Course Name: Optimization Theory and Algorithms

Professor Name: Dr. Uday K. Khankhoje

Department Name: Electrical Engineering

Institute Name: Indian Institute of Technology Madras

Week – 01

Lecture - 01

Introduction to the course - 1 - Prerequisites, key elements

Okay, so we'll have tutorials throughout the semester. There will not be any tutorial tests. Okay, so it's really how much you learn in the course is up to you. Okay, we'll have regular quiz one, quiz two, and sem and likely some kind of a project. Okay, so that's and that project will likely be, you know, maybe groups of two. So, in terms of your you know, your evaluation pattern, it's very standard and very regular, nothing extraordinary about it.

It's easier than last year because last year we also had tutorial tests. This time there are no tutorial tests. Okay. Um, so that's about evaluation.

I'll, I'll put up, so there'll be, apart from Google classroom, I'll also keep my class website on my website updated. So, what content we covered on a particular day, I'll post it over there. Just the titles. Okay. I am also, I will also be maintaining class notes, which have been like typewritten notes, not handwritten notes.

They're already there on the website. Many of you may have already seen it from the previous year's website. Now here is again something I would like from you all. Those notes are actually the closest to what we are doing in the class, more so than any other textbook. So please refer to those notes when you want to revise the material, prepare for exams, tutorials, whatever.

And because I wrote them about, I started writing them two years ago, then they got refined last year, but there are still some typos. So, if you come across some typos or you come across something that doesn't make sense, let me know. It needs to be fixed. It's a work in progress. Okay, so now let's talk about prerequisites.

I think this is going to be something which will be something important for all of you. Linear algebra. Who has not yet completed a course in linear algebra? Okay. Few hands have gone up.

Okay. So, is your plan to do it concurrently or so you're currently doing it in parallel? Okay. All right. So those of you who have not yet done linear algebra or are doing it in parallel, let me warn you in the beginning only, you may find it difficult. Okay, because I assume the whole, I mean this, that is why it's a prerequisite. I assume that you're familiar with the language of vectors, vector spaces, norms, all of those things.

There is going to be, the start of the course will have a revision module, but that revision is revision. I'm not going to start from, you know, everything, right? So, it is for you to see your comfort level. If you are finding it difficult, you can either struggle or you may have to drop it. Okay. In terms of the concepts of linear algebra that we need, unfortunately what will come up again and again is something that is taught at the end of linear algebra, which is singular value decomposition.

Okay, so it's, you know, by the time you come to that in your linear algebra course, this course will be over. So, you'll have to jump up ahead. So that's as far as prerequisite. Okay, the other prerequisite, I wouldn't say prerequisite, but very useful is if you have done programming. Okay, because I'll give you some programming assignments.

They will not be evaluated, but they are really important in really solidifying your understanding of the subject. There are, of course, so many choices of languages. So just for sake of uniformity, I'm going to go with MATLAB. Okay, because most people understand MATLAB and you don't have to, you know, start from scratch like in Python from importing libraries and everything. Everything is already inbuilt over there.

And we're doing something pretty basic. So, you don't need anything fancy. So, MATLAB is what we use. First row space is there, come. Okay, so as far as textbook goes, there is a very nice book by Nocedal and Wright.

I'll just mention it over here. This book has a, there is a e-version of this book available. I'll put it. I'll give you the link on the class website.

You can download it. So, the course is largely based on this. I mean, first part of the course is largely based on this textbook. Like I mentioned, you don't really need to go read this book because the class notes that I will be putting on the website should be sufficient. OK, but it's always good to read a broader book to get an idea of the things that I'm not going to mention in class.

OK. Anything else? Okay, so now questions from your side. We are not yet recording, so you can be free. You can ask whatever you want. Any expectations, questions, wish list? What would you like? Come, come, come.

