Informal Notes for Mathematics for Machine Learning
Imperial College London

Lecture 4: Gradient Descent, Convexity, and Convergence

Lecturer: Matthew Wicker

1 Learning Objectives

In our last lectures, we looked at how vector calculus can help us directly solve for the best parameter in linear models. We then looked at how we can use automatic differentiation to compute directional derivatives and from there compute numerical gradients in order to learn when symbolic differentiation is infeasible. In this lecture we will complete the picture of learning in models where directly solving for the model parameters is not straight forward. We will start in these notes by introducing the gradient descent algorithm, and analyzing its convergence properties in our analytically tractable linear regime. We then will turn our attention to definitions of convexity and proving more general convergence results.

2 What is convergence in machine learning?

There are many formal definitions of convergence (e.g., convergence in probability, convergence to distribution) each with their own specific criteria and use cases. The most widely known form of convergence, that most students study around their second year of calculus is something like the following:

Definition 2.1. Convergence A series $x_1, x_2, \ldots, x_n$ is said to *converge* to a limit L if for any $\epsilon > 0$ we have an integer K such that $\forall M > K, |x_M - L| < \epsilon$.

In calculus, this definition is typically investigated in the setting of arithmetic or geometric series. But how can the mathematical idea of convergence be employed as a tool for analysis or reasoning about our models? In order to think about this, let us pose *learning* as an iterative algorithm:

Algorithm 1 Gradient Descent
Input: X - Inputs, Y - Labels, γ - Learning rate, K - Number of iterations

$\theta_1 \leftarrow$ Random Initialization
for $i \in [K]$ **do**
  $l = \mathcal{L}(\boldsymbol{y}, f^{\theta}(\boldsymbol{X}))$
  $\theta_{i+1} \leftarrow \theta_i - \gamma \nabla_{\theta} l$
end for
return θ_K

The structure of the above algorithm (gradient descent) is the at the core of many machine learning models across different settings and application domains. One thing to notice, that links it to the idea of convergence is that it produces a series. In this case, a series of parameters $\theta_1, \theta_2, \ldots, \theta_K$. So, some a natural questions arise: *Under what conditions does this series converge? What does the series converge to? What do we want the series converge to?*

The last question is a good starting point. In particular, we would like for our parameter to converge to

$$\theta^{\star} := \underset{\theta}{\operatorname{argmin}} \left[\mathbb{E}_{(x,y)\sim\mathcal{D}} \mathcal{L}(y, \hat{y}) \right]$$

That is, we would like that for some finite K that $|\theta_K - \theta^{\star}| < \epsilon$. This seems like quite a tall order, so before tackling this when/if this can happen in general terms, lets analyze the sequence of parameter in the case when we know precisely what we would like to converge to.

3 Revisiting our Linear Regression Model

We return to our linear regression model in the supervised learning setting. Recall that we assume we have input features $\boldsymbol{x} \in \mathbb{R}^{D\times 1}$ and labels $y \in \mathbb{R}$. Recall the model is:

$$f(\boldsymbol{x}, \theta) = \boldsymbol{x}^{\top}\theta, \quad y = f(\boldsymbol{x}, \theta)$$

And we can express the loss with:

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - f(\boldsymbol{x}^{(i)}, \theta))^2$$

We can rewrite our loss in matrix form:

$$\mathcal{L}(\theta) = \frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\theta||_2^2$$

Now that we have a particular form for our loss, we can recall that if our gradient descent algorithm converges we would like for it to converge to the value $(\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}$. In addition, given that we have the exact loss we want, we can be more specific in our analysis of Algorithm 1 by plugging in

the closed form of the gradient. We now isolate the update equation in our algorithm (line 4) and expand out the gradient of the loss term:

$$\begin{aligned}\theta_{i+1} &= \theta_i - \gamma_i \nabla_\theta \mathcal{L}(\mathbf{y}, f^{\theta_i}(\mathbf{X})) \\ &= \theta_i - \gamma \nabla_{\theta_i} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\theta_i||_2^2 \\ &= \theta_i - \gamma \boldsymbol{X}^\top (\boldsymbol{X}\theta_i - \boldsymbol{y})\end{aligned}$$

