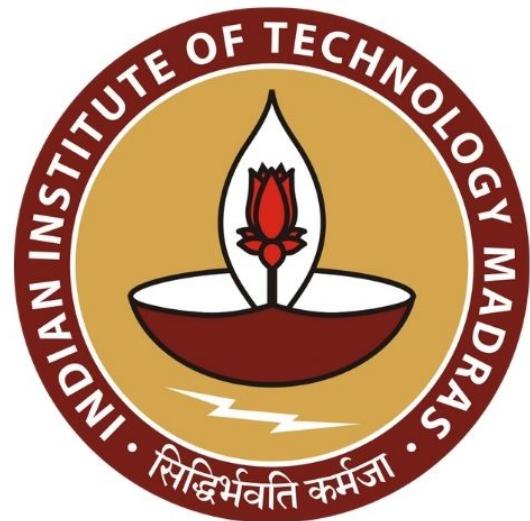




INTRODUCTION TO MACHINE LEARNING

Prof. Balaraman Ravindran
Computer Science
and Engineering
IIT Madras



INDEX

S. No	Topic	Page No.
	<i>Week 1</i>	
1	A brief introduction to machine learning	1
2	Supervised Learning	11
3	Unsupervised Learning	29
4	Reinforcement Learning	37
	<i>Week 2</i>	
5	Probability Basics - 1	44
6	Probability Basics - 2	64
	<i>Week 3</i>	
7	Linear Algebra - 1	89
8	Linear Algebra - 2	104
	<i>Week 4</i>	
9	Statistical Decision Theory - Regression	116
10	Statistical Decision Theory - Classification	131
11	Bias-Variance	139
	<i>Week 5</i>	
12	Linear Regression	144
13	Multivariate Regression	153
	<i>Week 6</i>	
14	Subset Selection 1	163
15	Subset Selection 2	170
16	Shrinkage Methods	181
17	Principal Components Regression	188
18	Partial Least Squares	197
	<i>Week 7</i>	
19	Linear Classification	202
20	Logistic Regression	210
21	Linear Discriminant Analysis 1	226
22	Linear Discriminant Analysis 2	235
23	Linear Discriminant Analysis 3	243
	<i>Week 8</i>	
24	Optimization	253
	<i>Week 9</i>	

25	Perceptron Learning	265
26	SVM - Formulation	293
27	SVM - Interpretation & Analysis	305
28	SVMs for Linearly Non Separable Data	311
29	SVM Kernels	319
30	SVM - Hinge Loss Formulation	330
31	Weka Tutorial	338

Week 10

32	Early Models	345
33	Backpropogation I	354
34	Backpropogation II	362
35	Initialization, Training & Validation	368

Week 11

36	Maximum Likelihood Estimate	380
37	Priors & MAP Estimate	384
38	Bayesian Parameter Estimation	389

Week 12

39	Introduction	397
40	Regression Trees	404
41	Stopping Criteria & Pruning	413
42	Loss Functions for Classification	421
43	Categorical Attributes	427
44	Multiway Splits	433
45	Missing Values, Imputation & Surrogate Splits	440
46	Instability, Smoothness & Repeated Subtrees	449
47	Tutorial	455

Week 13

48	Evaluation Measures I	481
49	Bootstrapping & Cross Validation	488
50	2 Class Evaluation Measures	495
51	The ROC Curve	504
52	Minimum Description Length & Exploratory Analysis	513

Week 14

53	Introduction to Hypothesis Testing	518
54	Basic Concepts	526
55	Sampling Distributions & the Z Test	536

56	Student's t-test	545
57	The Two Sample & Paired Sample t-tests	551
58	Confidence Intervals	558

Week 15

59	Bagging, Committee Machines & Stacking	565
60	Boosting	577
61	Gradient Boosting	588
62	Random Forest	601

Week 16

63	Naive Bayes	604
64	Bayesian Networks	618
65	Undirected Graphical Models - Introduction	634
66	Undirected Graphical Models - Potential Functions	647
67	Hidden Markov Models	660
68	Variable Elimination	665
69	Belief Propagation	679

Week 17

70	Partitional Clustering	687
71	Hierarchical Clustering	708
72	Threshold Graphs	716
73	The BIRCH Algorithm	728
74	The CURE Algorithm	739
75	Density Based Clustering	749

Week 18

76	Gaussian Mixture Models	758
77	Expectation Maximization	790
78	Expectation Maximization Continued	819

Week 19

79	Spectral Clustering	855
----	---------------------	-----

Week 20

80	Learning Theory	882
----	-----------------	-----

Week 21

81	Frequent Itemset Mining	911
82	The Apriori Property	923

Week 22

83	Introduction to Reinforcement Learning	936
----	--	-----

84	RL Framework and TD Learning	951
85	Solution Methods & Applications	970

Week 23

86	Multi-class Classification	980
----	----------------------------	-----

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 1

**Prof. Balaraman Ravindran
Computer Scince and Engineering
Indian Institute of Technology Madras**

Introduction to Machine Learning

Hello everyone and welcome to this NPTEL course on an introduction to machine learning in this course we will have a quick introduction to machine learning and this will not be very deep in a mathematical sense but it will have some amount of mathematical rigor. And what we will be doing in this course is covering different paradigms of machine learning and with special emphasis on classification and regression tasks and also we will introduce you to various other machine learning paradigms.

In this introductory lecture set of lectures I will give a very quick overview of the different kinds of machine learning paradigms and therefore I call this lectures machine learning, a brief introduction with emphasis on brief

(Refer Slide Time: 01:03)



Machine Learning

A Brief Introduction



So the rest of the course would be a more elongated introduction to machine learning right.

(Refer Slide Time: 01:16)



What is Machine Learning?

- “... said to learn from experience with respect to some class of tasks, and a performance measure P , if [the learner’s] performance at tasks in the class, as measured by P , improves with experience.”

Tom Mitchell 1997.



Inductive Learning

Introduction to Machine Learning

3

So what is machine learning? So I will start off with a canonical definition put out by Tom Mitchell in 97 and so a machine or an agent I deliberately leave the beginning undefined because you could also apply this to non machines like biological agents so an agent is said to learn from

experience with respect to some class of tasks right and the performance measure P if the learners performance tasks in the class as measured by P improves with experience.

So what we get from this first thing is we have to define learning with respect to a specific class of tasks right it could be answering exams in a particular subject right or it could be diagnosing patients of a specific illness right. So but we have to be very careful about defining the set of tasks on which we are going to define this learning right, and the second thing we need is of a performance measure P right so in the absence of a performance measure P you would start to make vague statement like oh I think something is happening right that seems to be a change and something learned; there is some learning going on and stuff like that.

So if you want to be more clear about measuring whether learning is happening or not you first need to define some kind of a performance criteria right. So for example if you talk about answering questions in an exam your performance criterion could very well be the number of marks that you get or if you talk about diagnosing illness then your performance measure would be the number of patients that you say are the number of patients who did not have adverse reaction to the drugs you gave them there could be variety of ways of defining performance measures depending on what you are looking for right and the third important component here is experience right.

So with experience the performance has to improve and so what we mean by experience here in the case of writing exams it could be writing more exams right so the more the number of exams you write the better you write it, the better you get it – test taking or it could be a patient's in the case of diagnosing illnesses like the more patients that you look at the better you become at diagnosing illness right.

So these are the three components so you need a class of tasks you need a performance measure and you need some well-defined experience so this kind of learning right where you are learning to improve your performance based on experience is known as a; this kind of learning where you are trying to where you learn to improve your performance with experience is known as inductive learning.

And then the basis of inductive learning goes back several centuries people have been debating about inductive learning for hundreds of years now and are only more recently we have started to have more quantified mechanisms of learning right. So but one thing I always point out to people is that if you take this definition with a pinch of salt, so for example you could think about the task as fitting your foot comfortably right.

So you could talk about whether a slipper fits your foot comfortably or let me put so I always say that you should take this definition with a pinch of salt because take the example of a slipper you know, so the slipper is supposed to give protection to your foot right and a performance measure for the slipper would be whether it is fitting the leg comfortably or not or whether it is you know as people say there is biting your leg or is it chafing your feet right and with experience you know as the slipper knows more and more about your foot as you keep wearing the slippers for longer periods of time it becomes better at the task of fitting your foot right as measured by whether it is chafing your foot or whether it is biting your foot or not right.

So would you say that the slipper is learned to fit to your foot well by this definition yes right so we have to take this with a pinch of salt and so not every system that confirms to this definition of learning can be said to learn usually okay.

(Refer Slide Time: 06:11)



ML Paradigms

- Supervised Learning
 - Learn an input and output map
 - Classification: categorical output
 - Regression: continuous output
- Unsupervised Learning
 - Discover patterns in the data
 - Clustering: cohesive grouping
 - Association: frequent cooccurrence
- Reinforcement Learning
 - Learning Control



So going on so there are different machine learning paradigms that we will talk about and the first one is supervised learning where you learn an input to output map right so you are given some kind of an input it could be a description of the patient who comes to comes to the clinic and the output that have to produce is whether the patient has a certain disease or not so this we have to learn this kind of an input to output map or the input could be some kind of equation right and then output would be the answer to the question or it could be a true or false question I give you a description of the question you have to give me true or false as the output.

And in supervised learning what you essentially do is on a mapping from this input to the required output right. If the output that you are looking for happens to be a categorical output like whether he has a disease or does not have a disease or whether the answer is true or false then the supervised learning problem is called the classification problem right and if the output happens to be a continuous value like, so how long will this product last before it fails right or what is the expected rainfall tomorrow right so these kinds of problems they would be called as regression problems.

These are supervised learning problems where the output is a continuous value and these are called as regression problems. So we will look at in more detail classification and regression as we go on right. So the second class of problems are known as unsupervised learning problems

right where the goal is not really to produce an output in response to an input but given a set of in data right we have to discover patterns in the data right. So that is more of; that is called unsupervised learning – there is no real desired output that we are looking for right we are more interested in finding patterns in the data.

So clustering right is one task one unsupervised learning task where you are interested in finding cohesive groups among the input pattern right, for example I might be looking at customers who come to my shop right and I want to figure out if there are categories of customers like so maybe college students could be one category and IT professionals could be another category and so on so forth and when I'm looking at these kinds of grouping in my data, so I would call that a clustering task right.

So the other popular unsupervised learning paradigm is known as the Association rule mining or frequent pattern mining where you are interested in finding a frequent co-occurrence of items right in the data that is given to you so whenever A comes to my shop B also comes to my shop right. So those kinds of co-occurrence so I can always say that okay if I see A then there is likely very likely that B is also in my shop somewhere you know so I can learn these kinds of associations between data right.

And again we look at this later in more detail these are I mean there are many different variants on supervised and unsupervised learning but these are the main ones that we look at so the third form of learning which is called reinforcement learning it is neither supervised or unsupervised in nature and typically these are problems where you are learning to control the behavior of a system and I will give you more intuition into reinforcement learning now in one of the later modules.

(Refer Slide Time: 09:33)



Machine Learning Tasks

Task	Measure
• Classification	error
• Regression	error
• Clustering	scatter/purity
• Associations	support/confidence
• Reinforcement Learning	cost/reward



So like I said earlier, for every task right, so you need to have some kind of a performance measure so if you are looking at classification the performance measure is going to be classification error so typically right. So we will talk about many, many different performance measures in the duration of this course but the typical performance measure you would want to use this classification error – it's how many of the items or how many of the patients did I get incorrect so how many of them who are not having the disease did I predict they had the disease and how many of them that had the disease that I missed right.

So that would be one of the measures that I would use and that would be *the* measure that we want to use but we will see later that often that is not is not possible to actually learn directly with respect to this measure. So we use other forms right and likewise for regression again so we have the prediction error suppose I say it is going to rain like 23 millimeters and then it ends up raining like 49centimeters I do not know so that is a huge prediction error right and in terms of clustering so this is a little becomes a little trickier to define performance measures we don't know what is a good clustering algorithm because we do not know what how to measure the quality of clusters.

So people come up with all different kinds of measures and so one of the more popular ones is a scatter or spread of the cluster that essentially tells you how spread out the points are that belong

to a single group if you remember we are supposed to find cohesive groups, so if the group is not that cohesive it's not all of them are not together then you would say the clustering is of a poorer quality and if you have other ways of measuring things like I was telling you, so if you know that people are college students right and then you can figure out that how many what fraction of your cluster or college students.

So you can do this kind of external evaluations so one measure that people use popularly there is known as purity right. And in the Association rule mining we use variety of measures called support and confidence that takes a little bit of work to explain support in confidence so I will defer it and I talked about Association rules in detail. And in more in the reinforcement learning tasks so if we remember I told you it is learning to control so you are going to have a cost for controlling the system and also the measure here is cost and you would like to minimize the cost that you are going to accrue while controlling the system. So these are the basic machine learning tasks.

(Refer Slide Time: 12:11)



Challenges

- How good is a model?
- How do I choose a model?
- Do I have enough data?
- Is the data of sufficient quality?
 - Errors in data. Ex: Age=225; noise in low resolution images
 - Missing Values
- How confident can I be of the results?
- Am I describing the data correctly?
 - Are Age and Income enough? Should I look at Gender also?
 - How should I represent age? As a number, or as young, middle age, old?



So there are several challenges when you are trying to build a machine learning solution right so a few of these I have listed on this slide right the first one is you have to think about how good is a model that you have learned right so I talked about a few measures on the previous slide but often those are not sufficient there are other practical considerations that come into play and we will look at some of these towards the middle of the course somewhere right and the bulk of the time would be spent on answering the second question which is how do I choose a model right.

So given some kind of data which will be the experience that we are talking about so given this experience how would I choose how would I choose a model right that somehow learns what I want to do right so how that improves itself with experience and so on so how do I choose this model and how do I actually find the parameters of the model that gives me the right answer right.

So this is what we will spend much of our time on in this course and then there are a whole bunch of other things that you really have to answer to be able to build a useful machine learning, useful data analytics or data mining solutions questions like do I have enough data do I have enough experience to say that my model is good right. Is the data of sufficient quality, there could be errors in the data right. Suppose I have medical data and age is recorded as 225, so what

does that mean it could be 225 days in which case it is a reasonable number it could be 22.5 years again is a reasonable number or 22.5 months is reasonable.

But if it is 225 years it's not a reasonable number so there is something wrong in the data right so how do you handle these things or noise in images right or missing values so I will talk briefly about handling missing values later in the course but this is as I mentioned in the beginning is a machine learning course right and this is primarily concerned about the algorithms of machine learning and the math and the intuition behind those and not necessarily about the questions of building a practical systems based on this.

So I will be talking about many of these issues during the course but just that I want to reiterate that will not be the focus right. And so the next challenge I have listed here is how confident can I be of the results and about that I certainly we will talk a little bit because the whole premise of reporting machine learning results depends on how confident you can be of the results right and the last question am I describing the data correctly.

So that is a very, very domain dependent and the question that you can answer only with your experience as a machine learning or a data sciences professional or with time right, so but there are typical questions that you would like to ask that are there on the slides so from the next in the next module we look at the different learning paradigms in slightly more detail.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

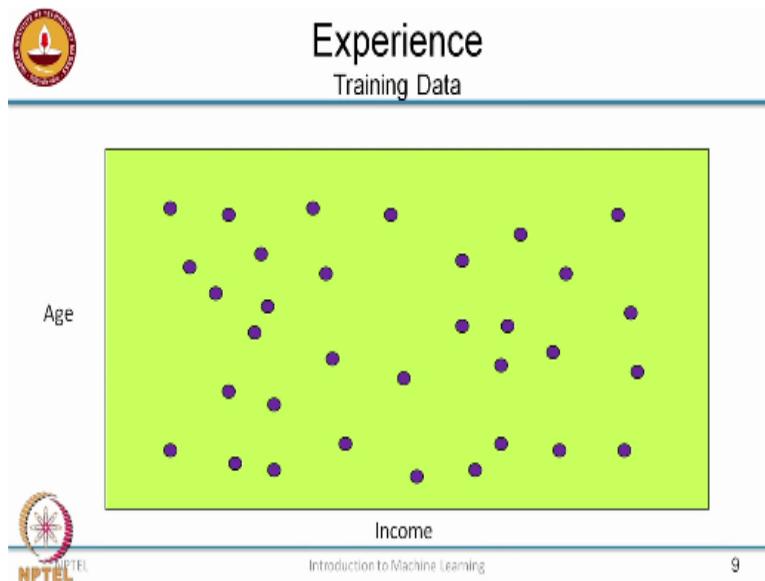
Lecture 2

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Supervised Learning

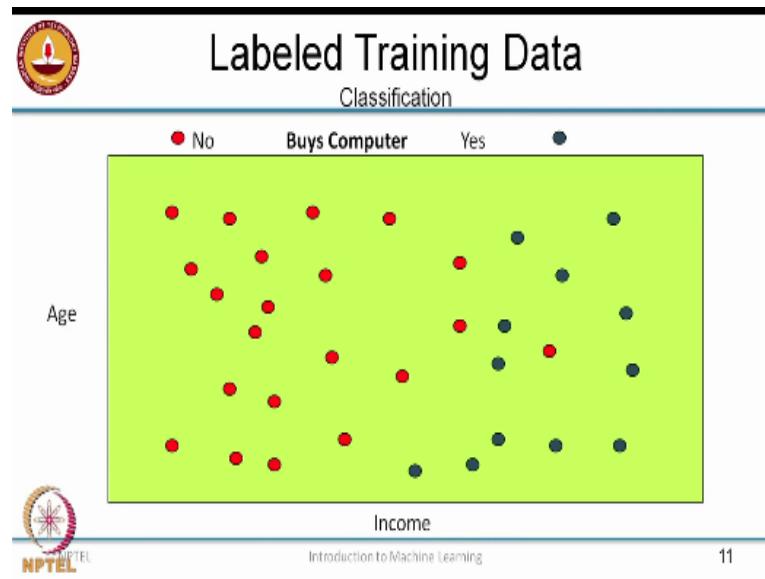
So in this module we will look at supervised learning right.

(Refer Slide Time: 00:21)



If you remember in supervised learning we talked about experience right where you have some kind of a description of the data. So in this case let us assume that I have a customer database and I am describing that by two attributes here, age and income. So I have each customer that comes to my shop I know the age of the customer and the income level of the customers right.

(Refer Slide Time: 00:48)



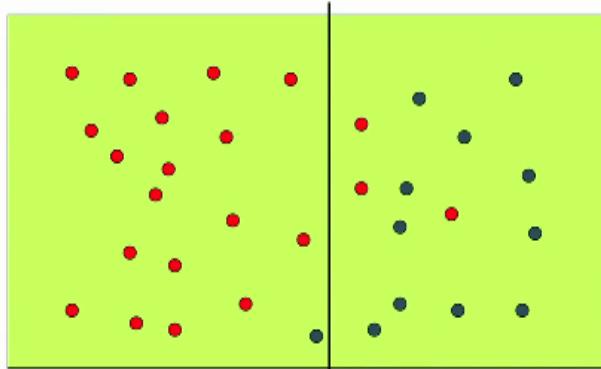
And my goal is to predict whether the customer will buy a computer or not buy a computer right. So I have this kind of labeled data that is given to me for building a classifier right, remember we talked about classification where the output is a discrete value in this case it is yes or no, yes this is the person will buy a computer, no the person will not buy a computer. And the way I describe the input is through a set of attributes in this case we are looking at age and income as the attributes that describe the customer right.

And so now the goal is to come up with a function right, come up with a mapping that will take the age and income as the input and it will give you an output that says the person will buy the computer or not buy the computer. So there are many different ways in which you can create this function.

(Refer Slide Time: 01:57)



Possible Classifiers



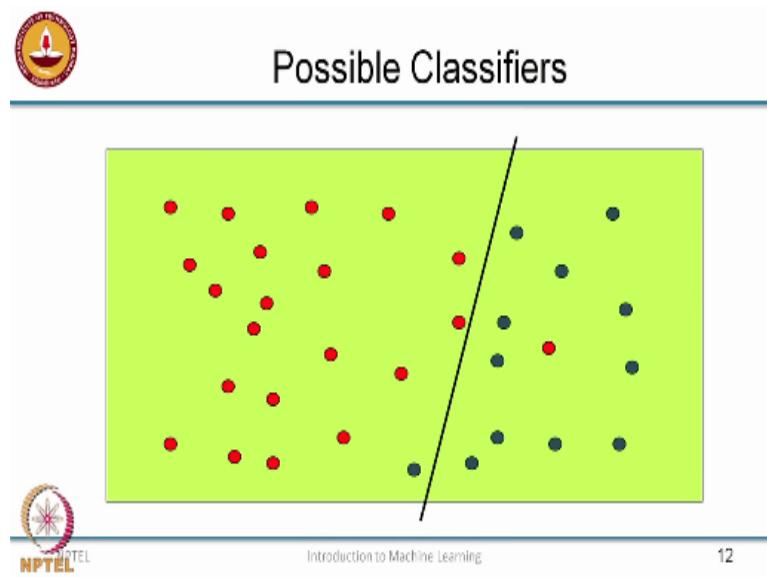
And given that we are actually looking at a geometric interpretation of the data, I am looking at data as points in space, the one of the most natural ways of thinking about defining this function is by drawing lines or curves on the input space right. So here is one possible example, so here I have drawn a line and everything to the left of the line right. So these are points that are red right, so everything to the left of the line would be classified as will not buy a computer, everything to the right of the line where the predominantly the data points are blue will be classified as will buy a computer.

So how would the function look like, it will look like something like if the income of a person remember that the x-axis is income and the y-axis is age. So in this case it basically says that if the income of the person is less than some value right, less than some X then the person will not buy a computer. If the income is greater than X the person will buy your computer. So that is the kind of a simple function that we will define.

Just notice that way we completely ignore one of the variables here which is the age. So we are just going by income, if the income is less than some X then the person will not buy a computer, if the income is greater than X the person will buy a computer. So is this a good rule more or less I mean we get most of the points correct right except a few right. So it looks like yeah, we can

we can survive with this rule right. So this is not too bad right, but then you can do slightly better.

(Refer Slide Time: 03:29)



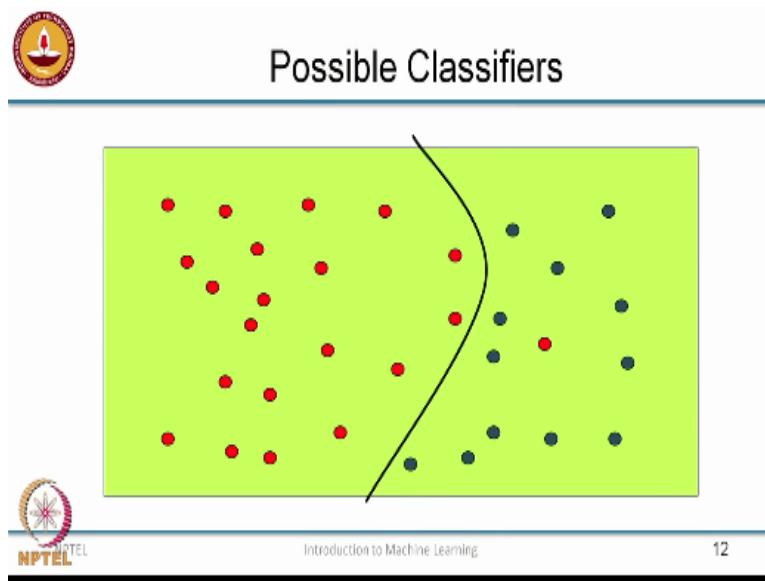
All right, so now we got those two red points that those just keep that points are on the wrong side of the line earlier now seem to be on the right side right. So everything to the left of this line will not buy a computer, everything to the right will buy a computer right, everyone moves to the right will buy a computer. So if you think about what has happened here, so we have improved our performance measure right.

So the cost of something, so what is the cost here. So earlier we are only paying attention to the income right, but now we have to pay attention to the age as well right. So the older you are right, so the income threshold at which we will buy a computer is higher right. So the younger you are, younger means lower on the y axis, so the younger you are the income threshold at which you will buy a computer is lower right.

So is that clear, so the older you are, the income threshold is shifted to the right here. So the older you are, so you need to have a higher income before you buy a computer and the younger

you are your income threshold is lower, so you do not mind buying a computer even if your income is slightly lesser. So now we have to start paying attention to the age right, but then the advantage is you get much better performance right.

(Refer Slide Time: 04:54)

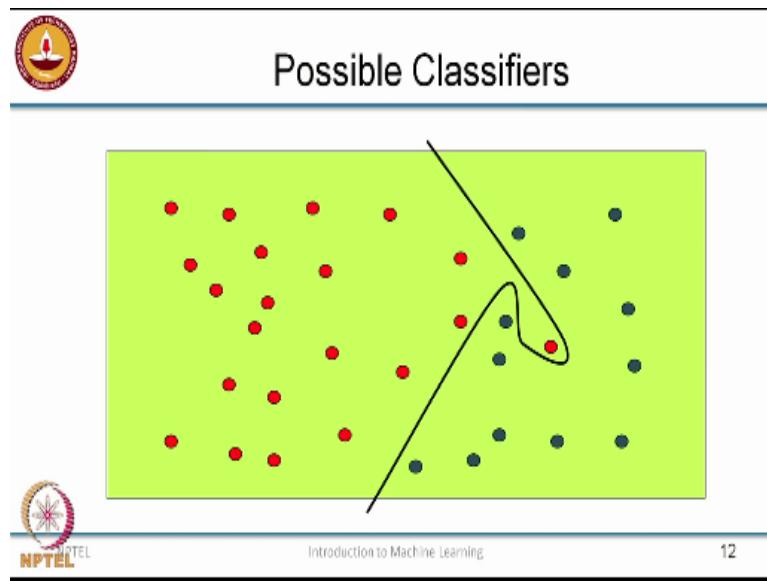


Can you do better than this yes? Now almost everything is correct except that one pesky red point, but everything else is correct. And so what has happened here we get much better performance, but at the cost of having a more complex classifier right.

So earlier if you thought about it in geometric terms, so first you had a line that was parallel to the y-axis therefore, I just needed to define a intercept on the x-axis right. So if X is less than some value then it was one class was greater than some value was another class. Then the second function it was actually a slanting line like that, so I needed to define both the intercept and the slope right.

And now here it is now a quadratic so I have to define three parameters right. So I have to define something like $ax^2 + bx + c$, so I have defined the a, b, c – the three parameters in order to find the quadratic, and I am getting better performance.

(Refer Slide Time: 05:57)



So can you do better than this? Okay that somehow does not seem right correct seems to be too complex a function just to be getting this one point there right. And I am not sure I am not even sure how many parameters you need for drawing that because Microsoft use some kind of spline PowerPoint use some kind of spline interpolation to draw this curve I am pretty sure that it is got lot more parameters than it is worth. Another thing to note here is that that particular red point that you see is actually surrounded by a sea of blue right.

So it is quite likely that there was some glitch there either the person actually bought a computer and we never we have not recorded it has been having bought a computer or there are some extremist reason the person comes into the shop sure that is going to buy a computer but then gets a phone call saying that some emergency please come out immediately and therefore he left without buying a computer right there could be variety of reasons for why that noise occurred and this will probably be the more appropriate classifier right.

So these are the kinds of issues I would like to think about what is the complexity of the classifier that I would like to have right and versus the accuracy of the classifier, so how good is the classifier in actually recovering the right input output map and or their noise data in the in the input in the experience that I am getting is it clean or is there noise on it and if so how do I handle that noise these are the kinds of issues that we have to look at okay.

(Refer Slide Time: 07:31)



Inductive Bias

- Need to generalize → Assumptions about lines!
- In general, **Inductive bias**
 - Language bias
 - Search bias



Introduction to Machine Learning 13

So these kinds of lines that we drew right kind of hiding one assumption that we are making so the thing is the data that comes to me comes as discrete points in the space right and from these discrete points in the space I need to generalize and be able to say something about the entire state space right so I do not care where the data point is on the x and y-axis right I should be able to give a label to that right.

If I do not have some kind of assumption about these lines right and if you do not have some kind of assumptions about these lines the only thing I can do is if the same customer comes again or somebody who has exact same age and income as that cause customer comes again I can tell

you whether the person is going to buy a computer or not buy a computer but I will not be able to tell you about anything else outside of the experience right.

So the assumption we made is everything to the left of a line is going to do one thing or the other; so everything to the left of the line will not buy the computer everything to the right or everyone to the right will buy a computer this is an assumption I made the assumption was the lines are able to segregate people who buy from who do not buy the lines or the curves were able to segregate people who will buy from who will not buy so that is a kind of an assumption I made about the distribution of the input data and the class labels.

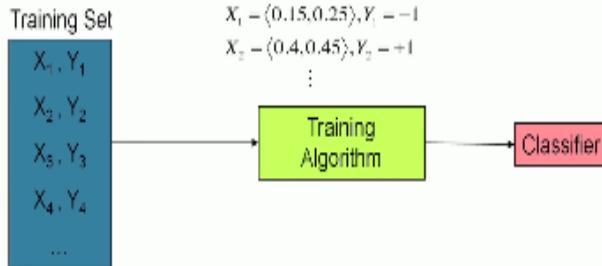
So this kind of assumptions that we make about these lines are known as inductive biases in general inductive bias has like two different categories one is called language bias which is essentially the type of lines that I am going to draw. Am I gonna draw straight lines or am I going to draw curves and what order polynomials am I going to look at and so on so forth these for my language bias. And search bias is the other form of inductive bias that tells me how in what order am I going to examine all these possible lines right.

So that gives me the gives me a search bias right, so putting these things together we are able to generalize from a few training points to the entire space of inputs right I will make this more formal as we go on and then in the next set of modules right.

(Refer Slide Time: 10:01)



The Process



And so here is one way of looking at the whole process so I am going to be giving you a set of data which we will call the training set. So the training set will consist of say as an input which we'll call as X and an output which we call as Y right, so I am going to have a set of inputs I have X_1 , X_2 , X_3 , X_4 likewise I will have Y_1 , Y_2 , Y_3 , Y_4 and this data is fed into a training algorithm right and so the data is going to look like this in our case right.

So remember our X's are the input variable X's are the inputs so in this case that should have the income and the age, so x_1 is like 30,000 and 25 and x_2 is like 80,000 and 45 and so on so forth and the y's or the labels they correspond to the colors in the previous picture right so y_1 does not buy a computer y_2 buys a computer and so on so forth so this essentially gives me the color coding so y_1 is essentially red and y_2 is blue right and I really if I am going to use something numeric this is what we will be doing later on I really cannot be using these values. First of all wise or not numeric and the X's vary too much right.

So the first coordinate in the X is like 30,000 and 80,000 and so on so forth and the second coordinate is like 25 and 45 so that is a lot a lot smaller in magnitude so this will lead to some kind of numerical instabilities. So what will typically end up doing is normalizing these so that they form approximately in the same range so you can see that I have try to normalize these X values between 0 and 1 right.

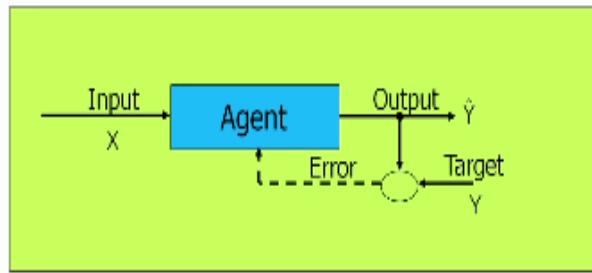
So have chosen an income level of say 2 lakhs it is the maximum and age of 100 and you can see the normalized values and likewise for buys and not buy I have taken not by as - 1 and by as computer is + 1. These are arbitrary choices, now but later on you will see that there are specific reasons for wanting to choose this encoding in this way. And then the training algorithm chugs over this data right and it will produce a classifier so now this classifier I do not know whether it is good or bad right so we had a straight line in the first case right an axis parallel line if we did not know the good or bad and we needed to have some mechanism by which we evaluate this right.

So how do we do the evaluation typically is that you have what is called a test set or a validation set right so this is another set of x and y pairs like we had in the training set, so again in the test set we know what the labels are it is just that we are not showing it to the training algorithm we know what the labels are because we need to use the correct labels to evaluate whether your training algorithm is doing good or bad right so, so this process by which this evaluation happens is called validation. Then at the end of the validation, if you are happy with the quality of the classifier we can keep it. If you are not happy then go back to the training algorithm and say hey I am not happy with what you produced give me something different right, so we have to either iterate over the algorithm again we will go over the data again and try to refine the parameter estimation or we could even think of changing some parameter values and then trying to redo the training algorithm all over again. But this is the general process and we will see that many of the different algorithms that we look, look at in the course; during the course of these lectures actually follow this kind of a process .

(Refer Slide Time: 13:48)



Training



So what happens inside that green box? So inside the training algorithm is that there will be this learning agent right which will take an input and it will produce an output \hat{Y} which it thinks is the correct output right but it will compare it against the actual target Y it was given for the in the training right, so in the training you actually have a target Y so it will compare it against a target why right and then figure out what the error is and use the error to change the agent right so then it can produce the right output next time around. This is essentially an iterative process so you see that input okay produce an output \hat{Y} and then you take the target Y , you can compare it to the \hat{Y} , figure out what is the error and use the error to change the agent again right. And this is by and large the way most of the learning algorithms will operate; most of the classification algorithms or even regression algorithms will operate and we will see how each of this works as, we go on right.

(Refer Slide Time: 14:46)



Applications

- Credit Card fraud detection
 - Valid transaction or not
- Sentiment Analysis
 - Opinion mining; buzz analysis; etc.
- Churn prediction
 - Potential churner or not
- Medical diagnoses
 - Risk analysis

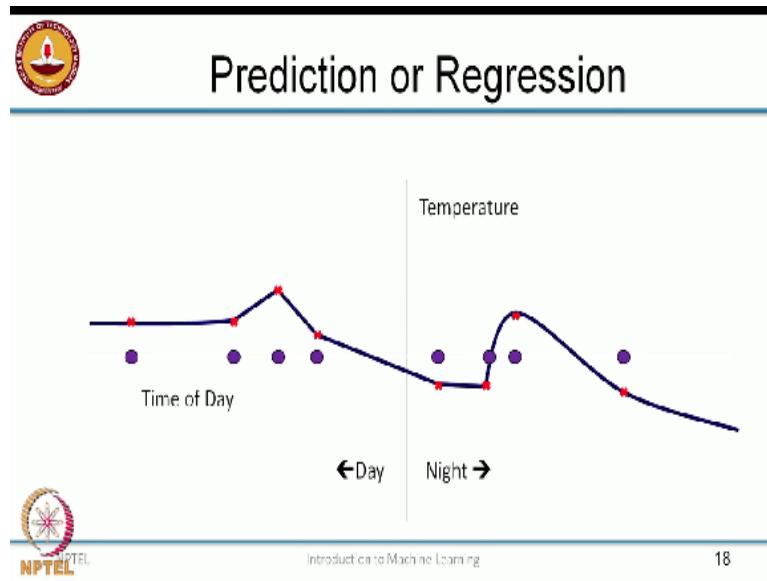


There are many, many applications. I mean this is too numerous to list. Here are a few examples you could look at say a fraud detection right, so we have some data where the input is a set of transactions made by a user and then you can flag each transaction as a valid transaction or not. You could look at sentiment analysis you know variedly called as opinion mining or buzz analysis etc., where I give you a piece of text or a review written about a product or a movie and then you tell me whether the movies whether the review is positive or whether is negative and what are the negative points that people are mentioning about and so on so forth and this again a classification task. Or you could use it for doing churn prediction where you are going to say whether a customer who is in the system is likely to leave your system or is going to continue using your product or using your service for a longer period of time, so this is essentially churn so when a person leaves your services you call the person chunner and you can label what the person is chunner or not. And I have been giving you examples from medical diagnosis all through apart from actually diagnosing whether a person has a disease or not you could also use it for risk analysis in the slightly indirect way I will talk about that when we when we do the algorithms for classification.

So we talked about how we are interested in learning different lines or curves that can separate different classes in supervised learning and, so this curves can be represented using different structures and throughout the course we will be looking at different kinds of learning

mechanisms like artificial neural networks support vector machines decision trees nearest neighbors and Bayesian networks and these are some of the popular ones and we look at these in more detail as the course progresses.

(Refer Slide Time: 16:45)



So another supervised learning problem is the one of prediction or regression where the output that you are going to predict is no longer a discrete value it is not like – will buy a computer or does not buy a computer – it is more of a continuous value so here is an example, where at different times of day you have recorded the temperature so the input to the system is going to be the time of day and the output from the system is going to be the temperature that was measured at a particular point at the time right. So you are going to get your experience or your training data is going to take this form so the blue points would be your input and the red points would be the outputs that you are expected to predict.

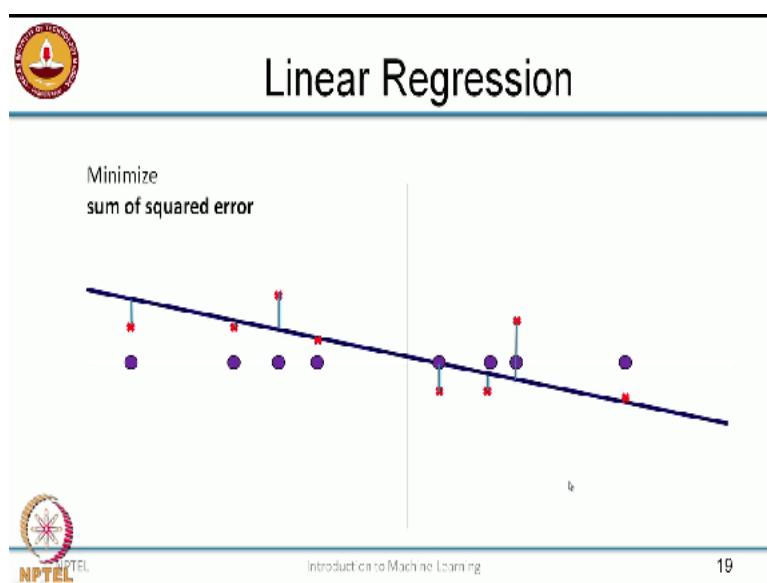
So note here that the outputs are continuous or real value right and so you could think of this in this toy example as points to the left being day and the points to the right being night right. And just as in the previous case of classification, so we could try to do these simple as possible fit in

this case which would be to draw a straight line that is as close as possible to these points now you do see that like in the classification case when it choose a simple solution there are certain points at which we are making large errors right so we could try to fix that.

And try to do something more fancy. But you could see that while the day time temperatures are more or less fine but with the night times we seem to be doing something really off right because we are going off too much to the right-hand side. How if you could do something more complex just like in the classification case where we wanted to get that one point right so we could try and fit all these temperatures that were given to us by looking at a sufficiently complex curve.

And again this as we discussed earlier is probably not the right answer and you are probably in this case surprisingly or better off fitting the straight line right. So these kinds of solutions where we try to fit the noise in the data we are trying to make the solution predict the noise in the training data correctly are known as over fitting, over fit solutions and one of the things that we look to avoid in, in machine learning is to over fit to the training data. So we will talk about this again in due course.

(Refer Slide Time: 19:21)



So what we do is typically we would like to do what is called linear regression some of you might have come across this under different circumstances and the typical aim in linear regression is to say take the error that your line is making so if you take an example point, let us say I take any I take an example point somewhere here right.

So this is the actual training data that is given to you and this is the prediction that your line is making at this point so this quantity is essentially the, the prediction error that this line is making and so what you do is you try to find that line that has the least prediction error right so you take the square of the errors that your prediction is making and then you try to minimize the, the sum of the squares of the errors. Why do we take the squares? Because errors could be both positive or negative and we want to make sure that you are minimizing that regardless of the sign of the error okay.

(Refer Slide Time: 20:31)



Linear Regression

- Minimize sum squared error
- With sufficient data simple enough
- With many dimensions, challenge is to avoid over fitting
 - Regularization
- Higher order functions?
 - Basis transformations
 - Ex: $(x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1x_2, x_1, x_2)$



NPTEL
Introduction to Machine Learning
20

So with sufficient data right so a linear regression is simple enough you could just solve it using matrix inversions as we will see later but with many dimensions the challenge is to avoid over fitting like we talked about earlier and then there are many ways of avoiding this.

And so I will again talk about this in detail when we look at linear regression right. So one point that I want to make is that linear regression is not as simple as it sounds right. So here is an example so I have two input variables x_1 and x_2 right and if I try to fit a straight line with x_1 and x_2 I will probably end up with something like

$$a_1 x_1 + a_2 x_2$$

and that looks like a plane in two dimensions right.

But then if I just take these two dimensions and then transform them transform the input so instead of saying just the x_1 and x_2 if I say my input is going to look like x_1^2 , x_2^2 , x_1x_2 and then the x_1 and x_2 as it was in the beginning so instead of looking at a two-dimensional input if I am going to look at a 5 dimensional input right, and now I am going to fit a line or a linear plane in this 5 dimensional input so that will be like

$$a_1 x_1^2 + a_2 x_2^2 + a_3 x_1 x_2 + a_4 x_1 + a_5 x_2$$

Now that is no longer the equation of a line in two dimensions right so that is the equation of a second-order polynomial in two dimensions but I can still think of this as doing linear regression because I am only fitting a function that is going to be linear in the input variables right.

(Refer Slide Time: 22:38)



Applications

- Time series predictions
 - Rainfall in a certain region
 - Spend on voice calls
- Classification!
- Data reduction
- Trend analysis
 - Linear or exponential
- Risk factor analysis
 - Factors contributing most to output



So by choosing an appropriate transformation of the inputs, I can fit any higher-order function so I could solve very complex problems using linear regression and so it is not really a weak method as you would think at first, first glance. Again, we will look at this in slightly more detail in the later lectures right and regression our prediction can be applied in a variety of places – one popular place is in time series prediction you could think about predicting rainfall in a certain region or how much you are going to spend on your telephone calls you could think of doing even classification using this, if you think of; you remember our encoding of +1 and -1 for the class labels. So you could think of +1 and -1 as the outputs right and then you can fit a regression line regression curve to that and if the output > 0 you would say this class is +1 its output < 0 you see the class is -1. So it could use the regression ideas to solve the classification problem and you could also do data reduction. So I really do not want to give you all the millions of data points that I have in my data set but what I would do is essentially fit the curve to that and then give you just the coefficients of the curve right.

And more often than not that is sufficient for us to get a sense of the data and that brings us to the next application I have listed there which is trend analysis so I am not really interested in; quite many times, I am not interested in the actual values of the data but more in the, the trends so for example I have a solution that I am trying to measure the running times off and I am not really

interested in the actual running time because with 37seconds to 38 seconds is not going to tell me much.