Come to the first row. There's space here. Anyone? Sir, what will be the project? What will the project entail? So, what projects have happened in the past is we want to look at some engineering papers, some engineering applications in research or industry at the heart of which is optimization. So, you have taken the course, you know some concepts, you want to express what that paper or that product is doing in the language of what you have learned. So, then we can explain it to other people. So, we've had like a course project means people have given presentations, they’ve also had poster day, so we'll see this time what we'll do.

It all depends on the class strength. If it's very large, then it's a little limited, but we'll see. Any other questions? will you cover sparse optimization? I would like to, but I will not be able to. This is our first course, so sparse optimization we will not be able to do. Any other questions? It's very similar.

So, have you done that course? Okay. So, if anyone has done this, what is it called? It's called nonlinear optimization. If anyone has done the CS course on nonlinear optimization, usually Professor Harish offers it. And we both discuss our course content. So, they are actually quite well aligned.

So, if you have done that course, there is absolutely no need for you to do this course. It's very similar. Okay. Good question. Anything else? Okay, so should we get started? So, we'll have feedback.

Anant, okay, all the TAs, not in this class, but in subsequent classes, the TAs will give out these feedback chits at the end of every class. Okay, so these feedback chits are like a quick way for me to know doubts that you have had at the end of each class. So, I'm going to read everyone's chit at the end of every class. You may or may not write your name on it.

It's up to you. And it's mostly for me to know, for example, if a large number of chits convey the same doubt, it means that I messed up. So, I will clarify it in the next class. So, it's feedback for me. And then, of course, it will be helpful for you all.

So, we'll do that from next time. Now, one thing I would like all of you to do is take notes. OK, this is I mean, all the notes are there online on my website and all. But we may have progressed a lot in technology where you get the YouTube recording and all of this. But between the eyes and the hand, there is this little thing called the brain. If you bypass that, stuff doesn't stay, you know.

Within a few days of the class, you will forget what has happened. So, it may seem very old fashioned. Just take notes. In exams, typically, I don't like to do closed book. So having your notes will help you in your exams also.

So, okay, so let's get started. We can start the recording. Yeah. Of course, of course that is a, maybe you came in late. I would like you to ask live doubts.

Don't worry about the camera at all. I may just repeat the question. So that it gets captured. Live doubts are much better than, uh, uh, doubts at the, chits at the end of the class. Okay. Sometimes these chits help for students who are shy and all but if that's not an issue It's better to ask because there's you know, something has clarified immediately.

It's better. Okay, so let's start with the introduction. Right, so the motivation for optimization is something which is, I mean, it's evident all around us in nature. For example, when you look at whether it's animals that are trying to get food or for example, when you see a drop of water which is falling during rain, it doesn't take any arbitrary shape. What kind of a shape does it take? nearly spherical, I mean if there were no gravity, it would be a perfect sphere, but in the presence of gravity, it's like a elongated kind of a thing but of all the possible shapes. For example, it could have been a cube, right? But we never see cubes of rain, we may see cubes of hail, but we don't see cubes of rain so it's not an accident, there is something happening at the heart of nature where one thing is preferred over the other and usually that principle can anyone given give a one-line intuition of why that is happening? Yeah, for example raindrops.

So, there is the answer is that for a given mass there is a certain shape which has a minimum surface area. Minimum surface area we are linking to surface tension, surface energy. So, here comes the main one of the main themes in nature is energy. We want to conserve energy and we see that in all living beings all around us right that there is somehow an effort to conserve energy. So, that survival is somehow extended or you know not in threat.

So, this is one of the central themes of nature, it is not wasteful. And when we are trying to solve problems, I mean human beings are not apart from nature, we are also as much nature as anything. In trying to solve our problems also there is a strong effort to somehow be optimum, right. We use the word without formally knowing what it means. We are trying to be as efficient; we will use words like efficient instead of optimal, right.

