of $I - \alpha G$?)*

Solution:

This was only deemed out of scope because it involves reasoning about how to minimize the maximum here.

For a discrete system, the stability criteria is:

$$|\lambda_i| < 1$$

For $\Delta\mathbf{x}(t)$, the eigenvalues are $\lambda_i = 1 - \alpha\lambda_{i\{A^TA\}}$, where $\lambda_{i\{A^TA\}}$ are the eigenvalues of A^TA. Note that A^TA is a symmetric matrix with all eigenvalues $\lambda_{i\{A^TA\}} \geq 0$:

$$-1 < 1 - \alpha\lambda_{i\{A^TA\}} < 1$$

If we meet the condition

$$-1 < 1 - \alpha\lambda_{max\{A^TA\}} < 1$$

where $\lambda_{max\{A^TA\}}$ is the largest eigenvalue of A^TA, then we will meet the stability criteria for all eigenvalues. With a little bit of algebraic manipulation, we can get the range of α that makes the system stable:

$$0 < \alpha < \frac{2}{\lambda_{max\{A^TA\}}}$$

If we only cared about the largest eigenvalue of A^TA, then for this system to converge the fastest, we would want $1 - \alpha\lambda_{max\{A^TA\}} = 0$, which means we would choose:

$$\alpha = \frac{1}{\lambda_{max\{A^TA\}}}$$

However, we also need to think about the other eigenvalues of A^TA. With $\alpha = \frac{1}{\lambda_{max\{A^TA\}}}$, there will be other eigenvalues of the error system larger than 0 if not all eigenvalues of A^TA are the same value. The largest eigenvalue of the error corresponds to the minimum eigenvalue of A^TA, which we'll call $\lambda_{min\{A^TA\}}$. As we increase α past $\frac{1}{\lambda_{max\{A^TA\}}}$, the error eigenvalue corresponding to $\lambda_{min\{A^TA\}}$ will decrease in magnitude, but the eigenvalue corresponding to $\lambda_{max\{A^TA\}}$ will increase (since the eigenvalue starts going negative). To find the optimal α, we set the minimum and maximum eigenvalues' magnitudes equal to each other:

$$1 - \alpha\lambda_{min\{A^TA\}} = \alpha\lambda_{max\{A^TA\}} - 1$$

Note that the eigenvalue corresponding to $\lambda_{max\{A^TA\}}$ flipped signs since α was large enough to make the eigenvalue negative. This gives us an optimal step size of:

$$\alpha = \frac{2}{\lambda_{min\{A^TA\}} + \lambda_{max\{A^TA\}}}$$

It turns out that this is related to a concept in numerical linear algebra called the condition number for a matrix. When the matrix A is well conditioned, then we can get faster convergence to the solution. Basically, this requires the ratio of the maximum eigenvalue of A^TA to the minimum eigenvalue of A^TA to be small. When the eigenvalues are closer to each other, the whole cluster can be made to be around zero in the diagrams we have plotted for you.

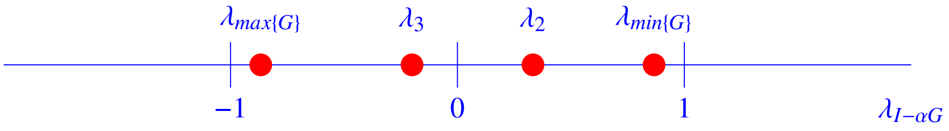


Figure 5: Plot of $\lambda_{I-\alpha G}$ for $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}}$. Note that the largest and smallest eigenvalues are centered around 0. They have the same magnitude, so the convergence will be the fastest. Any bigger α would make the one corresponding to λ_{max} closer to -1, thereby slowing down convergence. Meanwhile, any smaller α would make the one corresponding to λ_{min} closer to $+1$ which would also slow down convergence. This particular α is just right.

(h) **Play with the given jupyter notebook and comment on what you observe.** Consider how the different step sizes relate to recurrence relations, and how a sufficiently small step size can approach a continuous solution.

Solution: The most important thing to notice is that plotting on the semilog scale clearly shows the exponential speed of convergence. Having a learning rate that is too high indeed makes the estimates go unstable. While the optimal learning rate really does converge faster. Pretty much any observations count for full credit. You will learn much more about these things in 127.

3 HW Party Walkthrough

The last few minutes of this discussion will be spent having the GSIs and Readers walk through the process for HW parties.

Contributors:

- Aditya Arun
- Anant Sahai
- Christina Baek
- Josh Sanz
- Kyle Tanghe
- Miki Lustig
- Nathan Lambert
- Sidney Buchbinder