But I would really like to know if the running time scales linearly or exponentially with the size of the important all right so those kinds of analysis again can be done using regression. And in the last one here is again risk factor analysis like we had in classification and you can look at which are the factors that contribute most to the output so that brings us to the end of this module on supervised learning.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development

Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

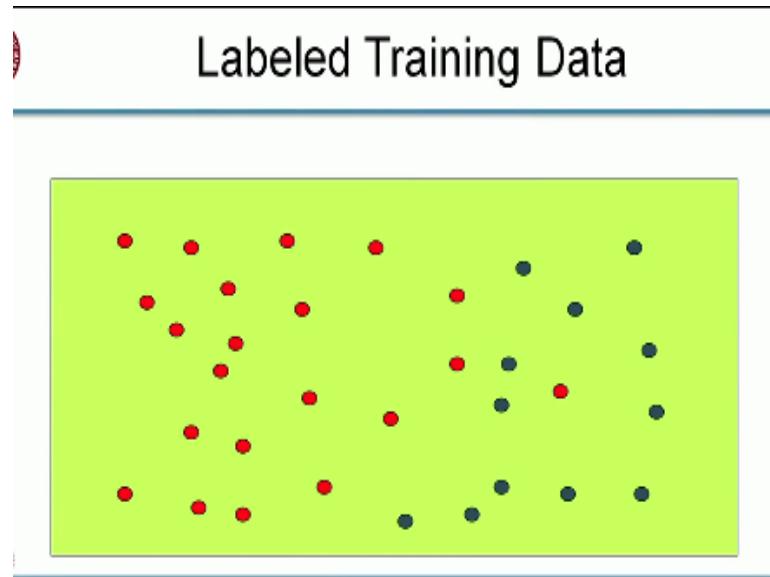
Lecture 3

**Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Unsupervised Learning

Hello and welcome to this module on introduction to unsupervised learning, right. So in supervised learning we looked at how you will handle training data that had labels on it.

(Refer Slide Time: 00:26)



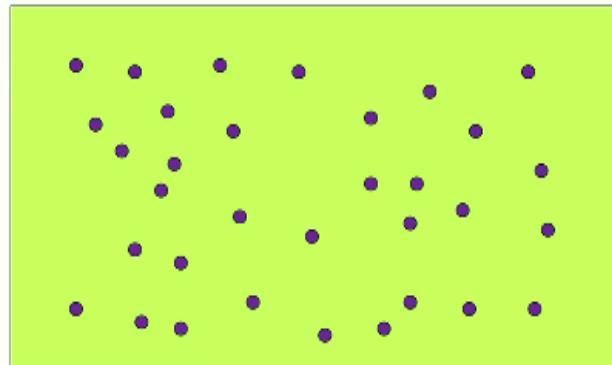
So this is this particular place this is a classification data set where red denotes one class and blue denotes the other class right.

(Refer Slide Time: 00:35)



Unlabelled Training Data

Clustering



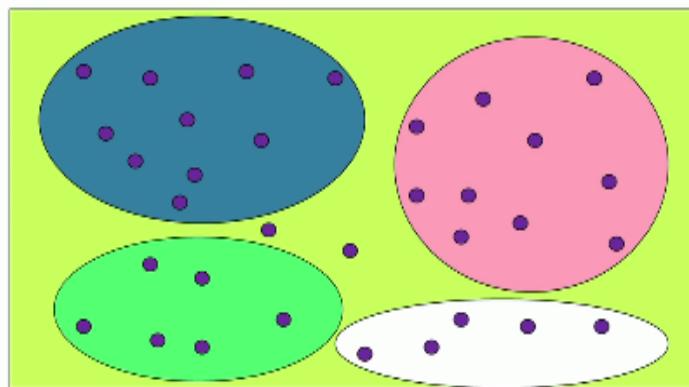
Faculty of Computer Application

34

And in unsupervised learning right so you basically have a lot of data that is given to you but they do not have any labels attached to them right. So we look at first at the problem of clustering where your goal is to find groups of coherent or cohesive data points in this input space right so here is an example of possible clusters.

(Refer Slide Time: 00:57)

Possible Clusters



So those set of data points could form a cluster right and again now those set of data points could form a cluster and again those and those. So there are like four clusters that we have identified in this in this setup. So one thing to note here is that even in something like clustering so I need to have some form of a bias right so in this case the bias that I am having is in the shape of the cluster so I am assuming that the clusters are all ellipsoids right and therefore you know I have been drawing a specific shape curves for representing the clusters.

And also note that not all data points need to fall into clusters and there are a couple of points there that do not fall into any of the clusters this is primarily a artifact of me assuming that they are ellipsoids but still there are points in the center is actually faraway from all the other points in the in the data set to be considered as what are known as outliers. So when you do clustering so there are two things so one is you are interested in finding cohesive groups of points and the second is you are also interested in finding data points that do not conform to the patterns in the input and these are known as outliers all right.

(Refer Slide Time: 02:23)



Applications

- Customer Data
 - Discover classes of customers
- Image pixels
 - Discover regions
- Words
 - Synonyms
- Documents
 - Topics



Image Courtesy: <http://is.brown.edu/~pfj/segment/>



And there are many many different ways in which you can accomplish clustering and we will look at a few in the course. And the applications are numerous right so here are a few representative ones. So one thing is to look at customer data right and try to discover classes of customers. So earlier we looked at in the supervised learning case we looked at is that a customer will buy a computer or will not buy a computer. As opposed to that we could just take all the customer data that you have and try to just group them into different kinds of customers who come to your shop and then you could do some kind of targeted promotions and different classes of customers right.

And this need not necessarily come with labels you know I am not going to tell you that okay this customer is class 1 that customer is class 2 you are just going to find out which of the customers are more similar with each other all right. And as the second application which we have illustrated here is that I could do clustering on image pixels so that you could discover different regions in the image and then you could do some segmentation based on that different region so for example here it have a picture of a beach scene and then you are able to figure out the clouds and the sand and the sea and the tree from the image. So that allows you to make more sense out of the image right.

Or you could do clustering on word usages right and you could discover synonyms and you could also do clustering on documents right and depending on which kind of documents are similar to each other; if I give you a collection of say 100,000 documents I might be able to figure out what are the different topics that are discussed in this collection of documents and many many ways in which you can use clustering.

(Refer Slide Time: 04:17)



Association Rule Mining

- Mining frequent patterns and rules
- Association rules: conditional dependencies
- Two stages
 - Find frequent patterns
 - Derive associations ($A \Rightarrow B$) from frequent patterns
- Find patterns in
 - Sequences (time series data, fault analysis)
 - Transactions (market basket data)
 - Graphs (social network analysis)



Rule mining: I should give you a side about the usage of the word mining here so many of you might have heard of the term data mining and more often than not the purported data mining tasks are essentially machine learning problems right so it could be classification regression and so on so forth. And the first problem that was essentially introduced as a mining problem and not as a learning problem was the one of mining frequent patterns and associations. And that is one of the reasons I call this Association rule mining as opposed to Association rule learning just to keep the historic connection intact right. So in Association rule mining we are interested in finding frequent patterns that occur in the input data and then we are looking at conditional dependencies among these patterns right.

So for example if A and B occur together often right then I could say something like if A happens then B will happen let us suppose that so you have customers that are coming to your shop and whenever customer A visits your shop custom B also tags along with him right, so the next time you find customary A somewhere in the shop so you can know that customer B is already there in the shop along with A.

Or with very high confidence you could say that B is also in the shop at some somewhere else may be not with A. But somewhere else in the shop all right, so these are the kinds of rules that we are looking at Association rules which are conditional dependencies – if A has come then B

is also there right and so the Association rule mining process usually goes in two stages so the first thing is we find all frequent patterns.

So A happens often so A is a customer that comes to my store often. And then I find that A and B are pairs of customers that come to my store often. So if I once I have that right A comes to my store often and A and B comes to my store often then I can derive associations from these kinds of frequent patterns. And also you could do this in the variety of different settings you could find sequences in time series data right and where you could look at triggers for certain events.

Or you could look at fault analysis right by looking at a sequence of events that happened and you can figure out which event occurs more often with a fault right or you could look at transactions data which is the most popular example given here is what is called Market Basket data. So you go to a shop and you buy a bunch of things together and you put them in your basket; so what is there in your basket right so this forms the transaction so you buy say eggs, milk and bread and so all of this go together in your basket.

And then you can find out what are the frequently occurring patterns in this purchase data and then you can make rules out of those or you could look at finding patterns and graphs that is typically used in social network analysis so which kind of interactions among entities happen often right so that is another question that is what we looking at right.

(Refer Slide Time: 07:31)



Mining Transactions

- Transaction is a collection of items bought together
 - A (sub)set of items is called an itemset
- Find frequent itemsets
- Itemset $A \Rightarrow$ Itemset B , if both A and $A \cup B$ are frequent itemsets.

So the most popular thing here is mining transactions so the most popular application here is mining transactions. And as I mentioned earlier transaction is a collection of items that are bought together right and so here is a little bit of terminology. A set or a subset of items is often called an item set in the Association rule mining community and so the first step that you have to do is find frequent item sets right.

And you can conclude that item set A , if it is frequent implies item set B if both A and $A \cup B$ or frequent item sets right so A and B are subset so $A \cup B$ is another subset so if both A and $A \cup B$ are frequent item sets then you can say that item set A implies item set B right. Like I mentioned earlier so there are many applications here so you could think of predicting co-occurrence of events.

(Refer Slide Time: 08:31)



Applications

- Predicting co-occurrence
- Market Basket analysis
- Time series analysis!
 - Trigger Events

And Market Basket analysis and Time Series analysis like I mentioned earlier you could think of trigger events or causes of Faults and so on so forth right so this brings us to the end of this module introducing unsupervised learning.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 4

Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Reinforcement Learning

Hi and welcome to this module that introduces reinforcement learning. So far we have been looking at popular models of machine learning such as supervised and unsupervised learning. In the supervised learning we looked at the classification in the regression problem and in unsupervised learning we looked at clustering and frequent pattern so on and so forth.

(Refer Slide Time: 00:32)



Learning to Control

- Popular models of machine learning
 - Supervised: Classification, Regression, etc.
 - Unsupervised: Clustering, Frequent patterns, etc.



And I have a question for you so. So how did you learn to cycle was it supervised learning or was it unsupervised learning right.

(Refer Slide Time: 00:45)



Learning to Control

- Popular models of machine learning
 - Supervised: Classification, Regression, etc.
 - Unsupervised: Clustering, Frequent patterns, etc.
- How did you learn to cycle?
 - Neither of the above
 - Trial and error!
 - Falling down hurts!



Introduction to Machine Learning

31



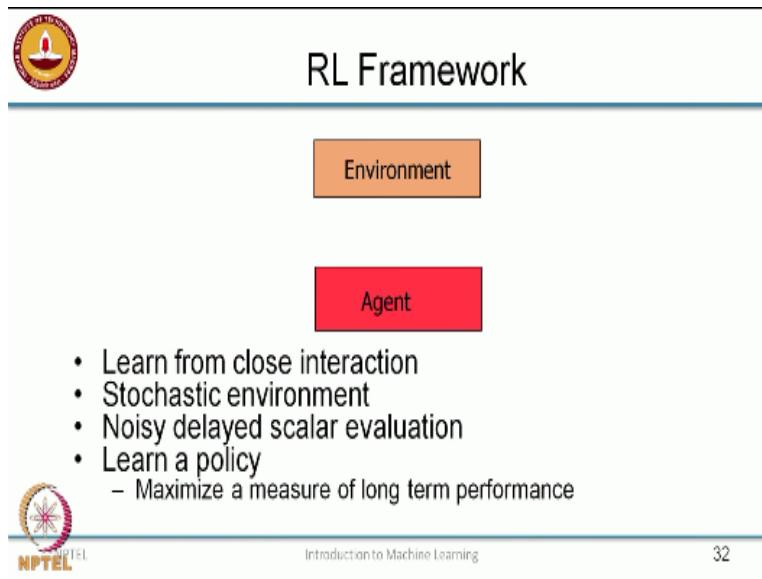
There is really no one telling you how you should cycle right. I mean how much how many pounds of pressure you should put with your left foot and what angle you should be leaning and so on so forth and if you think of it as a supervised learning problem that is how it should be and it was not completely unsupervised because it is not like you just watch people cycling and then figure out what the pattern that you should move in order to cycle and then you just magically got on a cycle and started cycling right.

So what was the crucial thing here? There is a trial and error component. So you have to get on the cycle. You had to try things out yourself before you could learn how to cycle in a acceptable manner right. So you have some kind of feedback it is not completely unsupervised right there will be somebody standing there and if you learn to cycle as a kid there was somebody standing there and clapping and saying hey great great good job. Come on go on go on or something like that and of course falling down hurts.

So you know that right so there is some amount of trial and error component and that is feedback that you are getting from the environment. So this kind of learning where you are learning to control a system through the trial and error and the minimal feedback is essentially what

reinforcement learning is. A mathematical formalization that captures this kind of learning is what we refer to as reinforcement learning right. So in the RL framework you typically think of a learning agent.

(Refer Slide Time: 02:12)



We already looked at learning agents, it could be the supervisor learner or it could be an unsupervised agent in this case you have a reinforcement learning agent that learns from close interaction with an environment that is outside the control of the agent right. The RL agent learns from close interaction with an environment. So what do I mean by close interaction here is that the agent senses the state in which the environment is right and it takes an action which it then applies to the environment which causes the state of the environment to change so thereby completing the interaction cycle. So the agent senses what is the state of the environment so if it is a cycle it is going to sense what angle is the cycle tilting in at what speed I am moving forward right and on what speed I am falling and so on so forth all this constitute the state of this system state of the environment. The agent is going to take an appropriate action which would be okay, lean to the right or push down with your right leg and then this action is then applied to the environment and that in turn changes the state of the environment. The agent learns from such close interaction with the environment and we typically assume that the environment is

stochastic so every time you take an action you are not going to get the same response from the environment so things could be slightly different right so there might be a small stone in the road that you did not have the last time you went over this place and therefore what was a smooth ride could suddenly turn bumpy and so on so forth. So you know that cycling always has some amount of noise and then you have to react to the noise.

So apart from this interaction the mathematical abstraction also assumes that there is some kind of an evaluation signal that is available from the environment that gives you some measure of how well you are performing in this particular task. If you remember we needed to have an evaluation measure for every task and we are assuming that this comes in the form of some kind of a scalar evaluation from the environment. It could be somebody clapping and saying that here you are doing well or it could be falling down and getting hurt; so all of this would be translated to some kind of a numeric scale.

And that's the mathematical abstraction that we make. So the goal of the agent is to learn a policy which is a kind of mapping from the states that you sense to the actions that you apply so as to maximize a measure of long-term performance so I'm not just interested in staying upright for the next two seconds but I am really interested in getting from point A to point B. So I have to make sure that I stay balanced throughout the entire duration of the ride. So this is the basic idea behind the reinforcement learning problem so each reinforcement algorithm the goal is to learn a policy that maximizes some measure of long-term performance right.

So there have been many successful applications of reinforcement learning so one of some of the marquee applications come from the domain of game playing like with many classical AI approaches.

(Refer Slide Time: 05:18)



Applications of RL

- Game playing
 - Backgammon – world's best player!
 - Atari games from scratch
- Autonomous agents
 - Robot navigation
- Adaptive control
 - Helicopter pilot!
- Combinatorial optimization
 - VLSI placement
- Intelligent Tutoring Systems

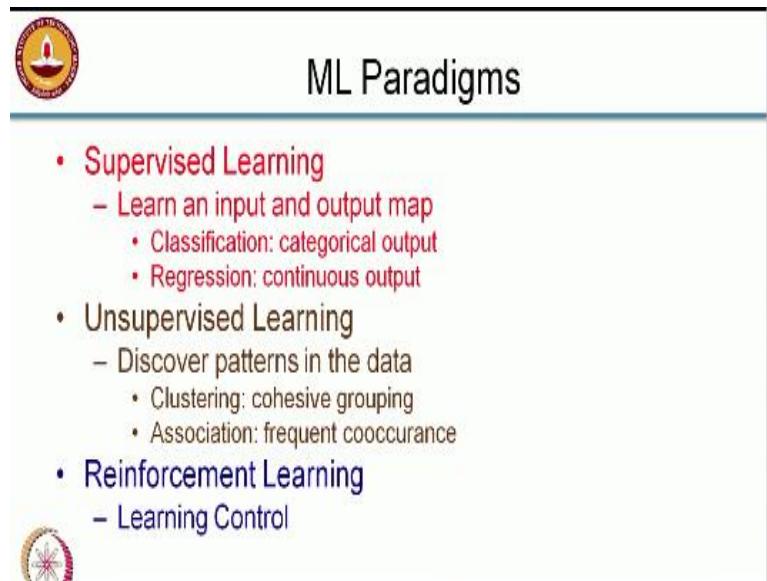


So backgammon is a board game based on die rolls. If people have not familiar with Backgammon, it is similar to the game Ludo but it's also got a rich history people have been playing it for several centuries and there are even world championships in backgammon. And the world's best player of backgammon is actually reinforcement learning engine. Ao notice that I did not qualify it saying the world's best computer player or anything so it was the world's best player and that managed to beat the world champion in backgammon over tournament.

More recent vintage so people have also gotten reinforcement learning agent to play at the video games, Atari video games from scratch. So the input to the system were like pixels from the screens right and the output from the system where the joystick controls and they managed to play this games from scratch right. And so in autonomous agents like in robots and other autonomous agents reinforcement learning is almost always the learning algorithm of choice and so in adaptive control and one of the again very prominent success stories of reinforcement learning is this helicopter pilot that was initially trained by Andrew Ng at Berkeley and later at Stanford where he could train the reinforcement learning algorithm to fly helicopter and at near human level competence. And there are other applications where people have looked at applying within combinatorial optimization problems solving really hard optimization problems and also in personalization and in adaptive systems like intelligent tutoring systems right.

And so to wrap up the set of introductory modules just wanted to recap the different machine learning paradigms that will be covering in the course. So the first one we will be looking at is supervised learning where we will be looking at learning in input-output map.

(Refer Slide Time: 07:31)



The slide has a blue header bar. On the left side of the header is a circular logo with a yellow center containing a red lamp (diya) and a green outer ring with text. To the right of the logo, the text "ML Paradigms" is written in white. Below the header, there is a white content area with a blue vertical decorative bar on the right edge. The content is organized into a bulleted list:

- **Supervised Learning**
 - Learn an input and output map
 - Classification: categorical output
 - Regression: continuous output
- **Unsupervised Learning**
 - Discover patterns in the data
 - Clustering: cohesive grouping
 - Association: frequent cooccurrence
- **Reinforcement Learning**
 - Learning Control

A small decorative icon consisting of a stylized sunburst or flower shape with radiating lines and dots, located at the bottom left of the slide area.

And so the classes the tasks that we look here or classification where the outputs that we are looking to predict or categorical outputs like yes or no or blue or red or buy a computer or not buy a computer. And the second supervised learning problem we look at is a regression where the output is continuous output. And the second class of problems we look at are unsupervised learning problems where we are interested in discovering patterns in the data.

Not necessarily in predicting a specific output and the canonical task we look at here are clustering where we are interested in finding cohesive groups in the data and also Association rules where we are interested in finding frequently occurring patterns right and the third paradigm which we will spend very little time on is reinforcement learning where you are interested in learning control or learning to control a system based on minimal feedback so from the next module onwards we will start looking at taking a little bit more mathematically rigorous look at machine learning.

IIT Madras Production

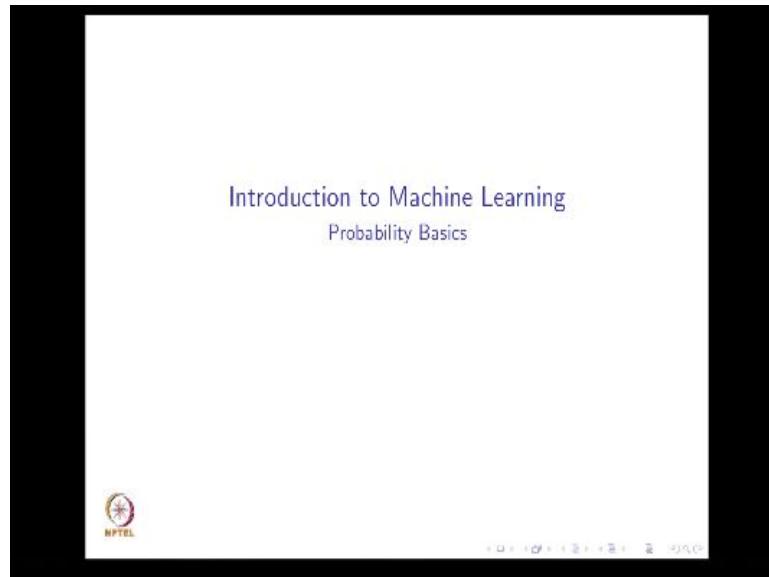
Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL
NPTEL ONLINE CERTIFICATION COURSE

(Refer Slide Time: 00:18)



Hello and welcome to the first tutorial in the introduction of machine learning course. My name is Priyatosh. I am one of the teaching assistants for this course.

In this tutorial, we will be looking at some of the basics of probability theory. Before we start let us discuss the objectives of this tutorial. The aim here is not to teach the concepts of probability theory in any great detail. Instead we will just be providing a high-level overview of the concepts that will be encountered later on in the course. The idea here is that for those of you who have done a course in probability theory or are otherwise familiar with the content this tutorial should act as a refresher. For others who may find some of the concepts unfamiliar, we recommend that you go back and prepare those concepts from say an introductory textbook or any other resource so that when you encounter those concepts later on in the course you should be comfortable with them.

(Refer Slide Time: 01:13)

Sample Space

Sample Space: The set of all possible outcomes of an experiment is called the sample space and is denoted by Ω . Individual elements are denoted by ω and are termed elementary outcomes.

Examples:

- ▶ (Finite) A single roll of an ordinary die. Here, $\Omega = \{1, 2, 3, 4, 5, 6\}$.
- ▶ (Countable) Infinite number of coin tosses in order to study, say, the number of tosses before 5 consecutive heads are observed. Here, $\Omega = \{H, T\}^\infty$.
- ▶ (Uncountable) Speed of a vehicle measured with infinite precision. Here, $\Omega = \mathbb{R}$.



Okay to start this tutorial we look at the definitions of some of the fundamental concepts. The first one to consider is that of the sample space. The set of all possible outcomes of an experiment is called the sample space and is denoted by Ω individual elements are denoted by ω and are termed elementary outcomes. Let us consider some examples in the first example the experiment consists of rolling an ordinary die: the sample space here is the set of numbers between one and six each individual element here represents one of the six possible outcomes of rolling a die.

Note that in this example the sample space is finite. In the second example the experiment consists of tossing a coin repeatedly until the specified condition is observed. Here we are looking to observe five consecutive heads before terminating the experiment. The sample space here is countable infinite. The individual elements are represented using a sequence of the H's and T's where H and T stand for heads and tails respectively. In the final example the experiment consists of measuring the speed of a vehicle with infinite precision. Assuming that the vehicle speeds can be negative the sample space is clearly the set of real numbers. Here we observe that the sample space can be uncountable.

(Refer Slide Time: 02:37)

Event

Event: An event is any collection of possible outcomes of an experiment, that is, any subset of Ω .

In most experiments we are generally more interested in observing the occurrence of particular events rather than the elementary outcomes. For example, on rolling a die, we may be interested in observing whether the outcome was even (event $E = \{2, 4, 6\}$) or odd (event $O = \{1, 3, 5\}$).



Navigation icons: back, forward, search, etc.

The next concept we look at is that of an event. An event is any collection of possible outcomes of an experiment that is any subset of the sample space. The reason why events are important to us is because in general when we conduct an experiment we are not really that interested in the elementary outcomes rather we are more interested in some subsets of the elementary outcomes.

For example on rolling a die, we might be interested in observing whether the outcome was even or odd. So for example on a specific role of a die let us say we observe that the outcome was odd. In this scenario, whether the outcome was actually a one or a three or a five is not as important to us as the fact that it was odd. Since we are considering sets in terms of sample spaces and events we will quickly go to the basic set theory notations. As usual capital letters indicate sets and small letters indicate set elements.

We first look at the subset relation. For all X ,

$$\begin{aligned}
 A \subseteq B &\Leftrightarrow x \in A \Rightarrow x \in B \\
 A = B &\Leftrightarrow A \subseteq B \text{ and } B \subseteq A \\
 A \cup B &= \{x : x \in A \text{ or } x \in B\} \\
 A \cap B &= \{x : x \in A \text{ and } x \in B\} \\
 A^c &= \{x : x \notin A\}
 \end{aligned}$$

In our case the universal set is essentially the sample space.

(Refer Slide Time: 04:28)

Properties of Set Operations

Commutativity

$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$

Associativity

$$A \cup (B \cup C) = (A \cup B) \cup C$$
$$A \cap (B \cap C) = (A \cap B) \cap C$$

Distributivity

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

DeMorgan's Laws



$$(A \cup B)^c = A^c \cap B^c$$
$$(A \cap B)^c = A^c \cup B^c$$

This slide lists out the different properties of set operations such as commutativity, associativity, and distributivity which you should all be familiar with. It also lists out the De Morgan's laws which can be very useful. According to the De Morgan's laws,

$$(A \cup B)^c = A^c \cap B^c$$
$$(A \cap B)^c = A^c \cup B^c$$

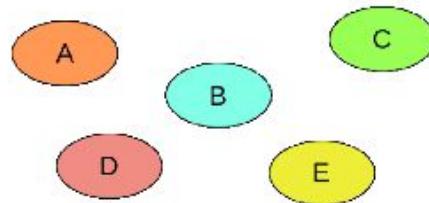
The De Morgan's laws presented here are for two sets. They can easily be extended for more than two sets.

(Refer Slide Time: 05:05)

Disjoint Events

Two events A and B are disjoint (or mutually exclusive) if $A \cap B = \emptyset$.

A sequence of events A_1, A_2, A_3, \dots are pair-wise disjoint if $A_i \cap A_j = \emptyset$ for all $i \neq j$.



NPTEL - National Programme on Technology Enhanced Learning

Coming back to events two events A and B are said to be disjoint or mutually exclusive if the intersection of two sets is empty. Extending this concept to multiple sets we say that a sequence of events A_1, A_2, A_3 and so on are pair wise disjoint if

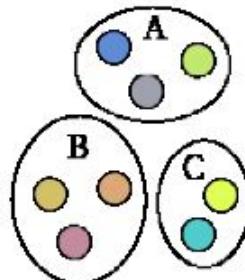
$$A_i \cap A_j = \emptyset \forall i \neq j$$

In the example below if each of the letters represents an event, then the sequence of events A through E are pair wise disjoint since the intersection of any pair is empty.

(Refer Slide Time: 05:39)

Partition

If A_1, A_2, \dots are pair-wise disjoint and $\bigcup_{i=1}^{\infty} A_i = \Omega$, then the collection A_1, A_2, \dots forms a partition of Ω .



NPTEL | NPTEL | NPTEL | NPTEL | NPTEL | NPTEL

If events A_1, A_2, A_3 and so on are pair wise disjoint and the union of the sequence of events gives rise to the sample space, then the collection A_1, A_2 and so on is said to form a partition of the sample space Ω . This is illustrated in the figure below.

(Refer Slide Time: 06:00)

Sigma Algebra

Given a sample space Ω , a σ -algebra is a collection \mathcal{F} of subsets of Ω , with the following properties:

- (a) $\Phi \in \mathcal{F}$.
- (b) If $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$.
- (c) If $A_i \in \mathcal{F}$ for every $i \in \mathbb{N}$, then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$.

A set A that belongs to \mathcal{F} is called an \mathcal{F} -measurable set (event).

Example: Consider $\Omega = \{1, 2, 3\}$.

$$\mathcal{F}_1 = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

$$\mathcal{F}_2 = \{\emptyset, \{1, 2, 3\}\}.$$



NPTEL | NPTEL | NPTEL | NPTEL | NPTEL | NPTEL

Next we come to the concept of a σ -algebra. Given a sample space Ω a σ -algebra is a collection \mathcal{F} of subsets of the sample space with the following properties.

- (a) $\emptyset \in F$
- (b) If $A \in F$, then $A^c \in F$
- (c) If $A_i \in F \forall i \in N$, then $\bigcup_{i=1}^{\infty} A_i \in F$

A set A that belongs to F is called an F measurable set. This is what we naturally understand as an event. So going back to the third property what this essentially says is that if there are a number of events which belong in the σ -algebra then the countable union of these events also belongs in the σ -algebra. Let us consider an example; consider Ω equals to 1, 2, 3 this is our sample space with this sample space we can construct a number of different σ -algebra. Here the σ -algebra F_1 is essentially the power set of the sample space. All possible events are present in the first σ -algebra.

However if we look at F_2 , in this case there are only two events the null set or the sample space itself. You should verify that for both F_1 and F_2 , all three properties listed above are satisfied. Now that we know what a σ -algebra is let us try and understand how this concept is useful. First of all for any Ω countable or uncountable the power set is always a σ -algebra for example for the sample space comprising of two elements H, T a feasible σ -algebra is the power set. This is not the only feasible σ -algebra, as we have seen in the previous example but always the power set will give you up feasible σ -algebra

(Refer Slide Time: 07:39)

Sample Space Size Considerations

For any Ω (countable or uncountable) 2^Ω is always a σ -algebra.

For example, for $\Omega = \{H, T\}$, a feasible σ -algebra is the power set, i.e., $\mathcal{F} = \{\emptyset, \{H\}, \{T\}, \{H, T\}\}$.

However, if Ω is uncountable, then probabilities cannot be assigned to every subset of 2^Ω .



Navigation icons for a presentation slide, including arrows for navigation and symbols for search and refresh.

However, if Ω is uncountable then probabilities cannot be assigned to every subset of the power set. This is the crucial point which is why we need the concept of σ -algebras. So just to recap if the sample space is finite or countable then we can kind of ignore the concept of σ -algebra because in such a scenario we can consider all possible events that is the power set of the sample space and meaningfully apply probabilities to each of these events.

However this cannot be done when the sample space is uncountable that is if Ω is uncountable then probabilities cannot be assigned to every subset of 2^Ω . This is where the concept of σ -algebra shows its use. When we have an experiment in which the sample space is uncountable.

For example let us say the sample space is the set of real numbers in such a scenario we have to identify the events which are of importance to us and use this along with the three properties listed in the previous slide to construct a σ -algebra and probabilities will then be assigned to the collection of sets in the σ -algebra.

(Refer Slide Time: 09:32)

Probability Measure & Probability Space

A probability measure \mathcal{P} on (Ω, \mathcal{F}) is a function $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ satisfying

- (a) $\mathcal{P}(\emptyset) = 0, \quad \mathcal{P}(\Omega) = 1;$
- (b) if A_1, A_2, \dots is a collection of pair-wise disjoint members of \mathcal{F} , then

$$\mathcal{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathcal{P}(A_i)$$

The triple $(\Omega, \mathcal{F}, \mathcal{P})$, comprising a set Ω , a σ -algebra \mathcal{F} of subsets of Ω , and a probability measure \mathcal{P} on (Ω, \mathcal{F}) , is called a **probability space**.



Navigation icons: back, forward, search, etc.

Next we look at the important concepts of probability measure in probability space. A probability measure P on a specific sample space Ω and σ -algebra F is a function from F to the closed interval 0 comma 1 which satisfies the following properties:

$$(a) P(\emptyset) = 0, \quad P(\Omega) = 1$$

- (b) if A_1, A_2 and so on is a collection of pair wise disjoint members of F then

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

Note that this holds because the sequence A_1, A_2 is pair wise disjoint. The triple (Ω, F, P) comprising a sample space Ω , a σ -algebra F which are subsets of Ω and a probability measure P defined on Ω, F : this is called a probability space. For every probability problem that we come across there exists a probability space comprising of the triple (Ω, F, P) .

(Refer Slide Time: 10:47)

Example

Consider a simple experiment of rolling an ordinary die in which we want to identify whether the outcome results in a prime number or not.

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

$$\mathcal{F} = [\emptyset, \{1, 4, 6\}, \{2, 3, 5\}, \{1, 2, 3, 4, 5, 6\}]$$

$$P : \mathcal{F} \rightarrow [0, 1]$$

- $P(\emptyset) = 0$
- $P(\{1, 4, 6\}) = 0.5$
- $P(\{2, 3, 5\}) = 0.5$
- $P(\Omega) = 1$



Even though we may not always explicitly take into consideration this probability space when you solve the problem, it should always remain in the back of our heads. Let us now look at an example where we do consider the probability space involved in the problem consider a simple experiment of rolling an ordinary in which we want to identify whether the outcomes ends in a prime number or not. The first thing to consider is the sample space. Since there are only six possible outcomes in our experiment, the sample space here consists of the numbers between one to six. Next we look at the σ -algebra. Note that since the sample space is finite you might as well consider all possible events that is the power set of the sample space. However note that the problem dictates that we are only interested in two possible events that is whether a number whether the outcome is prime or not.

Thus restricting ourselves to these two events we can construct a simpler σ -algebra here we have two events which correspond to the events we are interested in and the remaining two events follow from the properties which is σ -algebra has to follow. Please verify that the σ -algebra listed here does actually satisfy the three properties that we had discussed about. The final component is the probability measure the probability measure assigns a value between zero and one that is the probability value to each of the components of the σ -algebra. Here the values listed assumes that the die is fair in the sense that the probability of each face is equal to 1 by 6.

Having covered some of the very basics of probability in the next few slides, we look at some rules which allow us to estimate probability values.

(Refer Slide Time: 12:24)

Bonferroni's Inequality

$$P(A \cap B) > P(A) - P(B) - 1$$

General form:

$$P(\bigcap_{i=1}^n A_i) \geq \sum_{i=1}^n P(A_i) - (n - 1)$$

Gives a lower bound on the intersection probability which is useful when this probability is hard to calculate.

Only useful if the probabilities of individual events are sufficiently large.



The first thing we look at is known as the Bonferroni's inequality. According to this inequality,

$$P(A \cap B) > P(A) - P(B) - 1$$

The general form of this inequality is also listed. What this inequality allows us to do is to give a lower bound on the intersection probability.

This is useful when the intersection probability is hard to calculate however if you notice the right hand side of the inequality we should observe that this result is only meaningful when the individual probabilities are sufficiently large for example if the probability of A and the probability of B both values are very small when this -1 term dominates and the result doesn't make much sense.

(Refer Slide Time: 13:17)

Boole's Inequality

$$P(\cup_{i=1}^{\infty} A_i) \leq \sum_{i=1}^{\infty} P(A_i), \text{ for any sets } A_1, A_2, \dots$$

Gives a useful upper bound for the probability of the union of events.



According to the Boole's inequality for any sets A_1, A_2 and so on

$$P(\cup_{i=1}^{\infty} A_i) \leq \sum_{i=1}^{\infty} P(A_i)$$

Clearly this gives us a useful upper bound for the probability of the union events. Notice that this equality will only hold when these sets are pair wise disjoint.

(Refer Slide Time: 13:43)

Conditional Probability

Given two events A and B , if $P(B) > 0$, then the conditional probability that A occurs given that B occurs is defined to be

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Essentially, since event B has occurred, it becomes the new sample space.

Conditional probabilities are useful when reasoning in the sense that once we have observed some event, our beliefs or predictions of related events can be updated/improved.



NPTEL - National Programme on Technology Enhanced Learning

Next we look at conditional property given two events A and B. If $P(B) > 0$, then

$$P(A|B) = P\left(\frac{(A \cap B)}{P}(B)\right)$$

the conditional probability that A occurs given that B occurs is defined to be probability of A given B is equal to probability of A intersection B by probability of B.

Essentially since event B has occurred it becomes a new sample space and now the probability of A is accordingly modified. Conditional probabilities are very useful when reasoning in the sense that once we have observed some event our beliefs or predictions of related events can be updated on or improved.

(Refer Slide Time: 14:13)

is we want the probability that both tosses resulted heads given that the first toss resulted in a heads. Does this change the problem?

(Refer Slide Time: 16:07)

Bayes' Rule

We have:

$$\begin{aligned} P(A|B) &= \frac{P(A \cap B)}{P(B)} \\ P(A \cap B) &= P(A|B)P(B) \\ P(A \cap B) &= P(B|A)P(A) \\ P(A|B)P(B) &= P(B|A)P(A) \\ P(A|B) &= \frac{P(B|A)P(A)}{P(B)} \quad (\text{Bayes' Rule}) \end{aligned}$$



NPTEL

Next we come to a very important theorem called the Bayes' theorem or the Bayes' rule. We start with the equation for the conditional probability,

$$P(A|B) = P\left(\frac{A \cap B}{B}\right)$$

Rearranging we have,

$$P(A \cap B) = P(A|B) \cdot P(B)$$

Now instead of starting with probability of A given B, if I started with probability of B given A, you would have got

$$P(A \cap B) = P(B|A) \cdot P(A)$$

These two right hand sides can be equated to get

$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

Now taking this probability of B to the right hand side we get,

$$P(A) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

This is what is known as the Bayes' rule. Note that what it essentially says is if I want to find the probability of A given that B happened, I can use the information of probability of B given A along with the knowledge of probability of A and probability of B to get this value.

(Refer Slide Time: 17:29)

Bayes' Rule

Let A_1, A_2, \dots be a partition of the sample space, and let B be any subset of the sample space. Then, for each $i = 1, 2, \dots$,

$$P(A_i | B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^n P(B|A_j)P(A_j)}$$

Bayes' rule is important in that it allows us to compute the conditional probability $P(A|B)$ from the "inverse" conditional probability $P(B|A)$.



NPTEL - National Programme on Technology Enhanced Learning

As you will see this is a very important formula. Here we again present the Bayes' rule in an expanded form where A_1, A_2 and so on form partition of the sample space. As mentioned Bayes' rule is important in that it allows us to compute the conditional probability of A given B from the inverse conditional probability, the probability of B given A.

(Refer Slide Time: 17:50)

Example

Q. To answer a multiple choice question, a student may either know the answer or may guess it. Assume that with probability p the student knows the answer to a question, and with probability q , the student guesses the right answer to a question she does not know. What is the probability that for a question the student answers correctly, she actually knew the answer to the question?

Sol. Let K be the event that the student knows the question, and C be the event that the student answers the question correctly.

We have $P(K) = p$, $P(\neg K) = 1 - p$, $P(C|K) = 1$, $P(C|\neg K) = q$

$$P(K|C) = \frac{P(C|K)P(K)}{P(C)} = \frac{P(C|K)P(K)}{P(K)P(C|K) + P(\neg K)P(C|\neg K)}$$



Let us look at a problem in which the Bayes' rule is applicable. To answer a multiple choice question or student may either know the answer or may guess it. Assume that with probability P the student knows the answer to a question and the probability Q the student guesses the right answer to a question she does not know. What is the probability that for a question the student answers correctly she actually knew the answer to the question again pause the video here and try solving the problem yourself.

Okay let us first assume that K is the event that the student knows the question and let C be the event that the student answers the question correctly. Now from the question we can gather the following information the probability that the student knows the question is P hence the probability that the student does not know the question is goes to 1 minus P . The probability that the student answers the question correctly given that she knows the question is equals to 1 because if she knows the question she will definitely answer it correctly. Finally the probability that the student answers the question correctly given that she makes a guess that is she does not know the question is Q we are interested in the probability of the student knowing the question given that she has answered it correctly. Applying Bayes' rule we have

$$P(K|C) = \frac{P(C|K) \cdot P(K)}{P(C)}$$

The probability of answering the question correctly can be expanded in the denominator to consider the two situations: probability of answering the question correctly given that the student knows the question and probability of answering the question correctly when the student does not know question. Now using the values which mean have gathered from the question we can arrive at the answer,

$$P(K|C) = \frac{P}{P+Q \cdot (1-P)}$$

Note here that the Bayes' rule is essential to solve this problem because while from the question itself we have a handle on this value probability of C given K there is no direct way to arrive at the value of probably of K given C.

(Refer Slide Time: 20:13)

Independent Events

Two events, A and B , are said to be independent if

$$P(A \cap B) = P(A)P(B)$$

More generally, a family $A_i : i \in I$ is called independent if

$$P(\bigcap_{i \in J} A_i) = \prod_{i \in J} P(A_i)$$

for all finite subsets J of I .

From the above, it should be clear that pair-wise independence does not imply independence.



Two events A and B are set to be independent if probability of A intersection B is equal to probability of A into probability of B . More generally a family of events A_i where i is an element of the integers is called independent if probability of some subset of the events A_i is equal to the

product of the probabilities of those events. Essentially what, what we are trying to say here is that if you have a family of events A_i then the independence condition holds only if for any subset of those events, this condition holds. From this should be clear that pair wise independence does not imply independence that is pair wise independence is a weaker condition.

(Refer Slide Time: 21:03)

Conditional Independence

Let A , B , and C be three events with $P(C) > 0$. The events A and B are called conditionally independent given C if

$$P(A \cap B|C) = P(A|C)P(B|C)$$

or equivalently

$$P(A|B \cap C) = P(A|C)$$

Example: Assume that admission into the M.Tech. programme at IITM & IITB is based solely on candidate's GATE score. Then

$$P(IITM|IITB \cap GATE) = P(IITM|GATE)$$



Navigation icons: back, forward, search, etc.

Explaining the notion of independence of events we can also consider conditional independence let A , B and C be three events with probability of C strictly greater than zero. The events A and B are called conditionally independent given C if

$$P(A \cap B|C) = P(A|C) \cdot P(B|C)$$

Equivalently the events A and B are conditionally independent given C if

$$P(A|B \cap C) = P(A|C)$$

This latter condition is quite informative. What it says is that the probability of A calculated after knowing the occurrence of event C is same as the probability of A calculated after having

knowledge of occurrence of both events B and C thus observing the occurrence or non-occurrence of B does not provide any extra information. And thus we can conclude that the events A and B are conditions independent given C.

Let us consider an example assume that admission into the M.Tech program at IIT Madras in IIT Bombay is based solely on candidates GATE score. Then probability of admission into IIT Madras given knowledge of the candidate's admission status in IIT Bombay as well as the candidate's GATE score is equivalent to the probability calculated simply knowing the candidate's GATE score. Thus knowing the status of the candidates admission into IIT Bombay does not provide any extra information. Hence since the condition is satisfied we can say that admission into the program at IIT Madras and admission into the program at IIT Bombay are independent events given knowledge of the candidates GATE score.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

(Refer Slide Time: 00:16)

Random Variable

A random variable is a function $X : \Omega \rightarrow \mathbb{R}$, i.e., it is a function from the sample space to the real numbers.