Now, for example, If I think of your Flipkart or Amazon delivery person, right? If there are within IIT, let's say 10 addresses where this person has to drop off a package, will this person go randomly? No, right? That doesn't make any sense. So, there is intuitively there is an optimization happening where this person, what is the quantity that they are trying to conserve or optimize in your, time? Distance, okay, but distance is a proxy for what? fuel right so it's fuel or time or and time depends depends on who's paying the bill, right. So, I may want and this gives rise to all sorts of you can see this this simple principle in when it comes to apps. For example, you book a Uber app. You don't mind paying more money, but getting a cab sooner if you have a flight to catch.

Right? And apps know that and they give you these options correspondingly underneath the hood. What are these algorithms doing? They're trying to optimize, one or the other or several variables all at the same time. So, this the more you look at engineering, the more you realize optimization is literally everywhere. Now, the purpose of taking this course is to understand those principles formally, rather than coming at it from an intuitive perspective.

Intuition will only carry you so much. After that, you need formal rigor so that you can get a solution. Not just a solution, the whole framework of optimization also gives you sometimes, not always, a way to check your solution. How optimal is it? How good is it? You may have got a solution that works, but you may want to know is it possible for there to be a better solution. So that you know that your next, that OLA cannot offer a solution fundamentally better than yours.

Because then they will take away the market share. So, things like that. So, let's write down the sort of key elements. So, the first is the motivation was motivation from nature, right? In simple words we can just say be efficient, OK. So, whatever we have spoken about intuitively, let's just try to make it a little bit more formal.

Like this example of the delivery person. We said that we were either fuel or money or both. So, in the language of optimization, what would we call it? We would call it an objective function. So, every optimization problem at the heart of it has a objective function. and, you will notice something that we will do throughout the semester, we will write something we will first try to define it in plain English before we come to some complicated mathematical definition. So, an objective function is something we want to maximize or minimize, right? Now it's important to always in any concept know the plain English wording of it for a very simple reason.

Optimization is something common to all engineering professions. Every single engineering profession when it comes down to you know being implemented there is optimization there. Now, we all come from certain streams of engineering. For example, I'm from electrical.

Someone here is from computer science. I believe there is someone from physics also and so on. You're from physics, right? So, we have a range of different, different expertise. Everyone will be expert in their problems. So, if someone from aeronautical comes and gives, shows me a fluid dynamics problem, I’m not going to understand anything. But one of the beauties of optimization is, that I sometimes I can get away with not knowing the details.

If I can, what do they call abstract it out? There is something which needs to be maximized or minimized. And I'm going to learn the machinery by which I can do this. So, if I'm, if I've understood my optimization correctly, I can ask intelligent questions without having to take three years of undergrad in aerospace engineering. that's going to be our objective.

And the best example is a software like MATLAB, right. MATLAB allows you to, for example, solve optimization problems. And what does it do? It will say define an objective function. So, you use whatever thing you want, you know, whatever stream of engineering you're doing, define an objective function, we'll work with it. Okay. What would be the next kind of a formal thing you expect in an optimization problem? We have defined an objective function.

ok what would you think should be next the parameters what is it that I am trying to what is in my hand? for example in the Uber, let us say the courier delivery example what is in my hand? path is in my hand, amount of fuel may be in my hand right, right So, we can say so what would very simple we will call them the variables of the problem. The third thing which is not so obvious immediately, but this determines how I solve the problem which is constraints. So, let us make it a little bit more general than that not just constraints, but the nature of the variables. So, I will talk a little bit more about this the nature of the variables.

Okay. So, an example would be are the variables of the problem constrained or they are free to take on any values. So, let us write this as constrained or unconstrained. Do you think there will be different strategies for solving the problem if it is constrained or unconstrained? Intuitively, what do you think? It should be, right? Not just constrained or unconstrained. So, look at before we go on to other things, what can you think of again from engineering point of view, what might be the most basic constraints that come to your mind? Think of some optimization problem.

Okay Positive, maximum value, minimum value right. So, these are the basic things. For example, how much fuel? I want to optimize the amount of fuel that I consume. So, the variable if fuel is a variable, I know from common sense that fuel cannot be a negative quantity. I know from also from common sense that I cannot expect to carry more fuel than the tank capacity. So, there is a lower limit, there is a upper limit, okay? or I can carry a tank on the side.