This formula is almost what we want, but not quite. Typically when analyzing sequences, we want to simplify the expression with respect to the value of interest, in our case θ and by doing so for the above we get:

$$= (\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})\theta_i + \gamma \boldsymbol{X}^\top \boldsymbol{y}$$

This is a key step in our analysis because it allows us to exactly identify what kind of sequence we are looking at, in this case an arithmetico-geometric sequence. And so, we can bring tools we have learned from calculus to bear on this problem. To start, we want to look at exactly what form our update takes. In this case, the sequence is a matrix multiplication and a vector addition so we could abstract our update to:

$$\theta_{t+1} = \mathbf{B}\theta_t + \mathbf{c}, \quad t \geq 0,$$

While this is a great observation to make, the next step in our analysis would be to understand if there is a clean way to reason about what happens when we apply our update multiple times. In this case, our sequence would expand:

$$\mathbf{B}\big(\mathbf{B}(\mathbf{B}\theta_0 + \mathbf{c}) + \mathbf{c}\big) + \mathbf{c}$$

and so on. We can immediately see that this sequence would be much nicer to analyze if we could get the sequence a form such that the additive structure imposed by $+\mathbf{c}$ would cancel out. This would be the case if we could express our sequence in the following form:

$$\theta_{t+1} = \mathbf{A}(\theta_t + \beta) - \beta, \quad \text{for some } \mathbf{A}, \beta.$$

The key idea here is that all of the β terms cancel when we expand out and so we are left with simply t matrix multiplications and a single addition and subtraction after t iterations of our update. Thus, we would have the form:

$$\theta_t = \mathbf{A}^t(\theta_1 + \beta) - \beta$$

after t gradient descent updates. That is, expressing the sequence in the above form allow us to neatly express the sequence in terms of a simple matrix power operation. Readers can expand out the newly suggested form to check for themselves that this is the case. Of course, we want our update in this form, but we will need to do some work to get there. In particular, we must solve for $\mathbf{A}$ and β:

$$\begin{aligned}\theta_{t+1} =& \mathbf{A}(\theta_t + \boldsymbol{\beta}) - \boldsymbol{\beta} = \mathbf{B}\theta_t + \boldsymbol{c} \\ \Leftrightarrow \quad & \mathbf{A}\theta_t + (\mathbf{A} - \mathbf{I})\boldsymbol{\beta} = \mathbf{B}\theta_t + \boldsymbol{c} \\ \Leftrightarrow \quad & \mathbf{A} = \mathbf{B}, \quad \boldsymbol{\beta} = (\mathbf{B} - \mathbf{I})^{-1}\boldsymbol{c}\end{aligned}$$

Now that we have solved for $\mathbf{A}$ and β in terms of B and c (recall that this is just changing the sequence from one form to another), we can simply plug our values of $\mathbf{A}$ and β into get:

$$\Rightarrow \quad \theta_{t+1} = \mathbf{B}(\theta_t + (\mathbf{B}-\mathbf{I})^{-1}\boldsymbol{c}) - (\mathbf{B}-\mathbf{I})^{-1}\boldsymbol{c}$$

$$\Rightarrow \quad \theta_t = \mathbf{B}^t(\theta_0 + (\mathbf{B}-\mathbf{I})^{-1}\boldsymbol{c}) - (\mathbf{B}-\mathbf{I})^{-1}\boldsymbol{c}$$

Great, now let us recall what the form of $\mathbf{B}$ and $\boldsymbol{c}$ were from our red equation (our gradient descent update rule for linear regression), specifically:

$$\theta_{t+1} = (\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})\theta_t + \gamma \boldsymbol{X}^\top \boldsymbol{y}$$

So we have that:

$$\mathbf{A} = (\mathbf{I} - \gamma \mathbf{X}^\top \mathbf{X})$$

$$\beta = ((\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X}) - \mathbf{I})^{-1}\gamma \boldsymbol{X}^\top \boldsymbol{y}$$

Of course this is just the result of plugging in with no simplification, so we ought to simplify β (we include the algebra for completeness):

$$\begin{aligned}\beta = ((\mathbf{I} - \gamma \mathbf{X}^\top \mathbf{X}) - \mathbf{I})^{-1}\gamma \mathbf{X}^\top \mathbf{y} &= (-\gamma \mathbf{X}^\top \mathbf{X})^{-1}\gamma \mathbf{X}^\top \mathbf{y} \\ &= -(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}\end{aligned}$$