Examples:

- ▶ The sum of outcomes on rolling 3 dice.
- ▶ The number of heads observed when tossing a fair coin 3 times.



One of the important concepts in probability theory is that of the random variable.

A random variable is a variable whose value is subjected to variations. That is, a random variable can take on a set of possible different values each with an associated probability. Mathematically a random variable is a function from the sample space to the real numbers $X : \Omega \rightarrow \mathbb{R}$.

Let us consider some examples. Suppose we conduct an experiment in which we roll three dice and are interested in the sum of the outcomes, the sum of fives, say. It can be observed if two of the dice show up two each and the other die shows up as one. Alternatively the sum of five can also be observed if one die shows up as three and the other two dice show up one each. Since we are interested in only the sum and not the individual results of the dice rolls we can define a random variable which maps the elementary outcomes that is the outcomes of each die roll to the sum of the three rolls.

Similarly in the next example we can define a random variable which counts the number of heads observed when tossing a fair coin three times. Note that in this example the random

variable can take values between 0 and 3 whereas in the previous example the range of the random variable is between 3 and 18 corresponding to all dice showing up one and all dice showing up six.

(Refer Slide Time: 01:44)

Induced Probability Function

Consider the previous example experiment of tossing a fair coin 3 times. Let X be the number of heads obtained in the three tosses. Enumerating the elementary outcomes, we observe the value of X as

ω	HHH	HHT	HTH	THH	TTH	THT	HTT	TTT
$X(\omega)$	3	2	2	2	1	1	1	0

Instead of using the probability measure defined on the elementary outcomes or events, we would ideally like to measure the probability of the random variable taking on values in its range.

x	0	1	2	3
$P_X(X = x)$	1/8	3/8	3/8	1/8

Consider the previous example experiment of tossing a fair coin 3 times. Let X be the number of heads obtained in the three tosses, that is X is a random variable which maps each elementary outcome to a real number representing the number of heads observed in that outcome as it is shown in the first table. The first row lists out each elementary outcome and the second row lists out the corresponding real number value to which that elementary outcome is mapped that is the number of heads observed in that outcome.

Now instead of using the probability measure defined on the elementary outcomes or events we would ideally like to measure the probability of the random variable taking on values in its range. What we are trying to say here is that when we define probability measure we were associating each event that is the subset of the sample space with a probability measure. When we consider random variables the events correspond to different subsets of the sample space which mapped to different values of the random variable.

This is illustrated in the second table, the first row lists out the different values that the random variable X can take and the second row lists out the corresponding probability values assuming that the coin toss is a fair coin. This table describes the notion of the induced probability function which maps each possible value of the random variable to its associated probability value. For example, in the table the probability of the random variable taking on the value of 1 is given as 3/8. Since there are three elementary outcomes in which only one head is observed, each of these elementary outcomes has a probability of 1/8.

(Refer Slide Time: 03:30)

Induced Probability Function

Let $\Omega = \{\omega_1, \omega_2, \dots\}$ be a sample space and P be a probability measure (function).

Let X be a random variable with range $\mathcal{X} = \{x_1, x_2, \dots, x_m\}$.

We define the induced probability function P_X on \mathcal{X} as

$$P_X(X = x_i) = P(\{\omega_j \in \Omega : X(\omega_j) = x_i\})$$



From the previous example we can define the concept of the induced probability function. Let Ω be a sample space and P be a probability measure. Let X be a random variable which takes values with range $X = \{x_1, x_2, \dots, x_m\}$. The induced probability function P_x on X is defined as

$P_x(X = x_i) = P(\{\omega_j \in \Omega : X(\omega_j) = x_i\})$ or it equals to the probability of the event comprising of the elementary outcomes ω_j such that the random variable X map ω_j to the value x_i .

(Refer Slide Time: 04:10)

Cumulative Distribution Function

The cumulative distribution function or cdf of a random variable X, denoted by $F_X(x)$, is defined by

$$F_X(x) = P(X \leq x), \text{ for all } x$$

Example:

x	($-\infty, 0]$	($-\infty, 1]$	($-\infty, 2]$	($-\infty, 3]$	($-\infty, \infty$)
$F_X(x)$	1/8	1/2	7/8	1	1



Navigation icons: back, forward, search, etc.

The cumulative distribution function or CDF of a random variable X denoted by $F_x(x)$ is defined as $F_x(x) = P_x(X \leq x) \forall x$

For example, going back to the previous random variable which counts the number of heads observed in three tosses of a fair coin. The following table shows the intervals corresponding to the different values of the random variable X along with the corresponding values of the cumulative distribution function.

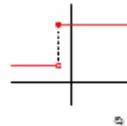
For example, $F_x(x) = F_x(1) = 1/2$, because the probability that the random variable X has a value of 1 is 3/8, and the probability that the random variable X has a value of 0 is 1/8. And therefore, the probability that the random variable X takes on a value with less than or equal to 1 is $1/8 + 3/8 = 4/8$ or 1/2.

(Refer Slide Time: 05:28)

Properties of cdf

A function $F_X(x)$ is a cdf iff the following three conditions hold:

- ▶ (Monotonicity) If $x \leq y$, then $F_X(x) \leq F_X(y)$
- ▶ (Limiting values) $\lim_{x \rightarrow -\infty} F_X(x) = 0$ and $\lim_{x \rightarrow \infty} F_X(x) = 1$
- ▶ (Right-continuity) For every x , we have $\lim_{y \downarrow x} F_X(y) = F_X(x)$



A function is a valid cumulative distribution function only if it satisfies the following properties.

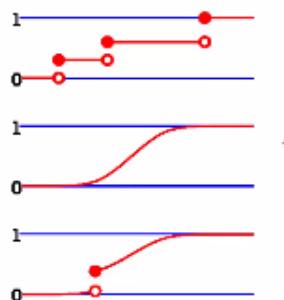
1. The first property simply states that the cumulative distribution function is a non decreasing function.
2. The second property specifies the limiting values, $\lim_{x \rightarrow -\infty} F_x(x) = 0$ and $\lim_{x \rightarrow \infty} F_x(x) = 1$
3. The third property specifies right continuity that is no jump occurs when the limit point is approached from the right and is also shown in the plot.

(Refer Slide Time: 06:07)

Continuous & Discrete Random Variables

A random variable X is continuous if $F_X(x)$ is a continuous function of x .

A random variable X is discrete if $F_X(x)$ is a step function of x .



A random variable X is continuous if its corresponding cumulative distribution function is a continuous function of X . This is shown in the second part of the diagram above.

A random variable X is discrete if its CDF is a step function of X this is shown in the first part of the diagram.

The third part of the diagram shows the cumulative distribution function for a random variable which has both continuous and discrete parts.

(Refer Slide Time: 06:33)

Probability Mass Function

The probability mass function or pmf of a discrete random variable X is given by

$$f_X(x) = P(X = x), \text{ for all } x$$

Example: For a geometric random variable X with parameter p ,

$$f_X(x) = \begin{cases} (1 - p)^{x-1} p & \text{for } x = 1, 2, \dots \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The probability mass function or PMF of a discrete random variable X is given by

$$f_x(x) = P(X = x), \forall x$$

Thus for a discrete random variable the probability mass function of that variable gives the probability that the random variable is equal to some value.

For example, for a geometric random variable X with parameter p the PMF is given as

$$f_x(x) = (1 - p)^{x-1} p \text{ for } x = 1, 2, \dots$$

And for other values of x the PMF = 0.

A function is a valid probability mass function if it satisfies the following two properties.

1. First of all the function must be non-negative.

2. Secondly $\sum_x f_x(x) = 1$

Probability Density Function

The probability density function or pdf of a continuous random variable is the function $f_X(x)$ which satisfies

$$F_X(x) = \int_{-\infty}^x f_X(t)dt, \text{ for all } x$$

Properties:

- ▶ $f_X(x) > 0$, for all x
- ▶ $\int_{-\infty}^{\infty} f_X(x)dx = 1$



Navigation icons: back, forward, search, etc.

For continuous random variables we consider the probability density function. The probability density function or PDF over a continuous random variable is the function $F_x(x)$ which satisfies the following.

$$F_x(x) = \int_{-\infty}^x f_x(t)dt, \forall x$$

Similar to the PMF, the probability density function should also satisfy the following properties.

1. First of all the probability density function should be non-negative for all values of x or $f_x(x) \geq 0, \forall x$.
2. $\int_{-\infty}^{\infty} f_x(x)dx = 1$

(Refer Slide Time: 08:21)

Expectation

The expected value or mean of a random variable X , denoted by $E[X]$, is given by

$$E[X] = \int_{-\infty}^{\infty} xf_X(x)dx \text{ (continuous RV)}$$

$$E[X] = \sum_{x:P(x)>0} xf_X(x) = \sum_{x:P(x)>0} xP(X=x) \text{ (discrete RV)}$$



Let us now look at expectations of random variables. The expected value or mean of a random

variable X denoted by $E[X]$ is given by integral $E[X] = \int_{-\infty}^{\infty} xf_x(x)dx$ (continuous RV).

Note that $f_x(x)$ here is the probability density function associated with random variable X . This definition holds when X is a continuous random variable. In case that X is a discrete random variable we use the following definition. $E[X] = \sum_{x:P(x)>0} xf_x(x) = \sum_{x:P(x)>0} xP(X=x)$ (discrete RV).

Here $f_x(x)$ is the probability mass function of the random variable X which essentially gives the associated probability for a particular value of the random variable thus leading to this definition.

(Refer Slide Time: 09:17)

Example

Q. Let the random variable X take values -2, -1, 1, 3 with probabilities $1/4, 1/8, 1/4, 3/8$ respectively. What is the expectation of the random variable $Y = X^2$?

Sol. The random variable Y takes on the values 1, 4, 9 with probabilities $3/8, 1/4, 3/8$ respectively.
Hence,

$$E(Y) = \sum_x xP(Y=x) = 1 \cdot \frac{3}{8} + 4 \cdot \frac{1}{4} + 9 \cdot \frac{3}{8} = \frac{19}{4}$$

Alternatively,

$$E(Y) = E(X^2) = \sum_x x^2 P(X=x) = 4 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} + 1 \cdot \frac{1}{4} + 9 \cdot \frac{3}{8} = \frac{19}{4}$$



Navigation icons: back, forward, search, etc.

Let us now look at an example in which we calculate expectations.

Problem: Let the random variable X take values -2, -1, 1 and 3 with probabilities $1/4, 1/8, 1/4, 3/8$ respectively. What is the expectation of the random variable $Y = X^2$?

Solution: So what we can do is we can calculate the values that the random variable Y takes along with associated probabilities, since we are aware of the relation between Y and X . Thus we have Y taking on the values 1, 4 and 9 with probabilities $3/8, 1/4$, and $3/8$ respectively. Given this information we can simply apply the formula for expectation and calculate the expectation on the random variable Y giving a result of $19/4$.

Another way to approach this problem is to directly use the relation $Y = X^2$ in calculating expectation.

$$E(Y) = E(X^2) = \sum_x x^2 P(X=x) = 4 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} + 1 \cdot \frac{1}{4} + 9 \cdot \frac{3}{8} = \frac{19}{4}$$

(Refer Slide Time: 10:54)

Properties of Expectations

Let X be a random variable and let a, b, c be constants. Then, for functions $g_1(X)$ and $g_2(X)$ whose expectations exist

- $E(ag_1(X) + bg_2(X) + c) = aEg_1(X) + bEg_2(X) + c$
- If $g_1(X) \geq 0$ for all x , then $Eg_1(X) \geq 0$
- If $g_1(X) \geq g_2(X)$ for all x , then $Eg_1(X) \geq Eg_2(X)$
- If $a < g_1(X) < b$, for all x , then $a < Eg_1(X) < b$



Let us now look at the properties of expectations.

Let X be a random variable and let a, b, c be constants. Then for functions $g_1(X)$ and $g_2(X)$ are functions of random variable X , such that their expectations exist that is they have finite expectations.

1. $E(ag_1(X) + bg_2(X) + c) = aEg_1(X) + bEg_2(X) + c$. This is called the linearity of expectations. There are actually a few things to note here first of all expectation of a constant is equal to the constant itself expectation of a constant times the random variable is equal to the constant into the expectation of the random variable and the expectation of the sum of two random variables can also be represented as the sum of the expectations of the two random variables. Note that here the two random variables need not be statistically independent.
2. According to the next property if a random variable is ≥ 0 at all points then the expectation of that random variable is also ≥ 0
3. Similarly if one random variable is $>$ another random variable at all points then the expectation of those random variables also follow the same constraint.
4. Finally if a random variable has values which lie between two constants then the expectation of that random variable will also lie between those two constants.

(Refer Slide Time: 12:34)

Moments

For each integer n , the n^{th} moment of X is

$$\mu'_n = E X^n$$

The n^{th} central moment of X is

$$\mu_n = E(X - \mu)^n$$



Let us now define moments. For each integer n the n^{th} moment of X is $\mu'_n = E X^n$. Also the n^{th} central moment of X is $\mu_n = E(X - \mu)^n$. So the difference between moment and central moment is that in central moment we subtract the random variable by the mean of the random variable or expected value. The two moments that find most common use are the first moment which is nothing but $\mu_1 = E[X]$ that is the mean of the random variable X and the second central moment which is $\mu_2 = E(X - \mu)^2$ which is the variance of the random variable X .

(Refer Slide Time: 13:23)

Variance

The variance of a random variable X is its second central moment,

$$VarX = E(X - \mu)^2 = E(X - EX)^2 = EX^2 - (EX)^2$$

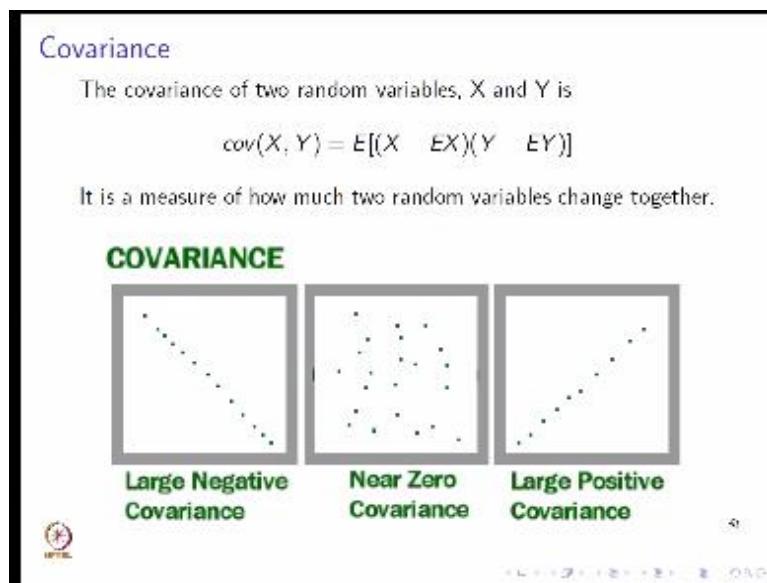
The positive square root of $VarX$ is the standard deviation of X .

Note: $Var(aX + b) = a^2 VarX$
where a, b are constants



Thus the variance of a random variable x is a second central moment. Variance of $X = E(X - \mu)^2$. Note that μ is just the first moment which can be replaced by $E[X]$. Thus we have $VarX = E(X - \mu)^2 = E(X - EX)^2 = EX^2 - (EX)^2$. Another very useful relation to remember is $Var(aX + b) = a^2VarX$ where a and b are constants.

(Refer Slide Time: 14:17)



The covariance of two random variables X and Y is $\text{cov}(X, Y) = E[(X - EX)(Y - EY)]$.

Remember that the variance of our random variable X is nothing but the second central moment thus the variance of a random variable measures the amount of separation in the values of the random variable when compared to the mean of the random variable. For covariance the calculation is done on a pair of random variables and it measures how much two random variables change together. Consider the diagram above. In the first part assume that the random variable X is on the x-axis and the random variable Y is on the y-axis. We note that as the value of X increases the value of Y seems to be decreasing thus for this relationship we will observe a large negative covariance. Similarly in the third part of the diagram we can see that as the variable value of variable X increases, so does the value of the variable y . Thus we see a large positive covariance. However in the middle diagram we cannot make any such statement because

as X increases there is no clear relationship as to how Y changes thus this kind of a relationship will give zero covariance.

Now from the diagram it should immediately be clear that covariance is a very important term in machine learning because we are often interested in predicting the value of one variable by looking at the value of another variable we will come to that in further classes.

(Refer Slide Time: 16:04)

Correlation

The correlation of two random variables, X and Y is

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}}$$

Note:

- ▶ For correlation to be defined, individual variances must be non-zero and finite
- ▶ $\rho(X, Y)$ lies between -1 and +1



Closely related to the concept of covariance is the concept of correlation. The correlation of two random variables X and Y is nothing but the covariance of the two random variables X and Y divided by the square root of the product of their individual variances.

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}}$$

Basically correlation is a normalized version of covariance. So the correlation will always be between -1 and 1 also since we used the variance of the individual random variables in the denominator for correlation to be defined, individual variances must be nonzero and finite.

(Refer Slide Time: 16:42)

Probability Distributions

Consider two variables X and Y , and suppose we know the corresponding probability mass functions f_X and f_Y .

Can we answer the following question:

$$P(X = x \text{ and } Y = y) = ?$$

5



Navigation icons

In the final part of this tutorial on probability theory we will talk about probability distributions and list out some of the more common distribution that you are going to encounter in the course. Before we proceed let us consider this question. Consider two variables X and Y and suppose we know the corresponding probability mass function f_x and f_y . Corresponding to the variables X and Y , can we answer the following question: $P(X=x \text{ and } Y=y)=?$

What is the probability that X takes a certain value small x and Y takes a certain value small y ? Think about this question. If you answered no then you're correct let us see why.

(Refer Slide Time: 17:25)

Joint Distributions

To capture the properties of two random variables X and Y , we use the joint PMF

$$f_{X,Y} : \mathbb{R}^2 \rightarrow [0, 1], \text{ defined by } f_{X,Y}(x, y) = P(X = x, Y = y)$$

48



Navigation icons: back, forward, search, etc.

Essentially what we were looking for in the previous question was the Joint Distribution which captures the properties of both the random variables the individual PMFs or PDFs. In case the random variables are continuous they capture the properties of the individual random variables only but miss out on how the two variables are related thus we define the joint PMF or PDF as $f_{X,Y}$.

(Refer Slide Time: 18:02)

Marginal Distributions

Suppose we are given the joint PMF

$$f_{X,Y}(x, y) = P(X = x, Y = y)$$

From this joint PMF, we can obtain the PMF's of the two random variables

$$\begin{aligned} f_X &= \sum_y f_{X,Y}(x, y) && (\text{marginal PMF of R.V. X}) \\ f_Y &= \sum_x f_{X,Y}(x, y) && (\text{marginal PMF of R.V. Y}) \end{aligned}$$



Navigation icons: back, forward, search, etc.

Suppose we are given the joint probability mass function of the random variables X and Y. What if we are interested in only the individual mass functions of either of the random variables. This can be obtained from the joint probability mass function by a process called marginalization. The individual probability mass function thus obtained is also referred to as a marginal probability mass function. Thus if you are interested in the marginal probability mass function of random variable X we can obtain this by summing the joint probability mass function overall values of Y. Similarly the marginal probability mass function of random variable Y can be obtained by summing the joint probability mass function over all values of X. Note that in case the random variables considered here are continuous we substitute summation by integration and PMF by PDF.

(Refer Slide Time: 19:02)

Conditional Distributions

Like joint distributions, we can also consider conditional distributions

$$f_{X|Y}(x|y) = P(X = x|Y = y)$$

Using conditional probability definition, we have

$$f_{X|Y}(x|y) = f_{X,Y}(x,y)/f_Y(y)$$

9

Note that the above conditional probability is undefined if $f_Y(y) = 0$.

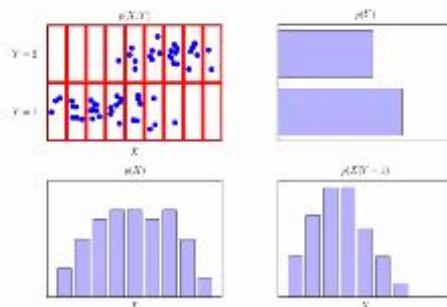


Like joint distributions we can also consider conditional distributions. For example here we have the conditional distribution $f_{X|Y}(x,y)$ which is the probability that the random variable X will take on some value small x given that the random variable Y has been observed to take on a specific value small y. The relation between conditional distributions, joint Distribution and marginal distributions are shown here. This relation should be familiar from the definition of

conditional probability that was seen earlier. Note that the marginal distribution $f_y(y)$ is in the denominator and hence it must not be equal to 0.

(Refer Slide Time: 19:45)

Example



The overall idea of joint marginal and conditional distributions is summarized in this figure. The top left figure shows the joint Distribution and describes how the random variable X which takes on 9 different values is related to the random variable Y which takes on two different values. The bottom left figure shows the marginal distribution of random variable X . As can be observed in this figure we ignore the information related to the random variable Y . Similarly the top-right figure shows the marginal distribution of random variable Y . Finally the bottom-right figure shows the conditional distribution of X given that the random variable Y takes on a value of 1. Looking at this figure and comparing it with a joint distribution we observe that in the bottom-right figure is simply ignore all the values of X for which y equals to 2 that is the top half of the joint distribution.

(Refer Slide Time: 20:46)

Bernoulli Distribution

Consider a random variable X taking one of two possible values (either 0 or 1). Let the PMF of X be given by

$$f_X(0) = P(X = 0) = 1 - p \quad (0 \leq p \leq 1)$$
$$f_X(1) = P(X = 1) = p$$

This describes a Bernoulli distribution

$$E[X] = p$$
$$\text{var}(X) = p(1 - p)$$



In the next few slides we will present some specific distributions that you will be encountering in the machine learning course. We will present the definition and list out some important properties for each distribution. It would be a good exercise for you to work out the expressions for the PMFs or PDFs and the expectation and variances of these distributions on your own. We start with the Bernoulli distribution. Consider a random variable X taking one of two possible values either 0 or 1.

$$f_x(0) = P(X = 0) = 1 - p \quad (0 \leq p \leq 1)$$

$$f_x(1) = P(X = 1) = p$$

Here p is the parameter associated with the Bernoulli distribution. It generally refers to the probability of success so in our definition we are assuming that $X = 1$ indicates a successful trial and X equals to 0 indicates of failure. The expectation of a random variable following the Bernoulli distribution is p and the variance is $p(1 - p)$. The Bernoulli distribution is very useful to characterize experiments which have a binary outcome such as in tossing a coin we observe either heads or tails or say in writing an exam either pass or fail.

Such experiments can be modeled using the Bernoulli distribution next we look at the binomial distribution.

(Refer Slide Time: 22:18)

Binomial Distribution

Consider the situation where we perform n independent Bernoulli trials where

- ▶ probability of success (for each trial) = p
- ▶ probability of failure = $1 - p$

Let X be the number of successes in the n trials, then we have

$$P(X = x | n, p) = \binom{n}{x} p^x (1-p)^{n-x}$$

where $\binom{n}{x} = \frac{n!}{(n-x)!x!}$
and $0 \leq x \leq n$

$$E[X] = np$$

$$\text{var}(X) = np(1-p)$$



Navigation icons: back, forward, search, etc.

Consider the situation where we perform n independent Bernoulli trials where the probability of success for each trial equals to p and the probability of failure for each trial equals to $1 - p$.

Let X be the random variable which represents the number of successes in the end trials then we have probability that the random variable X will take on a specific value of small x given the

parameters n and p is $P(X = x | n, p) = \binom{n}{x} p^x (1-p)^{n-x}$

Note that here X is going to be a number between 0 and n . The expectation of a random variable following the binomial distribution equals to np and the variance equals to $np(1-p)$. The binomial distribution is useful in any scenario where we are conducting multiple Bernoulli trials that is experiments in which the outcome is binary. For example suppose we have a coin and we toss the coin 10 times. We want to know the probability of observing three heads given the probability of observing a head in an individual trial. We can apply the binomial distribution to find out the required probability.

(Refer Slide Time: 23:38)

Geometric Distribution

Suppose we perform a series of independent Bernoulli trials, each with a probability p of success. Let X represent the number of trials before the first success, then we have

$$P(X = x | p) = (1 - p)^{x-1} p \quad x = 1, 2, 3, \dots$$

$$\begin{aligned} E[X] &= 1/p \\ \text{var}(X) &= (1 - p)/p^2 \end{aligned}$$

*



Navigation icons: back, forward, search, etc.

Suppose we perform a series of independent Bernoulli trials each with the probability p of success. Let X represent the number of trials before the first success. Then we have probability that the random variable X will take a value small x given the parameter p is $P(X = x | p) = (1 - p)^{x-1} p$. Essentially we are trying to calculate the probability that it takes us small X number of trials before observing the first success. This can happen if the first $x-1$ trials failed that is with probability $1 - p$ and the last succeeded with probability p .

A random variable which has this probability mass function follows the geometric distribution. For the geometric distribution the expectation of the random variable equals to $1 / p$ and the variance equals to $\frac{1-p}{p^2}$.

(Refer Slide Time: 24:38)

Uniform Distribution

A continuous random variable X is said to be uniformly distributed on an interval $[a, b]$ if its PDF is given by

$$f_X(x|a,b) = \begin{cases} \frac{1}{b-a} & \text{if } x \in [a,b] \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\begin{aligned} E[X] &= (a+b)/2 \\ \text{var}(X) &= (b-a)^2/12 \end{aligned}$$



http://www.eeengr.com/EEENGRTutorial.htm

Many situations we initially do not know the probability distribution of the random variable under consideration but can perform experiments which will gradually reveal the nature of the distribution. In such a scenario we can use the uniform distribution to assign uniform probabilities to all values of the random variable which are then later updated. In the discrete case, say the random variable can take n different values then we simply assign a probability of $1/n$ to each of the n values. In the continuous case if the random variable X takes values in the closed interval (a,b) then its PDF is given by

$$f_x(x|a,b) = \frac{1}{b-a} \text{ if } x \in [a,b]$$

For a random variable following the uniform distribution the expectation of the random variable X is $(a+b)/2$ and the variance equal to $(b-a)^2/12$.

(Refer Slide Time: 25:48)

Normal Distribution

A continuous random variable X is said to be normally distributed with parameters μ and σ^2 if the density of X is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \quad -\infty < x < \infty$$



Navigation icons: back, forward, search, etc.

A continuous random variable X is said to be normally distributed with parameters μ and σ^2 , if the PDF of the random variable X is given by the expression above. The normal distribution is also known as the Gaussian distribution and is one of the most important distributions that we will be using. The diagram represents the famous bell-shaped curve associated with the normal distribution.

(Refer Slide Time: 26:15)

Importance of Normal Distribution

Roughly, the central limit theorem states that the distribution of the sum (or average) of a large number of independent, identically distributed variables will be approximately normal, regardless of the underlying distribution.

Multivariate Normal Distribution

$$N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

where

- ▶ $\boldsymbol{\mu}$ is the D -dimensional mean vector,
- ▶ $\boldsymbol{\Sigma}$ is the $D \times D$ covariance matrix, and
- ▶ $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix



Navigation icons: back, forward, search, etc.

The importance of the normal distribution is due to the central limit theorem. Without going into the details the central limit theorem roughly states that the distribution of the sum of a large number of independent identically distributed variables will be approximately normal regardless of the underlying distribution. Due to this theorem many physical quantities that are the sum of many independent processes often have distributions that can be modeled using the normal distribution.

Also in the machine learning course we will be often using the normal distribution in its multivariate form here we have presented the expression of the multivariate normal distribution where μ is the D dimensional mean vector and Σ is the $D \times D$ covariance matrix.

(Refer Slide Time: 27:06)

Beta Distribution

The pdf of the beta distribution in the range [0,1], with shape parameters α, β , is given by

$$f(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

where the gamma function is an extension of the factorial function.

$$\begin{aligned} E[X] &= \alpha / (\alpha + \beta) \\ \text{var}(X) &= \alpha\beta / ((\alpha + \beta)^2(\alpha + \beta + 1)) \end{aligned}$$

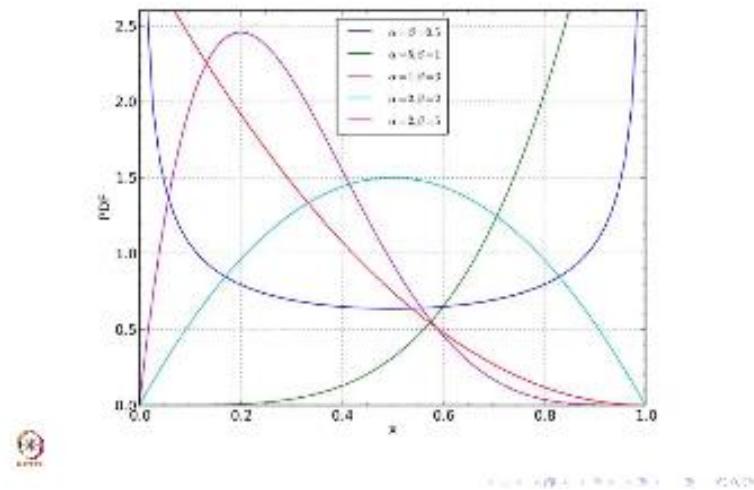


Navigation icons for a presentation slide, including arrows for navigation and symbols for search and refresh.

The PDF of the β distribution in the range 0 to 1 with shape parameters α and β is given by the following expression, where the λ function is an extension of the factorial function. The expectation of a random variable following the β distribution is given by $\alpha / (\alpha + \beta)$ and the variance is given by $\alpha\beta / ((\alpha + \beta)^2(\alpha + \beta + 1))$.

(Refer Slide Time: 27:32)

Beta Distribution



This diagram illustrates the β distribution similar to the normal distribution in which the shape and position of the bell curve is controlled by the parameters μ and σ^2 . In the β distribution the shape of the distribution is controlled by the parameters α and β and in the diagram we can see a few instances of the β distribution for different values of the shape parameters. Note that unlike the normal distribution a random variable following the beta distribution takes values only in a fixed interval.

Thus in this example the probability that the random variable takes a value less than 0 or greater than 1 is 0.

This ends the first tutorial on the basics of probability theory. If you have any doubts or seek clarifications regarding the material covered in this tutorial please make use of the forum to ask questions. As mentioned in the beginning if you are not comfortable with any of the concepts presented here do go back and read up on it there will be some questions from probability theory in the first assignment. So hopefully going through this tutorial will help you in answering those questions and note that there we will be having another tutorial next week on linear algebra.

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Linear Algebra Tutorial CS5011 – Machine Learning

Abhinav Garlapati Varun Gangal

Department of Computer Science
IIT Madras

January 17, 2016

Hi everyone welcome to the second tutorial of the introduction to machine learning course. So in this tutorial we shall be taking a tour of the aspects of linear algebra which you would need for the course. We will cover a variety of concepts such as subspaces, basis, span, decomposition, eigenvalues and eigenvectors over the course of the tutorial.

(Refer Slide Time: 00:47)

What is Linear Algebra

Linear Algebra

Linear algebra is the branch of mathematics concerning vector spaces and linear mappings between such spaces. It includes the study of lines, planes, and subspaces, but is also concerned with properties common to all vector spaces.

Why do we study Linear Algebra?

- Provides a way to compactly represent & operate on sets of linear equations.
- In machine learning, we represent data as matrices and hence it is natural to use notions and formalisms developed in Linear Algebra.

NPTEL

Abhinav Garlapati, Varun Gangal

Linear Algebra Tutorial

January 17, 2016

2 / 31

So the first question one would ask is why we need linear algebra at all and what is linear algebra? So you may have come across this in school or your +1 or +2 level but just to recap

linear algebra is the branch of mathematics which deals with vectors and vector spaces and linear mappings between these spaces. So, why do we study linear algebra here? Especially in the context of machine learning, firstly it gives us a way to manipulate, represent and operate sets of linear equations. And why do these linear equations pop up in machine learning in the first place? So the reason for that is in machine learning we represent our data as a $n \times p$ matrix where n is the number of data points and p is the number of features. So it is natural we have to use notions and formalisms developed in linear algebra. The data or the parameter is you use are represented as vectors so as a result linear algebra has an important role to play in machine learning.

(Refer Slide Time: 02:05)

Introduction to LinAl

- Consider the following system of equations:

$$\begin{aligned} 4x_1 - 5x_2 &= -13 \\ -2x_1 + 3x_2 &= 9 \end{aligned}$$

- In matrix notation, the system is more compactly represented as:

$$Ax = b$$

$$A = \begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix}$$

$$b = \begin{bmatrix} -13 \\ 9 \end{bmatrix}$$


Abhinav Garapati, Varun Gangal

Linear Algebra Tutorial

January 17, 2016

3 / 31

So you can see here a system of linear equations with two equations and two variables

$$4x_1 - 5x_2 = -13$$

$$-2x_1 + 3x_2 = 9$$

So we can right away see here the advantages of a matrix notation. If you see below you can represent the same system of two equations directly as one equation in the form of $Ax = b$ where A is the set of coefficients and x is the 2×1 matrix or you may also call it a 2-dimensional vector (x_1, x_2) .

So we can see that when you multiply the matrix A with (x_1, x_2) the 2×1 matrix you will get back the same L.H.S or the set of two L.H.S and b . The matrix b represents the RHS so it is very easy to verify that if you represent this in matrices you can get back the original representation. So to solving this in general, without matrices would require first solving for one variable and then substitute to get the other. But using matrices you can even solve this directly, so just multiply both the sides by a inverse.

So you would get $x = A^{-1}b$. Of course you would have to care about the fact that all matrices do not have an inverse. But in most cases they do, so in that case you can directly get the solution of x in the form of $x = A^{-1}b$. As said earlier, linear algebra gives us this freedom to manipulate several equations at once and multiple variables.

(Refer Slide Time: 04:10)

Vector Space

Definition

A set V with two operations $+$ and \cdot is said to be a **vector space** if it is closed under both these operations and satisfies the following eight axioms.

- ① Commutative Law
$$x + y = y + x, \quad \forall x, y \in V$$
- ② Associative Law
$$(x + y) + z = x + (y + z), \quad \forall x, y, z \in V$$
- ③ Additive identity
$$\exists 0 \in V \text{ s.t } x + 0 = x, \quad \forall x \in V$$
- ④ Additive inverse
$$\forall x \in V, \exists \bar{x} \text{ s.t } x + \bar{x} = 0$$

 IITMPT

Abhisav Garlapati, Varun Gangal | Linear Algebra Tutorial | January 19, 2016 | 4 / 31

A fundamental definition in linear algebra is that of a vector space. A set of vectors V is said to be a vector space if it is closed under the operations of vector addition and scalar multiplication and in addition satisfies the axioms we have listed here. Like if we take two elements from this set X & Y then $X + Y$ will also lie in the set V . In addition to this, if we take a scalar α (a real

number) and multiply a vector from this set y with it then αy also belongs to V . If both these properties are satisfied then the set of vectors is said to be closed with respect to vector addition and scalar multiplication. Now let us have a look at the axioms.

1. The first one is a commutative law, the commutative law states that if you pick any two elements from the set V , x & y then $x+y=y+x$.
2. The associative law says that if you pick any point from this set x, y, z ,

$$(x+y)+z=x+(y+z)$$
3. The additive identity law states that there exists an additive identity or a 0 such that if you pick any element from the set and add this 0 to it you get back the same element.
4. The additive inverse law states that there exists for every element x a corresponding \bar{x} such that $x+\bar{x}=0$.

(Refer Slide Time: 06:16)

Vector Space (Contd..)

- ➊ Distributive Law

$$\alpha \cdot (x + y) = \alpha \cdot x + \alpha \cdot y, \quad \forall \alpha \in \mathbb{R}, x, y \in V$$
- ➋ Distributive Law

$$(\alpha + \beta) \cdot x = \alpha \cdot x + \beta \cdot x, \quad \forall \alpha, \beta \in \mathbb{R}, x \in V$$
- ➌ Associative Law

$$(\alpha\beta) \cdot x = \alpha \cdot (\beta \cdot x), \quad \forall \alpha, \beta \in \mathbb{R}, x \in V$$
- ➍ Unitary Law

$$1 \cdot x = x, \quad \forall x \in V$$

MPTEL
Abhinav Garlapati, Varun Gangal
Linear Algebra Tutorial
January 19, 2016
5 / 31

5. The fifth law is the distributive law. This law says that, if you have a real scalar α with which you multiply the sum of two vectors $x+y$, then that should be equal to $\alpha x+\alpha y$. The second distributive law says that if you have two scalars α, β and vector x , then

$$(\alpha + \beta)x = \alpha x + \beta x$$

6. The associative law says that if you will first multiply two scalars α, β and then multiplying the vector x with them that should be equal to multiplying the vector x first with the second scalar β and then with α . Or, $(\alpha\beta).x = \alpha.(\beta.x)$
7. The unitary law says that on multiplication by the scalar real number 1, you get back the same vector. This is important because you would not want multiplication to force any unexpected scaling like if you multiplied a vector x by the scalar k then you need to be sure that it will be exactly k times the initial value and should not be say \sqrt{k} .

(Refer Slide Time: 07:35)

Subspace

Definition

Let W be a subset of a vector space V . Then W is called a **subspace** of V if W is a vector space.

- Do we have to verify all 8 conditions to check whether a given subset of a vector space is a subspace?
- **Theorem:** Let W be a subset of a vector space V . Then W is a subspace of V if and only if W is non-empty and $x + \alpha y \in W, \forall x, y \in W, \alpha \in \mathbb{R}$



IITTEL
Abhinav Garlapati, Varun Gangal
Linear Algebra Tutorial
January 19, 2016
6 / 31

A second related definition is that of a subspace. A subset W of a vector space V is said to be a subspace, if W is a vector space. Now this means that W should be closed under vector addition and scalar multiplication. It should also satisfy the eight axioms we stated earlier. Now the question that arises is do we need to verify all these eight conditions given that we know that W is already a subset of a vector space. No, it is enough to check just for the following two conditions firstly that W is non empty that in other words it has at least a single element. And secondly that if I pick any two elements $x & y$ from this set and any real number α then $\alpha x + \alpha y \in W$.

(Refer Slide Time: 08:39)

Norm

Definition

Norm is any function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying:

- ① $\forall x \in \mathbb{R}^n, f(x) \geq 0$ (non-negativity)
- ② $f(x) = 0$ iff $x = 0$ (definiteness)
- ③ $\forall x \in \mathbb{R}^n, f(tx) = |t|f(x)$ (homogeneity)
- ④ $\forall x, y \in \mathbb{R}^n, f(x + y) \leq f(x) + f(y)$ (triangle inequality)

- Example - l_p norm

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

- Matrices can have norms too - e.g., Frobenius norm

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2} = \sqrt{\text{tr}(A^T A)} \quad (1)$$

 IIT-BHU
Abhinav Garlapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016

7 / 31

Now let us have a look at the definition of a norm. So intuitively a norm is a measure of the length of a vector or its magnitude. It is a function from a vector space which mostly happens to be \mathbb{R}^n where n is a dimension of a vector to the space of real numbers \mathbb{R} . So for a function to be a norm it should satisfy the four conditions which we have given here. Firstly it should be always be non-negative. Secondly, it should be zero if and only if the vector is zero. Thirdly for every vector if you multiplied by a scalar its norm should get multiplied by the modulus of the scalar. By modulus here we mean the absolute value. The fourth being that if we take any pair of vectors in our vector space which is \mathbb{R}^n , the norm of the sum of these two vectors should be less than the sum of their norms, which this is also known as the triangle inequality. So this is namely related to the fact that the third side of a triangle should always be less than sum of the other two sides.

Now an example of a norm is the l_p norm there you sum up the absolute values along each dimension raised to p and then take the $1/p$ th root of this. So when $p = 2$ you get the L2 norm which is the magnitude of a vector as we have learnt in our earlier studies $\sqrt{x^2 + y^2}$ if you are looking at just the space \mathbb{R}^2 . There are other norms for instance there are norms defined for

matrices as well. Here we had defined all norm for vectors, so the Frobenius norm is a matrix norm. So what it does is it essentially sums up the squares of all the elements and then takes the root of that. So this also happens to be equal to the trace of $A^T A$. Now the trace of a matrix is simply the sum of its diagonal elements.

(Refer Slide Time: 11:12)

Range Of A Matrix

- The **span** of a set of vectors $X = \{x_1, x_2, \dots, x_n\}$ is the set of all vectors that can be expressed as a linear combination of the vectors in X .
In other words, set of all vectors v such that $v = \sum_{i=1}^{i=|X|} \alpha_i x_i, \alpha_i \in R$
- The **range** or **columnspace** of a matrix A , denoted by $R(A)$ is the span of its columns. In other words, it contains all linear combinations of the columns of A . For instance, the columnspace of $A = \begin{bmatrix} 1 & 0 \\ 5 & 4 \\ 2 & 4 \end{bmatrix}$ is the plane spanned by the vectors $\begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 4 \\ 4 \end{bmatrix}$.



Abhinav Garapati, Varun Gangal
Linear Algebra Tutorial
January 19, 2016
8 / 31

The span of a set of vectors is the set of all vectors which can be composed using these vectors by using the operations of vector addition and scalar multiplication. So the name span comes from the fact that this set of vectors spans a potentially larger set of vectors which is then called a

span. So to define this more formally it is the set of all vectors V such that $v = \sum_{i=1}^{|x|} \alpha_i x_i, \alpha_i \in \mathbb{R}$.

Now a related definition is that of range or column space. So if we think of a matrix each of its columns is a vector so the set of all columns of a matrix is like a set of vectors. Now the span of this set of vectors is called the range or column space of that matrix. So if you consider the matrix given here the columns of this matrix A are $(1, 5, 2)$ and $(0, 4, 4)$, so what would be this column space of this matrix, it would be this the span of the vectors $(1, 5, 2)$ and $(0, 4, 4)$ which is essentially the plane which is spanned by these two vectors.

(Refer Slide Time: 12:56)

Nullspace Of A Matrix

Definition

The nullspace $N(A)$ of a matrix $A \in \mathbb{R}^{m \times n}$ is the set of all vectors that equal 0 when multiplied by A . The dimensionality of the nullspace is also referred to as the **nullity** of A .

$$N(A) = \{x \in \mathbb{R}^n : Ax = 0\}$$

- Note that vectors in $N(A)$ are of dimension n , while those in $R(A)$ are of size m , so vectors in $R(A^T)$ and $N(A)$ are both of dimension n .