So, these are the sorts of ways in which now if there are constraints on the problem, then I have to factor that into this, into how I solve the course. It is not going to be that I just solve it and I get the answer and it is somehow magically going to work out. That happens, if that happens it is a fortunate thing, okay? For example, let us say that your objective function happened to be a quadratic function.

ok. Now, and your solution are the roots of the quadratic function. The roots of the quadratic function, one may be very nice, the other one may be larger than your maximum permissible value, right? So, you will have to discard one. So, constraints are important. In fact, they are so important that the second half of the course is going to be devoted to only constraint optimization. The first half is the sort of easier part which is unconstrained optimization. There are no restrictions on the variables of the problem, they can take any value.

Once we get used to that, then we will move to the framework of constraint optimization, okay? Turns out most engineering problems are constraint optimization problems. I have for example, you know this sensor nodes or IOT nodes, they all have a fixed amount of power, a maximum amount of power that they can use and you cannot exceed that. So, if your algorithm says that these nodes can consume you know 5 watts of power whereas your battery only gives 1 watt of power, what will you do? are the things that we will cover in the second half of the course, okay. So, I would say that these three key elements, this would be what we would call the model of the problem.

These define the problem model. So, what happens after this? I have defined the problem model, what happens after this? Look for the right variable values, can we say it in even simpler language? We solve the problem; we have defined the problem now we solve it right. So, we solve it. Solving it implies choosing an algorithm. There may be several ways of solving a certain problem right and the more complicated the problem the more you may have to look for a particular algorithm that suits you, right? So, this would be involved choosing the right algorithm, okay. Now, this point over here, this is actually a little bit of a subtle point because how we model the problem also dictates what kind of solution strategy or algorithm can be used.

I will give you an example. Supposing I you know this is a common kind of a thing with people who come to optimization for the first time. they want to model the problem in its full complexity, right? For example, this delivery courier delivery problem, it may be a simple what are the input variables, these are the drop off locations, these are the package weights. Okay, now, if I want to really be crazy, I can go and model the health of the person, the stamina of the person, the weather conditions, everything. I can take everything into account.

I get a nice, beautiful problem, which turns out is impossible to solve. So, there is always going to be tradeoff between complexity of the model and complexity of the solution and as engineers you have to learn to make this tradeoff. So, what you should start off with is just hack away all the inessential parts come up with a basic model which is not totally unrealistic, but it captures the main essence of the problem solve it. Then if you find the solution not satisfactory, then you can start adding the complexity one by one. You should always resist the urge to make your problem as realistic as possible on the first shot.

Guaranteed for not getting a solution. From experience I can tell you that. It just doesn't work. So, start with something solution. So, this is this is not a science, this is an art and the more experience you get the more you will be able to pick up this art, okay. So, this is something to keep in mind let me make a note of that over here tradeoff between complexity, right? model versus algorithm.

Okay. So, let us say that everything has worked out so far, I have solved my problem, are we done? We have to check. We have to check. What do we have to check? Right. So, we have to once we have got the solution, we have to check that is this good enough? does it solve my problem? it seems kind of that if I solved it obviously, I have got the correct answer right I mean what is there to check what we will see why we have to check. So, let us make a note of that you have to check for optimality conditions check for optimality, ok.

A little later in the class I will give you an example of when this will become very very useful, ok. So, we will just put that on hold for now, ok. So, this is in a nutshell, this entire course is based on just these principles, modeling the problem finding an algorithm that solves it, and then checking is the solution that I have got is it optimum another word for optimum is it good is it best you will find different different words which convey the same thing is this what I wanted? okay? or is there a way is there a better solution possible is just that my algorithm did not arrive at it? That is also possible, okay there may be another solution. So, let us now look at types of problems, okay. I have already given you a idea of types of problems in terms of constrained and unconstrained, but there are more. So, let us talk about more types of problems.