This quantity is very familiar to us indeed as at the in Lecture 2 we derived this as the least squares estimate for our linear model. By calling this value θ^*, we have arrived at the following expression for our update:

$$\theta_t = (\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})^t(\theta_0 - \theta^*) + \theta^*, \quad \theta^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$$

Thus, we can observe that our sequence will converge to the desired value if $(\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})^t(\theta_0 - \theta^*) \to \mathbf{0}$. But does this occur? In order to find out lets write out the formula for the distance between the result of our algorithm and the desired minimizer, that is he ℓ_2 distance between θ_t and θ^*:

$$\begin{aligned}||\theta_t - \theta^*||_2^2 &= ||(\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})^t(\theta_0 - \theta^*)||_2^2 \\ &= |(\theta_0 - \theta^*)^\top (\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})^{2t}(\theta_0 - \theta^*)|\end{aligned}$$

Recall from our review of linear algebra that we can use eigen decomposition in order to bound the above product. The relevant fact is: $\lambda_{min}(\mathbf{A})||\boldsymbol{x}||_2^2 \leq \boldsymbol{x}^\top \mathbf{A}\boldsymbol{x} \leq \lambda_{max}(\mathbf{A})||\boldsymbol{x}||_2^2$. When substituting for the relevant terms we get:

$$||\theta_t - \theta^*||_2^2 \geq \lambda_{min}((\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})^{2t})||\theta_0 - \theta^*||_2^2$$

$$||\theta_t - \theta^*||_2^2 \leq \lambda_{max}((\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X})^{2t})||\theta_0 - \theta^*||_2^2$$

Convergence properties in difference cases:

1. $\lambda_{max} < 1$: always converge

2. $\lambda_{min} \geq 1$: always diverge

3. $\lambda_{min} < 1$ but $\lambda_{max} \geq 1$: convergence depending on θ_0

While this gives us a guide for when our algorithm will converge, we would ideally like to try to control the maximum eigenvalue such that we guarantee convergence. So, lets use our knowledge of eigenvalues to figure out how we can manipulate γ in order to ensure convergence. Lets start by assuming that we have a value λ that is an eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$. Lets review some simple facts about eigenvalues by thinking about an eigenvalue λ of a matrix A. We know that λ^2 is an eigenvalue of A^2, we know that $c\lambda$ is an eigenvalue of cA and we know that $1 - \lambda$ is an eigenvalue of $I - A$. Putting all of these together we get that if λ is an eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$, then we have that $\left(1 - \gamma\lambda\right)^2$ is an eigenvalue of $\left(\mathbf{I} - \gamma \boldsymbol{X}^\top \boldsymbol{X}\right)^2$. Now that we have the form of an eigenvalue of our matrix of interest, we need to isolate γ. Recalling that $X^\top X$ is PSD, we have that:

$$\gamma < \frac{2}{\lambda_{max}(\boldsymbol{X}^\top \boldsymbol{X})}$$

4 Taste of Convex Optimization

Above we have observed something that is intuitively very nice. Namely, that in our linear-in-the-parameters model we can use gradient descent to converge to the least squares estimate. But this doesn't tell us anything about more general models which might not be as analytically nice. In fact, there are very few models that are as analytically tractable as our linear-in-the-parameters model. The critical piece of information is whether of not the function we would like to optimize is *convexity*.

Definition 4.1. Convex Function We say that a function $\mathcal{L}(\theta) : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if and only if:

$$\mathcal{L}(\alpha\theta + (1-\alpha)\theta') \leq \alpha\mathcal{L}(\theta) + (1-\alpha)f(\theta')$$

This definition is formulated in terms of the secant of a function and essentially says that every function value between $\mathcal{L}(\theta)$ and $\mathcal{L}(\theta')$ must lie below or on the secant. We also require that the domain of the function is a convex set. A convex set is defined as:

Definition 4.2. Convex Set We say that a set $C \subseteq \mathbb{R}^n$ is convex if $\forall x, y \in C$ and $\forall \alpha \in [0, 1]$:

$$(\alpha x + (1-\alpha)y) \in C$$

In our lecture slides we cover strict convexity as well as convexity defined for differentiable functions (given in terms of the tangent rather than the secant) as well as convexity in terms of the Hessian. We also provide a two equation slide that makes the critical point that for convex functions whose domain is also a convex set it is easy to prove that a local minimum must also be a global minimum. Please be sure to study those terms and equations as well.