MPTEL

Abhinav Gorlapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016 9 / 31

If we have matrix A of dimensions $m \times n$ then the null space is the set of $n \times 1$ vectors which gives $m \times 1$ zero vector on being multiplied by A . In other $N(A) = \{x \in \mathbb{R}^n : Ax = 0\}$. Nullity is the rank or dimensionality of the null space. We will revisit the definition of nullity later once you defined rank more clearly. Another interesting fact about null spaces is that the null space of A is of dimension n while the range of A or the column space as we defined it earlier is of dimension m or $m \times 1$. This means that vectors in $R(A^T)$ so note that A^T is $n \times m$. So vector in the range of A^T will be of dimension n similar to the vectors in the null space of A . So this means that the vectors in range of A^T and null space of A would both be of the dimension $n \times 1$.

(Refer Slide Time: 14:15)

Example

Consider the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 5 & 4 \\ 2 & 4 \end{bmatrix}$$

The nullspace of A is made up of vectors x of the form $\begin{bmatrix} u \\ v \end{bmatrix}$, such that

$$\begin{bmatrix} 1 & 0 \\ 5 & 4 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The nullspace here only contains the vector (0,0).



MPTSEL

Abhinav Garlapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016

10 / 31

Let us consider an example to illustrate the concept of a null space. Consider the matrix A given here, A is a 3×2 matrix hence the null space of A will be made up of vectors of dimension 2 or 2×1 . Now we see that the on solving being we get $u = 0, v = 0$. This means that the null space only contains the 0 vector the two dimensional 0 vector (0, 0).

(Refer Slide Time: 14:52)

Another Example

Now, consider the matrix

$$B = \begin{bmatrix} 1 & 0 & 1 \\ 5 & 4 & 9 \\ 2 & 4 & 6 \end{bmatrix}$$

Here, the third column is a linear combination of the first two columns.
Here, the nullspace is the line of all points $x = c, y = c, z = -c$.



Abhinav Garapati, Varun Gangal

Linear Algebra Tutorial

Navigation icons: back, forward, search, etc.

January 19, 2016

11 / 31

Let us consider another example to illustrate null spaces better. Take the matrix B which is a 3 x 3 matrix. The null space would consist of 3 x 1 vectors. We leave the finding of the null space to the audiences as an exercise. However when on solving we get the null space to be the set of all vectors of the form $x = c, y = c$ and $z = -c$, where c is any real number and x, y, z referred to the first second and third dimension respectively.

(Refer Slide Time: 15:35)

Linear Independence and Rank

Definition

A set of vectors $\{x_1, x_2, \dots, x_n\} \in \mathbb{R}^n$ is said to be **(linearly) independent** if no vector can be represented as a linear combination of the remaining vectors.

- i.e., if $x_n = \sum_{i=1}^{n-1} \alpha_i x_i$ for some scalar values $\alpha_1, \dots, \alpha_{n-1} \in \mathbb{R}$, then we say that the vectors $\{x_1, x_2, \dots, x_n\}$ are linearly dependent; otherwise, the vectors are linearly independent
- The **column rank** of a matrix $A \in \mathbb{R}^{m \times n}$ is the size of the largest subset of columns of A that constitute a linearly independent set
- Similarly, **row rank** of a matrix is the largest number of rows of A that constitute a linearly independent set



Abhinav Garlapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016

12 / 31

Before we dive into defining linear independence, recollect how we define the linear combination. A set of vectors is linearly independent if no vector in the set can be produced using the linear combination of the other vectors in the set. Now let us have a look at the related concept of rank. So the column rank of $m \times n$ matrix A is the size of the largest linearly independent subset of columns. Note that our columns here are $m \times 1$ vectors. The row rank is defined in a similar way for rows.

(Refer Slide Time: 16:22)

Properties Of Ranks

- For any matrix $A \in \mathbb{R}^{m \times n}$, it turns out that the column rank of A is equal to the row rank of A , collectively as the rank of A , denoted as $\text{rank}(A)$
- Some basic properties of the rank:
 - For $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) \leq \min(m, n)$.
If $\text{rank}(A) = \min(m, n)$, A is said to be **full rank**
 - For $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = \text{rank}(A^T)$
 - For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$
 - For $A, B \in \mathbb{R}^{m \times n}$, $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$



Abhinav Garapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016

13 / 31

Let us walk through some interesting properties of ranks. For any $m \times n$ matrix of real numbers the column rank is equal to the row rank. We refer to this quantity as the rank of the matrix. Earlier we had looked at the quantity called nullity. Nullity is the rank of the null space of A . Some other interesting properties of ranks are also listed here. For instance the rank of a matrix is at most the minimum of its two dimensions, the row dimension and the column dimension. Secondly the rank of a matrix is the same as the rank of its transpose. Thirdly if you multiply two matrices A and B the rank of the resultant matrix is at most the minimum of the ranks of A and B . If you add up two matrices the rank of the resultant matrix is at most the sum of the ranks of A and B .

(Refer Slide Time: 17:33)

Orthogonal Matrices

- A square matrix $U \in R^{n \times n}$ is **orthogonal** iff
 - All columns are mutually orthogonal $v_i^T v_j = 0, \forall i \neq j$
 - All columns are normalized $v_i^T v_i = 1, \forall i$
- If U is orthogonal, $UU^T = U^T U = I$. This also implies that the inverse of U happens to be its transpose.
- Another salient property of orthogonal matrices is that **they do not change** the Euclidean norm of a vector when they operate on it, i.e $\|Ux\|_2 = \|x\|_2$.
Multiplication by an orthogonal matrix can be thought of as a pure rotation, i.e., it does not change the magnitude of the vector, but changes the direction.



Abhinav Garlapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016

14 / 31

A square matrix U of dimension $n \times n$ is defined to be orthogonal if and only if the following two conditions hold. Firstly all pairs of distinct columns should be orthogonal. By columns being orthogonal we mean that the dot product of any pair of distinct column vectors is zero. In other words the $v_i^T v_j = 0, \forall i \neq j$. The second condition is that the dot product of any column with itself or $v_i^T v_i = 1, \forall i$.

In other words all the column vectors should be normalized. An interesting implication of a matrix being orthogonal is that UU^T and $U^T U$ both end up being equal to the $n \times n$ identity matrix I . This also means that $U^T = U^{-1}$. Or the transpose of such an orthogonal matrix is also its inverse. An additional interesting property is seen when we multiply a $m \times 1$ vector x by $m \times m$ orthogonal matrix U . The Euclidean or L2 norm of such a vector x remains the same on multiplication by U . Intuitively we can understand this as orthogonal matrices U performing only pure rotation on multiplying the vector x . In other words they only change the direction of a vector but do not change its magnitude.

(Refer Slide Time: 19:27)

Quadratic Form of Matrices

- Given a square matrix $A \in \mathbb{R}^{n \times n}$ and a vector $x \in \mathbb{R}^n$, the scalar value $x^T Ax$ is called a **quadratic form**.
- A symmetric matrix $A \in \mathbb{S}^n$ is positive definite (PD) if for all non-zero vectors $x \in \mathbb{R}^n$, $x^T Ax > 0$.
- Similarly, positive semidefinite if $x^T Ax \geq 0$, negative definite if $x^T Ax < 0$ and negative semidefinite if $x^T Ax \leq 0$.
- One important property of positive definite and negative definite matrices is that they are always full rank, and hence, invertible.
- Gram matrix:** Given any matrix $A \in \mathbb{R}^{m \times n}$, matrix $G = A^T A$ is always positive semidefinite. Further if $m \geq n$, then G is positive definite.



Abhinav Garlapati, Varun Gangal

Linear Algebra Tutorial

January 19, 2016

15 / 31

We often encounter the quadratic form which is the vector equivalent of a quadratic function. The quadratic form with respect to the matrix A of a vector x where the matrix A is $m \times n$ and the vector x is $n \times 1$ is given by the real number $x^T Ax$. Based on the quadratic forms of matrices we can classify them as positive definite, negative definite, positive semi definite and negative semi definite.

A matrix A is said to be positive definite if its quadratic form is greater than zero for any vector x . Similarly we can define it to be negative definite. A matrix A is positive semi definite if the quadratic form is greater than equal to 0 for any vector x . Note that equality with 0 may also hold here. One important property of positive and negative definite matrices is that they are always full rank. An implication of this is that A^{-1} always exists. For a matrix A , which is of dimension $m \times n$, one can define a special matrix called a gram matrix. The gram matrix is given by $A^T A$. One of the property of the gram matrix is that it is always positive semi definite Moreover if the number of rows exceeds the number of columns in other words if $m >= n$ this means that G is positive definite.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 8

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology
Linear Algebra (2)

(Refer Slide Time: 00:15)

Eigenvalues & Eigenvectors

- Given a square matrix $A \in \mathbb{R}^{n \times n}$, λ is said to be an eigenvalue of A and vector \vec{x} the corresponding eigenvector if

$$A\vec{x} = \lambda\vec{x}$$

- Geometrical interpretation**
We can think of the eigenvectors of a matrix A as those vectors which upon being operated by A are only scaled but not rotated.
- Example**

$$A = \begin{bmatrix} 6 & 5 \\ 1 & 2 \end{bmatrix}, \vec{x} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$
$$A\vec{x} = \begin{bmatrix} 35 \\ 7 \end{bmatrix} = 7\vec{x}$$


AKHILARAJA VENKATESWARAN
Linear Algebra Tutorial
January 09, 2014 14 / 20

Eigenvectors and eigenvalues of A are tied together, which means that every eigenvector has an associated eigenvalue. We often characterize square matrices in terms of their eigenvectors. One way of looking at eigenvectors is as follows: \vec{x} can be thought of as a vector and in \mathbb{R}^n and the square matrix A acts like an operator which transforms \vec{x} into another n -dimensional vector

\vec{Ax} . Now the Eigen vectors of A are those vectors which on being transformed by A or operated upon by A are only scaled by λ but not rotated in other words their direction does not change.

We can have a look at this example here the 2×2 matrix A. On multiplying the vector $X = (5, 1)$ gives back the vector X multiplied by the real value 7. So here X is an eigenvector of A and 7 is an eigenvalue of A.

(Refer Slide Time: 01:40)

Characteristic Equation

- Trivially, the 0 vector would always be an eigenvector of any matrix.
Hence, we only refer only to non-zero vectors as eigenvectors.
- Given a matrix A, how do we find all eigenvalue-eigenvector pairs?

$$A\vec{x} = \lambda\vec{x}$$

$$A\vec{x} - \lambda I\vec{x} = 0$$

$$(A - \lambda I)\vec{x} = 0$$

The above will hold iff $|(A - \lambda I)| = 0$

This equation is also referred to as the characteristic equation of A. Solving the equation gives us all the eigenvalues λ of A. Note that these eigenvalues can be complex.

 IIT Roorkee, Varanasi, India
Linear Algebra Tutorial January 19, 2018 47 / 48

We can see that zero would always be an eigenvector of any matrix if we simply go by the $\vec{Ax} = \lambda \vec{x}$ definition. Hence we only refer to nonzero vectors as eigenvectors. So the question is given a matrix A how does one find all the eigenvalue eigenvector pairs. By simplifying $\vec{Ax} = \lambda \vec{x}$ we get $(A - \lambda I) \vec{x} = 0$. Now since we are only looking at nonzero vectors $\det(X)$ cannot be 0 and X can't be a zero vector which means that $\det(A - \lambda I)$ should be zero. So the equation $\det(A - \lambda I) = 0$ is called a characteristic equation of A and solving this equation gives us all the

eigenvalues of A. One thing you note that even though all the values of A are real and A is a real matrix the eigenvalues can be complex.

(Refer Slide Time: 02:54)

The slide has a blue header bar with the title 'Properties'. Below the header, there is a bulleted list of properties of matrices:

- The trace $\text{tr}(A)$ of a matrix A also equals the sum of its n eigenvalues.

Below this point, there is a mathematical equation:

$$\text{tr}(A) = \sum_{i=1}^n \lambda_i$$

- The determinant $|A|$ is equal to the product of the eigenvalues.

Below this point, there is a mathematical equation:

$$|A| = \prod_{i=1}^n \lambda_i$$

- The rank of a matrix is equal to the number of non zero eigenvalues of A.
- If A is invertible, then the eigenvalues of A^{-1} are of form $\frac{1}{\lambda_i}$, where λ_i are the eigenvalues of A.

At the bottom of the slide, there is a footer bar with the following information:

- NPTEL
- Allison Subedi, Nitin Gangal
- Linear Algebra, Tatyana
- January 20, 2018
- 18 / 21

There are interesting relations between some properties of a matrix and its eigenvalues. For instance,

1. The trace of a matrix is equal to the sum of its eigenvalues $\text{tr}(A) = \sum_{i=1}^n \lambda_i$
2. The determinant of a matrix is equal to the product of its eigenvalues $\det(A) = \prod_{i=1}^n \lambda_i$

The rank of a matrix is equal to the number of nonzero eigenvalues. Note that if an eigenvalue has multiplicity greater than 1, for instance, if two distinct eigenvectors x_1, x_2 both have eigenvalue λ we would count λ twice. Also we can describe the eigenvalues of A^{-1} in

terms of the eigenvalues of A provided of course A is invertible. The eigenvalues of A^{-1} will be of the form $\frac{1}{\lambda_i}$ where λ_i is an eigenvalue of A.

(Refer Slide Time: 03:55)

Proof

$$\begin{aligned} \sum_{i=1}^{j=k} a_i v_i &= \vec{0} \\ (A - \lambda_k I) \sum_{i=1}^{j=k} a_i v_i &= \vec{0} \\ \sum_{i=1}^{j=k} (A - \lambda_k I) a_i v_i &= \vec{0} \quad \text{?} \\ \sum_{i=1}^{j=k} a_i (\lambda_i - \lambda_k) v_i &= \vec{0} \end{aligned}$$

Since the eigenvalues are distinct, $\lambda_i \neq \lambda_k \forall i \neq k$. Thus the set of $(k-1)$ eigenvectors is also linearly dependent, violating our assumption of it being the smallest such set. This is a result of our incorrect starting assumption.

Hence proved by contradiction.

Now let us have a look at an interesting theorem about eigenvalues and eigenvectors. The theorem goes as follows:

If a matrix has all its eigenvalues distinct then its eigenvectors are linearly independent.

We shall prove this by what is called a proof by contradiction. If this theorem does not hold that means there is a set of k eigenvectors such that it is linearly dependent. Let the i^{th} vector in the set be v_i and the corresponding eigenvalue be λ_i . Note that we are considering the smallest such set. Since the set is linearly dependent this means there exists real constants a_i such

that $\sum_{i=1}^{i=k} a_i v_i = \vec{0}$. Now let us multiply both sides of the equation by $A - \lambda_k I$. Since v_k is an

eigenvector of A , $(A - \lambda_k I)v_k = 0$. We can understand this from the characteristic equation hence the term corresponding to v_k disappears from the equation since it goes to zero.

Now for the remaining eigenvalues since we know they are distinct, the term $\lambda_i - \lambda_k \neq 0$. Note that $(A - \lambda_k I)v_i$ simplifies to $(\lambda_i - \lambda_k)v_i$ since $Av_i = \lambda_i v_i$. However now we can think of

$a_i(\lambda_i - \lambda_k)$ as a new constant b_i . This means now that we have $\sum_{i=1}^{k-1} b_i v_i = 0$. However, we had

assumed that the set of size k was the smallest set of linearly dependent eigenvectors however now we have an even smaller set. This contradicts our starting assumption. Hence such a set of k linearly dependent eigenvectors cannot exist for any $k \geq 2$. Hence all our eigenvectors are linearly independent hence our theorem stands true.

(Refer Slide Time: 06:54)

Diagonalization

Given a matrix A , we consider the matrix S with each column being an eigenvector of A

$$S = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$$AS = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 \vec{v}_1 & \lambda_2 \vec{v}_2 & \dots & \lambda_n \vec{v}_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$$AS = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots \\ 0 & \ddots & \dots \\ 0 & \dots & \lambda_n \end{bmatrix}$$



 Akhilesh Gopal, Vinita Gangal Linear Algebra Tutorial January 20, 2015 21 / 24

Diagonalization gives us a way of representing a matrix in terms of its eigenvalues and eigenvectors. Let us consider a $n \times n$ square matrix A. We denote the matrix where every column is an eigenvector of A by S. On multiplying S by A each column would get multiplied by λ_i since the column itself is an eigenvector of A.

This right hand side can then be simplified as the product of two matrices the first one being S itself while the second one being the diagonal matrix where the i^{th} diagonal element is the eigenvalue λ_i . Remember that the LHS is AS. Now we have the equation $AS = S\Lambda$ where Λ is the diagonal matrix of eigenvalues. On simplifying this we get $A = S\Lambda S^{-1}$. This is a diagonalization of A. Note that $S^{-1}AS$ is a diagonal matrix since $S^{-1}AS$ is nothing but Λ , the diagonal matrix of Eigen values.

This result is dependent on S being invertible. It will hold if the eigenvalues of a matrix are distinct since the eigenvectors would then be linearly independent. This would mean the columns of S would be linearly independent and hence S would be full ranked and as a consequence invertible.

(Refer Slide Time: 08:58)

Properties of Diagonalization

- ① A square matrix A is said to be **diagonalizable** if there exists such that $A = SAS^{-1}$.
- ② Diagonalization can be used to simplify computation of the higher powers of a matrix A , if the diagonalized form is available.

$$A^n = (SAS^{-1})(SAS^{-1}) \dots (SAS^{-1})$$
$$A^n = SA^n S^{-1}$$

A^n is simple to compute since it is a diagonal matrix.



Then do we say that the square matrix is diagonalizable? Well when such a diagonalization exists we saw that we needed S to be invertible for the diagonalization to exist.

Another advantage of diagonalization is that it simplifies the process of computing A^n . We first represent every A in diagonalized form now we can see that the S^{-1} of the first term and the S of the second term would multiply to give us I . Similarly for the second third fourth and so on in this way by regrouping the terms we get $A^n = S\Lambda^n S^{-1}$. Note that it is very easy to compute the n^{th} power of a diagonal matrix since you just have to raise every diagonal element to the power of n . In this way the diagonalization has helped us simplify the process of computing A^n . Without this simplification we would have needed to multiply non-diagonal matrix n times.

(Refer Slide Time: 10:31)

Eigenvalues & Eigenvectors of Symmetric Matrices

- Two important properties for a symmetric matrix A :
 - All the eigenvalues of A are real.
 - The eigenvectors of A are orthonormal, i.e., matrix S is orthogonal. Thus, $A = SAS^T$.
- Definiteness of a symmetric matrix depends entirely on the sign of its eigenvalues. Suppose $A = SAS^T$, then

$$x^T Ax = x^T SAS^T x = y^T \Lambda y = \sum_{i=1}^n \lambda_i y_i^2$$

- Since $y_i^2 \geq 0$, sign of expression depends entirely on the λ_i 's. For example, if all $\lambda_i > 0$, then matrix A is positive definite.



If a matrix is symmetric then all its eigenvalues are real numbers. Also its eigenvectors are orthonormal, that is they are mutually orthogonal and normalized. This means that the matrix of eigenvectors S is also orthogonal. We have seen that for orthogonal matrices the inverse and the transpose are the same, hence, we can write $A = S\Lambda S^T$ as per the diagonalization we defined earlier. For symmetric matrices their definiteness can be inferred from the signs of their eigenvalues. Suppose that $A = S\Lambda S^T$, now taking the quadratic form with respect to A for vector x , $x^T Ax$ simplifies to $y^T \Lambda y$, where y is $S^T x$. This further simplifies to $\sum_{i=1}^n \lambda_i y_i^2$. Now for a

matrix to be positive definite this term must always be positive. Since $y_i^2 \geq 0$, anyway the sign of this term depends on the eigenvalues.

If all the eigenvalues are positive the matrix is positive definite. If we know that the matrix is positive semi definite or PSD then what can we say about its eigenvalues. Since the quadratic form of a PSD matrix is non-negative for any vector x , this should hold for the eigenvectors too. Now since $Ax = \lambda x$, $x^T Ax$ simplifies to $\lambda \|x\|^2 \geq 0$.

Since eigenvectors are nonzero by definition the square of the norm is always positive. This means that every eigenvalue of A is non-negative.

(Refer Slide Time: 13:04)

Singular Value Decomposition

- We saw that diagonalization is applicable only to square matrices. We need some analogue for rectangular matrices too, since we often encounter them, e.g the Document-Term matrix. For a rectangular matrix, we consider left singular and right singular vectors as two bases instead of a single base of eigenvectors for square matrices.
- The Singular Value Decomposition is given by $A = U\Sigma V^T$ where $U \in R^{m \times m}$, $\Sigma \in R^{m \times n}$ and $V \in R^{n \times n}$.



IIT Kharagpur, India | Last Update: January 10, 2018 | Page: 112

We learnt about diagonalization which took in a square matrix of size $n \times n$ and represented it in terms of its eigenvectors. However we cannot directly apply the same diagonalization for rectangular matrices since the notion of eigenvector is defined only for a square matrix. We need a diagonalization for rectangular matrices since we come to them often. For instance the matrix of n data points or features or the matrix of n documents and r terms. For the rectangular matrix A of size $m \times n$ they can be represented in terms of the eigenvectors of AA^T and A^TA both of which are square matrices. This is known as the singular value decomposition. A is represented as $A = U\Sigma V^T$, where $U \in R^{m \times m}$, $\Sigma \in R^{m \times n}$ and $V \in R^{n \times n}$

(Refer Slide Time: 14:22)

Singular Value Decomposition

- ① U is such that the m columns of U are the eigenvectors of AA^T , also known as the left singular vectors of A .
- ② V is such that the n columns of V are the eigenvectors of A^TA , also known as the right singular vectors of A .
- ③ Σ is a rectangular diagonal matrix with each element being the square root of an eigenvalue of AA^T or A^TA .

Significance: SVD allows us to construct a lower rank approximation of a rectangular matrix. We choose only the top r singular values in Σ , and the corresponding columns in U and rows in V^T .



National Programme on Technology Enhanced Learning
Linear Algebra Tutorial
January 09, 2018 22 / 20

The three elements U , Σ and V are as follows. In U every column represents an eigenvector of AA^T . In V every column represents an eigenvector of A^TA and Σ is a rectangular diagonal matrix with each element being the square root of an eigenvalue of AA^T or A^TA . Now look at AA^T or A^TA have different eigenvectors but the set of eigenvalues is the same. This is because suppose $A^TAx = \lambda x$ for some eigenvector and eigenvalue λ . Now multiplying both sides by A we get AA transpose times $AA^TAx = \lambda Ax$. Hence Ax is an eigenvector of AA^T and λ is also an eigenvalue of AA^T .

This is why AA^T and A^TA have the same set of eigenvalues. The significance of this decomposition is that if we ordered U , V and Σ such that the eigenvalues whose magnitude is large will come first both in U and V in the column order also along the diagonal in Σ . Then we can drop everything greater than index r to get a r dimensional low rank approximation of the original matrix A . This approximate form of A will be represented as $U\Sigma V^T$ which is an $m \times r$ matrix.

Consider a function f which takes in matrix of dimension $m \times n$ and outputs real numbers. The gradient is the matrix of partial derivatives then $(\nabla_A f(A))_{ij} = \frac{\partial f(A)}{\partial A_{ij}}$. Consider a different type of function which takes in a n dimensional vector and returns a real number. The Hessian for this function is defined as follows the $(\nabla^2_x f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$. You can see that hessian would be an $n \times n$ matrix.

Now let us study how we can find the gradient for some simple vector functions consider the function $f(x) = b^T x$, where x is an n dimensional vector and b is also an n -dimensional vector.

Then $f(x) = \sum_{i=1}^{i=n} b_i x_i$. On differentiating this with respect to the k th component of the vector X we get $\frac{\partial f(x)}{x_k} = b_k$. Hence the gradient of $f(x)$ is given by the vector b . We can see how this intuitively relates to the first derivative of the scalar function $f(x) = Ax$ which is equal to A .

We had earlier looked at a type of function called the quadratic form defined for a $n \times n$ matrix A . The quadratic form with respect to matrix A is a function $f(x) = x^T Ax$ which takes in an n -dimensional vector x . Now let's have a look at how one can find the gradient and hessian on the quadratic form of a known symmetric matrix A . They can write down $f(x) = \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j$. We can split-up this summation into four terms based on whether i and j are equal or not equal to k .
(Refer Slide Time: 18:33)

Differentiating Linear and Quadratic Functions

Consider the function $f(x) = x^T Ax$ where $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$ is a known symmetric matrix.

$$\begin{aligned} f(x) &= \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} A_{ij}x_i x_j \\ \frac{\partial f(x)}{\partial x_k} &= \frac{\partial}{\partial x_k} \left[\sum_{i \neq k} \sum_{j \neq k} A_{ij}x_i x_j + \sum_{i \neq k} A_{ik}x_i x_k + \sum_{j \neq k} A_{kj}x_k x_j + A_{kk}x_k^2 \right] \\ \frac{\partial f(x)}{\partial x_k} &= \sum_{i \neq k} A_{ik}x_i + \sum_{j \neq k} A_{kj}x_j + 2A_{kk}x_k \\ \frac{\partial f(x)}{\partial x_k} &= \sum_{i=1}^n A_{ki}x_i + \sum_{j=1}^n A_{kj}x_j \\ \frac{\partial f(x)}{\partial x_k} &= 2 \sum_{i=1}^n A_{ki}x_i \end{aligned}$$



National Programme on Technology Enhanced Learning

Linear Algebra (TUTORIAL)

January 23, 2014 26 / 34

Finally, $\frac{\partial f(x)}{\partial x_k} = 2 \sum_{i=1}^n A_{ki}x_i$. Note that the simplification from the second last step to the last step

can only be done if A is symmetric thus we get the gradient of $x^T Ax = 2Ax$. Similarly on further differentiating every element of the gradient by x_k we can derive the hessian of the function. The hessian of this function comes out to be $2A$.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction of Machine Learning

Lecture 9

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Statistical Decision Theory – Regression

Hello and welcome to this module on statistical decision theory. So the goal here is try to give you a framework that we will keep using for the rest of the course or at least for the majority of the rest of the course and introduce you to some of the basic notations and also to talk about some kind of a unifying idea behind what we will look at in different classification algorithms and regression algorithms.

To set the tone let's consider the inputs which will denote by X as being drawn from some p dimensional space so which we will call \mathbb{R}^p . So if you think about what we did in the previous modules, we talked about input that had age and income as the attributes, so that would mean that p was two dimensions. So one of the dimensions represented age and the other dimension represented income.

So what we are doing here now is trying to move to a more general setting where I am talking about any kind of a p dimensional space and what p could be much larger than two and the output that we are going to be looking at least in the initial regression case that we will see I will assume that the output is drawn from the real numbers again. So this will be like the temperature that we saw in this second example in the previous modules.

(Refer Slide Time: 01:37)

Handwritten notes on a chalkboard:

- $X \in \mathbb{R}^p$ Input
- $Y \in \mathbb{R}$ Output Regression
- $\Pr(X, Y)$
- $f(x) : \mathbb{R}^p \rightarrow \mathbb{R}$
- $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$
- $x = (x_1, x_2, \dots, x_p)$
- $f(x) = \beta_0 + \sum_{i=1}^p \beta_i x_i$
- $\text{Let } x_0 = 1$
- $f(x) = \sum_{i=0}^p \beta_i x_i$

So the input X is drawn from a p dimensional real space and the output Y is drawn again from the real numbers and in the case of regression. So the case of classification will see the little bit later the output will come from a discrete space and we will also make an assumption that the data comes to you from some kind of a problem joint probability distribution $\Pr(X, y)$.

So you do not know this joint distribution apriori and so nobody tells you what is the distribution from which the data is coming. But the assumption that we are going to make is that there is an underlying data distribution like a joint distribution over the inputs and the outputs and that it is fixed. You are going to be given samples drawn from this probability distribution $\Pr(X, y)$. So this will be your training data which you will use for both training and possibly for validation if required.

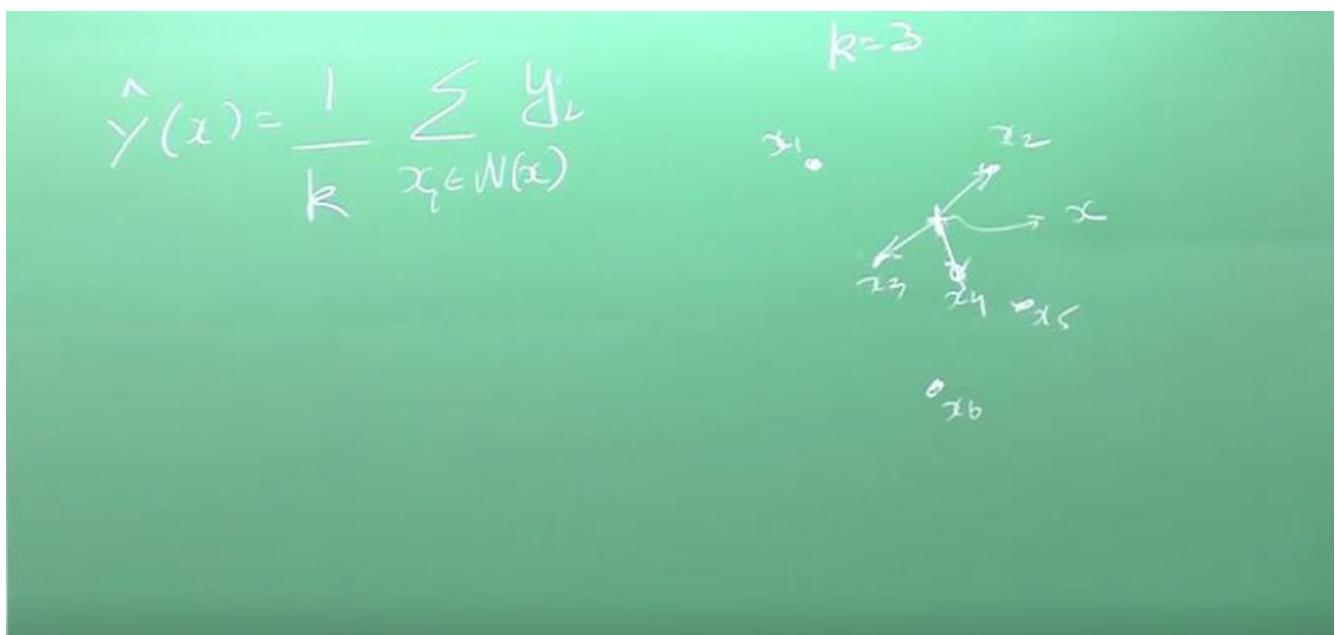
So you are going to get an x_1 with a corresponding y_1 , x_2 with the corresponding y_2 and so on so forth. So the goal is given such a set of training data we have to learn a function $f(x)$ that goes from a p -dimensional space to the real line, where the p dimensional space essentially corresponds to a point in the input space and the real line corresponds to the output space. So the function f is going to take any input that is given to it and produce a number. So the f could take

different forms for instance we looked at f being a straight line in the example that we saw earlier.

So one example of f would be saying that I am going to predict y where $y = f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$. So one thing which I want you to note here is that this (x_1, x_2, \dots, x_p) are essentially the coordinates of X . So when I say X here, X essentially comprises of age, income etc. One of these corresponds to a different attribute that describes the data.

So I can look at this and then I can write $y = f(x) = \beta_0 + \sum_{j=1}^p x_j \beta_j$. An alternate way of writing this is to set $x_0 = 1$, and then I can just remove this special treatment of β_0 and I can just write it as $y = f(x) = \sum_{j=0}^p x_j \beta_j$. So this is essentially what do you do in when you are doing linear regression. So another example of doing this is a very popular classifier which we will call the nearest neighbor classifier.

(Refer to slide time 8.32)



Here, $y = \frac{1}{k} \sum_{x_i \in N(x)} y_i$. So let us assume that my training data looks something like this. Take note

of the different points labeled as x_i 's. Let us say my k is 3, so if I get a query point (middle x) say somewhere here I get a query so this is my x and my training data the x_1 to x_6 are my training data and x is the point for which I want to predict the output. So these are the places which I have already measured it. This is a new point and I want to produce the output here so in this case what do I do? I pick the three nearest neighbors because k is 3, I pick the 3 points that are closest to this data point right find the corresponding y 's. So in this case I will pick y_2 , y_3 and y_4 and I will take the average of these three points and I will report the value of the function at x . That will be the average of this three point. So this is called the k -nearest neighbor regressor.

So I will just take the average of the outputs of y_2 , y_3 and y_4 and report that as the value of the x . So depending on where x is, I will be picking different three neighbors and reporting their values. This is the k -nearest neighbor algorithm. So there are different ways in which you can define this function f but remember that we had this discussion in the last set of modules that unless you make an assumption about the form of f you really cannot do any generalization. We needed to talk about lines in the previous class but now we are talking about different assumptions for the function f need not necessarily be lines. In the previous case it is a straight line but in this case it is an average. It is a local averager and that gives me the function that I want to learn.

So how do you choose this function? So there could be many different ways in which you can define the β 's. Given that I have chosen that this is the way to model the function how do I pick the β ? So how do I even choose this form for my predictor and how do I know that this is a valid form to choose? So we have to look at some performance measure which we will consider in this case as the loss function. This will compare the true output y right with the predicted output $f(x)$. I have the true output y and I have a particular $f(x)$, so I will have some loss function that compares $f(x)$ with y . My goal is to find an $f(x)$ such that this loss function is minimized. One of the most popular loss functions that people use in the literature is known as the squared error. So basically I am interested in is the expected prediction error of the function f . That is equal to the expected value of $y - f(x)^2$. In the case of squared error so the expected prediction error is $E\{(y - f(x))^2\}$.

(Refer to slide time 12.55)

$$\begin{aligned} \text{Lossfn: } & L(Y, f(x)) \\ \text{Sq. Error: } & (Y - f(x))^2 \\ EPE(f) &= E \left\{ (Y - f(x))^2 \right\} \\ &= \int (y - f(x))^2 \cdot P_{Y|X}(dy | dx) \end{aligned}$$

So what is the distribution with respect to which you are taking this expectation? Whenever we talk about the expectation of a random variable we want to talk about the underlying distribution. So what is the distribution with respect to which you are taking this expectation? The answer is exactly the joint distribution of X and Y or $P(X, Y)$. So I can do a little bit more sleight of hand here right and talk about the conditional distribution. If you remember $P(X, Y) = P(Y | X)P(X)$. This is just the product rule in probability. So what does this tell me that okay there is some chance with which I can choose a data point X and having chosen a data point X so what is the probability of seeing a particular output value.

So why are we looking at probabilities here again? So this helps us to kind of you know model a variety of different scenarios. The first one is that if there is noise in the measurement then we should we do?

I am talking about $P(Y | X)$, suppose I am telling you that I am measuring the temperature at 3 o'clock every day so there will be some kind of a natural variation in the temperature is measured at 3 in the morning right. So that is modeled by this probability and there will be some set of temperatures that are very probable and some set of temperatures that are not. So for example if I am measuring temperature at 3 a.m, 40^0 C is not a probable value. So those will have lower

probabilities and then say something in the 20s will have a higher probability. So, I am talking about Chennai if people are wondering how you are getting 20^0 early in the morning.

The second factor is that this allows us to look at is our ignorance about the whole system. So I might have just chosen the time of day maybe there are other factors I should have taken into consideration while I am forming my data, so these factors about which I do not know anything will appear as noise. So it is not important whether I take the temperature at 3:00 a.m, maybe it is important where in the building I do the measurements. Maybe I am measuring it next to the kitchen where things will be warmer or maybe I am measuring it next to an air-conditioner where things would be actually warmer. If I am measuring it on the external of the building or it could be measuring it on the inside of an air-conditioned room the temperatures could be lower so even though I say I measure it at 3 a.m. there could be many such factors for natural variations which I have not modeled. So this is beyond the natural variations in the system. One way of arguing about it could be to say that hey the natural variations are due to factors that you do not know anything about. So that is a valid argument so it could very well be that. So one might argue that really there is nothing like a natural variation and there is no real noise so all the uncertainty in the data arises from my lack of knowledge but that is a philosophical question.

So there are things that are measurable which we do not measure right and that I would call as lack of knowledge and things which are immeasurable which I would call as noise. There could be both of these sources which introduced the probability into system. So it is not just a mathematical whimsy that we model this as a joint distribution but there is an actual practical reason for talking about probability distributions. So now I can go back and write my expected prediction error as $EPE = E_x E_{Y|X} ([y - f(x)]^2 | X)$.

It is the same quantity earlier the only difference is now I am conditioning it on the value of X . So what this expression says is I will tell you what X is, then you tell me what the error will be? So the uncertainty here is over the value of Y . I will give you X , you tell me what Y is? I am going to look at the error just after conditioning on X this will only look at the variation on Y and the outer expectation gives me the variation of X .

Now I can try to find the minimum of this prediction error by conditioning on a specific value for X . So I will not look at this expectation, I am not making any assumptions about f and I am just

assuming that f can be anything like any function in the world. What I want to do now is I want to look at each and every value of X and I want to say that I will pick an f such that for every value of x it makes the best possible prediction. So what does that mean? It will produce the prediction so $f(x)$ for a specific value of x , $f(x)$ will give the output such that this inner expectation is minimized.

(Refer to slide time 19.55)

$$\text{EPE}(f) = E \left\{ (y - f(x))^2 \right\}$$

$$= \int (y - f(x))^2 \cdot P_{Y|X}(dx, dy)$$

$$P_{Y|X}(x, y) = P_Y(y|x) \cdot P_X(x)$$

$$\text{EPE}(f) = E_x E_{y|x} ((y - f(x))^2 | x)$$

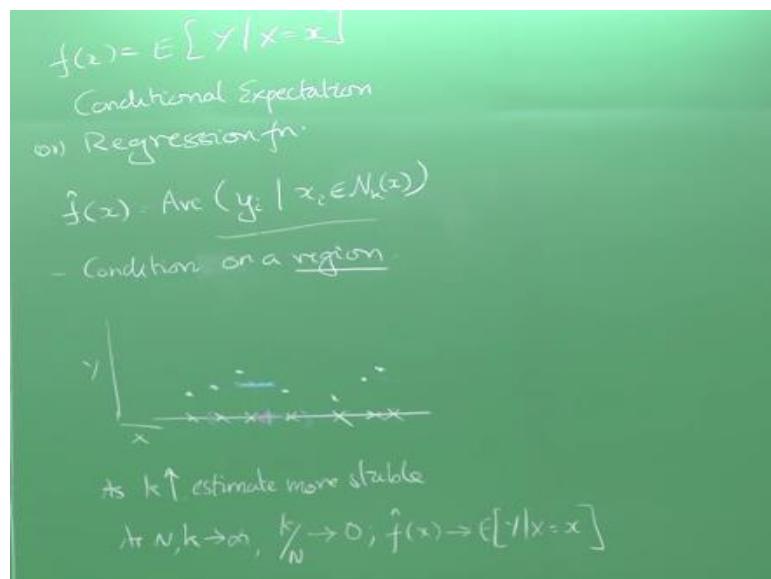
$$f(x) = \arg \min_c E_{y|x} ((y - c)^2 | x)$$

So I am going to write it down like this. For a specific value of x this is $f(x)$ for a specific x . I was writing capital X which is a random variable but here I am using a specific x . So given an x , $f(x)$ has to be a specific number. Let us say somebody with an age of 25 an income of 15,000 rupees walks into my shop I can only give one output. Whether that person is going to buy a computer or does not buy a computer or I am going to say I am measuring the temperature at 3 a.m what will be the value? And I can give you only one number since I have already fixed the input description, so I can give you one output corresponding to that input description. Let me call that output as c . So it should be such that the error which is $(y - c)^2$ is as small as possible. I am minimizing over the different possible values of c that I could assign for $f(x)$. I am trying to pick that c which gives me the smallest error. So $\arg \min$ means first minimize with respect to c .

and take that argument which achieve this minimum. If there are multiple values that gives you the minimum, I can pick any one right. So this is essentially called conditioning on a point. Instead of conditioning on the random variable X , where we conditioned on a specific point where $X=x$, I can find this. So now what happens for every possible input X that I could have small x , I will find the corresponding c and I will say $f(x)$ equal to that c . The thing to note here is I have not made any functional assumptions about what this what should f look like.

So f could be something really, really jagged I do not care. This is a recipe for disaster as we saw earlier that you might end up over fitting the data, but just work with me here because we are just trying to build some general principles. We can go little further right now that we have decided to say that the minimizer is the one that that you have assigned to $f(x)$ so what is the value of c that will minimize this expression? So I have to look at $(y - c)^2$ and I have to assign a single value for c . Suppose I give the input as x , I make a measurement let us call it say y_1 and I give it same input x again I make another measurement say y_2 . I give the same input x again I make another measurement y_3 . So I have three measurements y_1, y_2, y_3 for the same input x . Right now I am asking you to give me a prediction for what will be the output given x .

(Refer Slide Time: 23:38)



So what should your prediction be? It will be the average of these three or $(y_1 + y_2 + y_3)/3$. Why is that the case? Because we are talking about squared error, the quantity that will minimize the squared error is essentially the average. I will end up writing that $f(x) = E[Y | X = x]$. So this is essentially what my prediction would be. This is known as the conditional expectation or also sometimes called the regression function.

What are some of the problems with this equation?

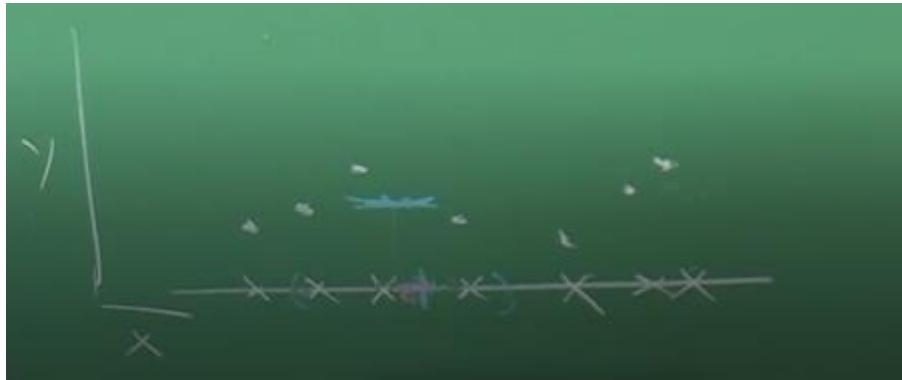
- a. I do not know the distribution with respect to which I am taking the expectation.
Or I do not know $\Pr(Y | X)$. If I know that, then my life would be a lot simpler.
I actually have to estimate it from the data. What is the data that is given to me?
I have this pairs of $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. That is the data that has been given to me and I have to do this estimate of this expectation from that data. So how will you do that? You know that you can always estimate the expectation by taking averages so what you would do is from your data you pick all the training data points that have this value of x right find the corresponding y take an average. So one simple way of thinking about it is to say that okay I cannot find the true f so I am going to find an estimate of that which is called as \hat{f} which is equal to average of all the y_i 's such that x_i equal to x or $\hat{f}(x) = \text{Ave}(y_i | x_i = x)$.