Let us now build some basic intuition for how to say things about function for which we do not have the closed form but know about its convexity. Before thinking about convergence, lets

first provide formal intuition that gradient descent makes things better in the first place, assuming a convex function. Though we give a sketch here, this sort of bound is known in the literature as a progress bound. We start by stating our gradient update as an equation as we did in the linear regression case:

$$\theta^{(k)} = \theta^{(k-1)} - t_k \nabla \mathcal{L}(\theta^{(k-1)})$$

where $k = 1, 2, \ldots$ is the iteration number, t_k is the step size (or step length) at iteration k, initial $\theta^{(0)} \in \mathbb{R}^n$ is usually given. We can prove that $\mathcal{L}(\theta^{(k)}) < \mathcal{L}(\theta^{(k-1)})$ by applying first-order approximation on the LHS as follows

$$\mathcal{L}(\theta^{(k)}) = \mathcal{L}(\theta^{(k-1)} - t_k \nabla \mathcal{L}(\theta^{(k-1)})) \approx \mathcal{L}(\theta^{(k-1)}) - t_k \nabla \mathcal{L}(\theta^{(k-1)})^\top \nabla \mathcal{L}(\theta^{(k-1)}) \leq \mathcal{L}(\theta^{(k-1)})$$

Basically, this says that at each iteration we get closer to the optimal solution. As a mini exercise, students may find it useful to make the above rigorous by making additional assumptions about the function.

Another critical aspect in optimization is the smoothness of the function we wish to optimize. In addition to convexity, we will here assume that the function is twice differentiable and has a Lipschtiz continuous gradient.

Definition 4.3. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be Lipschitz continuous with Lipschitz constant $L > 0$ if, for all x, y in its domain $\mathbb{R}^n$, the following inequality holds:

$$||f(x) - f(y)|| \leq L \cdot ||x - y||$$

In other words, f is Lipschitz continuous if there exists a positive constant L such that the change in the function's values is bounded by L times the change in its input. However, we do not require that the function itself is Lipschtiz, but that its gradient is Lipschitz continuous. Given both of these facets of a function, and without knowing its exact analytical form, we can still prove some useful convergence facts.

Assume that $\mathcal{L}$ is convex, differentiable with $\text{dom}(\mathcal{L}) = \mathbb{R}^n$ and Lipschitz gradient with constant $M > 0$. We have the following theorem.

Theorem 4.4. *Gradient descent with a fixed step size $t \leq \frac{1}{M}$ satisfies:*

$$\mathcal{L}(\theta^{(k)}) - \mathcal{L}^* \leq \frac{||\theta^{(0)} - \theta^*||^2}{2tk}$$

.

I have not had time to write out and fully explain the proof for this theorem, but for an unexplained but rigerous proof of this result please see the first two pages of the following lecture notes: https://www.stat.cmu.edu/~ryantibs/convexopt-F13/scribes/lec6.pdf

A Lecture 4: Gradient Descent Convergence

Question 1 (Rayleigh quotient)**.** The *Rayleigh quotient* is defined for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and a non-zero vector $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$:

$$R(\mathbf{A}, \boldsymbol{x}) = \frac{\boldsymbol{x}^\top \mathbf{A}\boldsymbol{x}}{||\boldsymbol{x}||_2^2}, \quad ||\boldsymbol{x}||_2^2 = \boldsymbol{x}^\top \boldsymbol{x}.$$

Show that $R(\mathbf{A}, \boldsymbol{x}) \in [\lambda_{min}(\mathbf{A}), \lambda_{max}(\mathbf{A})]$.

This result immediately indicates that $\lambda_{min}(\mathbf{A})||\boldsymbol{x}||_2^2 \leq \boldsymbol{x}^\top \mathbf{A}\boldsymbol{x} \leq \lambda_{max}(\mathbf{A})||\boldsymbol{x}||_2^2$, which is used to prove gradient descent convergence.

Question 2 (Gradient descent with pre-conditioning)**.** Consider the following update rule named *pre-conditioned gradient descent*:

$$\theta_{t+1} = \theta_t - \gamma_t \mathbf{P}_t^{-1} \nabla_\theta L(\theta_t).$$

Here $\mathbf{P}_t$ is called *pre-conditioner* at time step t. We consider linear regression as an example, and assume constant learning rate and pre-conditioner, i.e., $\gamma_t = \gamma$ and $\mathbf{P}_t = \mathbf{P}$ for all t. Show that with an appropriate choice of the pre-conditioner $\mathbf{P}$, we can achieve a robust selection of the learning rate γ, i.e., if the selected γ works for an initialisation θ_0, it will also work for all other initialisations.

Hints: you can follow the below steps to solve the question:

1. Work out the pre-conditioned gradient descent update in linear regression, and derive θ_t as a function of θ_0, γ, $\mathbf{P}$ and the dataset $(\boldsymbol{X}, \boldsymbol{y})$;
2. For a given $\mathbf{P}$, work out the learning rates γ_{min} and γ_{max} such that pre-conditioned gradient descent converges when $\gamma < \gamma_{min}$, or diverges when $\gamma \geq \gamma_{max}$;
3. Select $\mathbf{P}$ such that $\gamma_{min} = \gamma_{max}$, therefore there exist no interval (like $[\gamma_{min}, \gamma_{max})$) such that convergence depends on initialisation when γ falls into such interval.

Question 3 (Momentum gradient descent)**.** Consider the following update rule named *momemtum gradient descent*, with constant learning rate γ and momentum step-size α:

$$\theta_{t+1} = \theta_t - \gamma \nabla_\theta L(\theta_t) + \alpha \Delta\theta_t,$$
$$\Delta\theta_{t+1} = \theta_{t+1} - \theta_t, \quad \Delta\theta_0 = \mathbf{0}.$$

Show that solving linear regression using momemtum gradient descent, if converges, converges to $\theta^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$.

Hint: follow the below steps and practice your linear algebra skills :)

1. Write down the update equations for the parameters θ_t and the momentum $\Delta\theta_t$;
2. Collect both terms as a long vector $(\theta_t^\top, \Delta\theta_t^\top)^\top$, and merge the two linear update equations in step 1 into one "joint" linear equation using block matrices;
3. Apply the analysis techniques for gradient descent convergence for linear regression to show the converged solution (if converges).

Question 1 (Rayleigh quotient)**.** The *Rayleigh quotient* is defined for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and a non-zero vector $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$:

$$R(\mathbf{A}, \boldsymbol{x}) = \frac{\boldsymbol{x}^\top \mathbf{A}\boldsymbol{x}}{||\boldsymbol{x}||_2^2}, \quad ||\boldsymbol{x}||_2^2 = \boldsymbol{x}^\top \boldsymbol{x}.$$

Show that $R(\mathbf{A}, \boldsymbol{x}) \in [\lambda_{min}(\mathbf{A}), \lambda_{max}(\mathbf{A})]$.

This result immediately indicates that $\lambda_{min}(\mathbf{A})||\boldsymbol{x}||_2^2 \leq \boldsymbol{x}^\top \mathbf{A}\boldsymbol{x} \leq \lambda_{max}(\mathbf{A})||\boldsymbol{x}||_2^2$, which is used to prove gradient descent convergence.

$\lambda_{min}(A)\,||x||^2 \leq x^T A x \leq \lambda_{max}(A)\,||x||^2$

$\frac{x^T A x}{||x||^2} = \frac{x^T A x}{x^T x} = \frac{x^T Q \Lambda Q^T x}{x^T Q Q^T x}$, let $z = Q^T x \rightarrow \frac{z^T \Lambda z}{z^T z}$

$z^T \Lambda z = (\cdots) \begin{bmatrix} \lambda_1 & 0 & \cdots \\ 0 & \lambda_2 & 0 \\ \vdots & 0 & \lambda_n \end{bmatrix} \begin{pmatrix} \vdots \end{pmatrix} = \sum_{i=1}^{n} z_i^2 \lambda_i$