There is a problem here. First, how many samples do you think you are going to get of the same input x . Second, you are not going to be able to make a prediction for any data point which is not there in the input. We are trying to make an estimate of the expectation by using the averages but if you don't have enough measurements than your average is going to be bad and second thing is you are making an average at that point and if the point does not exist in your training data you are not going to be able to return an estimate for it. S

So we need to address this somehow. What we will do here is we will relax the conditioning. Instead of conditioning on a point we will be conditioning on a region. So I am taking the average here of all those data points for which x_i equal to x . Now that is not going to work because there are too few data points. What I am going to do is I am going to take this as the average of all the data points which belong to some region around x which is essentially the

neighborhood that we are talking about. That circle there would correspond to the neighborhood around x and I am going to be conditioning on this region which is given by this neighborhood around x . So we are not going to condition on the point we are conditioning around the region so the one assumption that we are making is an implicit one. Why are we conditioning around a region so that instead of taking an average of one data point, I have at least k data points of which I will be taking the average. That gives me a better estimate of the expectation. That is the reason we are doing this conditioning over a region but more importantly we are also making an implicit assumption. If you remember our inductive bias said that we needed to make assumptions. The one we are making here is that the output of the function over this region is going to be a constant. Let us let us try and do a little example so that becomes a thing clear to people. Let me go back to my one-dimensional example so it makes it easier for me to draw things. I have let us say I have multiple data points like this. I have a query point and then the corresponding outputs. So these are the y 's this is x and that is y so these are my x_i 's and y_i 's. This is the training data I have and now given a query point let us say I am given a query point here. I want to know what is the output value for this x . Let us say I pick my three nearest neighbors which would be these three data points right and then I will try to take the average of this, this and this which will be somewhere here.

(Refer to slide time 31.40)



I am going to make cases for variety of data points but one thing which I want to point out here is that I assume that my data point lies here so what if I had assumed that my data point was here if my query point was here so what would have been the output. So my neighbors remain the same. The three neighbors do not change and these are my three nearest neighbors whether the query was here whether the query was here or the query was here my nearest neighbors do not change. So what will be the output for this input point? I will be taking the average of these three

points so the output will be the same. In fact for a certain region around here where these three are the nearest points the output will be a constant. I said output will always be a constant so this is what I mean by saying that we make the assumption that the output is constant in a region. For all those data points for which these three are the nearest neighbors, the output is going to be here. So this is essentially the assumption we are making that the output is going to be consistent over a region. I can write an expectation over the region as my substitute. If you think about it so what have we come up with here this is essentially your nearest neighbor classifier. You take the generic idea of minimizing the expected prediction error and then add certain conditions to it. You are going to take averages and you cannot do an average on the training data and therefore you are going to do an average over a region assuming that the output is constant over a region. Relaxing the conditioning on a data point principle gives us conditioning on a region. This is also known as the nearest neighbor classifier. So in some sense you can argue that one way of minimizing the expected prediction error yields a nearest neighbor classifier.

In fact it is a very powerful classifier and you can show that as k increases so the estimate becomes more and more stable. So for small changes in the input data the classifier does not change tremendously and so as and as n and k tends to infinity or they become large , your ratio at $\frac{k}{n} \rightarrow 0$ in such a case your $\hat{f}(x) \rightarrow E[Y | X = x]$. As K increases the estimate becomes more stable in particular as k and n becomes large and my number of data points is very, very large, the number of points I can look in the neighborhood also becomes larger and larger such that the data points have to grow at a faster rate than the size of the neighborhood. That is what $\frac{k}{n}$ means. In which case I can show that my actual prediction I make using this average actually approaches the true prediction that I am interested in.

There are a few caveats here that I need to point out. I assuming I am saying that n goes to infinity. That is a pretty blank statement to make because n rarely goes to infinity. In fact coming up with large data sets are hard except for very rare cases, therefore you cannot really have a classifier that gives you the right output.

And the other problem is as p becomes larger as the dimensionality becomes larger. Generally the data tends to becomes sparse, so if I am looking at k neighbors like a thousand dimensional space, the area or the volume covered by these k neighbors would be very large because they are very sparse space and it is usually not a good assumption to make that the output is a constant over this large area.

One thing is if p is large then if the dimension of the input is large if you have like 10,000 dimension vector as your input then using k nearest neighbors is not really a good idea. Alternatively you should also remember that in some cases having a little bit of a bias is actually not a bad thing and therefore we have to look at an appropriate way of representing the function f . Remember we did not make any assumptions about the function f so the function f could change as drastically as we want. That means that we are trying to keep the bias as minimal as possible.

(Refer Slide Time: 37:03)

$$f(x) = x^T \beta$$

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & \dots & \dots & x_{2p} \\ \vdots & & & \vdots \\ x_{n1} & \dots & \dots & x_{np} \end{bmatrix}$$

$$(y - X\beta)^T (y - X\beta) = \text{EPE}(\hat{\beta})$$

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



We would like to remove that assumption. Moving on let us look at the linear regression case where we actually made a significant assumption about the form of the function f . Specifically we assume that f is going to be linear in the input parameter. So f can be written as $y = f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$. Essentially $f(x) = x^T \beta$. And so if you look at it from the training data point of view I can think of having a vector notation for this. I can think of a matrix in which each row corresponds to an input x_i so X will be something like
(Refer to slide time 37.59)

$$f(x) = x^T \beta$$

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & \dots & \dots & x_{2p} \\ \vdots & & & \vdots \\ x_{n1} & \dots & \dots & x_{np} \end{bmatrix}$$

β would be a vector of the coefficients so $\{\beta_0 \dots \beta_p\}$ and the zeroth dimension is going to be 1. So the final equation would be $(Y - X\beta)^2 = EPE(\hat{f})$. I can minimize this by taking the derivative and then I can equate it to zero. I can do some minimization to get the value of β and that is essentially taking the differential of this equate it to zero simplify for β . I am going to get $\hat{\beta} = (X^T X)^{-1} X^T Y$. Remember that the X that we put in here is essentially a matrix where the rows are the data points and the columns are the features.

So this will be the like the age of every customer that comes and this will be the income of every customer and so on so forth and each row is a complete data point. So what we have done here is make the assumption that my function is globally linear and then I have tried to solve for it to give you the parameters $\hat{\beta}$. In the nearest neighbor case we made the assumption that my function is locally constant.

So we start off with the same formulation we wanted to minimize the expected prediction error and we make different assumptions. One assumption leads us to linear regression the assumption that we made was the data is going to be globally linear and another assumption that we made where the data is going to be locally constant gives us k nearest neighbor.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

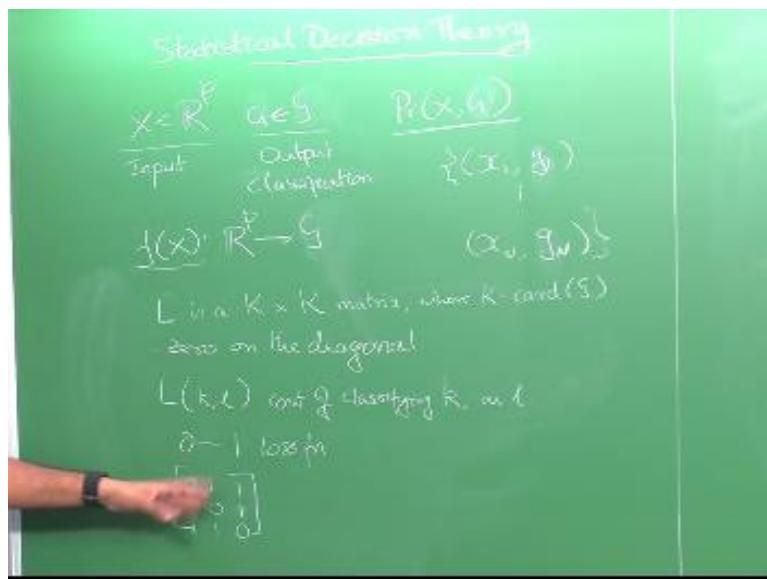
Introduction to Machine Learning

Lecture 10

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Statistical Decision Theory –Classification

(Refer Slide Time: 00:16)



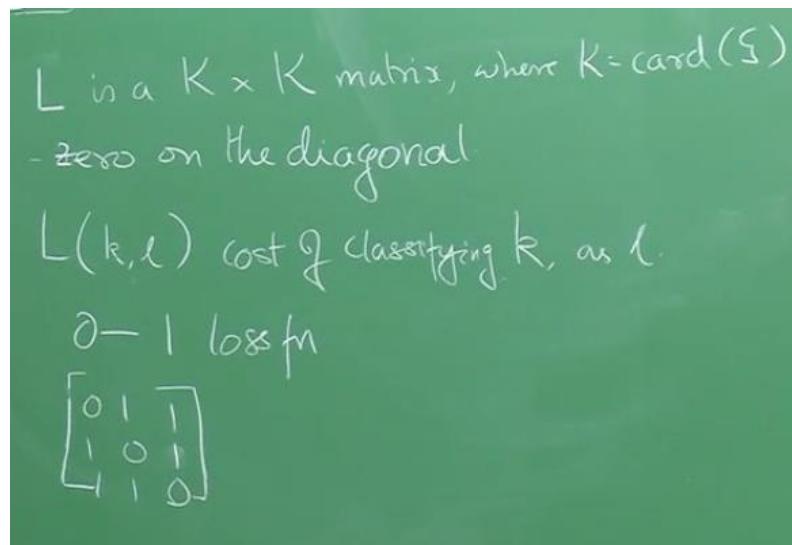
In this module we are going to look at the case where the output variable is drawn from a discrete space or in other words we are going to look at the classification problem. As before the input is coming from a p dimensional space R^p and the output which I am denoting by g here, I am going to assume is coming from some space G which is a discrete value. It could be “Buys a computer” or “does not buy a computer” so g could just consist of buys a computer does not buy computer or it should consist of like 5 different outcomes “has the disease”, “a mild form of the disease”, “a severe form of the disease”, “does not have the disease” and so on so forth right.

It could be a variety of outcomes but it is a small discrete set. So that space is denoted G which is the random variable corresponding to the output, then like before we are going to have a joint distribution on the input on the output. The training data is going to consist of $\{(x_1, g_1), (x_2, g_2), \dots, (x_n, g_n)\}$ and the goal here is to learn a function $f(x)$ that is going to take you from a p -dimensional input space R^p to the discrete space G .

And so the thing that we have to look at now is what is an appropriate loss function in this case. What is an appropriate loss function in this case since we are talking about the discrete output? So I really cannot talk about squared error as a loss function even though in cases where the discrete values have been encoded as numeric outputs people do use squared error and we will see that later. So people do use squared error is an appropriate measure as long as your space G has been encoded numerically.

So but in general so we are going to define the loss in as a $k \times k$ matrix where k is the cardinality of the discrete space G that we are looking at. Suppose there are 5 classes then my last matrix is going to be a 5 by 5 matrix. so it will have zeros on the diagonal and so the kl^{th} entry $L(k, l)$ in the loss matrix essentially is the cost that you incur of classifying the output k as l . So the true output is k but you output you say l so that is essentially the cost of classifying k as l . That is denoted by the kl^{th} entry of the loss matrix.

(Refer slide time 3.58)



So frequently the most popular loss function that you use is known as the $0 - 1$ loss function. The $0-1$ loss function essentially says that suppose I have three classes, so my loss function would

look like this, $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. So if I if I classified to the right class I get a penalty of zero but if I classify to the wrong class right I get a penalty of one regardless of which wrong class I classify to. So this entry says that okay the data point actually belongs to class one I have classified it as class two what is the penalty so 1 data point belongs to class 1 I classify it as class 3 what is the penalty one and so on so forth. This is called the $0-1$ loss function because all the entries in the loss matrix are either 0 or 1.

(Refer Slide Time: 04:26)

$$\text{EPE}(f) = E_{x \sim P_X} [L(f(x), g(x))]$$

$$= E_{x \sim P_X} \sum_{k=1}^K L(g_k(x), f_k(x))$$

$$f(x) = \arg \min_{\Delta} \sum_{k=1}^K L(g_k(x), f_k(x))$$

0-1 Loss

<u>3 classes</u>	
$P_1(x=1 x) = 0.6$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$
$P_2(x=2 x) = 0.2$	
$P_3(x=3 x) = 0.2$	

$$g(1) = \frac{0+2}{3} = 0.67$$

$$g(2) = \frac{1+0}{2} = 0.5$$

$$g(3) = \frac{1+1}{2} = 0.9$$

So what we are again going to look at is the expected prediction error of \hat{f} , or,

$$EPE(\hat{f}) = E[L(G, \hat{f})].$$

We can do the same thing that we did earlier so I can start conditioning it on x and the expected prediction error and then the expectation of g given x which essentially becomes

$$EPE(\hat{f}) = E_x E_{G|x} \{L[G, \hat{f}] | x\}.$$

So the loss of g , \hat{f} given that the input is x but if you think about it this is not a continuous distribution this is actually a discrete distribution because G can take only finitely many values. So instead of writing it out as this expectation I can actually simplify that and write it as

$$E_x = \sum_{k=1}^K L[k, \hat{f}(x)] \Pr(k | x). \text{ So this is a loss that I will incur if } k \text{ was the true class and my}$$

prediction was $\hat{f}(x)$ times the probability that k is the true class given the input x . So here I am essentially writing out the expectation because it is a discrete distribution. I am able to write it out in a compact form and again I can do point wise minimization of this like we talked about earlier. So point wise would mean that I make a specific assumption about what is the value of x .

$$\text{So I am going to look at } \hat{f}(x) = \arg \min_g \sum_{k=1}^K L(k, g) \Pr(k | X = x)$$

We are essentially following the same treatment that we did with the regression case except that we are using a discrete output space since of a continuous output space. So this essentially says that I am going to pick the prediction g that gives me the smallest expected error. Suppose, I have the 0-1 loss function, so what does this mean? I should essentially set my g to be that k which has the highest probability. So if we think about it this probability term $\Pr(k | X = x)$, contributes to every element in the summation. So what I can do is among all these probability terms I can pick one term and set it to 0 by my choice of g . Suppose I choose g to be 1 then my $l(1,1)$ will become 0 and but my $l(2,1), l(3,1)$ so on so forth will all be 1. What will happen then is that $\Pr(2 | x), \Pr(3 | x)$ all of this will actually appear in this summation.

(Refer to slide time 8.30)

$$\begin{aligned} EPE(\hat{f}) &= E[L(g, \hat{f})] \\ EPE(\hat{f}) &= E_x \left[\sum_k L[k, \hat{f}(x)] \Pr(k | x) \right] \\ &= E_x \sum_{k=1}^K L[k, \hat{f}(x)] \Pr(k | x) \\ \hat{f}(x) &= \arg \min_g \sum_{k=1}^K L(k, g) \Pr(k | x) \end{aligned}$$

So if I set my g to that value of k which has the highest probability then that will yield the best possible solution here. If you are not able to see that let us assume that there are 3 classes. I assume that there are 3 classes so and my true distribution is says that the $\Pr(1|x) = 0.6, \Pr(2|x) = 0.2, \Pr(3|x) = 0.2$ and of course my loss function is going to be such that

$$\text{and } \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

So if I guess that my class label is going to be 2 let us say so I said $g = 2$, what is going to happen? If the class label is 1 right so I am going to look at the loss corresponding to 1, 2 which is the loss matrix entry (2,1), so I will get 1 times 0.6 then if the class label is 2 so I will be looking at loss matrix entry (2,2) I will get 0 times 0.2 + if the class label is 3 I look at loss matrix entry (2,3) I will get 1 times 0.2, so I will get a score of 0.8.

So as you can see depending on which value I choose, say, if I choose $g = 2$ then I will be zeroing out the second entry if I choose $g = 1$ I will be zeroing out the 1st entry and by choosing g equal to 1 I will basically get a score of 0.4. So what I have to do in order to get the minimum here is to pick that g for which this probability is the highest.

(Refer Slide Time: 11:31)

The image shows a handwritten derivation of the Bayes Optimal Classifier formula. It starts with the expression for Expected Posterior Error (EPE) under 0-1 loss:

$$EPE(g) = E_x [L(x, f(x))]$$

This is expanded using the law of total expectation:

$$= E_x \sum_{k=1}^K L\left[k, f(x)\right] p_k(k|x)$$

Then, it is shown that this is equivalent to the expected loss given the classifier $f(x)$:

$$f(x) = \arg \min_g \sum_{k=1}^K L(k, g) p_k(k|x)$$

Below this, the 0-1 loss is defined:

0-1 loss.

The final result is the Bayes Optimal Classifier formula:

$$\hat{f}(x) = \arg \max_g \Pr(g|x)$$

Below this, the text "Bayes Optimal Classifier" is written.

At the bottom, there is a note about k-NN:

k-NN = Pick k nearest neighbours & take majority.

So now can you realize why the min here became the max? This is based on the argument that we just did, so this is essentially saying that from your training data classify it to the most probable class and if I knew this probability, what will I do? I can set it to the most probable output. So this is this kind of a classifier which is known as the Bayes optimal classifier. It says that I can look at the conditional distribution given x look at the probability of g , take the g that has the highest probability and assign it as the output so this is essentially what the Bayes optimal classifier would say.

But then you do not know g . So what we have to do is you have estimate this probability. How would you estimate this probability? Do we know of any method for estimating this probability? Of course we do. We know how to do nearest neighbor, so what you would do in this case is that instead of taking the average over the neighbors like we did in the regression case you would do estimate the probabilities in the neighborhood. What you would do is that you will take a data point look at the k nearest neighbors of the data point find out what their class labels are and then for each label count the number of occurrences of that label in the k neighbors and divide by k . This will give you the probability of the class label in the neighborhood but we really do not have to do this much work because we are not interested in the actual probability. All we need is the one that has the maximum probability since the denominator is going to be k for all the probabilities we can ignore the denominator we can just look at the numerator.

So what we can do is we can count the occurrences of the class label in the neighborhood and whichever occurs more often we can assign that as the class label. Think about it for a minute. What we are essentially doing when we take the majority is actually estimating this probability and taking the max probability so take the majority label in the neighborhood and use that as your prediction. So this essentially gives you the k -nearest neighbor classifier.

Like earlier, all the caveats that we talked about for the k nearest neighbor regressor apply to the k nearest neighbor classifier as well. You have to be careful about using it in very high dimensions and you really need large values of k and large values of n before you can get stable estimates. But having said all that I should say that it turns out to be a really powerful classifier in practice and we will come back to that a little later as to why it is such a powerful classifier.

Now can we use linear regression or the linearity assumption here? It turns out that you could use linear regression in almost directly for solving this problem. So the way you do it is the following. You take this data set $\{(x_1, g_1), (x_2, g_2), \dots, (x_n, g_n)\}$ and convert it into a data set suitable for doing regression, so how do I do that? I take that (x_1, g_1)

(Refer Slide Time: 16:10)

$$\begin{aligned} & (x_1, 1) \\ & (x_2, 0) \\ & (x_3, 1) \\ & (x_4, 0) \\ & \vdots \\ & (x_n, 0) \end{aligned}$$

$$\hat{f}(x) \geq 0.5 \text{ class 1}$$

$$\hat{f}(x) < 0.5 \text{ class 0}$$

Let us say that I have only two classes for simplicity sake let us say I have only two classes. I have g_1 and g_2 . I will say I will say that 0 or 1 right so instead of having some arbitrary classes I am going to say it is 0 or 1. so what I am going to now do is my thing will become something like this right so instead of having some arbitrary symbols g 's (g_1, g_2), I am going to have $(x_1, 0), (x_2, 1), (x_3, 1), \dots, (x_n, 0)$. What I can do is I can solve this as a regression problem I can just solve this as a regression problem and whatever output I get I can treat that as an estimate of the

$\Pr(g = 0 | x)$ or $\Pr(g = 1 | x)$. So how to find the probability of $g = 1$ given x . For the same value of x , say, if there are multiple ones. Suppose I the same value x occurs say 5 times in my training data 3 of the times it was 1 and 2 of the times it was 0 , when I am trying to do a prediction I would expect to end up at the average of this prediction right, so just be like $3/5$ and it also turns out to be the probability with which the output is 1 given an x . So if I do regression with this as my training data, what I will be learning is the probability that $g = 1$ given x . Roughly there are lot of caveats in this which we will look at when we do regression later obviously you cannot treat this directly as probabilities because the regression curve can become negative.

You cannot really treat it as probabilities but it is a useful intuition to have. So the output that you learn here is $\hat{f}(x)$. In this case if it is ≥ 0.5 then you say in the class is 1 if it is $<$ than 0.5 you say the class is 0. Let us say can use a linear regression to solve this as well so what we have done in this couple of modules is to look at a unifying formulation for classification and regression problem and looked at a couple of different classifiers that arise out of making certain assumptions about classifiers and regressors that arise out of making certain assumptions about the function that we are trying to learn.

In the subsequent classes we will start looking at each of these in more detail starting off with linear regression. We'll look at this different classifiers in greater detail thank you.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 11

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Bias-Variance

(Refer Slide Time: 00:15)

$$\begin{aligned}
 J &= E[(y - f(x))^2] \\
 &= E\left(\hat{f}(x) - f(x)\right)^2 + \underbrace{\left(E[\hat{f}(x)] - f(x)\right)^2}_{\text{Bias}^2} + \underbrace{\sigma^2}_{\text{Variance}} \\
 &= \frac{\sigma^2}{k} + \left[f(x) - E\left\{\frac{1}{k} \sum_{i=1}^k \hat{f}(x_i)\right\}\right]^2 + \sigma^2
 \end{aligned}$$

I will give a very preliminary introduction on bias variance in this class and later on, as we progress we will come back to this. So let us start off with the assumption that and in many cases I will be looking at regression because it is easier to write but you can extend similar concepts for classification also. I am going to assume that your actual data is being generated by a system of this form $y = f(x) + \varepsilon$. So there is a function f which is what you are trying to learn about but the data that is given to you or the y 's that are given to you are actually corrupted by some kind of noise.

If you remember from the last class or at least the last class I was teaching, we were talking about a joint distribution over y and x . You do not know what the joint distribution is, you are

only given samples drawn from that distribution. Here we are making a specific assumption about the form of the joint distribution. I am assuming that there is some kind of an underlying deterministic function f here which is operating on my input x .

But then it is corrupted by some stochastic noise which we will call epsilon. And that gives me the joint distribution over X and Y and we are going to assume that $E[\varepsilon] = 0, \text{Var}[\varepsilon] = \sigma^2$.

So the expected prediction error at some point x_0 right is $EPE[x_0] = E[(y - \hat{f}(x_0))^2 | X = x_0]$. So it turns out that I can rewrite this expectation as a sum of three terms. So what are the three terms? The first term is essentially the error that I am going to see by looking at the estimate that I will get from a specific data instance or from the estimate that I will get as an expectation over the sample from which the training data is being drawn. So if I build a classifier multiple times so this ($E[\hat{f}(x_0)]$) is the expected output that I am going to get for x_0 and this ($\hat{f}(x_0)$) is the output I am getting for this specific instance of data that I have. So that is one component of the error the other component is, look at the expected prediction I will make for x_0 taken over multiple training instances ($E[\hat{f}(x_0)]$) minus the expected output I am going to get or the true output ($E[y]$) I am going to get. What is expected true output I will get? That is $E[y]$ and what will be the $E[y]$ in this case? That is $f(x_0)$. Then there is an underlying error σ^2 , that just comes from the fact that I have a variance of σ^2 . I am going to make any single prediction even if I am going to give you the output as $f(x)$, there will be an expected error or σ^2 because my y has that noise in it. So this term $E[\hat{f}(x_0)] - f(x_0)]^2$ is typically called the bias. And this term $E(E[\hat{f}(x_0)] - \hat{f}(x_0))^2$ is typically called the variance of the estimator.

Or $EPE[x_0] = E[(y - \hat{f}(x_0))^2 | X = x_0] = E(E[\hat{f}(x_0)] - \hat{f}(x_0))^2 + [E[\hat{f}(x_0)] - f(x_0)]^2 + \sigma^2$

So, one way to think about it is the following. So f is my true function, and regardless of how much ever data I am getting, or regardless of whatever data I get I expect to make at least this much error from the true function f . So that is the bias and the variance. So it is essentially given a specific instance of the training data, what is the expected error I am going to make. You cannot do anything about the noise, I mean regardless of how powerful your classifier is you cannot get rid of that σ^2 because that is inherent noise in the data. So now by choosing your classifier appropriately you can trade-off between the bias and the variance. I will just for

simplicity sake take the example of our k -nearest neighbor classifier. All of you know about K-nns right. It is very easy to talk about bias and variance in KNNs. So what do you think this variance term will be for the KNN case? Since I am looking at a prediction I am making over many, many instances right and the specific prediction I make for one training set, what would that be? If you think about it, the prediction I am making is essentially just the mean of k numbers. What will be the variance of that prediction from many many different samples drawn? It will be the base variance divided by the number of samples. You should have seen that in probability theory course if you have not okay, later I will be doing a session on statistics, and a little bit on hypothesis testing and so on. So further at that point we will go back and look at it.

So I have some distribution and I take samples from it. I draw samples from that distribution p and I try to estimate the mean and variance of that distribution through those samples. So now the variance of the estimates of the mean made from this samples is essentially given by the variance of the underlying distribution divided by the number of samples which you are drawing every time. This assumes that you have drawn the k samples many times and they have made an estimate of what the mean will be right.

And this is the specific estimate of the mean this essentially the variance of that estimate right, so the variance is σ^2 / k . Now what about the bias²? $[f(x_0) - E\{\frac{1}{k} \sum_{l=1}^k f(x_{(l)})\}]^2$

So this is essentially my expected prediction this there should be an expectation here over the training data so this is the expected prediction I am going to make. So I am going to take the all the K nearest neighbors of a data point then take the average of that and that will be the prediction I am making so this is the prediction that I am going to make.

Now let us try and look at this what happens when I change my k . If I increase my k , what will happen to the variance? It will decrease. What will happen to the bias? This is an interesting question. If increase my k , essentially what is going to happen is that I am going to start pulling in data points that are further and further from x_0 therefore my estimate of $f(x_0)$ is going to be an average of a lot of dissimilar data points, so the error is going to be higher. So for a fixed dimension increasing k , is going to essentially pull in data points that are more and more

dissimilar than the query point so I am going to go further out and therefore this will essentially become larger. So as K becomes larger my bias increases in KNN and my variance decreases. Variance decreases just because I am taking an average of more data points.

There is nothing to tell you that the average is correct. Its just that I am telling you the average will look the same even if I change the training data. Because I am averaging so many data points and this part of course we cannot do anything about this is the irreducible bias (σ^2). So last class we had this discussion about increasing k . What did we say when K becomes larger I did not say anything about it becoming more correct, I said that it will look more stable.

Why does it look more stable because my variance goes down. So when I say that classifier is a more stable estimator because the variance has gone down. Also, if you look at the classification surface that you will get, the separation surface that you will get will be a lot smoother if k is very large. But I told you when k is 1, you are going to get lot of isolated islands of different classes and so on so forth and for small values of k , you will find that the classification surface is very complex like it is not like a linear thing or you can predict very complex functions also.

Easy to think of the complexity in terms of the classification surface but function wise also you can think of very complex functions if $k = 1$. If k is larger and larger, the function has to be smoother and smoother. It cannot have rapid variations in the function. That essentially means that when k becomes larger your function class becomes simpler. This kind of looks counterintuitive. I am giving you a lot of k but then your function class typically becomes simpler because it has to have all the smoothness constraints on it and as k becomes smaller, then your function class can be larger. So your regressor or your classifier is more complex. If k is smaller, it is less complex. If k is larger and in general that is the case that if your classifier is more complex your variance will be higher and bias will be lower. If your classifier is less complex the bias will be higher and their variance will be lower. So this is usually the case and this also lets us understand why k-means does not perform that well in high dimensions. Why is that? So if you take a very high dimensional data, then you can, with a little bit of analysis show that the very high probability, the nearest neighbors will be far away from any query point. If you take any query point and draw a ball around it the ball is more likely to be empty then filled.

So the radius of the ball depends on the dimension. It becomes larger and larger as the p becomes larger and so essentially it means that even for small values the bias will be high even for small k the bias will be high because the expected distance to the nearest point will be larger in the high dimensional space. So not only will the variance be high because you have small k , the bias will be high now. Basically increasing k is not helping you in that case.

This is just pretty rudimentary discussion on bias and variance tradeoff. But I just wanted to give you a feel of that and you have to keep this in mind later on as we are looking at every classifier that we will see and specifically now we are going to go into linear regression. So, what about bias and variance in linear regression? Does linear regression have any bias? It must be right as it seems to be a very simple classifier. I will talk about that later but the point is yes so any classifier that you are going to be building or thinking about in the future, you will have to start thinking about what the bias and variance okay is. Is it appropriate to use this classifier in this setting and things like that.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 8

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Linear Regression

(Refer Slide Time: 00:55)

$$f(x) = \beta_0 + \sum_{j=1}^p x_j \beta_j$$

$E\{Y|X\}$ is linear.

So there is a basic assumption that we had earlier. $f(x) = \beta_0 + \sum_{j=1}^p x_j \beta_j$ So we are going to assume

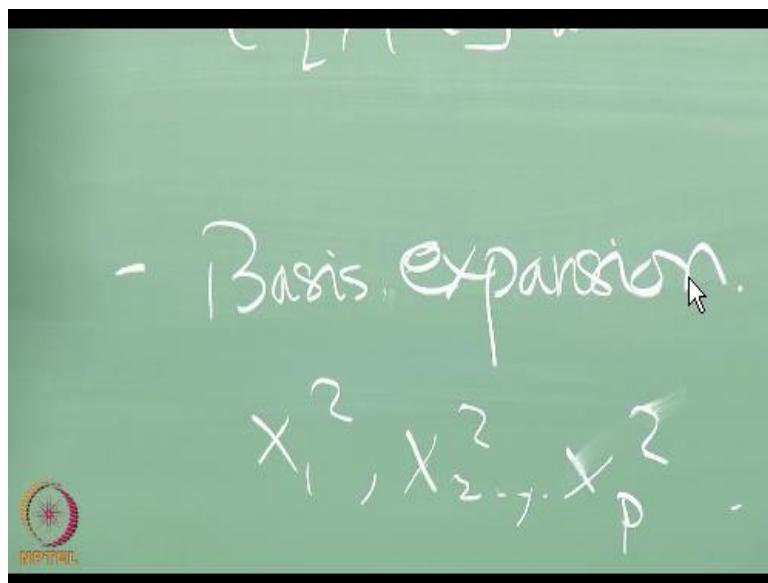
that $E[Y | X]$ is linear. So what does this mean? $f(x)$ is the expected value of y . So there is some kind of a noise corrupted training data that is given to you and $E[Y | X]$. So we had this discussion earlier, the nice thing is linear regression is not as weak as you think.

So X can be a variety of different things, X need not just be real valued inputs. They are assumed to be drawn from R^p or X are assumed to be drawn from some p dimensional real valued space.,

but they did not just be real valid inputs. They could be any kind of encoding. We talked about basis expansions, which essentially is blowing up your input space by some kind of transformation of the input variables. So if my original data is $x_1 \dots x_p$, basis expansion will be:

$x_1^2, x_2^2, \dots, x_p^2$. I could also think of interaction terms such as $x_1x_2, x_1x_3\dots$

(Refer Slide Time: 2:26)



I could think of more complex transformations $\sin(x_1)$ etc. X could be qualitative inputs as well. What I mean by that: It can be (hot, cold), (tall, short, medium height) etc. How I would handle that? Weights has levels in the input or it could be just (red, blue, green) meaning it does not really correspond to any level. I mean young and old we can think of saying 'young' is 1 and 'old' is 2 and middle age is '1.5', say, but what about (red, blue and green)? We have to encode each color. But how do you do the encoding? I could do some kind of binary encoding, so I can think of saying that okay I have four colors, so I will have two bits to encode the four colors. Four gets translated into two bits. It turns out that that is usually too much of a compression in the encoding right and if you have four possible values this thing can take it is better to sometimes use four bits. So it is sometimes called 'one of n' encoding. So, only one of those four bits will be one for any input. 'Red' means the first bit will be one blue means the second bit will be one and so on and so forth. Or sometimes it is called 'one hot encoding' because one of

the inputs will be hot the others will all be cold. So sometimes it is called ‘one of n’ or ‘one hot encoding’. So you could take care of qualitative or categorical inputs like that. And whatever you do or however you have expanded your basis or doing your encoding, finally the model you fit will be linear. Its just that if your original dimension was 1 in this case because I had a one color input, it could take four values now my input dimensions become 4. Similarly I had p input earlier now input dimensions has become the case above. It really depends on whether I am feeding in $x_1 \dots x_p$ also. If I only feed in the second order terms it is still p but the class of functions I can model is restricted. And if I feed in $x_1 \dots x_p$ as well as the squares then it is $2p$ and the class of functions I can model will become larger.

So that is basically the underlying set up. The model is still linear. Why is four-bit encoding better than 2-bit encodings? The point is so when I have two bits encoding so there will be the same input variable that gets activated for two different colors. Suppose I am using red this 01 okay and blue is 11 so that 11 will get activated for both red then blue. And likewise when there is 10 and 11 right. So so the same bit gets activated for multiple inputs. And that gives you some amount of interference in the training. We can still train it with two bits you probably need a lot more training data to take care of the interference from one to the other. When you have these kinds of 4 bits essentially you have independent weights modeling the influence of each of the levels. So for red there will be one weight and by weight I mean one β here. So for red there will be one β that will be modeling the effect of red and for blue there will be another β modeling the effect of blue. That way there will not be much interference between the variables. So technically you can model it with two bits and get away with it is that you will probably need more data for the estimation. That is why I say in practice four bits is better.

Let us continue looking at this. So my training data is : $(x_1, y_1), \dots, (x_n, y_n)$ $x_i = (x_{i1} \dots x_{ip})$.I am going to assume this of the form and that each x_i . I am going to assume that I have n data points each of the form $(x_1, y_1), \dots, (x_n, y_n)$.

(Refer Slide Time: 08:09)

Least Squares :

$$RSS(\beta) = \sum_{i=1}^N (y_i - f(x_i))^2$$

X is $N \times (P+1)$

And the way we are going to fix this is using least squares. So we are going to translate this into matrix notation to show you some things and in matrix notation when I write X at least for today it is an $N \times (P+1)$ matrix, where the first column is all ones. So we have seen this already so it is $N \times P$ matrix where the first column is all ones. So I can write it like this in matrix form as below.

(Refer to slide time 10.09)

Least Squares :

$$RSS(\beta) = \sum_{i=1}^N (y_i - f(x_i))^2$$

X is $N \times (P+1)$; rows are data points | 1st col is 1

$$RSS(\beta) = (Y - X\beta)^T (Y - X\beta)$$

The square in the first equation becomes that way in the last equation. Because $f(x)$ now becomes just $X\beta$. What would that be? It will be $\frac{\partial \text{RSS}(\beta)}{\partial \beta} = -2X^T(Y - X\beta)$

So I am going to let you think about this if you cannot see that immediately. I am going to let you work it out yourself. We should get really familiar with doing this kind of derivatives of matrices. Because we will be using this quite often whenever we write this kinds of error functions in terms of matrices be ready to use this.

Intuitively you can see it but you just need to work out the math here. $\frac{\partial^2 \text{RSS}(\beta)}{\partial \beta^2} = -2X^T X$.

And at least this seems easy enough I am taking a derivative of this with respect to β and the only term where β appears is $X^T X$. So if X has full column rank, $X^T X$ will be positive definite. So it will certainly be invertible. So it will be and no it is not just invertible it will be positive definite and therefore we can assume that it has a maxima or minima.

Now anyways so if I equate this to 0, I will get an extremism point, I will get either a maximum or a minimum. So what would it be? Anyway think about it I am going to anyway minimize the error that that should give you a clue right. So essentially I have to set this to 0 if I want to find the minima of the error and this is going to give me the following result $\hat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y$. (Refer Slide Time: 13:26)

$$\begin{aligned} & \text{Minimizing} \\ & X^T(Y - X\beta) \approx 0 \\ & \hat{\beta} = (X^T X)^{-1} X^T Y \end{aligned}$$

(Refer Slide Time: 14:16)

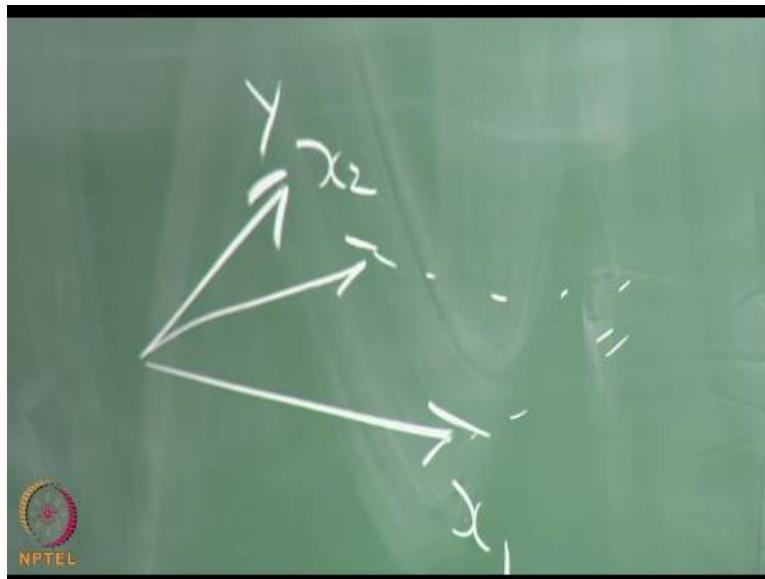
$$\hat{Y} = X\hat{\beta} = X \underbrace{(X^T X)^{-1} X^T}_{\text{"Hat" Matrix}} Y$$

This is all standard if you already know what the solution of linear regression is we saw that in the last class and you should have revised things by now.

They tell you that if you read in the previous whatever we have covered till the previous class and come the next class will be easier. So we already seen the solution and so if I put this together I basically get $\hat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y$. As shown above, this expression is sometimes called the “hat matrix”. You know why? Because it takes Y and puts a hat on it. So it is called the “hat matrix” so hat essentially is the short hand for estimates. Hats are short hands for estimates and they denote that it is not the true quantity so Y is the true random variable and \hat{Y} is an estimate of the value of Y . So this is essentially the estimator matrix. So in that sense you can think of it as a hat matrix, so another way of thinking about it is the following. So what can we say about Y ? The vector Y and not the output random variable Y . I am talking about the vector Y , I should say that. So X is $(n \times p + 1)$ and Y is $(n \times 1)$. So Y is actually a point in $(n \times 1)$, so you can take the $(p + 1)$ columns of X and X is going to have $(p + 1)$ columns. You can take the $(p + 1)$ columns of X as the set of basis vectors. So what is the dimensionality of each column? It is n . So each column is a vector in R^n and I have $(p + 1)$ such vector in R^n .

Now I can think of these vectors as a set of basis function or basis vectors. So ideally I would like them to span a $p + 1$ dimensional subspace of R^n . It is where all the linear algebra tutorial supposed to help. So you have a $p + 1$ subspace of R^n , and your $X\beta$ will be a point in that $p + 1$ dimensional subspace. Because X are my basis vectors and I am combining the basis vectors by some set of scalars β like $(\beta_1 \ \beta_2)$, all those scalars just am getting just getting a linear combination of my basis vectors, so it is going to give me some point in the $p + 1$ dimensional space. In fact if I am doing linear regression all I can do is express a point in that $p + 1$ dimensional space. If I take the columns of my X matrix any output that I can learn will be a point in that $p + 1$ dimensional space. This makes sense. So what is the best possible point in that $p + 1$ dimensional space that I can predict? So let us say I have two vectors X_1, X_2 . These are not the data points. These are the column vectors. Let us suppose that I have a vector Y which is in the n dimensional space. What is the best prediction I can make?

(Refer Slide Time: 18:45)



So Y is in R^3 and if you can buy into my drawing skills, so X_1 and X_2 span that two dimensional subspace of R^3 and Y is a point in R^3 . So that is what Y is. What is the best prediction that I can make that fits into the $X_1 X_2$ space? It will be the projection of Y . Am I making the prediction? Yes, because if you look at the error, or $Y - \hat{Y}$ is essentially orthogonal to the space spanned by

X. So that is what our minimization condition is telling us. $X^T Y - X\beta = 0$. So essentially it is telling us that ,this vector this vector is orthogonal to the plane spanned by X. That is essentially what the minimizing condition is telling. So this is the best possible estimate that you can make for y given that you are restricted to the plane spanned by the columns of X.

That make sense so this is a geometric interpretation of what linear regression is doing. Lets us think about some other things. So what happens if X is not full rank. That would mean that some of the columns are dependent on each other or linearly dependent on each other. That essentially means that it is not really spanning a $p+1$ dimensional space its spanning a smaller subspace. It is spanning a smaller dimensional subspace therefore your approximation is going to be worse. That is one part of it. Then anything else the formula would not be valid. So we have to think of different ways of doing it. So that is the next thing but still regardless of that the best fit that you can get will still be the projection of your Y onto the space spanned by the Xs. You have to have to come up with different ways finding it. But it will still be the projection so that is the thing. So one of the easiest ways of doing it is what? Now we know exactly, that it is in the space that is spanned by these vectors and we are supposed to find the projection onto the space. And if there are redundant vectors that will not help us define the space. We can throw them out but even though I have all this $p+1$ dimension. Whatever is redundant that is not helping me define the subspace I can throw them out.

So there are some very simple checks that you can do. In fact if you use some standard tools like R and you are trying to do linear regression unless you explicitly tell it not to it will automatically do the check for you it will automatically do the check and throw out the independent columns. It will pick some subset of independent bases and then use that to figure out what the projection should be. So what about the case with the number of dimensions is much larger than the number of data points? Do you think that will happen? Yes? No? Possible? How many of you here work with images or have done any work with image data? So more often than not that is the case right. So because image data is very high dimensional and unless you are able to generate huge volumes of such data and more often than not p will be greater than n .

So you have to think of some kind of regularizing the fit so that you get actually a valid answer. If p is larger essentially what it means that you have a much larger space and Y actually exists in a smaller space than what is given to you. So we have to figure out a way of regularizing the problems so that adding additional constraints on what kind of projections you are looking for. Because otherwise it does not make sense to talk about the projection of Y on this P plus 1 dimensional space okay.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Higher Education
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

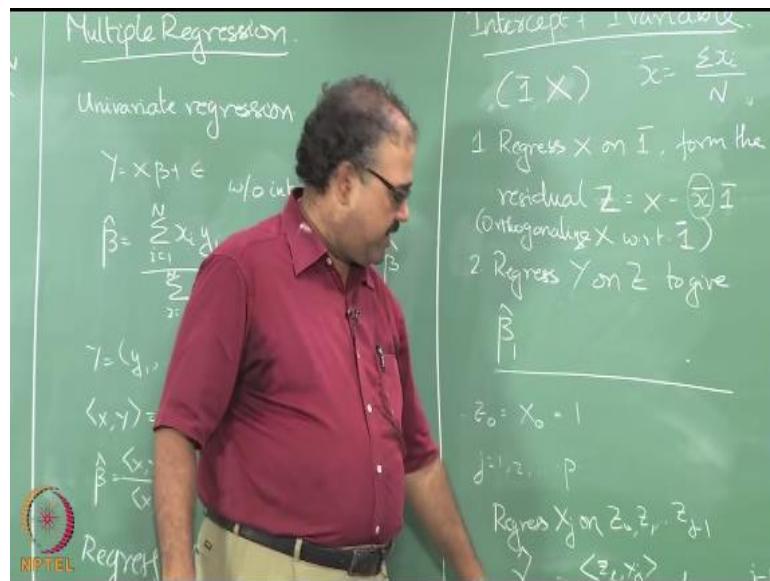
Lecture 13

Prof. Balaraman Ravibdran
Computer Science and Engineering
Indian institute of technology

Multivariate Regression

So far I have really not worried about the fact that we have multiple dimensions in the input space. That we just had this way of handling it. But then if you actually look at how statisticians typically present linear regression, they will start off with a univariate regression or they start off with one input variable and one output variable or one independent variable and one dependent variable, so the independent variable is the input variable.

(Refer Slide Time: 00:54)



So whatever we have looked at so far is usually called multiple regression. We will still typically start off with univariate regression as people usually start off with univariate regression because

it is easier to analyze you can derive a lot of intuition into what exactly is happening with the regression. In fact if you think about it, this picture I drew for you is a univariate regression with an intercept so that there is a column of ones and then there is one other variable that is all.

This is essentially univariate in the regression with the bias term. So you can very easily develop all kinds of intuitions and also the analysis will be very clean and more importantly you can understand multivariate regression by a series of univariate regressions. So let us look at it very quickly and then we will see what happens. So this is the basic model that we have $Y = X\beta + \varepsilon$. But here we are going to assume that X is a single number so it is a single vector now. So my data will be of the form some (x, y) . This is not a that is not a vector is just a simple x .

So why is it called the intercept? So the constant value you add is what the value of where it will cut the y -axis. That is why it is intercept. So I have no intercept that means there is no β_0 here.

So now this one β is given by $\beta = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2}$. Essentially our original case or

$\hat{\beta} = (X^T X)^{-1} X^T Y$ for a univariate case. I am going to denote by r_i the residual error that I am making, or $r_i = y_i - x_i \hat{\beta}$. So I made the prediction $x_i \hat{\beta}$ and y_i is the actual output that I saw in the training data, so $r_i = y_i - x_i \hat{\beta}$ is the residual error.

I hope people are familiar with the inner product notation of this form essentially. Now can you tell me what $\hat{\beta}$ should be? This is fairly simple, so one thing I just point out in passing here right I am not going to cover it if you want you can go through Hastie Tshibrani Friedman later chapters.

(Refer to slide time: 5.44)

Y = $(y_1, \dots, y_N)^T$, X = $(x_1, \dots, x_N)^T$

$\langle x, y \rangle = x^T y$ Y = $(y_1, \dots, y_N)^T$

$\hat{\beta} = \frac{\langle x, y \rangle}{\langle x, x \rangle}$ $\hat{Y} = Y - x \hat{\beta}$

Activate W
Go to Settings

But the fact that I am using inner products here should tell you that I can apply the ideas of linear regression on any inner product space and not just in real number space. So I will leave it at that gives you a good generalization. So what we are doing here I will call this as, "Regressing Y on X" and we get the coefficient $\hat{\beta}$. So what we are talking about so far is a univariate regression no intercept nothing.

Suppose that your columns are all orthogonal. Not only are they independent they are all orthogonal with a little bit of thought you can convince yourself that each β , so $\beta_1 \beta_2 \beta_3$ and so on so forth are just given by regressing Y on X_1, X_2, X_3 and so on so forth. So β_1 this regression of Y on X_1 , β_2 is regression of Y on X_2 . Why is that the case? So now my X_1 and X_2 are orthogonal they are actually the orthogonal basis an orthogonal basis for the p dimensional space the $p + 1$ dimensional space I am talking about and each coefficient that I am going to get essentially would mean will be the intercept on each of the individual dimensions.

The projection on each of the individual dimensions because they are orthogonal in the lower space. So that is easy to convince yourself. What is interesting is what happens if the Xs are not orthogonal, they are independent or they are still spanning a $p + 1$ dimensional space but they are not orthogonal. So what do the coefficients represent? In that case so that is essentially what we are going to look at. So we will start off by taking one step at a time.

Look at the intercept plus one variable. So far I said that is one variable without intercept and now I am adding the intercept. So what will be what does it essentially mean for us? My X becomes $(1, x)$. So what I am going to consider is a column of ones and my original vector x . This is my new vector that I am going to consider. So what I am going to do is the first step, I am going to do is tell you about that, so this upper case X is the actual column vector X_s of consists

of x okay this is the actual input I am going to look at. So let me define $\bar{x} = \frac{\sum x_i}{N}$ as the average of the all the inputs I have seen all the inputs I have received is my training data.

So I regress X on 1 right and form the residual. So what will the residual be? But in this case what would it be I am saying. Because I am regressing on one all ones. So if all ones is the only input variable I have what should be the best possible prediction I can give? It will be \bar{x} . So \bar{x} is the only output I can give that will be the one that minimizes the prediction error. Because I am looking at squared error the output should be \bar{x} . So my $\hat{\beta}$ will be \bar{x} in this case.

So the residual which I will denote by is by $z = x - \bar{x}\bar{1}$. This one bar is just to indicate that it is a vector of ones. So this x is a vector so this \bar{x} is a scalar value which is the average of all the inputs and $\bar{1}$ is the vector of ones so that this gives you the residual. Does it make sense? This is the vector of residuals, so I usually put the bar on the x and the z then the middle to differentiate it from two. But sometimes it looks like lower case z , so is it fine let us adopt the convention that even if I put the bar there it is still uppercase z ? Because either way I will have to be very careful about distinguishing 2 and Z .

So the second thing is shown below. 2 separate univariate regressions (1) and (2)

(Refer to slide time 12.40)

1 Regress x on $\bar{1}$, form the residual $z = x - \bar{x} \bar{1}$

2. Regress y on z to give $\hat{\beta}_1$

This tells me that I have taken the average value out of the input variables because the average value can be used to predict the average output if I want I have taken out the average value so whatever is left okay is the individual variations on the data point okay and use that to predict my output value y .

So this essentially means that so given that there are two dimensions 1 and x . So the $\hat{\beta}_1$ tells me what is the contribution of x after I have used one to explain the output. So given I have taken care of one already what is x ? So if you think about what I have done here this is essentially making it orthogonal to the 1 vector, the z vector is essentially the x vector the component of x that is orthogonal to the 1 vector going back to how we did the univariate regression. That is what we have done here so does this remind you people of anything? We have already looked at Gram Schmidt orthogonalization. This is essentially something very similar to that.

So I start off with one bar and the x as the 2 vectors that span some space now I am orthogonalize I am essentially giving you an orthogonal basis now one is one vector and z this the other vector

but together they span the same space that was spanned originally by one and x . Except that they are orthogonal and people agreed with me earlier when I said that if the columns of x are orthogonal then they can independently do regression on each of those columns that is essentially what we are doing here. So I have done a regression on this to get me β_1 .

So going back to our picture here, so essentially I had some x_1 , I had some x_2 . So what I did was I first regress x_2 on this and so essentially I am getting so that is my z . I am getting a orthogonal component to that. So now I have x_1 I have z and they are spanning the same well yeah they are spanning the same space that z is a projection of x_2 . Imagine the plane is going into the board right so it does not look right to you but the plane is going into the board.

So z is actually perpendicular to x_1 and it is formed by projecting , by regressing x_2 on x_1 okay that is direction z . And my original y which was going out of the plane, now I essentially project it on z to get the coefficient. It does not matter see this is still going to project here okay so earlier when I wanted the coefficient for x_1 and x_2 . I would have looked at this these points here now I will basically look at these points that is essentially what I have done there is no change in the actual space. So the same \hat{y} is what I will get okay but the coefficients I will be using for representing the \hat{y} will be different. We can generalize this to p dimensions.

(Refer to slide time 17.51)

$$z_0 = x_0 = 1$$

$$j = 1, 2, \dots, p$$

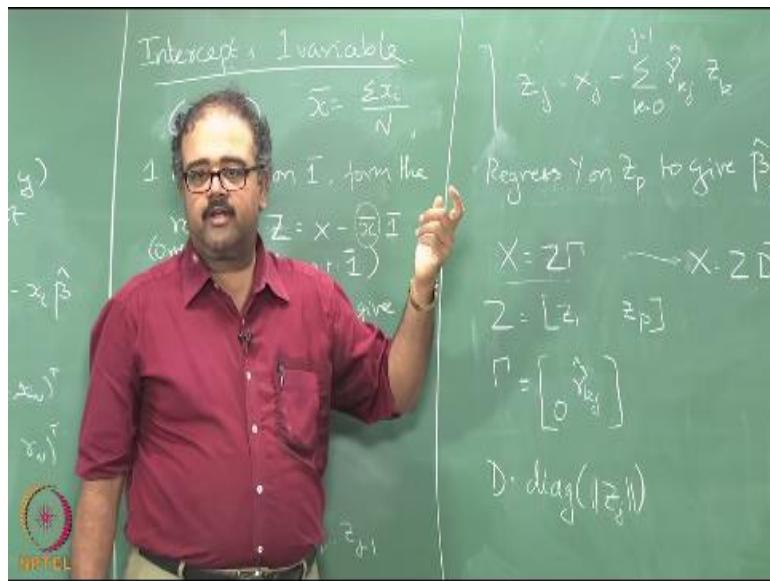
Regress X_j on z_0, z_1, \dots, z_{j-1}

So what will you get so j runs from 1 to p so I will regress x_j on all the previous z directions that I have determined. So how would I determine z_0 . I start off with z_1 would have been obtained by regressing x_1 on z_0 and then finding the residual. So that gives me so that is what we do here right, so I take \bar{x} right I basically I regressed x on $\bar{1}$ and then take the residual and make that as z . So likewise I will regress x_1 on z_0 take the residual and use that as z_1 . Then I will regress x_2 on z_0 and z_1 and then take the residual. So that is the γ coefficient which will so in this case it was \bar{x} . (Refer to slide time 19.48)

$$\begin{aligned} z_0 &= x_0 = 1 \\ j = 1, 2, \dots, p \\ \text{Regress } x_j \text{ on } z_0, z_1, \dots, z_{j-1} \\ \gamma_j &= \frac{\langle z_j, x_0 \rangle}{\langle z_j, z_j \rangle} \quad (j=0, \dots, j-1) \end{aligned}$$

So if you think about it. That will be \bar{x} , so the inner product with z_1 's when z is all ones is n and the z_1 inner product with x_1 when z it is all ones is just x_1 . So this essentially will be the summation of x_1 this will be just \bar{x} . The first case that is this is our gamma. It make sense or was it too quick? Yes or No. So I am saying this \bar{x} was derived by just using the same formula. This happens when I regress on the first variable so start off with z_0 is 1 I am regressing x_1 on z_0 when I regress x_1 on z_0 what do I get is z_0 inner product z_0 is n . Well just ones right and they sum up all the ones. So that is where dimension is n that will be n and z_0 inner product x_1 will be summation x_1 . So summation x_1 / n is essentially the average. So that is essentially what we had here. So that is the same formula. Now I am generalizing to other dimensions so I am still continuing the loop here okay, so that loop that runs for $J = 1$ to P .

(Refer Slide Time: 21:14)



So for every j and that is how I derive my z_j . So I take the current coordinate under consideration x_j subtract all the previous dimensions I have basically looked at. So what I am left with what am I left with the orthogonal component of x_j . Orthogonal with respect to the dimensions I have already looked at, so in some order I am considering, so once I have done this for in some more I am considering it in some order right when they come to the p th dimension, so what do I get, so what is $\hat{\beta}_p$? It is the actual coefficient that I will find for the p th variable if I had done the regression as we talked about earlier. If I had done that if you estimated my β like this okay.

This is essentially what I will end up with okay but because of the process we have generated it we can interpret it in a slightly different way which is essentially $\hat{\beta}_p$ tells you how much the p th variable contributes to the output given, that we have adjusted for all the other input variables given that we have adjusted for all the other input variables, how much does the p th variable contribute to the output, now we can go back and think about non independent vectors, if any of the variables is not independent right, so what will happen in this case the residue will be 0 and it essentially will be trying to find, how much would 0 contribute to the output okay that is not going to be a lot okay.

But it becomes even more interesting if my vectors are merely dependent but not exactly so what will happen is if I subtract out everything else from that vector right, so think of it like this right this is x_1 that is x_2 okay this is the 2D plane it is not like this is the plane right, so x_1 and x_2 are very close to each other there, so if I subtract out x_1 from x_2 I am going to get a small component that is orthogonal to this right, I am going to get something like this all right. Now if I take the inner product of that, so that will be a small number here, so this can become very large right.

So if my vectors are nearly dependent but not exactly so that the residual is not zero exactly but close to zero then the whole thing can become very unstable the estimate whole estimation process can become unstable, so that is essentially what will happen if even if you eliminate perfectly dependent columns right there could still be possibility of getting numerical instability so to avoid all of these things people have come up with various techniques, that of course one of them is to essentially get rid of all the correlated or the nearly correlated columns right, but there are other ways of actually trying to get this to be stable okay.

So just an aside, so let Z be the matrix that we create by taking z_1 to z_p columns. So I have done this Z_1 to Z_p in this elimination process in some order. So I will take this z_1 to z_p columns okay and γ is the matrix where I store all my γ hat k_j there is an upper triangular matrix right, so for every combination k_j , I will have 1 γ hat value I will just put it in the upper triangular part and the lower triangular side I will just keep it as zero. So an upper triangular matrix there and you can think about it you can write the x as z times γ right.

So essentially the I am just stacking all of these things you have done together and we are writing it as is z times γ and so D is a diagonal matrix where the diagonal entries are the norm of the inner product of z_j with itself right, so the j entry or the Z_j entry in the D matrix would be the inner product of z_j with itself that is the norm of z_j , so I can write it like this, so this is called the QR decomposition of x right, so the thing about Q is it is orthogonal right. Q is orthogonal and R is upper triangular okay.

So this kind of a representation for the data matrix, so this kind of a QR representation of the data matrix essentially gives you some kind of a orthonormal basis but Q is not just orthogonal is orthonormal why because I am dividing by the norm here okay, so it is so the product will be ones or zeros because they are orthogonal to begin with anyway okay I made them orthonormal so Q gives me an orthonormal basis and R is said upper triangular matrix that lets me reconstruct the inputs x and this kind of a decomposition is very convenient and it is used widely in other kinds of representation or transformation of the data and so on so forth.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

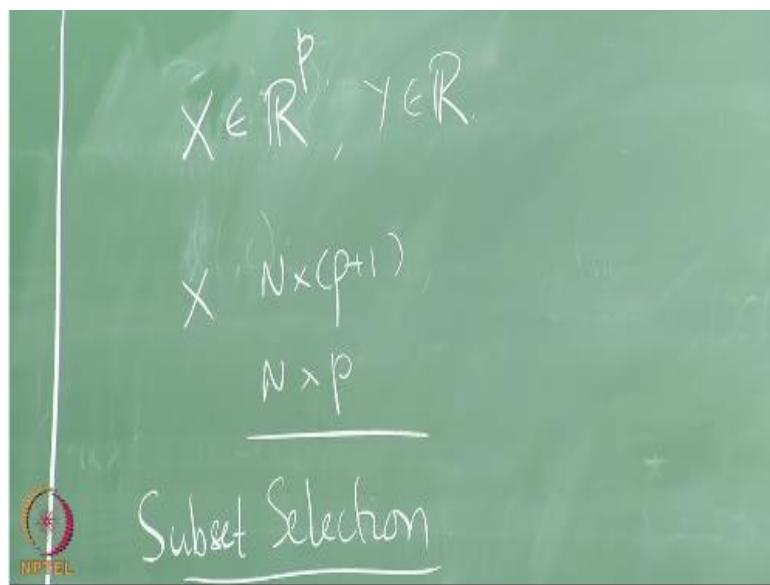
Introduction to Machine Learning

Lecture 14

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Subset Selections 1

(Refer Slide Time: 00:24)



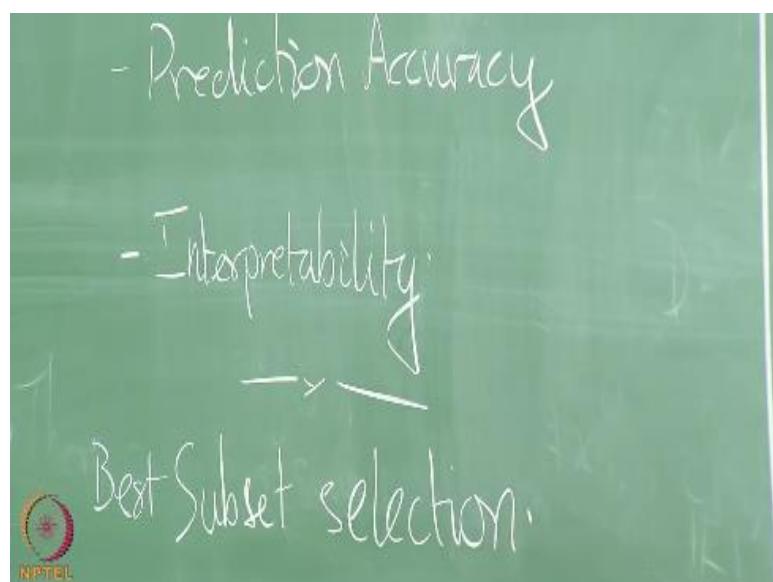
So we were looking at linear regression. We are assuming that the X is coming from \mathbb{R}^P but I told you that it's not necessary that it has to come from the set of real numbers and we talked about various ways in which you can encode the data and so on so forth. And then if we assume that the Y comes from \mathbb{R} and I told you depending on the circumstances, for example, so we will talk about the input matrix X okay which might be of $N \times p + 1$ or $N \times p$. So it will be $N \times p + 1$ when we actually have an explicit intercept or a β_0 term. For the β_0 term we will have a column of ones added to the data. The input instead of thinking of it as a p -dimensional vector we will

think of it as $p + 1$ dimensional vector and when I do not have the intercept, it just becomes a p -dimensional input. So that is that is the basic setup that we have and we are essentially looking at minimizing some squared error. So we looked at the simple linear regression we looked at the case where there were multiple inputs and we looked at how you can interpret that in terms of single variable regression univariate regression is essentially what we looked at in the last class. And so this class we will go on to look at a little more complex things that you can do with linear regression. So linear regression is great because it is so easy to solve and it is very efficient, runs very quickly and all, that but there are a couple of drawbacks to linear regression. So the first one is that if you remember I was mentioning in last class also that by doing this least squares fit you are actually getting a fit that has very low variance but it also has how much bias? What I said is that if you do least squares fit and if linear happens to be the right choice then least squares which gives you the 0 bias solution. So the least squares fit gives you 0 bias solution but the problem is the variance can be relatively high and it turns out that by not just doing the straightforward least squares fit, by doing more tricks with their data with the models that we have we can trade off a little bit bias. I am going to get a biased estimator for the fit for the line. I am still fitting straight lines and I have not done anything different I am still fitting straight lines but the fit I am getting will no longer be a bias-free fit but then I can reduce the variance a lot more by adding certain additional constraints to the problem. So essentially what I would really like to do is reduce the number of variables which I am trying to fit, Instead of trying to fit $p + 1$ variables, if you can somehow reduce the number of variables what does is equal to? It is equivalent to setting some of the β to 0. So if I can somehow set some of the β to 0 then essentially what it means is that I have lot fewer parameters that I am estimating . So the fewer the parameter set I am estimating the lower the variance still but because I am now restricting the class of models that I am going to be looking at, since, I have to set some numbers to 0 so my bias will go up slightly. This is assuming that I am fitting straight lines you know the straight lines are the right thing to do but still my bias will increase a little bit in this case. Of course there is always this residual bias because the language of straight lines is not powerful enough. So that bias is still there. I am saying even assuming straight lines are the right thing to do if I am going to force certain coefficients to be 0, I am increasing the bias in the estimator. But the variance will go down because I have a lot fewer parameters that I am going to estimate. So that is essentially what we are going to look at and so what I mean by subset selection here is that I am

going to select a subset of the input variables to use for fitting the line. So one thing is we can reduce the variance significantly and that is where we can improve the prediction accuracy of the model. That is one of the reasons we would like to do subset selection. Are there any other reason you can think of for wanting to work with the few smaller subset? Less computation but it is a question of where you have to do more computation to figure out what the subset is and so on so forth. But yeah less computation is one answer but there is another one. So related to this, there are variables which could potentially have high noise and so it will end up with a small coefficient. But then if I tell you okay here is this model M and it has like 135 coefficients and it becomes hard for you to even figure out what this means. Instead of that, if it is okay here is this model you gave me a 135 variables but these are the 4 important variables that I need for doing a linear fit.

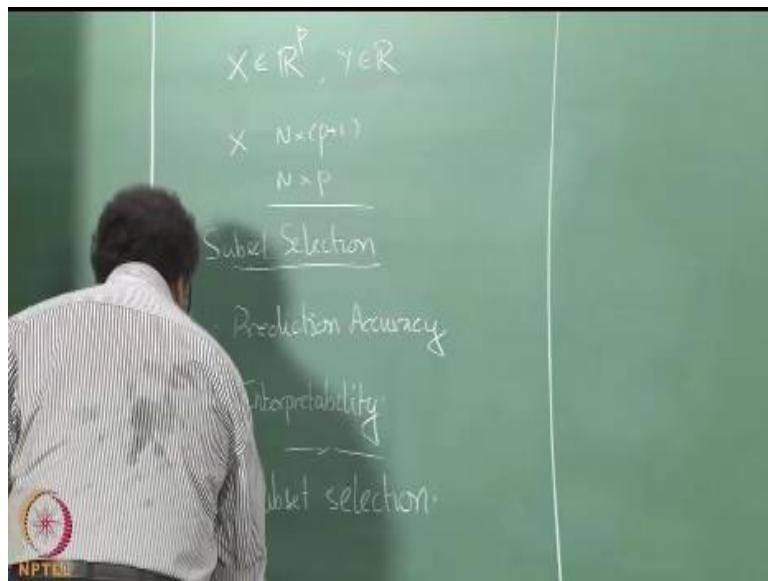
Now it is easy much easier for you to interpret what is going on. So interpretation is a very big component of any kind of data analytics that you want to do. Ultimately what you are doing with machine learning is trying to understand the data, so one of the things you would like to have this interpretability.

(Refer Slide Time: 07:00)



So there are many ways in doing this the simplest kind of approach is essentially to take “subset selection” literally and try to select from subsets of features. So why is that a simplistic approach? It is combinatorial.

(Refer Slide Time: 08:03)

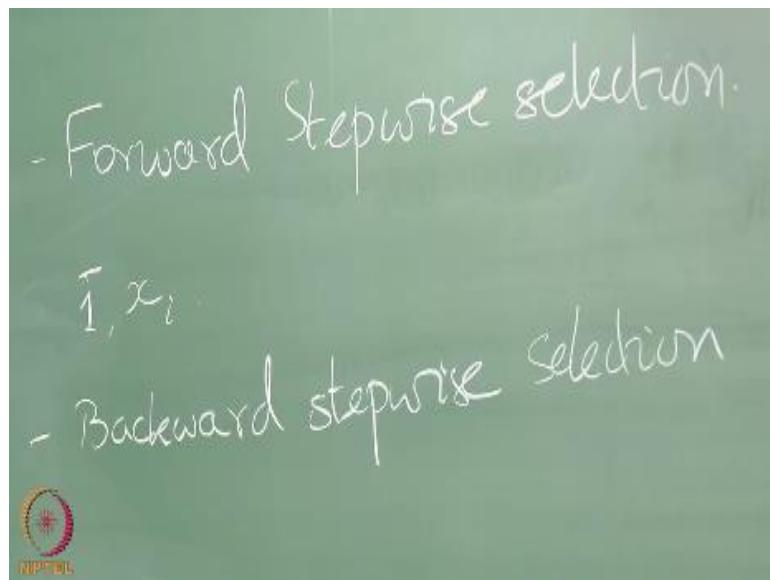


So this will just do subsets in a best subset selection. Essentially I would say that okay here first pick subsets of size one, subsets of size two and subsets of size 3 and so on so forth right, and it turns out that you can see yourself if you start playing around with some linear regression tools that what is the variables that go into the best subset of size one right basically the one best variable need not figure in the best subset of size two so it does not have a nice inclusion property.

So for it to have a nice inclusion property you need to have certain very nice conditions on the data set. So in general it does not have this inclusion property and you have to just redo the whole thing again. So it is not enough if you just do one, say okay you cannot be greedy. Basically I cannot say that I will do the first subset and then I just add the one best variable to it and then I will find the next one, so essentially you have to do a combinatorial selection so people have come up with a very clever ways of organizing this, say, using QR decomposition to do things more rapidly. I am not going to go into details of that but then up to like 30 or 40

variables you can scale well. But if you go for much larger variables, like many problems like in text or image domains then there is no hope to do an exhaustive search but that is a baseline which you can do people actually come up with algorithms which do this kind of a subset selection. So there is one very interestingly named algorithm called “leaps and bounds” which does pretty efficient subset selection but this is just a more of informational thing for you right.

(Refer Slide Time: 10:22)



Next is forward step-wise selection. So any guesses what that is? It is a greedy approach. It is just trying to do best subset selection by being greedy. So you start off what is the feature you for sure want to have? All ones or you need the intercept. Otherwise your line has to pass through the origin. So you need the intercepts you start with. So what should be the coefficient for the intercept? We already looked at it before now the mean of the y's right.

So that will be the coefficient? So we already have fitted that now what you do this you start off with that variable, then you add the next one let some x_i , add that as the next variable such that it gives you the best fit modulo the set they have already taken. So you are not going to disturb all the stages the variables you have taken to. In step k now, we will add a new variable such that

among all the variables I could add at this $k+1$ stage, this one gives me the maximum of improvement in the performance.

So how will I measure performance? Some kind of residual error on the test data. So that is how I measure performance. I keep doing this until a point where the error does not change much. Is there any other thing I can say? So the residual is orthogonal to any of the other directions I could add right there. So we know that at the end of the right fit you get the residual will be orthogonal to the space spanned by the x 's. That means individually if I take any of the x 's, the residual will be orthogonal to that individual direction as well. So when you find that none of the directions that are left have any kind of component along the direction of the residual then I can stop.

Or I mean that may take a long time to do because that might happen only when I have the full least squares fit. So I can stop and the residual drops below a certain threshold that you can say okay I am happy with the prediction accuracy I am getting and I will stop here. So there are many ways of doing. So the other way is to do what will you do in this case?

I will start with the fit that has all the variables okay and then I will keep dropping one by one right. So one thing to note is that you can do this if the number of data points is greater than the number of dimensions. So then you can actually find the fit. If $p > n$, as people pointed out last time, the formula we are using is no longer valid. So because if the matrix will not be invertible so we will have to think of other tricks for doing the fit and so on so forth. But forward stepwise selection you can do even if p is greater than n because I am anyway fitting one direction at a time right so it is fine I am adding one direction at a time I can keep going until I reach n directions at which point I should have a full least squares fit. So when people actually even come up with all kinds of variants on this. So one thing which you could do is think of some kind of hybrid approach where I keep adding and deleting directions right. So if you remember we talked about how greedy is not always the best way to grow things. So you can grow up to a certain level again maybe then dropping one of the earlier dimensions will actually give you a slightly better performance could be right so that is one possibility and right they could do some kind of a hybrid version of this. So one thing I should point out here, so you might think that forward stepwise selection because it is greedy is going to be much worse than best subset

selection. So it turns out that in many real cases or many real data sets that you work with right the greedy selection is actually not a bad thing to do. In fact, so much so many statistics packages like R will allow you to do this. So you would not find this in many of the machine learning tools like Weka, since they would not have this kind of a forward feature selection and things like that they have other ways of doing feature selection which we will talk about later right but then statistical packages actually have this stage wise edition of features because they seem to work well on a variety of data sets okay.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

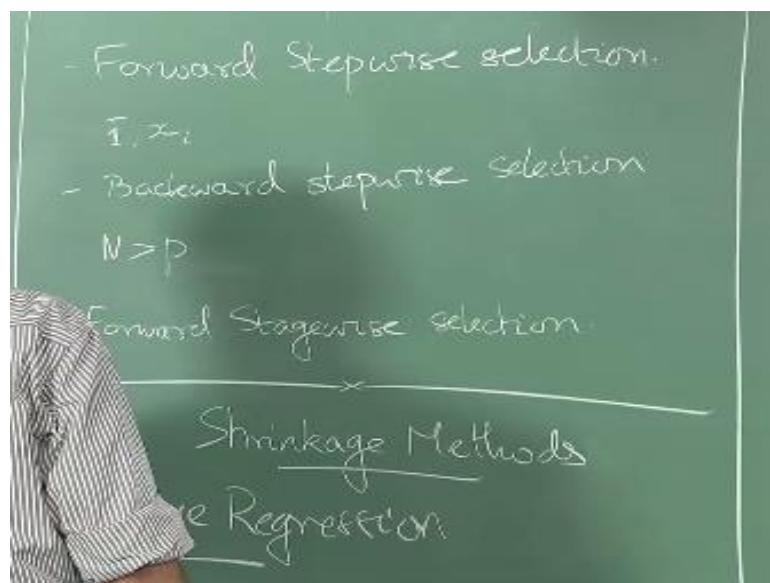
Introduction to Machine Learning

Lecture 15

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Subset Selection 2

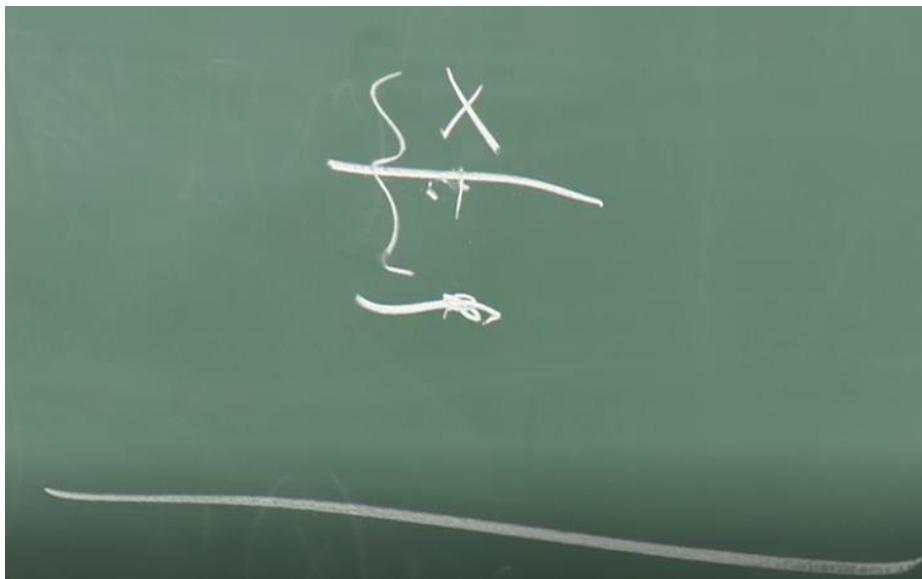
(Refer Slide Time: 00:21)



So now we will talk about forward stage wise selection, where at each stage you do the following. Let me rephrase it, on the first stage you do the following. So you pick the variable that is most correlated with the output and then you regress the output on that variable and find the residual. Now what you do is pick the variable that is most correlated with the residual, regress the residual on that variable. Now add it to your predictor. So what is your predictor? You already had one variable then you had a coefficient for that variable which you got by the first regression. Now you have a second variable and they have a coefficient for that variable which you got by regressing the residual on this variable. Essentially what you are trying to do is using

the first variable make some prediction. The second variable is going to try to predict what the error is, so essentially now I will be adding the error to the prediction of the first variable. Did that make sense? So the first variable, let us say, that is the true output that I want. So the first variable will make a prediction saying that okay this is the actual fitted value, and this is the residual. What I am trying to do with the second variable is actually to predict this gap.

(Refer to slide time 2.23)



So when I add the second variable with this coefficient to the first variable, what I am essentially doing is okay the first variable will give this as the output, the second variable make some other prediction, let us say that much so I will add the two, so the new output will be that right. Now I still have a residual left, so then I will pick a third variable which is maximally correlated with this residual. And now I add the output of all the three. And then I get my new predictor. Does it make sense? So at every stage I find the residual whatever has not been predicted correctly by the previous 'k' stages, whatever is the residual error after the previous 'k' stages and try to predict that using the new variable. Essentially I find the direction which is most correlated with this prediction and then I try to give you that. This is called forward stage wise selection.

So what is the advantage of stage wise selection? Can you think of any advantage of this?

Student: I am not picking variables randomly in forward stage wise selection.

I wasn't randomly picking a variable in the previous methods, I was picking greedily that was not random. Even in the previous case I only pick variables that gave me better fits right. In fact I will tell you that it will probably converge faster in forward step wise selection rather than forward stage wise.

But there is another significant advantage here. If you just thought about the process of fitting the coefficients, at every stage I do a univariate regression. At every stage I am just regressing the residual on one variable. In forward stepwise selection, so every stage I will add a new variable, but then I have to do a multivariate regression, I have to do the regression all over again, I am not able to reuse the coefficients from the previous step right.

So when I add a new variable I basically now I have $k+1$ variables and then do a new regression with $k+1$ variables. But in this case what is happening at every stage is that I just have to do a linear regression. I keep all the work that I have done so far intact. So in fact since we are doing this only one at a time, even though I might have k variables in the system.

But the coefficients I have for the k variables might not be the same k coefficients I would have gotten, if I started with this k variables and did a linear regression on it. So the coefficients could be different, if I take those k variables and do linear regression I will get a better fit rather than doing this stage wise fit. But we prefer to the stage wise, because it saves us a lot of computation.

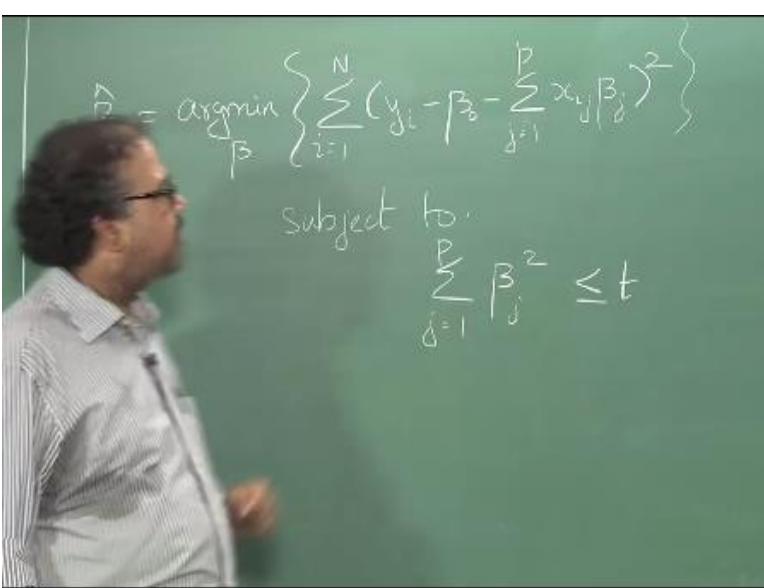
Eventually everything will catch up and we will get the same kind of prediction at the end of it, but you might end up adding a little bit more variables in this approach, but that is fine.

So the next class of methods we will look at are called shrinkage methods. The idea is to shrink some of the parameters to zero. So in the subset selection here essentially if you think about what we are doing is that all the variables that we did not select you are setting the coefficients to zero. But instead of doing an arbitrary greedy search or stage wise selection and so on so forth, in shrinkage methods what we do is we come up with a proper optimization formulation which allows us to shrink the unnecessary coordinates.

Ideally you would like to shrink them all the way to zero, but there are problems in doing that, but we will try to keep them as small as possible you can do some post-processing and then get rid of really small coordinates and things like that. But we really like to shrink these coordinates. So this is fine from the prediction accuracy point of view. From the interpretability point of view it still leaves a little bit to be desired, because you might have a lot of coefficients with I mean a lot of variables with very small coefficients back in the system.

So it is still a little bit of a thing, but mathematically this is a much sounder method than any of these things we have been talking about. And of course this is the soundest, but also impossible right. So the first thing we look at it is called ridge regression. The whole idea behind all of this shrinkage methods is that you are going to have your usual objective function which is the sum squared error that you are going to try and minimize. In addition you are going to impose a penalty on the size of the coefficients. So you want to reduce the error, but not at the cost of making some coefficient very large. You do not try and shrink the coefficients as much as possible, so what will happen is your optimization procedure will try to find solutions which have as smaller coefficient as possible and give you the similar kind of minimization in this squared error objective.

(Refer Slide Time: 08:52)



$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \right\}$$

Subject to:

$$\sum_{j=1}^p \beta_j^2 \leq t$$

So what is your normal objective function? $\hat{\beta} = \arg \min_{\beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \right\}$. So that is a

normal objective function for finding their β , and so your $\hat{\beta}$ is essentially this. So now what I am saying is, let us not do this, but let us do this with the constraint right. So what is a

constraint? $\hat{\beta} = \arg \min_{\beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \right\}$ subject to $\sum_{i=1}^p \beta_i^2 \leq t$

Fairly straight forward I have added a squared norm constraint.

So this is essentially the L2 norm. I am taking the root I am just leaving it as a square and it does not matter. So it is like an L2 norm constraint for my data.

(Refer Slide Time: 11:16)

$$\hat{\beta}_{\text{ridge}} = \underset{\beta}{\operatorname{arg\,min}} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

$$\nabla_{\beta} = (\gamma - X\beta)^T (\gamma - X\beta) + \lambda \beta^T \beta$$

$$= X^T X + \lambda I^{-1} X^T \gamma$$

$$= \lambda \beta$$

So I can make this into an unconstrained problem right. Because λ has to be greater than zero. Why do I want the β s to be small okay good question actually. So what we wanted to do was to make sure that you are reducing the variance of your model right, so that is essentially what we are trying to do now, all the subsets selection was we set the coefficient to zero, we said you have lot fewer parameters to estimate right.

So now what I am doing is that I am by imposing the size constraint on the parameters. The size constraints on the variables I am actually reducing the range over which these variables can actually move around. So if you think about it if I have moderately correlated input variables or correlated or anti-correlated input variables, so let us say I have two variables which x_1 and x_2 which are correlated and now I can have a large β_1 and a large negative β_2 , that essentially will cancel out each other in terms of the predictions I am making, because x_1 and x_2 are themselves correlated. So I can actually make my β_1 very large and my β_2 is largely negative right, so that it will just cancel out the actual effects of the two variables. So it essentially becomes a difference of β_1, β_2 that actually matters right, not necessarily the difference in magnitude of β_1, β_2 that matters not actually not the actual values. So I can basically have a large class of models which will give me the same exact output. This makes my problem much harder to control and then that increases the difficulty of the estimation problem. But now we are saying that no I cannot allow these things to become very large and then I am restricting the class of models I am going to be looking at.

So that is the reason why the decreasing size of β helps. I did not explain this completely last time so thanks for asking the question. So we just have to make sure that our λ are positive we know that little so lagrange multipliers have to be positive and so on so forth.

$$\hat{\beta} = \arg \min_{\beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

So now I can go ahead and minimize this right.

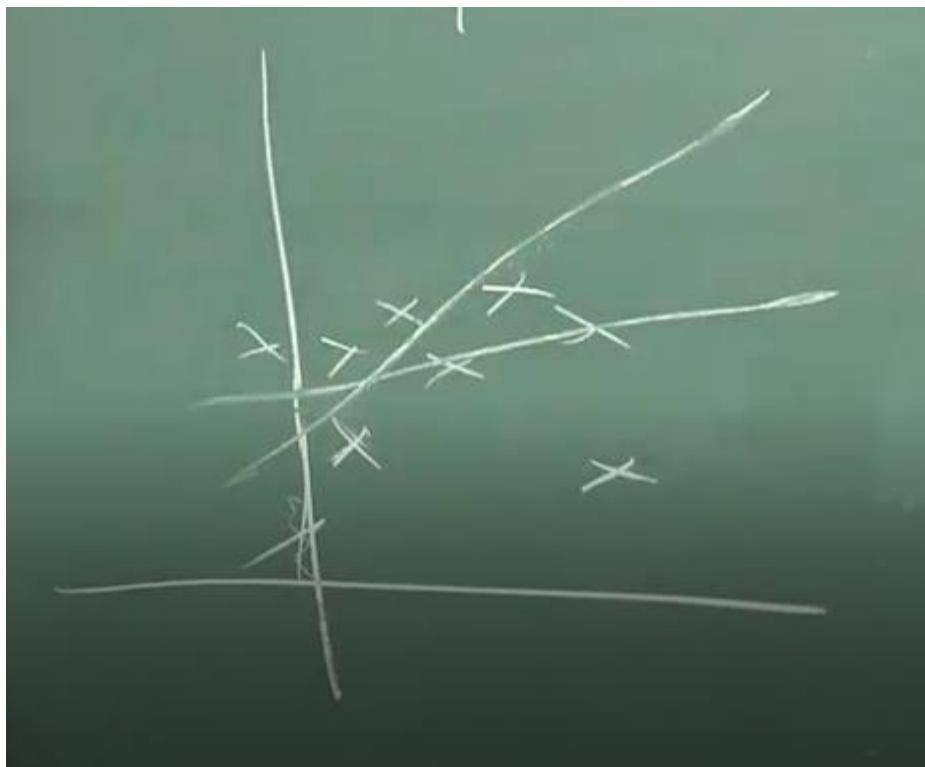
So a couple of things which I want to point out now, so one thing is if you notice the penalty here, so what do you notice about this? I am not including β_0 , see the sum runs from 1 to p it is not running from 0 to p also note that I actually explicitly wrote out β_0 here I did not squish it into the $p+1$ thing, because I am going to be treating β_0 specially here mainly, because if I penalize β_0 then what will happen is if I move my data up right, so let us say this is my X and Y axis and I have this is the data that I had right.