$R(A,x)$: $\frac{\sum_{i=1}^{n} z_i^2 \lambda_i}{||z||_2^2}$ / $\sum_{i=1}^{n} z_i^2 = ||z||_2^2 \rightarrow \sum_{i=1}^{n} \lambda_i$ $\therefore$ $R(A,x)$ is between λ_{min} & λ_{max}

Question 2 (Gradient descent with pre-conditioning)**.** Consider the following update rule named *pre-conditioned gradient descent*:

$$\theta_{t+1} = \theta_t - \gamma_t \mathbf{P}_t^{-1} \nabla_\theta L(\theta_t).$$

Here $\mathbf{P}_t$ is called *pre-conditioner* at time step t. We consider linear regression as an example, and assume constant learning rate and pre-conditioner, i.e., $\gamma_t = \gamma$ and $\mathbf{P}_t = \mathbf{P}$ for all t. Show that with an appropriate choice of the pre-conditioner $\mathbf{P}$, we can achieve a robust selection of the learning rate γ, i.e., if the selected γ works for an initialisation θ_0, it will also work for all other initialisations.

Hints: you can follow the below steps to solve the question:

1. Work out the pre-conditioned gradient descent update in linear regression, and derive θ_t as a function of θ_0, γ, $\mathbf{P}$ and the dataset $(\boldsymbol{X}, \boldsymbol{y})$;
2. For a given $\mathbf{P}$, work out the learning rates γ_{min} and γ_{max} such that pre-conditioned gradient descent converges when $\gamma < \gamma_{min}$, or diverges when $\gamma \geq \gamma_{max}$;
3. Select $\mathbf{P}$ such that $\gamma_{min} = \gamma_{max}$, therefore there exist no interval (like $[\gamma_{min}, \gamma_{max})$) such that convergence depends on initialisation when γ falls into such interval.

$\theta_{t+1} = \theta_t - \gamma P^{-1} \nabla_\theta L(\theta_t)$

Linear regression: $L = \frac{1}{2}||Y - X\theta||_2^2$

$\theta_{t+1} = \theta_t - \gamma P^{-1} \nabla_\theta \left(\frac{1}{2}||Y - X\theta||_2^2 \right)$

$\frac{1}{2}||Y - X\theta||_2^2 = \frac{1}{2}(Y - X\theta)^T (Y - X\theta) = \frac{1}{2}(Y^T - \theta^T X^T)(Y - X\theta) = \frac{1}{2}(Y^T Y - Y^T X\theta - \theta^T X^T Y + \theta^T X^T X\theta)$

$= \frac{1}{2}(Y^T Y - 2\theta^T X^T Y + \theta^T X^T X\theta)$

∇_θ: $-X^T Y + X^T X\theta = X^T (X\theta - Y)$

$\theta_{t+1} = \theta_t - \gamma P^{-1} X^T (X\theta_t - Y)$

$= \theta_t - \gamma P^{-1} X^T X \theta_t + \gamma P^{-1} X^T Y = (I - \gamma P^{-1} X^T X)\theta_t + \gamma P^{-1} X^T Y$

$= B\theta_t + C$, $B = I - \gamma P^{-1} X^T X$ & $C = \gamma P^{-1} X^T Y$

$$\theta_{k+1} = \theta_k - \alpha^{-1} X^T (X\theta_k - y)$$

$$= \theta_k - \alpha^{-1} X^T X \theta_k + \alpha^{-1} X^T y$$

$$= (I - \alpha^{-1} X^T X)\theta_k + \alpha^{-1} X^T y$$

$$= B\theta_k + C \qquad B = (\cdots), \quad C = \alpha^{-1} X^T y$$

$$A(\theta_{k+1} - \beta) \quad <- \quad A^k(\theta_0 + \beta) - \beta$$

$$A\theta_k + A\beta - \beta = B\theta_k + C \quad -> \quad A = B \qquad (A - I)\beta = C$$

$$\beta = (I - B)^{-1} C = (I - (I - V))^{-1} C$$

$$\beta = (\alpha^{-1} X^T X)^{-1} \alpha^{-1} X^T y$$

$$= (X^T X)^{-1} X^T y = \theta^*$$

$$\theta_{k+1} = A(\theta_k - \beta) + \beta \quad <- \quad (I - \alpha^{-1} X^T X)(\theta_k + \theta^*) + \theta^*$$

$$\theta_{k+1} = \underbrace{(I - \alpha^{-1} X^T X)^k (\theta_0 + \theta^*)}_{\text{want to go to } 0} + \theta^*$$

$\lambda_{min}(A)\|x\|_2^2 \le x^T A x \le \lambda_{max}(A)\|x\|_2^2$

conv: $\lambda_{max} < 1$

div: $\lambda_{min} > 1$

$$\|\theta_{k+1} - \theta^*\|_2^2 = \|(I - \alpha^{-1} X^T X)^k (\theta_0 + \theta^*)\|_2^2$$

$(AB)^T AB = B^T A^T A B$

$$= (\theta_0 + \theta^*)^T (I - \alpha^{-1} X^T X)^{2k} (\theta_0 + \theta^*)$$

$x^T \quad A \quad x$

$\lambda_{min}(A)\|x\|_2^2 \le x^T A x \le \lambda_{max}(A)\|x\|_2^2$

for $\lambda_{max} < 1$: converge.

λ: eigenval of $\alpha^{-1} X^T X$ -> $(1 - \alpha\lambda)^2$ for λ

need $(1-\eta\lambda)^2 < 1$: $1 - 2\eta\lambda + \eta^2\lambda^2 < 1$

$-2\eta\lambda + \eta^2\lambda^2 < 0$

$\eta^2\lambda^2 < 2\eta\lambda$

$\eta\lambda < 2$

$\eta < \frac{2}{\lambda_{max}}$ for conv.

for diverge, $(1-\eta\lambda_{min})^2 > 1 \rightarrow \eta > \frac{2}{\lambda_{min}}$

want P s.t $\lambda_{max} = \lambda_{min}$ for $P^{-1}\hat{H}P$

identity has $\lambda_{min} = \lambda_{max} \rightarrow$ do $P = \hat{H}$ so $P^{-1}\hat{H}P$ = identity

$B\theta_t + c$: $A(\theta_t + \beta) - \beta$

$= A\theta_t + A\beta - \beta = A\theta_t + (A - I)\beta$

$A = B$, $c = (A - I)\beta \Rightarrow \beta = (A - I)^{-1} c = (B - I)^{-1} c$

$\beta = (\alpha\beta\bar{x}\bar{x} - \alpha\beta)^{-1} \alpha\beta\bar{x}\bar{y} = -(\alpha\beta\bar{x}\bar{x})^{-1} \alpha\beta\bar{x}\bar{y}$

$= -(\bar{x}\bar{x})^{-1}\bar{x}\bar{y} = -\theta^*$

$\theta_{t+1} = A(\theta_t + \beta) - \beta$

$= B(\theta_t + (B - I)^{-1} c) - (B - I)^{-1} c$

$\theta_t = B^t(\theta_0 + (B - I)^{-1} c) - (B - I)^{-1} c$

$\theta_t = (I - \alpha\beta^{-1}\bar{x}\bar{x})^t (\theta_0 - \theta^*) + \theta^*$

Need convergence to θ^* so $(I - \alpha\beta^{-1}\bar{x}\bar{x})(\theta_0 - \theta^*) \to 0$

$\|(I - \alpha\beta^{-1}\bar{x}\bar{x})^t (\theta_0 - \theta^*)\|_2^2 = (\theta_0 - \theta^*)^T (I - \alpha\beta^{-1}\bar{x}\bar{x})^{2t} (\theta_0 - \theta^*)$

If λ is eigenval of $\beta^{-1}\bar{x}\bar{x}$, then $(1 - \alpha\lambda)^2$ is eigenval of $I - \alpha\beta^{-1}\bar{x}\bar{x}$

$$\lambda_{min}(\mathbf{A})\|\boldsymbol{x}\|_2^2 \leq \boldsymbol{x}^\top \mathbf{A}\boldsymbol{x} \leq \lambda_{max}(\mathbf{A})\|\boldsymbol{x}\|_2^2,$$

convergence: need $\lambda_{max}(A) < 1$

$(1 - \alpha\lambda)^2 < 1$

$\alpha^2\lambda^2 - 2\alpha\lambda + 1 < 1$

$\alpha^2\lambda^2 - 2\alpha\lambda < 0$

$\alpha\lambda - 2 < 0$

$\alpha < \frac{2}{\lambda_{max}}$

divergence: $\lambda_{min}(A) > 1$

$(1 - \alpha\lambda)^2 > 1$

$\alpha > \frac{2}{\lambda_{min}}$

Set $\lambda_{min} = \lambda_{max} \to$ need β s.t. $\lambda_{min}(\beta^{-1}\bar{x}\bar{x}) = \lambda_{max}(\beta^{-1}\bar{x}\bar{x})$