(Refer to slide time 15.45)



So now I have to fit that line through this right, it is a univariate regression problem. Y is my response and X is my input I have to fit a line. But now the same data points, if I shift them up, so shifting up the data points is hard, so I will just shift the origin. If I shift the origin what will happen if I penalize β_0 ? So if you penalize β_0 it will try to keep this intercept small, penalizing β_0 will try to keep the intercept small.

So earlier when I had that, if you look at the fit it will pass very close to the origin the intercept will be close to 0. Now when I shifted this, it is going to try and make the intercept small. There is this and line just shifting the slope of the line will change. It is the same data it has just shifted up a little bit, so the slope of the line will change, so it will try to go through somewhere here.



So essentially earlier when the line now the line will become like that because I am penalizing β_0 . So we do not want that to happen so just simple shifts in the data should not change the fit. So we do not penalize β_0 . Does it make sense? And anyway we know what β_0 should do, you know what β_0 should be the average of the outputs.

So one way which we can actually get rid of β_0 from this optimization problem is to say that we will center the inputs. So we will subtract the average from the y_i 's and likewise we will subtract the averages from all the x 's. So we will center the input and we will make all the X variables centered on zero. So we will subtract the mean from all the X 's, we will subtract the mean from the Y 's. So this will give me a centered input and then I will just do regression on this centered input well there will be no β_0 . So from now on when I write X it is a $n \times p$ matrix where the input has been centered. So essentially what I have done here is I have taken my data from there and shifted it so that the fit whatever is the fit I am going to get will pass through the origin.

So that is essentially what I have done. I have taken the original data translated it so that whatever fit will pass through the origin. And I will go back and add the β_0 later to get my original fit. Does that make sense? So in matrix form I write it like this,

(Refer to slide time 18.50)

$$RSS(\lambda) = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta}$$

So you can minimize this take the derivative and set it to zero solve for it you will get this. So here, so both my x and y are centered. So I subtracted the mean from Y, I subtracted the mean from the columns of x so they are all centered here. So just remember that once I get this centered values I can solve for it, this gives me $\hat{\boldsymbol{\beta}}_{ridge}$ for 1 to P in the β_0 I estimate as \bar{y} and that gives me the full solution.

(Refer to slide time 19.45)

$$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}$$

So one thing which I forgot to point out earlier you remember I had this variable ‘t’ here which was upper bound. I said subject to the constraint that it should not be larger than ‘t’, the ‘t’ has vanished but you can show that this λ and the ‘t’ are related.

So it does not matter, so for every choice of ‘t’, you have a choice of λ but typically what happens is you choose your appropriate lambda and then you work with it you do not worry about the ‘t’ formulation.

So this tells you why this is called “ridge regression”, because what they have done here is you essentially added a ridge to your data matrix you take the $X^T X$ and then you add a diagonal λ which is like adding a ridge of size λ to the diagonal elements of $X^T X$. So that is why it is called ridge regression. So why are you doing this and can you see one advantage of doing this?

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T Y$$

This whole thing becomes invertible right. So as soon as I add the λi , I am sure that this is non-singular. And even if $X^T X$ was originally singular and adding λi makes it nonsingular and it is invertible.

In fact this was the original motivation for ridge regression right, back in the I forgot, in the 50s when people came up with ridge regression the original motivation was $X^T X$ could be badly conditioned, even if it is non singular we talked about this in the last class. It could be that some variables are so highly correlated. So even if the matrix is invertible numerically you will get into problems I told you that the residual might be so small right.

So when you try to fit the coefficients you will get into problems. So numerically the inversion might be a problem right, even if the matrix is non-singular, but by adding this λ_i term to it you make sure that it is invertible and by controlling the size of the λ you can make sure that numerically also the problem is well behaved. So that is the idea behind with original motivation for ridge regression was essentially to make the problem first of all solvable right.

But then people went back and understood rigid regression in terms of shrinkage or variance reduction. And since it makes it convenient for us to talk about a whole class of problems, shrinkage problems, we motivate ridge regression from the view point of shrinkage as opposed to this inversion problem. Any questions? So I am going to encourage you to read the discussion that follows rigid regression in the book right. It requires you to work out some things along with the book you just cannot just sit there and passively read it okay, but then it draws a lot more connections from ridge regression to a variety of other statistical properties about the data which will be useful to know and I am going to make you read, I mean so go read it and I will ask you questions on it later. So go and read the discussion.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction of Machine Learning

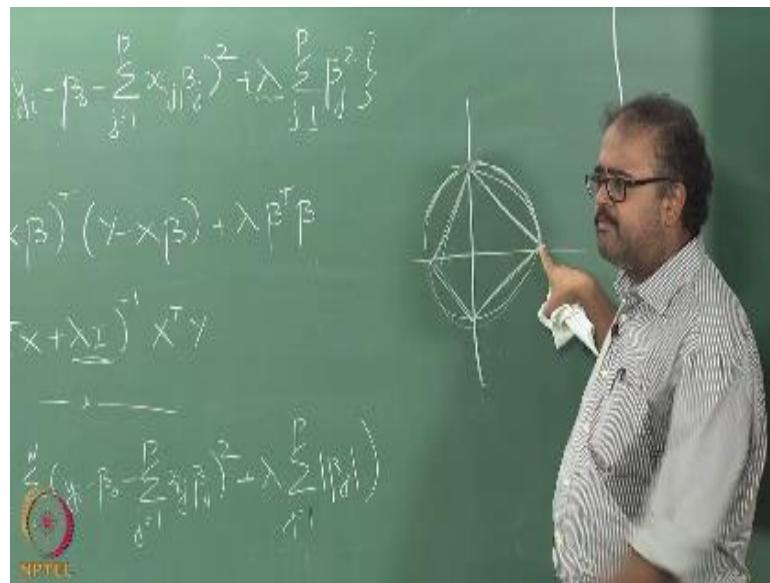
Lecture 16

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Shrinkage Methods

So what are some of the other shrinkage methods you think we can come up with? Each of β is closed right so we imposed a L_2 norm on the β . You can impose any other norm constraint on the β , say, I can I can impose an L_4 norm or more commonly I can impose a L_1 now it is called Lasso.

(Refer Slide Time: 01:00)



So lasso essentially says that let us just ignore the absolute value of the β . You sum up those and you want to keep them so you can write the same. We can write the constraint formulation where

I can say sum of β has to be less than some 't' or sum of $|\beta|$ has to be less than some 't', then I could just do this kind of a formulation right.

(Refer to slide time 2.15)

$$\hat{\beta}_{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \left(\sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^P x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^P |\beta_j| \right)$$

And so to impose a constraint on each individual β s would require you to know something about the variables themselves beforehand, otherwise if you constrain a very important variable to have a small coefficient then it becomes a problem. So you need to know something about the variables and you can say I know that these variables are very important make sure that the other variables I want to be more than 0.5 times the coefficient of these variables, so something like that. You can think of all kinds of complex constraints once you have knowledge about the system but typically you do not write in such cases you will have to have some kind of uniform constraints like this and so this is a very popular constraint. It actually makes life harder for us as it does not have such a nice closed form solution any more. I mean this is no longer differentiable. So I can't write your nice closed form solution like this in fact I have to work very hard to solve this. Although there are packages in R or WEKA where you can always run lasso on it will give you the nice fit. So what is the nice thing about lasso I will try to give you an intuition about it. So think of it this way so suppose I have a non-important coefficient, if I can reduce it from say at 1000 to 0.3 okay. Let us not even look at it that way. So I can reduce some coefficient from say 1000 to 999 and there is another coefficient which you can reduce from let us say 1 to 0. So there are many variables in my fit there is one variable whose coefficient I can reduce from 1000 to 999 there is another variable whose coefficient is one I can reduce it from 1 to 0. And both of them cost the same change in my squared error both of them contribute equally to the squared error and making this change will make the same change to the squared error.

So which one would ridge regression prefer? To reduce 1000 to 999 because that causes a much larger reduction in the squared penalty. And which one would lasso prefer to reduce? Doesn't matter. Either one but then I can make this thing slightly more contrived right now, say from 1.1 to 0. Which one would lasso prefer? So even though in absolute value terms this is a larger reduction, ridge would prefer still preferred 1000 to 999 right.

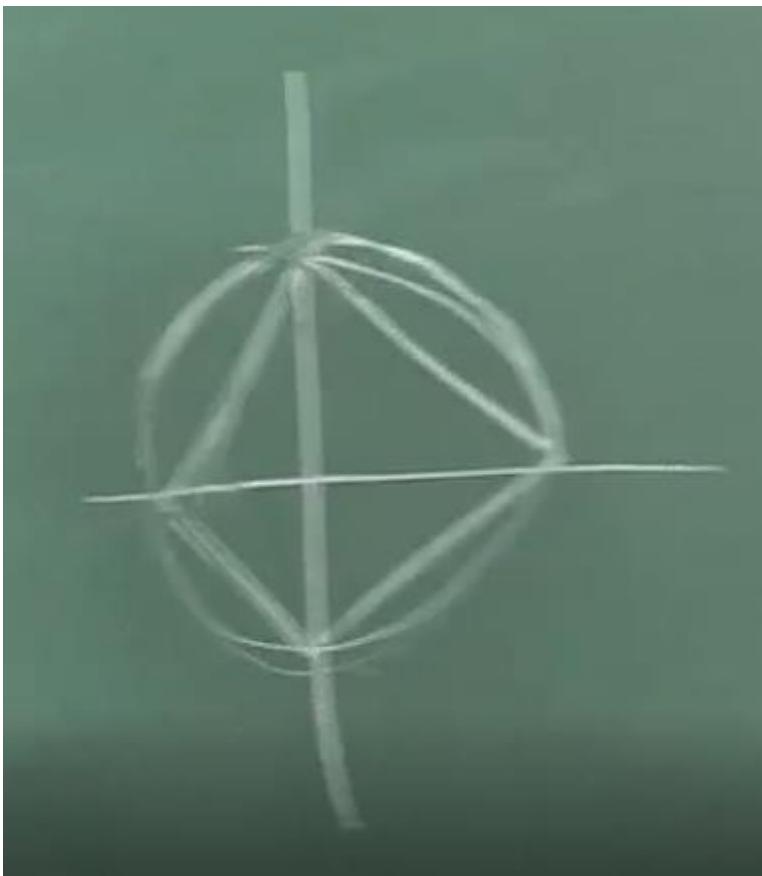
Because the fall is 1.1^2 to 0^2 vs 1000^2 quite to 999^2 still that is a larger reduction in error. So what is a take-home message here? LASSO is more likely to drive coefficients to zero than ridge. So ridge would happily leave the coefficient at 1.1 or even more dramatically it will happily leave coefficients at 0.3, 0.2, 0.8. So it will leave it at small values it will not drive it all the way to zero. Well lasso given an opportunity right we will drive the coefficients to zero we need not drive it to zero at the cost of minimizing the error. It will still try to minimize the error but given the chance it will more likely to drive coefficients to zero.

So, sometimes Lasso is also called sparse regression. Because this L1 norm constraint is also called a sparsity constraint because it makes your β vector to have more zeros. So if you know what a sparse matrix is so you have a matrix with a lot of zero entries in it and only few nonzero entries. So you really don't want to have an array representing your sparse matrix because most of the entries are 0 so typically what you do in a sparse matrix representation is you store the index of the nonzero entry and the non-zero entry itself. So that actually takes a lot less memory than actually having a large M by N array with lots of 0 so that's why its called sparse. So here the L1 regression has a tendency to make the β sparse or to have a lot of 0. So it is sometimes called the sparsity constraint.

Yeah no see if I take 0.01 and square it okay and the difference between that squared and 0^2 is lesser than 0.1 and 0. So but the drop in the value will be bigger in the Lasso than in Ridge. Now it depends on what other competing elements that you have. So lasso typically drives the coefficients to zero. Well ridge does not do this is that I was giving you an intuition as to why that is the case right. It is not mathematically a sound argument but you can give a mathematically sound argument also that LASSO is more likely to find sparse fix than ridge regression okay.

So I am being very careful here. So I mean I can also think of a geometric intuition for it so if you think about the the lasso constraints it will be something like this. Let me think about ridge constraints that suppose to be a circle. So the ridge constraint will be something like this so here if you know where the sum has to be a constant. So the sum has to be a constant in the sum of squares has to be a constant, so one will be a circle other one will not be a circle.

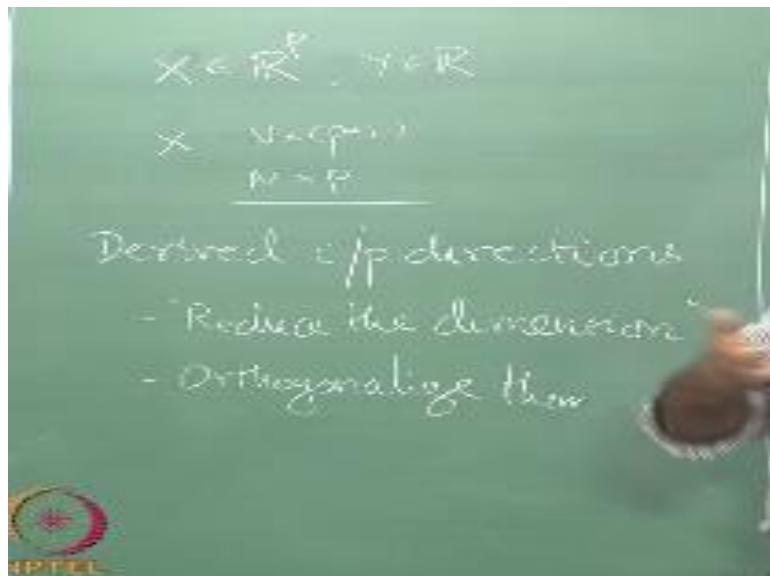
(Refer to slide time 10.02)



So when you're looking at the error surface corresponding to this, essentially you will have to find solutions that lie on this or lie within this for lasso and lie within this for ridge. And it turns out that you are more likely to hit a corner off you can show that more formally that you are more likely to hit a corner in the latter case than in the ridge case. So the probability of hitting something because this is the whole thing is convex the probability of hitting that side it is higher

This is just the very rough geometric intuition. I do not want to get into showing things formally but you can show that the probability that lasso will give you these kinds of corners in the fit corner obviously you can see that has one of the coefficients a 0, so that you will get a corner as a fit is much higher then you will get one of these axis points in the ridge. So in fact you can think of having higher-order penalties also. Like I said you can think of an L4 norm penalty. So far we looked at two methods for variance reduction. One was subset selection and the other one was shrinkage based methods. Now there is a third class of methods which people use for getting better fits with possibly fewer variables or fewer parameters.

(Refer Slide Time: 11:29)



This is based on the derived input directions. So we talked about reducing the number of variables so far at least in the subset selection part we retain some of the variables and then we ignore some of the other variables. Likewise whether we are doing implicit subset selection by doing lasso or ridge regression we are reducing the coefficients of some variables and retaining some other variables. But at all points we were operating with the original set of basis vectors that were given to you. So what were the basis vectors we are talking about here? The columns of the X matrix are the basis vectors.

So we are working with the original basis vectors we are working with the same columns that were given to us. In one case we picked some columns and threw out some of the other columns and in the other case we tried to continuously adjust the weights of the column so that some of them were given more weight and some were given less weight. So when we talk about derived input directions now I am not going to stick with the original columns. I am going to find a new set of columns and I am going to find a new set of features or new set of directions which I will then use for doing my regression okay.

So we actually talked a little bit about it in the when we looked at the orthogonalization. I told you that in orthogonalization essentially what you are doing is you are finding an orthogonal basis for the input space and then you are trying to find the coefficients there. So likewise what we will do here is we will reduce the dimensions. So what is the advantage of orthogonalizing the dimension well we could do univariate regression on each dimension separately and then that will give us the coefficients.

So we do not have to actually do a multivariate regression we can do univariate regression on each dimension because once I orthogonalize the directions they do not interfere with one another. So I can do univariate regression. So typically when I try to do these derived input directions I try to orthogonalize the directions and I also try to find a reduced set of dimensions that will give me the original fit or as close to the original fit as possible.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

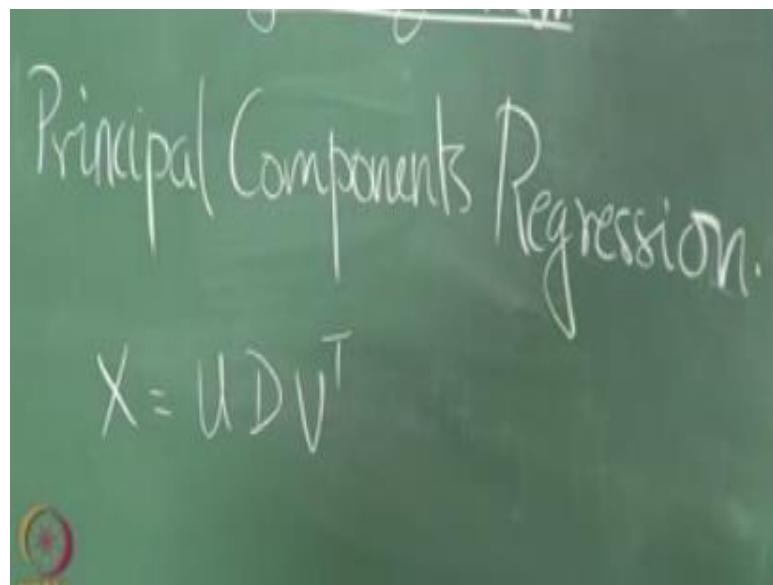
Introduction to Machine Learning

Lecture 17

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

**Principal Components
Regression**

(Refer Slide Time: 00:16)



So D is a diagonal matrix where the diagonal entries are your eigenvalues (ideally) or otherwise known as singular values. V is a P x P matrix which has your eigenvectors and U the N x P matrix which typically spans your column space as X . So this is essentially your singular value decomposition that we talked about.

(Refer Slide Time: 01:07)

$$S = \frac{(X - \mu)^T (X - \mu)}{N}$$

$$= \frac{X_c^T X_c}{N}$$


So if you look at singular value decomposition or what is called the principal component analysis literature, you will find the following. You will find that they will talk about the covariance matrix S . What is the covariance matrix? $S = (X - \mu)^T (X - \mu) / N = X_c^T X_c / N$

It is essentially if you think of whatever we have been doing so far what would be this centered data. So I take the centered data and I find the eigendecomposition of that. I find the Eigen decomposition of the covariance matrix.

(Refer Slide Time: 02:16)

$$= \frac{X_c^T X_c}{N}$$

$$X_c^T X_c = V D^2 V^T$$


So I can essentially write this as $X_c^T X_c = V D^2 V^T$. So the same V and D . So it is the same V and D that I wrote here. If I take X_c so basically I am going to get the same thing. So it is essentially like doing singular value decomposition and retrieving the V matrix. I am essentially taking the $X^T X$ which is the covariance matrix of the centered data and I am finding the Eigenvalue

decomposition of that. So D^2 would be the Eigen vectors of $X^T X$ so this is standard stuff you should know.

(Refer Slide Time: 03:09)

$$X_c^T X_c = V D^2 V^T$$

The columns of V are called
principal component directions of X .

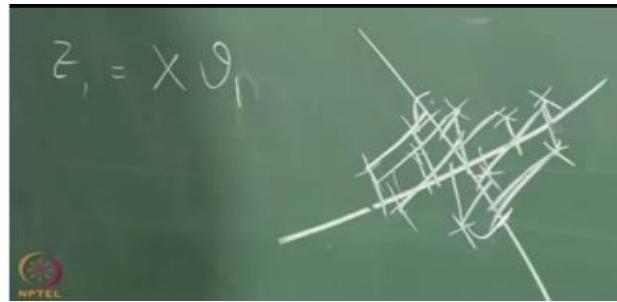
So the columns of V are called the principal component directions of X . So there are a couple of nice things about the principal component directions. We will talk about just one. So I will actually come back to PCA slightly later when I talk more about generally about feature selection not just in the context of regression but when I talk with generally about feature selection I will come back to PCA and tell you or at least show you why PCA is good. Right now I will just tell you why PCA is good I will come back later and then I will show you why PCA is good.

(Refer Slide Time: 04:33)

$$Z_i = X \phi_i$$

So suppose I take $z_1 = Xv_1$ where v_1 is the eigenvector corresponding to the first eigenvalue. So essentially what this means is that I am projecting my data X on the first eigenvector direction so the resulting vector z_1 will have the highest variance among all possible directions in which I can project X .

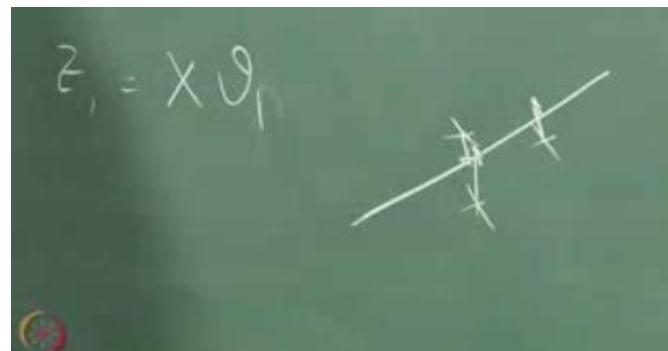
(Refer Slide Time: 05:17)



So what does that mean? Suppose this is X okay this is not x and y , so it is a two-dimensional X this is X now I am claiming that v_1 will be such that when I project X onto v_1 I will have the maximum variance. So in this case it will be some direction like this and projecting X onto this essentially means that so you can see that the data is pretty spread out it goes from here to here. On the other hand if I had taken a direction let us say that looks like that right so if I look at projection of the data right, so you can look at the spread it is a lot lesser in that direction than in the original direction I did the projection I know it looks pretty confusing to look at but the people can get my point. It is in the original direction that way the data was a lot more spread out as opposed to this direction where the data is lot more compact when I project it on to that direction.

So that is essentially what I am saying. So z_1 is essentially the projected data onto that direction onto X like z_1 actually has a highest variance among all the directions in which I can project the data and consequently you can also show things like if I am looking to reconstruct the original data and I say that you can only give me one coordinate, so you have to summarize the data in a single coordinate and now I am going to measure the data measure the error in reconstruction. If you looked at it so the error in reconstruction would have been these bars that I did the projection over.

(Refer Slide Time: 07:40)



That would be the error in reconstruction. So I have the original data so that is the data and now I will give you these coordinates. Now I have to reconstruct the data so essentially this will be the errors so the first principal component direction. The first principal component direction is the one that has the smallest reconstruction error. First principal component direction will be the one that has the smallest reconstruction error so we can show a lot of nice properties about this. So I will actually come back and do this later, when we talk about the general feature selection. But here the first thing you can see what each one v_1 to v_p will be orthogonal. So I have gotten my orthogonal directions and the thing to notice is a lot of the variation in the data is explained by v_1 or v_1 has the maximum variance likewise you take out v_1 and so now what you have your data lies in some kind of a $p - 1$ dimensional space and the direction in that the space which has the highest variance is v_2 it turns out that so v_1 has the highest variance over the data. So in this space orthogonal to v_1 , v_2 has the highest variance right in the space orthogonal to v_1 and v_2 , v_3 will have the highest variance and so on so forth. So essentially now what you can do is hey I am going to take all this directions one at a time and I will do my regression because each is orthogonal I can independently do the regression I can add the outputs and I can keep adding the dimensions until my residual becomes small enough that make sense. So I will just keep adding this orthogonal dimensions until my residual becomes small enough at that point I stop.

(Refer Slide Time: 09:44)

$$z_j = Xv_j$$
$$\hat{y} = \bar{y} \vec{1} + \sum_{m=1}^M \hat{\theta}_m z_m, \text{ where } \hat{\theta}_m = \frac{\langle z_m, y \rangle}{\langle z_m, z_m \rangle}$$

So this is essentially the idea behind PCR.

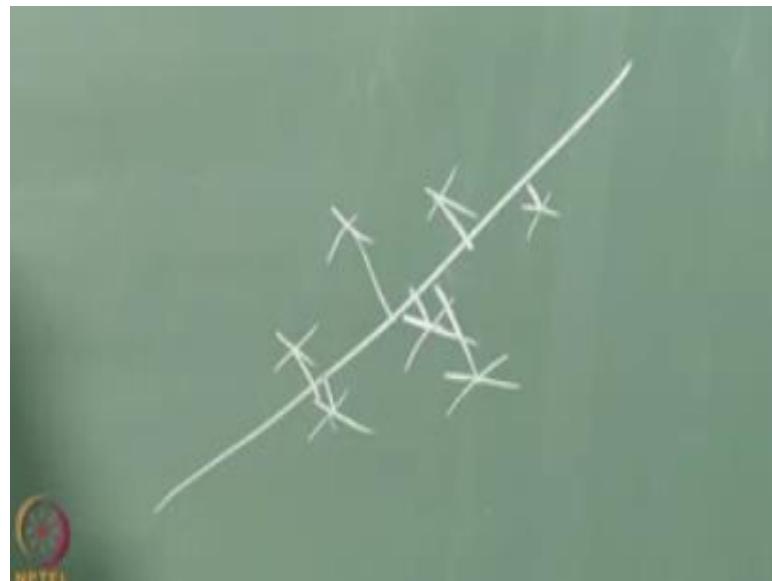
(Refer to slide time 10.39)

$$\hat{y} = \bar{y} \vec{1} + \sum_{m=1}^M \hat{\theta}_m z_m, \text{ where } \hat{\theta}_m = \frac{\langle z_m, y \rangle}{\langle z_m, z_m \rangle}$$

So remember we are working with the centered data a and you automatically add in your intercept which is \bar{y} . The coefficient is \bar{y} and then if you choose to take the first 'm' principal components your thing will be $\theta_m Z_m$ where Z_m is given by $z_j = Xv_j$ and θ_m is essentially regressing y on Z_m . So that is a univariate regression expression we know that well now. So this gives you the principal component regression fit. So one of the drawbacks of doing principal component regression is that I am only looking at the input right I am not looking at the output

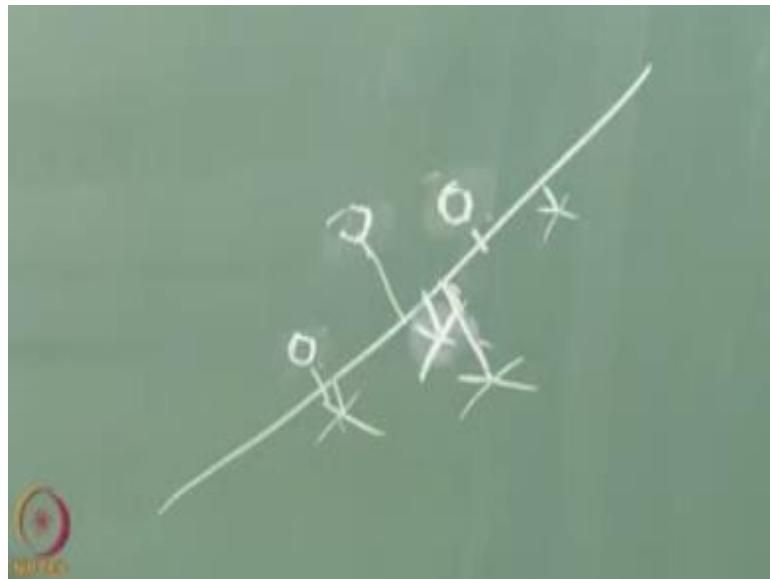
so it could very well be that once I consider what the output is right I might want to change the directions a little bit right, so I can give you an example is easier for me to draw if I think of classification.

(Refer Slide Time: 12:00)



Let us say this is the data and what would be the principal component direction you want to choose something like this. So that would be the ideal direction that you would want to choose okay so now what will happen the data will get we get projected like this right but suppose I tell you that.

(Refer Slide Time: 12:32)



Suppose I tell you that that is fine that these three were in a different class and if you want to think of it in terms of regression let us assume that these three have an output of -1 and these four have an output of +1. Now if you think of this direction so the +1 and -1 are hopelessly mixed up and I cannot draw or give a smooth prediction of which will be +1 which will be -1. On the other hand, if you project onto a direction like this right the variance is small. I agree the variance is much smaller. But if you think about it, all the -1 go to one side right all the +1 go to one side, so now if I want to do a prediction on this so it will be like okay this is this side is -1 and that side is +1. I can essentially do a fit like this which will give me a lot lesser error than the other case. So in cases where you are having an output that is specified for you already it might be beneficial to look at the output also when trying to derive directions as opposed to just looking at the input data. So in classification this will be say class 1 this will be class 2 and having this direction allows you to have a separating surface somewhere here right we talked about classification in the first class.

So just having a separating surface here will be great but in this case if I am projecting on this direction coming up with a linear separating surface is going to be hard as everything gets completely mixed up.

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

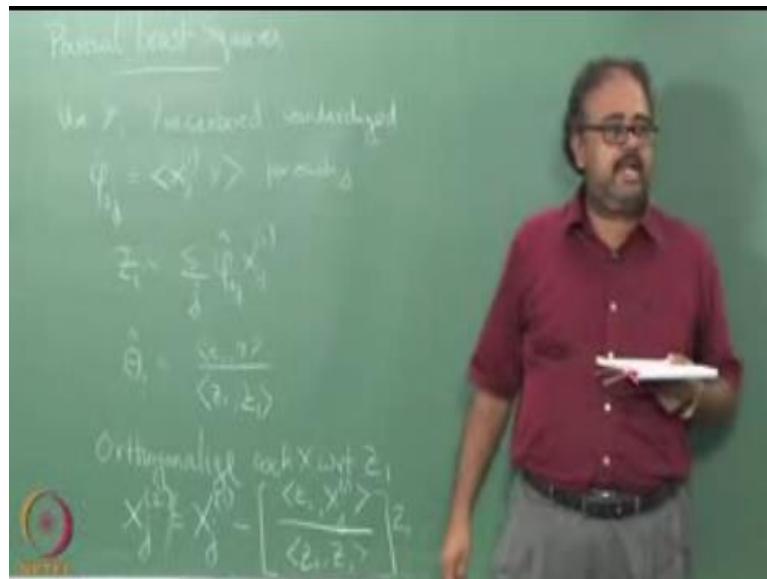
Introduction to Machine Learning

Lecture 18

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Partial Least Squares

(Refer Slide Time: 00:17)



Okay so we will continue from where we left off. As I promised, so we are looking at linear regression and we looked at subset selection and then we looked at the shrinkage methods and then, finally we came to derived directions. I said there are three classes of methods so we are looking at a couple of examples of each of those classes of methods the first one we looked at was subset selection so we looked at forward, backward selection and stage wise selection in step wise selection and all that and then we looked at shrinkage methods where we looked at ridge

regression and lasso and then we started looking at derived directions where we looked at principal component regression. I said the next one we look at is partial least squares and gave the motivation for looking at partial least squares. It is mainly because principal component regression only looks at the input data, does not pay attention to the output and therefore you might sometimes come up with really counterintuitive directions. Like an example I showed you with the +1 and -1 outputs, so the basic idea here is that we are going to use the Y also right.

Just like the usual case I am going to assume that Y is centered. And I am also going to assume that the inputs are standardized. This is something which you have to do for both PCA and partial least squares as they essentially assume that each column is going to have 0 mean and unit variance on the data that is given to you make it 0 mean unit variance. So that you are not having any magnitude related effects on the output. So what I am going to do is the following. If you remember how we did orthogonalization earlier something very similar so I am going to look at the projection of Y on X_j then I am going to create a derived direction which essentially sums up all of these projections. I have computed basically the projection of Y on x_j . So this is essentially the direction is a vectorized version of it then I am going to sum all of this up. So essentially what I am doing here is I am looking at each variable in turn. I take each X_j in turn I am seeing what is the effect on Y, so how much of Y, I am able to explain just by taking X_j alone and I am using all of that I am combining that and making that as my single direction so individually taking each one of this all by itself how much of Y can I explain. And that becomes my first derived direction that is my z_1 . So that is the coefficient for z_1 in my regression fit. For that one you can see what it is like so I have taken Y and regressed it on z_1 and that essentially gives me what the coefficient for z_1 . So I am looking at how much of Y is along each direction X_j , so in some sense you can think of it as if I have one variable X_j , how much of Y can be explained with that one variable X_j . I am looking at that and then my first direction z_1 is essentially summing that univariate contributions over all my input directions I suppose, I have two input directions. Unfortunately I have to do this in 3d suppose I have two input directions so what I am going to do is I am going to take my Y and project it on x_1 alone first and on x_2 alone.

(Refer to slide time 5.11)

Partial Least Squares

Use \mathbf{Y} : \mathbf{Y} is centered: standardized

$$\hat{\varphi}_{1j} = \langle \mathbf{x}_j, \mathbf{Y} \rangle \text{ for each } j$$

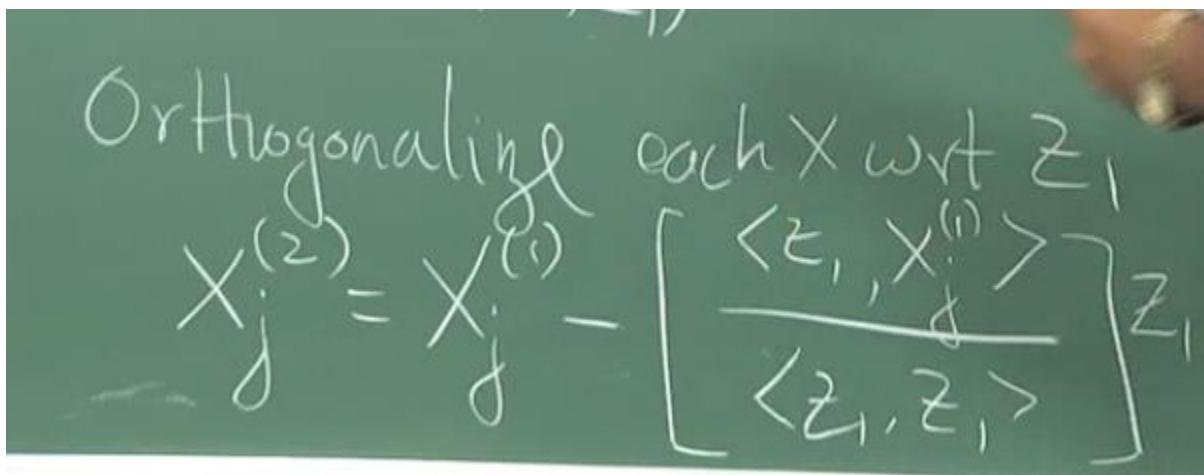
$$z_1 = \sum_j \hat{\varphi}_{1j} \mathbf{x}_j$$

$$\hat{\theta}_1 = \frac{\langle z_1, \mathbf{Y} \rangle}{\langle z_1, z_1 \rangle}$$

But the basic idea is I take \mathbf{Y} and I find the projection of \mathbf{Y} along \mathbf{x}_1 , then I find the projection of \mathbf{Y} along \mathbf{x}_2 . Now I am going to take the sum of these two and whatever is the resulting direction and I am going to use that as my first direction.

In PCR what we did was we first found directions \mathbf{X} which had the highest variance here we are not finding directions in \mathbf{X} with the highest variance but we are finding directions in \mathbf{X} . In some sense, components of \mathbf{X} which are more in the direction of the output variable \mathbf{Y} , so eventually you can show that which you are not going to do but you can show that the directions you pick that $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ that you pick or those which have high variance in the input space. But also have a high correlation with \mathbf{Y} . It is actually an objective function which tries to balance correlation with \mathbf{Y} and variance in the input space \mathbf{Y} . PCR does only variance in the input space does not worry about the correlation but partial least squares you can show that it actually worries about the correlation as well right. We find the first coordinate now what do you do you orthogonalize, so what should I do now I should regress \mathbf{x}_j on \mathbf{z}_1 .

(Refer to slide time 9.29)



The image shows a handwritten derivation on a green chalkboard. At the top, the text "Orthogonalizing each X w.r.t Z₁" is written. Below it, the formula $X_j^{(2)} = X_j^{(1)} - \left[\frac{\langle Z_1, X_j^{(1)} \rangle}{\langle Z_1, Z_1 \rangle} \right] Z_1$ is derived. The derivation involves several steps of cancellation and simplification of terms involving Z_1 .

This is how we did the orthogonalization earlier. So you find one direction then you regress everything else on that direction then subtract from it that gives you the orthogonal direction. So essentially that is what you are doing here. The expressions look big but then if you have been following the material from the previous classes then it is essentially whether they just reusing the univariate regression construction we had earlier right.

So now I have a new set of directions which I call $X_j^{(2)}$. I have a new set of directions which we will call $X_j^{(2)}$ and then I can keep repeating the whole process, I can take Y projected along $X_j^{(2)}$ and then combine that and get Z_2 and then regress Y on Z_2 to get θ_2 . So I can keep doing this until I get as many directions as I want. So what is the nice thing about Z_1, Z_2 other things. They themselves will be orthogonal because they are being constructed by individual vectors which are orthogonal with respect to their all the previous Zs that we have. Each one will be orthogonal and therefore I can essentially do univariate regression. So I do not have to worry about accommodating the previous variable, so when it when I want to fit the Z_k I can just do a univariate regression of Y on Z_k and I will get the coordinates θ_k . So once I get this θ_1 to θ_k how do I use it for prediction? Can I just do like $X\beta$ or can I do $X\theta$? I know what should I do well so I can do θZ and predict it but then I do not really want to construct this Z directions for every vector that I am going to get. So I do not want to project it along those Z directions, so instead of that what I can do if you think about it each of those Zs is actually composed of the original variables X right. So I can compute the θ and then I can just go back and derive coefficients for the Xs directly because all of these are linear computations I all I need to do is essentially figure

out how I am going to stack all the thetas so that I can derive the coefficients for the Xs okay think about it you can do it as a short exercise but I can eventually come up.

So where I can derive this coefficients $\hat{\beta}$ from these θ s. So I will derive $\theta_1, \theta_2, \theta_3$ so on so forth. I can just go back and do this computation. So you will have to think about it. Its very easy you can work it out and figure out what is the number should be. And what is the ‘m’ doing? That number of their directions, I actually derive the number of directions I derive from the PLS. So here the first direction I can keep going suppose I derive p directions what can you tell me about the fit for the data if I get p PLS directions? It essentially means that I will get as good a fit as the original least squares fit. So I essentially get the same fit as least squares fit and anything lesser than that is going to give me something different from the least squares fit. Here is a thought question if my X are originally, orthogonal to begin with X were actually orthogonal to begin with what will happen with PLS? Z will be the same as Xs and what will happen to Z_2 ? Can I do the Z_2 ? No. PLS will stop after one step because there will be no residuals after that. So I will essentially get my least squares fit in the first attempt itself okay so that is essentially what will happen. So we will stop with regression methods.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

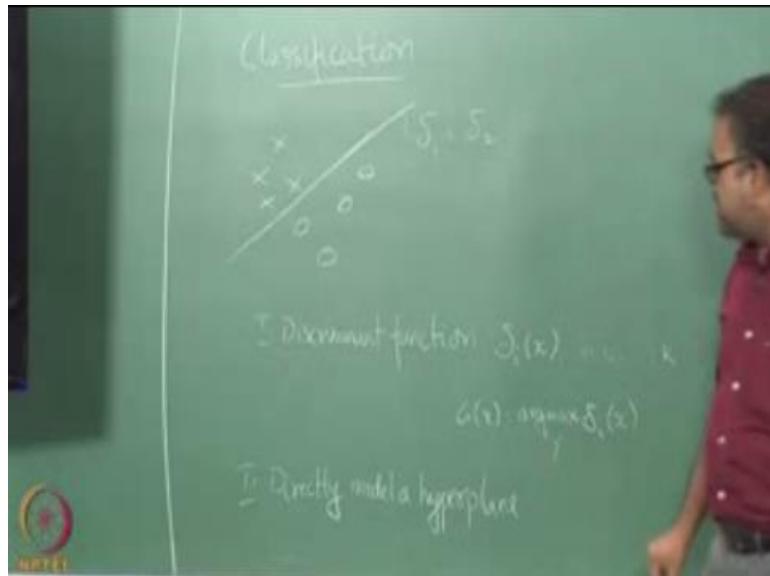
Introduction to Machine Learning

Lecture 19

Prof. Balaraman Ravibdran
Computer Science and Engineering
Indian institute of technology

Linear Classification

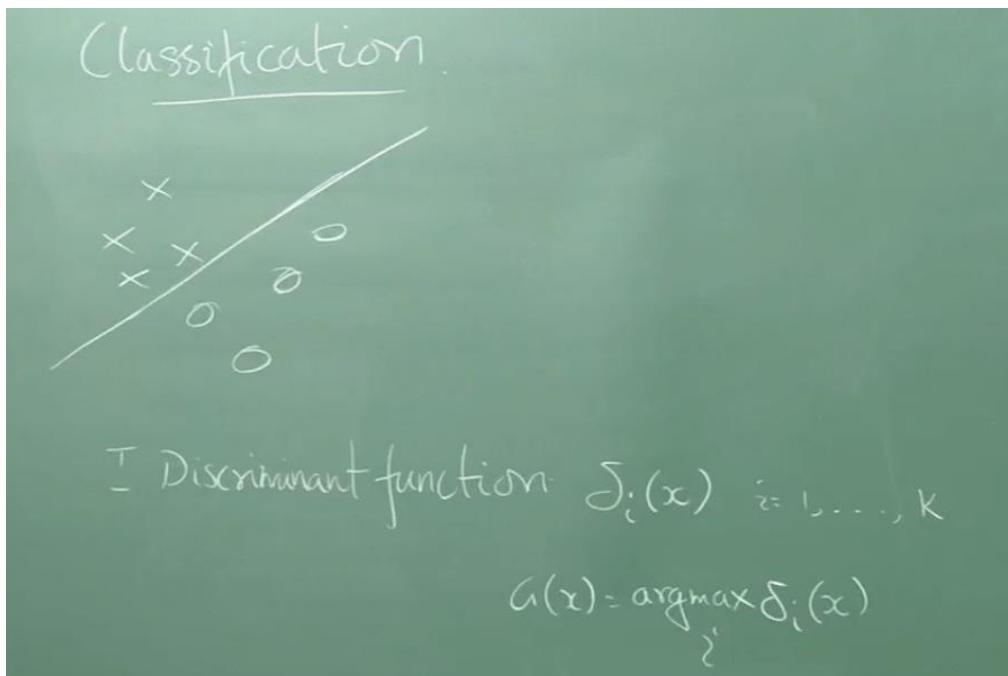
(Refer Slide Time: 00:16)



So we move on from linear methods for regression to linear methods for classification. So far we have been looking at linear methods for regression but I did tell you that you could do “nonlinear regression” also by doing appropriate basis transformations. So what do I mean by linear methods for classification? Linear regression you can understand that the response is going to be a linear function of the inputs, so what do I mean by linear classification? So when I am going to separate the positive classes or when I am going to separate two classes the boundary of separation between the two classes will be linear.