Can do $\beta \propto \bar{x}^T\bar{x}$

Question 3 (Momentum gradient descent)**.** Consider the following update rule named *momemtum gradient descent*, with constant learning rate γ and momentum step-size α:

$$\theta_{t+1} = \theta_t - \gamma \nabla_\theta L(\theta_t) + \alpha \Delta\theta_t,$$
$$\Delta\theta_{t+1} = \theta_{t+1} - \theta_t, \quad \Delta\theta_0 = \mathbf{0}.$$

Show that solving linear regression using momemtum gradient descent, if converges, converges to $\theta^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$.

Hint: follow the below steps and practice your linear algebra skills :)

1. Write down the update equations for the parameters θ_t and the momentum $\Delta\theta_t$;
2. Collect both terms as a long vector $(\theta_t^\top, \Delta\theta_t^\top)^\top$, and merge the two linear update equations in step 1 into one "joint" linear equation using block matrices;
3. Apply the analysis techniques for gradient descent convergence for linear regression to show the converged solution (if converges).

$\theta_{t+1} := \theta_t - \gamma \nabla_\theta L(\theta_t) + \alpha \Delta\theta_t$

$L(\theta_t) = \frac{1}{2}\|y - X\theta\|_2^2 = \frac{1}{2}(y-X\theta)^T(y-X\theta) = y^Ty - y^TX\theta - \theta^TX^Ty + \theta^TX^TX\theta$
$= y^Ty - 2\theta^TX^Ty + \theta^TX^TX\theta$

$\nabla L(\theta) = -2X^Ty + 2X^TX\theta = 2X^TX\theta - 2X^Ty = X^T(X\theta - y)$

$\theta_{t+1} = \theta_t - \gamma X^T(X\theta_t - y) + \alpha\Delta\theta_t = (I - \gamma X^TX)\theta_t + \gamma X^Ty + \alpha\Delta\theta_t$

$\Delta\theta_{t+1} := \theta_{t+1} - \theta_t = -\gamma X^T(X\theta_t - y) + \alpha\Delta\theta_t = -\gamma X^TX\theta_t + \gamma X^Ty + \alpha\Delta\theta_t$

$$\begin{bmatrix}\theta_{t+1} \\ \Delta\theta_{t+1}\end{bmatrix} = \begin{bmatrix} I - \gamma X^TX & \alpha I \\ -\gamma X^TX & \alpha I\end{bmatrix}\begin{bmatrix}\theta_t \\ \Delta\theta_t\end{bmatrix} + \begin{bmatrix}\gamma X^Ty \\ \gamma X^Ty\end{bmatrix}$$

$(I - \gamma X^TX)\theta_t + \gamma X^Ty + \alpha\Delta\theta_t$
$= B\theta_t + C$

$A[\theta_t; \beta] - \beta = A\theta_t + AB - B$
$A = B \qquad C = AB - B = (A-I)B \rightarrow B = (A-I)^{-1}C = (B-I)^{-1}C$

$A[\theta_t; \beta] - \beta \rightarrow (\lambda_1 X^T X + \gamma I)^{-1}(\lambda_1 X^T X + \gamma\theta) - (\lambda_1 X^T X + \gamma I)^{-1}\gamma\theta^*$, $\theta^* = (X^TX)^{-1}X^Ty$

$\theta_t \rightarrow \theta^* + (\theta_0 - \theta^*)\gamma^t$

$\Delta\theta_{t+1} = \lambda_1 X^T X\theta - \lambda_1 X^T X\theta^* + \alpha\Delta\theta_t$
$B\theta_t + C$

$\lambda_1 X^T(I - X\gamma) + (\lambda_1 X^T X - I)\gamma\theta) X^T X\theta^* =$
$(\gamma\theta\alpha I + \lambda_1 X^T X)^{-1}(I - X^T X)\gamma\theta + X^T X\theta^*$