So that is what I mean by linear classification. So this boundary that I draw between two classes will be linear, so you can think of when we did look at an example in the first class where we had drawn quadratics and phases and things like that. So instead of that we will assume that the separating surface will be a hyper plane. So there are two classes of approaches that we will look at for linear classification and the first one is essentially on modeling a discriminant function.

(Refer to slide time 3.18)



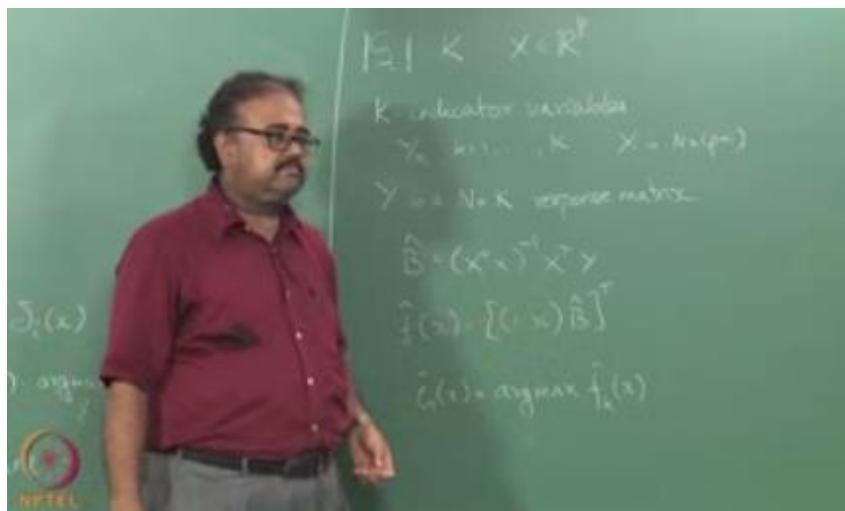
So one rough way of thinking about it is to say that I am going to have a function for each class and if the function for "class I" output for "class I" is higher than for all the other classes I will classify the data point as belonging to "class I". So I am going to have some function for each class and depending on whichever is the highest I will output it to that, so this is essentially the idea behind discriminant functions. I am going to have to figure out a way to learn this δ_i 's. Suppose let us just keep it simple think of a two class problem okay.

Here a question okay they think of a two class problem and I have δ_1 and δ_2 right so where will be by separating hyper plane? Wherever when $\delta_1 > \delta_2$ it is class 1, when $\delta_2 > \delta_1$ it is class 2 and wherever they are equal it will be this a boundary. So if I need this to be a linear surface what conditions should δ_1, δ_2 satisfy? Should they be linear? Not necessarily but this is sufficient condition if you are linear the surface will be linear.

So what else can they be? They can be non linear as long as I have some kind of a monotone transformation of them which will become linear. So we will see examples of this. We will actually look at discriminant functions or we look at the assumptions which will appear to be. We are doing something heavily nonlinear but at the end of the day you will find that the surface will be linear or the separating surface will be linear, so we look at that as we go along. So the few approaches that we look in this class are essentially linear regression. You could do a linear regression and try to treat that as your discriminant function it for each class you could do we talked about this in the very first class right or the 2nd class, where you could do a linear regression on an indicator variable, so that will give you a discriminant function or you could do logistic regression or it could do linear discriminant analysis which is like principal component regression but taking into account the class labels you will think of deriving directions and which will be doing the classification. You look at the 3 of those.

The second class of methods which will come to this is directly model the hyper plane. So it is related to this in some sense. If I give you the discriminant functions I can always recover the hyper plane but here instead of trying to do a class wise discriminant function will directly try to model a hyper plane. This is second class of problem we look at one classic approach for doing that which is the perceptron. We will also talk about some more recent well founded ways of doing that which is essentially looking at the question of what an optimal hyper plane is and trying to solve for it directly. So these are the two approaches we look at. So this basically just setting things up. People remember the basic set up for classification.

(Refer Slide Time: 07:05)



So I am going to assume that I have some space G which has K classes. I will first conveniently index them as 1 to K . X is going to come from R^p as before and the output is going to come from this space G . So that is our setup and so if there are K classes I am going to have when have K indicator variables.

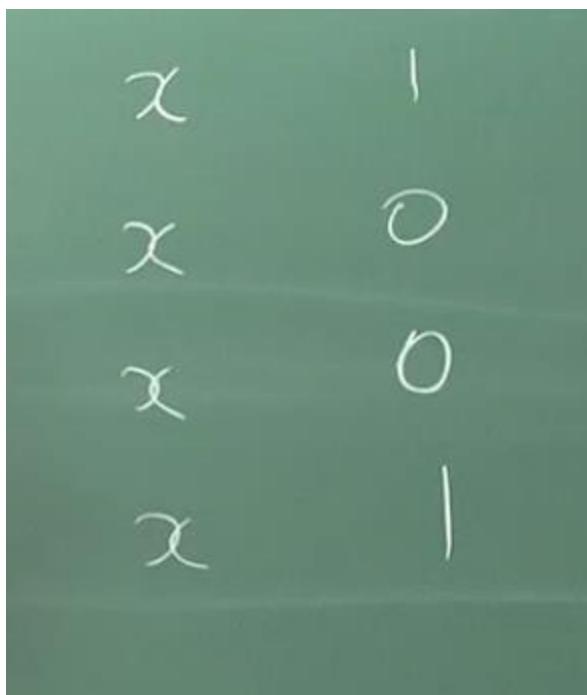
Remember when we talked about “one of K ” encoding, one of these K indicator variables will be one for any input, depending on what class that data point belongs to. I have augmented ones so my $\hat{\beta}$ is equal to $(X^T X)^{-1} X^T Y$. That is linear regression for you. I can just do linear regression on my response matrix. \hat{B} is capitalized here because it is also a matrix. So each class I have a set of β so I can produce a vector of outputs f given an input X by essentially taking the product with the β that gives me a vector of outputs f and finally class label that is sent to the data point this argmax of they have f . $\hat{f}(x) = [(1x)\hat{B}]^T$ and $\hat{G}(x) = \arg \max \hat{f}_k(x)$.

So I am going to get a vector of f s one for each class right and the one, that I assign finally is the one that gives me the maximum output. I am not doing that any complex math here at all, so only bit of math here we already saw in the very first linear regression fitting.

So X is the input data point I add a 1 to the front of it to for the bias. So what does it mean? What does this $f_k(x)$ mean? So if you think about it whenever the input take as of some classes let us pick a particular class let us call it j right let us have a class or even make it more confident and instead of ‘ j ’ lets consider class 3. So the input belongs to class 3 whenever, the input belongs to class 3, $y_3=1$ from the training data.

So if you think about it, and look at the expected output that you should get for a particular x the expected output you should get for particular x is : what is the average number of times it is going to be one? So I am going to see the x again and again and again right whenever that the x belongs to class 3 the output will be 1 and the x does not belong to class 3 the output will be 0. So what is the output I expect? It is the average of the outputs. The prediction should be the average of the outputs. Does it make sense? So I have many x,x,x there are different x they are the same x ok many times I am getting x again and again so sometimes it is class 3 sometimes it is not sometimes it is not sometimes it is class three okay.

(Refer to slide time 13.24)

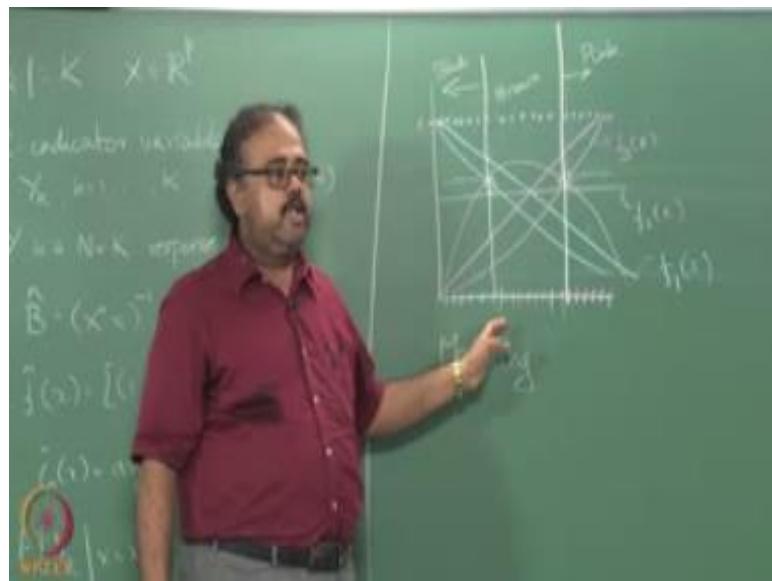


$E[Y_k | X = x] = \Pr(G = k | X = x)$. So if you take the average of all of this outputs what am I getting? Probability that x is class 3. We know that when I am trying to do the linear regression what I am trying to predict is the expected value. Ideally I should be trying to predict the expected value of this but since its linear you will not be able to get there but we are trying to do is probability of the class. There is a problem of using linear regression and that is what I am coming to it right.

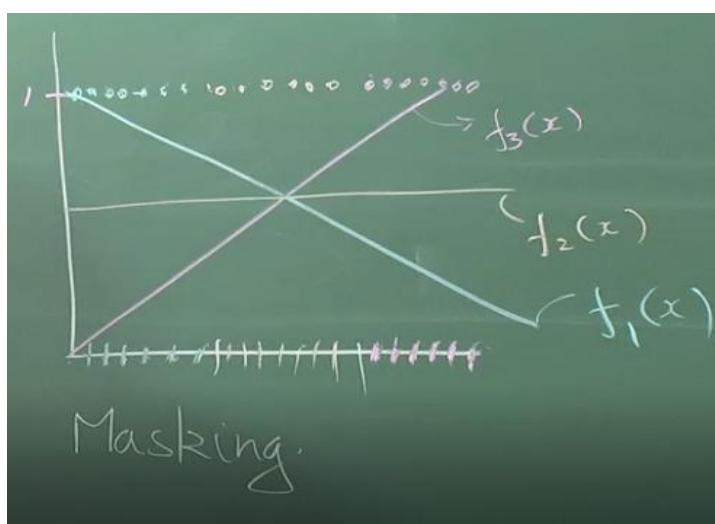
So you cannot really interpret these as probabilities because linear regression is not constrained. So we will come to that in a minute but this is one I am working upto that it is telling you the interpretation of what you want to do is that it is a probability? So I really would like to interpret this so the expected value of y_k given x you would ideally like it to be probability that the output is K given the input is X. This is ideally what you want and the linear regression gives you hope of getting there and people sometimes still use linear regression because it is easy to use.

We would have to think of other ways of getting to it. I will come to that in a minute, so before that I just want to point out one other pitfall of using linear regression for classification. But is it clear any questions? This is the same indicator variable thing, so it is either 0 or 1.

(Refer Slide Time: 16:50)



So I am going to assume there is a single input direction. So let us say that okay there are data points here that belong to one class and data points here that belong to another class r. So if you think about it let us say this is this is encoded by pink right so the training data right will look like this right and 0 elsewhere for pink training data for blue will look like this and 0 elsewhere.



(Refer to slide time 21.03)

So now if I try to fit a straight line to this so what do you think will happen? I will get a line that goes like that and I will probably get a line that goes like that. So this is essentially what your outputs will look like. Directly trying to interpret this as probabilities is not a good idea obviously right but you can see that wherever this is greater than this okay that should probably belong to class blue, where ever it is pink is greater than blue it should belong to class pink right. So at least this much you can conclude from the output of the linear regression.

So that is essentially how you would interpret the output? So whenever one output is greater than the other or greater than all the others you will assume that it is the correct class. Directly interpreting that as probability it is a problem, so this is what you would like to do. But you do not want to do this. Having said this let us see how visible this color is, suppose I have a 3 class okay they are sitting in the middle like this, so the outputs were this will be somewhere there right. Now if I try to fit a straight line for this what is going happen?

Remember the rest of the points are all sitting here right they are a bunch of 0 here a bunch of 0 here and a bunch of ones there, so I try to do linear regression on this so I am going to get that line like that I know. What is the problem with that? Blue and pink completely dominated, there is no part in the input space, where brown no part of the input space where brown actually dominates. The output of brown never dominates anywhere. So this is essentially what your f_1 f_2 f_3 will be so it turns out that for class two will never output any input point as class two okay, so this problem is called this problem is called masking. So this is one thing which you have to be aware of well you are doing linear regression for making your predictions. Is there any way to get to over masking? So instead of looking at pairs you just look, at higher order basis transformations. Instead of regressing on x , I could regress on x^2 .

So if I am going to do that essentially I am going to get curves that look like that, with interesting curve is this guy how is this brown curve going to look like. So these are the crossover points it is anything, that this side will be blue anything to that side will be pink and anything in between will be brown okay, you can see okay. So but remember the input space is just on this line okay, so here this is the output whatever is going up is the output, so the input is only on this line okay just a single dimensional input. So it is no region but say it is only a line segment here so in this

part of the input space it will be blue this part it will be brown in this part it will be pink thank the almost ideal except there is a small here.

That is the just drawing error so you can choose appropriate data points such that you can actually with the quadratic transformation if you regress on x^2 you can recover the actual boundaries okay so the rule of thumb is if you have K classes in your input data you need at least $K - 1$ basis transformations. So in fact with a lot of work you can show that even with x^2 regression you will have masking if you have four classes so in four classes you have to regress on the cubic transformation okay so that you can still get away with it.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 20

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Logistic Regression

So let us go back to whatever has been bothering all of us. So what we are essentially choosing when we did regression here was we are going to make sure that the output is either 1 or 0. And then we are trying to regress on that. So what do you want our function to do? Basically you want your $f(x)$ to give you $P(K|X)$ but then trying to do that is a little harder so what we are going to do is you are going to look at some kind of a transformation of the probability.

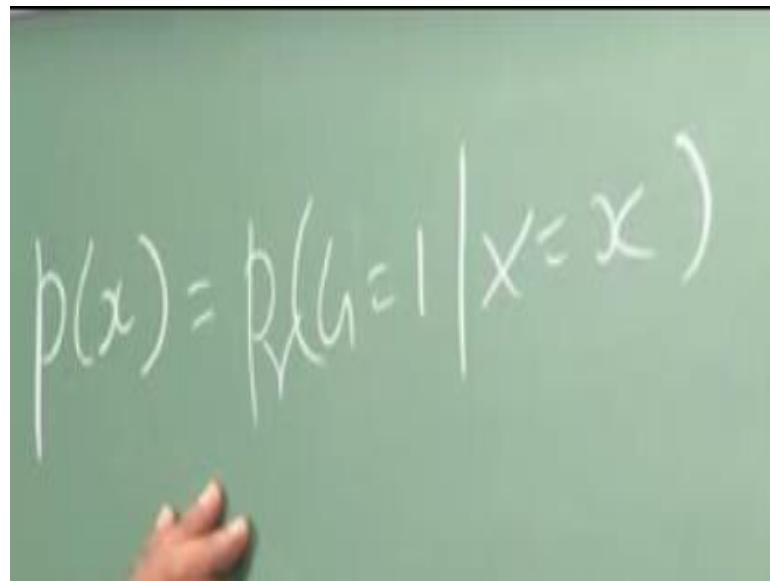
(Refer Slide Time: 00:49)

$$\text{Transformation}$$
$$\text{Logit} \quad \log \frac{p(x)}{1 - p(x)}$$

And we are going to try and fit that. Let me look at the logit transformation. This is essentially

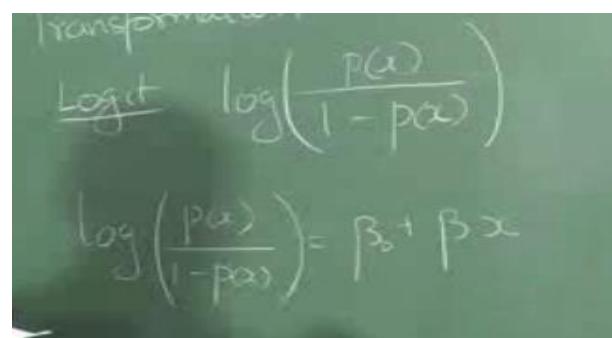
$$\log \frac{p(x)}{1-p(x)}.$$

(Refer Slide Time: 01:29)


$$p(x) = \Pr(G=1 \mid X=x)$$

To make my life easier for the next few minutes I am going to assume we are dealing with binary classification. So the class label is either 0 or 1 and $p(x)$ is essentially probability that the output is 1 given the input is X . So this makes my life a little easier when I write the next part.

(Refer Slide Time: 02:12)


$$\text{Logit } \log\left(\frac{p(x)}{1-p(x)}\right)$$
$$\log\left(\frac{p(x)}{1-p(x)}\right) \approx \beta_0 + \beta_1 x$$

So given that $p(x) = \Pr(G=1 \mid X=x)$, what is $1 - p(X)$? It is probability 0. We are talking about binary classes so this is sometimes called the probability of success divided by the probability of

failure or “odds”. So this is sometimes called the log odds function or the logit function. So this is essentially the transformation that we want to look at so what I am going to do is I am going to try fit a linear model to the log odds. So what does $p(x)$ in this case right.

(Refer Slide Time: 03:37)

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

So what is this function going to look like? That is a sigmoid right so essentially we are saying that my $p(x)$ is going to given by probability that x is 1 will be given by a sigmoid.

(Refer to slide time 4.35)

Transformation

$$\text{Logit } \log\left(\frac{p(x)}{1 - p(x)}\right)$$

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x$$

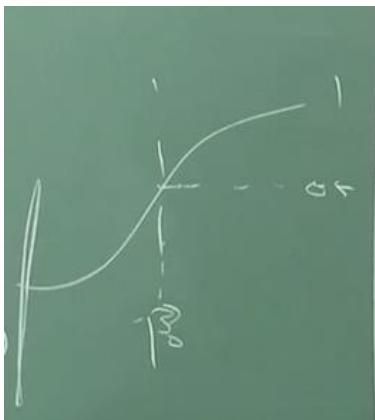
$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

(Refer Slide Time: 04:54)

$$\frac{p(x)}{1 - p(x)} = \beta_0 + \beta_1 x$$

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

That is the term in the power except that here it is a minus before that. So what do we have here? so what we do if $p(x)$ is greater than 0.5?



(Refer to slide time 7.46)

We will output this one and if $p(x)$ is less than 0.5 will output it as 0. So is it okay because even though I am doing linear regression and linear regression is unbounded I am going to plug it into this expression and therefore this will make sure that my probability is between 0 & 1. What is that point this 0.5 depends on what I had put for my β_0 . So what about the classifier that I am learning here? What is the separating surface or the decision boundary between class one and class two? Think you have $p(x) = 0.5$ but what does it mean? Look at the expression that we have here? So in $p(x)$ equal to 0.5 this be 1 we have log of that will be 0.

So essentially the thing is $\beta_0 + \beta x = 0$ and that is the straight line I mean assuming X is uni-dimensional that is a straight line right. So even though I did something complex, I used an exponential to define my probability, the decision surface turns out to be still hyper plane. So plug in $p(x)$ equal to 0.5 here and I'm going to get 0 on the left hand side. I am essentially solving $\beta_0 + \beta x = 0$. That is just a straight line assuming that it is a hyper plane. Maybe I should do the whole class in one dimension it makes it easier for people to visualize things. So one thing I should point out is that logistic regression looks simple right but it yields a very powerful classifier it works very well in practice.

And it is used for not just for building classification surfaces but it is also used a lot in what people sometimes call “sensitivity analysis”. So they look at how each factor contributes to the output and so how much is each factor important in predicting the class label so for doing that they do logistic regression and then they look at the β vector and figure out how much

each variable is going to be contributing to the output. So people use that a lot I mean of course you can use anything that we have seen for doing this kind of sensitivity analysis I am just telling you what people use in practice okay.

So logistic regression is something that is used very widely in practice both by machine learning folks and by statisticians in fact when I work with few doctors right it was almost impossible to get them to accept anything else other than logistic regression as a valid classifier because they were so sold on logistic regression and with good reason because it does work very well in very well in practice.

So that is for two classes, so what do you do for multiple classes? So multiple classes I'm essentially going to take recourse to this form. I am going to say keep the probability that the output is class 1 given X is given by an expression like this the probability that the output is class 2 given the input is X is given by another expression like this where which will have a different set of β_0 and β_s right.

Likewise for every "class I " am going to say is given by a different set of β_0 and β . So do we have to do that for all the k classes? I have to do it only for $k-1$ classes because the k th class probability will be automatically determined. I will have to have $k-1$ sets of β . If I have k classes I have to figure out how to estimate those.

(Refer Slide Time: 10:52)

$$P(Y=k|X=x) = \frac{e^{\beta_0^{(k)} + \beta^{(k)}x}}{1 + e^{\beta_0^{(k)} + \beta^{(k)}x}}$$

(Refer to slide time 11.46)


$$P(Y=k | X=x) = \frac{e^{\beta_0 + \beta_k x}}{1 + e^{\beta_0 + \beta_k x}}$$

So we are going to write like this for that K minus 1 classes by convention either the first class or the last class whichever arbitrary numbering you choose either the first class or the last class the coefficient is set to zero setting the coefficient to zero will essentially give you the answer that you want. So now you agree with me setting it to zero is fine so how do we estimate the parameters for logistic regression? So it's a little tricky since we are anyway trying to model directly the probabilities. What we are going to try and do is maximize the likelihood okay of the data so far we have always looked at some kind of error function and we have been trying to optimize the error function within those linear regression we looked at squared error and then we did the optimization and soon so forth but here we are going to look at a slightly different criterion we are going to optimize the likelihood of the data.

(Refer Slide Time: 13:14)

Likelihood

$$P_\theta(D|\theta) \triangleq L(\theta)$$

So just to keep it together I am going to do this today but I have a whole session planned on maximum likelihood and other ways of estimating parameters so when we come to that I will do maximum likelihood in more detail in a generic form. So right now I will just look at logistic regression and maximum likelihood. So what is likelihood? Suppose I have some training data D the training data has been given to me the probability of D given parameters θ is known as the likelihood of θ .

So D is fixed. Think about it. I am given a training data D, D is fixed what is it that I am actually looking to find? It is θ . So this I will write as likelihood of θ . So we are always used to thinking of something that comes after the slash as the conditioning variable and the one that comes before the slash is the actual argument in this case it turns out that theta is the argument and the probability of D given θ is the likelihood of θ . D is fixed I am really trying to find what θ is correct. So finding the likelihood of theta so the scoring function should be on θ and I am usually interested in the log of θ .

(Refer Slide Time: 15:05)

$$\begin{aligned}
 & \text{Likelihood} \quad D = \{(x_1, g_1), \\
 & \quad (x_2, g_2), \dots, (x_n, g_n)\} \\
 & P(D|\theta) \triangleq L(\theta) \\
 & \log L(\theta) \triangleq l(\theta) \\
 & L(\beta_0, \beta) = \prod_{i=1}^n p(x_i | \beta_0 + \beta_1 g_i) \quad g_i \in \{0, 1\}
 \end{aligned}$$

Because it allows me to simplify a lot of the distributions that I will be considering and we will denote this by "l" mostly. So what is the likelihood so in our case? θ is our β s so my input my D is going to consist of $\{(x_1, g_1), \dots, (x_n, g_n)\}$. It is going to consist of pairs of data points like this. So X is the input G is the output we are talking about classification so G is the output X is the input right so what is the likelihood? So I wanted to stay in the two class domain for y so G is either G belongs to 0 or 1 so 0 means is class 0, 1 means is class 1 okay.

(Refer Slide Time: 16:41)

$$\begin{aligned}
 & \text{Likelihood} \quad D = \{(x_1, g_1), \\
 & \quad (x_2, g_2), \dots, (x_n, g_n)\} \\
 & P(D|\theta) \triangleq L(\theta) \\
 & \log L(\theta) \triangleq l(\theta) \\
 & L(\beta_0, \beta) = \prod_{i=1}^n p(x_i | \beta_0 + \beta_1 g_i) \quad g_i \in \{0, 1\}
 \end{aligned}$$

This is a funky expression that is written and we will come back to this. We will see this again so this is the probability of one pair “xg” occurring? So what is this is the probability that the X has a label 1? This is the probability that X as the labels 0 and what is this? This is the actual label of X. If the actual label of X is 1 then the term $\log(p(x_i))$ will appear in the equation if the actual level of X is 0 then the term $\log(1-p(x_i))$ will appear in the equation this will become 1 right so if the actual level of X is 1 what should be the probability.

Probability that X equal to 1 right that is what I should be looking at so that is what this is the actual label is 0 then I should be looking at the probability that X is 0 that is what this term is right so you can see that this gives me the probability of one “XG” pair. I do this for all of them assuming that they are all sampled independently right because I am assuming they are independent I can take the product. So now we know why we love logarithms right. (Refer Slide Time: 18:26)

$$L(\beta_0, \beta) = \prod_{i=1}^N P(x_i | \beta_0 + \beta x_i)$$

$$\ell(\beta_0, \beta) = \sum_{i=1}^N [y_i \log P(x_i | \beta_0 + \beta x_i) + (1-y_i) \log(1 - P(x_i | \beta_0 + \beta x_i))]$$

So that is the expression and is simple enough so now comes the interesting part we want to do what we want to maximize likelihood. So we need to take the derivative of this and equate it to zero. It's fine right because log is a monotone transformation we can take the derivative of the log equate it to zero and then solve for β . Unfortunately life is not so simple. Let us try and do the simplification which I am multiplying this out and gathering the terms. Multiplying this out I gather the terms here and we know what that is what is that yet we already know that so that we can insert that and simplify that there and what about this guy one minus this right I can again write it in a simpler form write log of one minus z will give me okay.

(Refer Slide Time: 21:33)

$$= \sum_{i=1}^N -\log(1 + e^{\beta_0 + \beta x_i}) + \sum_{i=1}^N y_i(\beta_0 + \beta x_i)$$

So now I can take a derivative of that with respect to β and equate it to zero what do I get.
(Refer Slide Time: 22:11)

So take the derivative of this term right you are going to end up with minus P . So this will go down and I will get $e^{\beta_0 + \beta x}$ as the numerator. I will get minus P times X since I am doing it with respect to a specific β_j . I will get X_{ij} . Does it make sense? So this first term I will get minus P times X_{ij} and this one if I take the derivative of that I will get g_i times X_{ij} okay. So that essentially what I have done here looks like a nice and easy expression to solve but unfortunately it is not so because this will it is an exponential function here so it is not really easy to solve this you have to look at some other iterative method for solving this and the most popular method used is Newton-Raphson. I am not going to go into the depths of Newton-Raphson. People are encouraged to look it up.

(Refer Slide Time: 24:11)

$$\text{Newton-Raphson} \rightarrow \beta = \beta^{(old)} - \left(\frac{\partial^2 \ell}{\partial \beta \partial \beta} \right)^{-1} \frac{\partial \ell}{\partial \beta}$$

$X: N \times (p+1)$

$\bar{P}(i) \approx P(x_i)$

$\ell_{(i)} = P(x_i)(1 - P(x_i))$

But the basic idea is that so people were more comfortable looking at it this way so take the old estimate of your values or your take your old solution and look at the first derivative of the function that you are maximizing. ℓ' divided by ℓ'' so you adjust this by that so that is essentially the basic idea behind Newton-Raphson I am just defining some terms here so X is going to be my $(n \times p + 1)$ matrix as usual and my P is going to be a vector where each entry is going to be the probability of x_i . So what will be the dimensionality of P ? It will be n . So it is a n vector that tells me what is the probability of each x_i being one. So that is that is P and W is going to be a diagonal matrix where each diagonal entry is P into 1 minus P right for that particular at a point x_i so this makes it convenient to rewrite things and I am going to assume that g is the vector of outputs like zeros and ones depending on what class it is.

(Refer to slide time 27.13)

$$\text{Newton-Raphson} \rightarrow \beta = \beta^{(old)} - \left(\frac{\partial^2 \ell}{\partial \beta \partial \beta} \right)^{-1} \frac{\partial \ell}{\partial \beta}$$

$X: N \times (p+1)$

$\bar{P}(i) \approx P(x_i)$

$\frac{\partial \ell}{\partial \beta} = X^T(g - P)$

(Refer Slide Time: 24:46)

The image shows a handwritten derivation on a chalkboard. At the top, there is a partial derivative symbol followed by a fraction: $\frac{\partial^2 \ell}{\partial \beta \partial \beta}$. Below this, the derivative is written as $\frac{\partial \ell}{\partial \beta} = X^T(g - p)$. To the right of this, another partial derivative symbol is shown with a fraction: $\frac{\partial \ell}{\partial \beta} = -X^T W X$.

So I can write my $\partial L / \partial \beta$ is, it is $X^T(g - p)$. In terms of the matrices it is P and the vector Y is the vectors of zeros and ones corresponding to the class label and X is my input. So I am just basically written this in vector notation so you already found the derivative I have just rewritten it in vector notation that it make sense okay, so what about the second derivative so I am so I am not going to work it out. But it is $X^T W X$ okay and so W is essentially the diagonal matrix with this entries okay so that is my second order I made my second derivative so what do I get now putting this together.

(Refer Slide Time: 28:40)

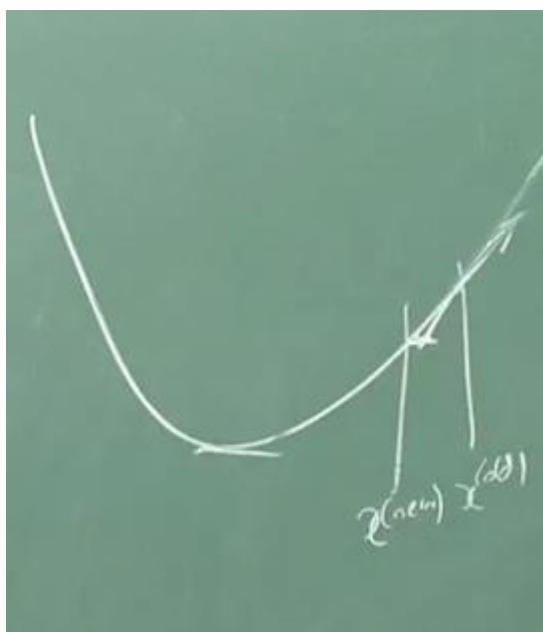
The image shows a handwritten update rule for β . It starts with β_{new} followed by an equals sign and a plus sign. Then there is a term $\beta_{\text{old}} + (X^T W X)^{-1} X^T(g - p)$. A hand is visible pointing towards the first term β_{old} .

The beginning look something like regression here alright you are getting your $(X^T X)^{-1} X^T$ and all that so we just have to do a little bit more work little bit of algebra to make it look more like regression so that's what we will do now. I just substituted the derivatives here, nothing fancy. So you want to solve this problem when this becomes 0 so you can see the β in here right no yes the β is in the P right so the right so I have erased the p off now but so β is in their P is the P is

$e^{\beta_0 + \beta x} / (1 + e^{\beta_0 + \beta x})$ is here I really want to solve for this right I want to find the 0 of this function right.

But it is not easy to do because of the fact that we have it exponential in there right so what we have to do is look at some kind of iterative method for solving this problem and so what the way we do this iterative approaches you start off with a guess called β_{old} . And then you do some computation you get a new guess call β_{new} . So one very popular way of doing this kind of iterative thing is to do “gradient following”. Have you have you looked at that? I mean you must have might have come across that this side so suppose I have a function like this, I am here this is my current solution right this is I will call this X_{old} okay, so now I will compute the gradient here right and I will move in the opposite direction of the gradient to find the minimum right so instead of going all the way I can go a small step that gives me the X_{new} right.

(Refer to slide time 31.22)



Normally what you will do is you will find the gradient try to equate it to 0 and get it but it can do this in iterative fashion also right you can take small steps in the direction of the gradient so likewise what we are going to do is we will start off with β_{old} which is some guess for this okay in fact β of all 0 actually works fine okay can start off by saying all my β at 0 okay and then try to find a β new. So what I will essentially be doing is I will find so β_0 will put me somewhere here on the L function right I will find out what is the first order and the second order derivatives

at this point with respect to β and then use that for changing my β values right so people agree with me so this is the $X^T W X^{-1}$. (Refer Slide Time: 32:08)

$$\begin{aligned}
 &= (X^T W X)^{-1} X^T W Z \\
 Z &= (X\beta_{\text{old}} + W^{-1}(g - p)) - \text{Adjusted response} \\
 &\text{argmin } (Z - X\beta)^T W (Z - X\beta)
 \end{aligned}$$

If you take the product here this will be $X^T W X$ so that is just the identity right and I take the product here I will get $X^T W^{-1}$ back this W^{-1} and will get cancelled out right so I have just done some algebra to get it this way think about it what is $X\beta_{\text{old}}$, so what is original I mean since it is like linear regression right it is like the original response I will get if β_{old} is my variables and I am actually prime making a linear prediction based on X right. So the $X\beta_{\text{old}}$ is this and I am essentially adjusting it you think this quantity so this is the prediction I make with my old parameters this is some kind of adjustment I am making to the prediction so this is called the “adjusted response” and this turns out to be the solution of something known as weighted linear regression it is in weighted linear regression essentially what you do.

So in linear regression that is what you are trying to minimize that is a square error right so linear regression this is what you are trying to minimize weighted linear regression you essentially have a waiting term in your error function right since I am just saying I am going to minimize the squared error for every term in the squared error I am going to assign a different weight right so for some data points I want to be more aggressive in minimizing the for some data point I want to be less aggressive.

So data points in which I have to be more aggressive I will have a higher weight for data points for which I want to be less aggressive I will have a lower weight so that will allow me to trade-off the importance of data points this is idea behind weighted linear regression right so this is essentially weighted linear regression so minimize be the minimizer of this right it is this okay

you can do the usual now you can take the derivative set it to zero and solve it this is easy enough to solve this is actual linear regression right.

So the minimize is $(X^T W X)^{-1} X^T W$ into Z right so essentially what we are saying is your β the β new is essentially solving a weighted linear regression or weighted least squares problem okay are solving a weighted least squares problem with this adjusted response so this is called iterative reweighted least squares this Is a separate algorithm called iterative reweighted least squares for solving logistic regression.

But all it does this essentially does Newton-Raphson essentially is doing Newton-Raphson but the way iterative rebated least squares is described to you is okay start off with a guess for β okay form the adjusted response okay as soon as I guess as soon as I have a value for β , I can find out what my P is so G is given to me already in the data and my W can be constructed once I know P . I make a guess for β , I construct my P . I construct my W .

Form this adjusted response solve this weighted least squares problem get a new β keep repeating this until my predictions are accurate enough okay so that is basically this is it is the most popular way of solving logistic regression but there are many other ways people have come up with more efficient ways of solving logistic regression actually and but if you pickup any popular package like R or something so IRLS is the base logistic regression solver that would be implemented okay. So this just to give you a flavor of how hard it can be to optimize things sometimes.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 21

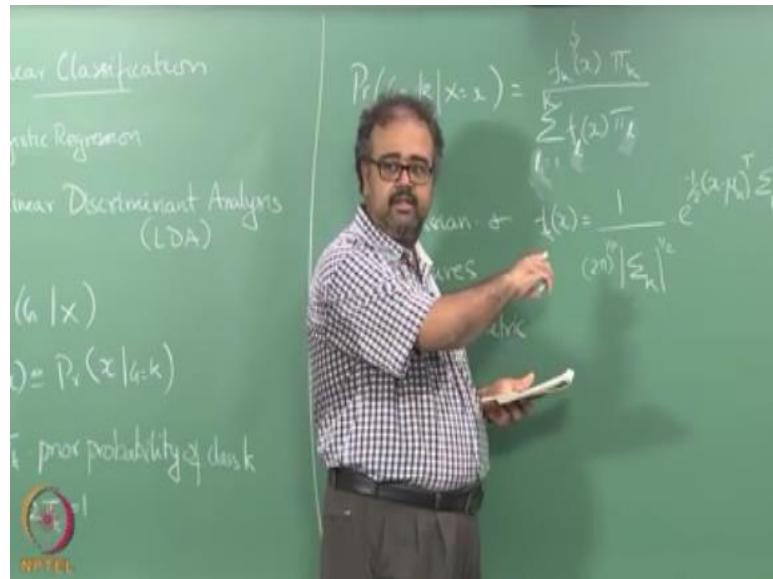
**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian institute of technology**

Linear Discriminant Analysis I

So we started looking at linear classification and in the last class we looked at logistic regression. So you remember the assumptions that we made for logistic regression. We assume that the log odds can be modeled as a linear function. So each of the individual the probabilities were given by sigmoid functions right but the log odds was assumed to be linear that is where we started off with and that gave us a linear decision boundary.

So the separating surface between two classes ended up being linear. If people remember what log odds was, it was probability of class one divided by probability of class zero and the log of that we assumed that was linear. So that's the assumption we made in logistic regression and today we look at another one of those discriminant based approaches. So we already looked at two, one was linear regression on an indicator variable and the second one was logistic regression. Now we look at the third popular classifier called Linear Discriminant Analysis.

(Refer Slide Time: 01:35)



And also known as LDA. Unfortunately in machine learning there are two very popular algorithms both of which are abbreviated as LDA so this was the older one okay linear discriminant analysis there's also something called Latent Dirichlet allocation which we will not get into which talks about a completely different approach to modeling distributions. It has nothing to do with classification. It is more on modeling distribution that's also sometimes abbreviated as LDA. So be context-sensitive when you use LDA. If you remember so we are really interested in the probability of a class given the data point. We are really interested in the probability of the class given the data point and you can get this using Bayes' rule. If you have the probability of the data point given a class and probability of the class. So probability of the data point given class times probability of the class divided by probability of the data point . So what we will do is we will start by making assumptions on probability of the data point given the class is k.

(Refer to slide time 4.13)

$$\Pr(g \mid x)$$
$$f_k(x) \triangleq \Pr(x \mid g=k)$$

π_k - prior probability of class k

$$\sum \pi_k = 1$$

So these are also known as class conditioned densities the class conditioned density of the data point apologies for that so I am going to denote by $f_k(x)$ probability of x given that the class was k. This is the class conditional density and I am going to assume that π_k is the prior probability of class k.

So we assumed that all data points belong to some class or the other. So that's going to be one.

So now I can write

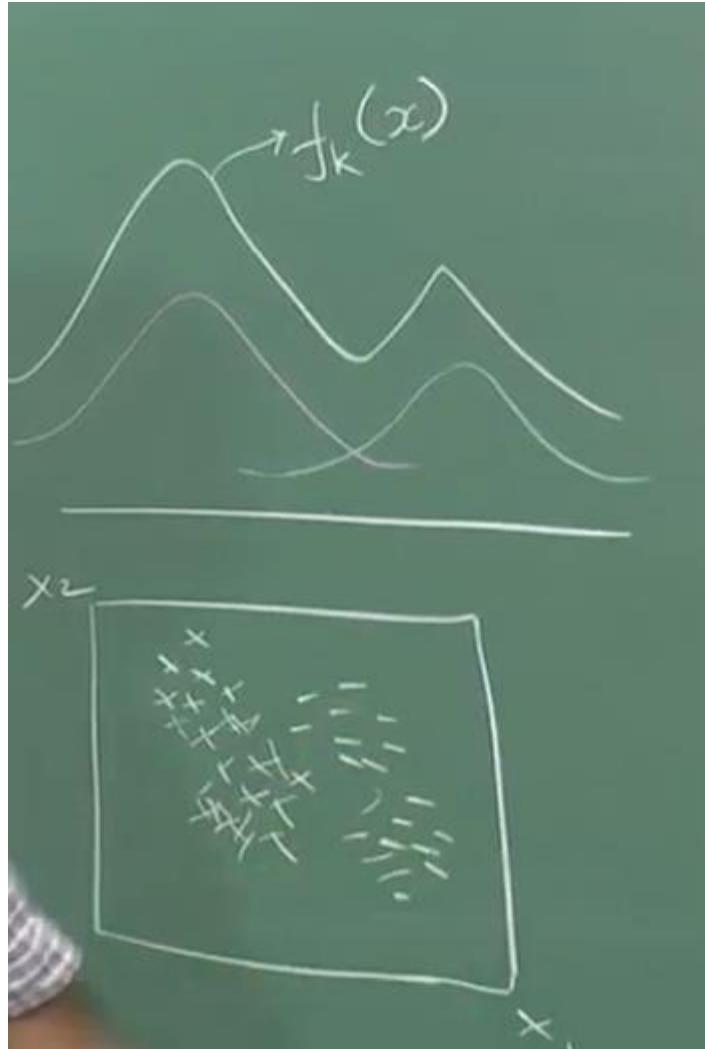
(Refer to slide time 5.26)

$$\Pr(g=k \mid x=x) = \frac{f_k(x) \pi_k}{\sum_{l=1}^K f_l(x) \pi_l}$$

So we have been using ‘1’ throughout so that makes sense. That is why I told you do not need the probability of the data. I can always fake that by saying that since the data has to belong to some class so I can just sum over all the classes I will get the probability of the data.

So this is essentially marginalizing over class and I get the probability of the data. And now depending on the kind of assumptions we make for our f_k the form of f_k , we will get different classifiers. So some of the most popular assumptions about f_k are that f_k is Gaussian for both LDA and a related method called QDA. Any guesses what QDA is? Quadratic discriminant analysis. Both of them assume that the class condition density f_k is given by a single multivariate gaussian. You could also assume that the class conditional densities come from mixtures so instead of a single Gaussian, you assume that there are multiple gaussians which jointly generate the data for you. So people are familiar with the concept of a mixture distribution? Very simple it is like let us do a little segue you here suppose I want to model this following distribution over univariate data right. So this single dimension and the axis actually tells me the probability of seeing something but what I want to do something like that can you think of a parametric form that will give me this kind of a distribution. Looks a little daunting right can you come up with like a closed form expression for this it looks little daunting right but if you think about it I can look at another Gaussian like that and I can suitably weigh that two of them and I can combine their distributions right. So the combined distribution will look like it has two peaks alright so this is essentially the idea behind mixture distributions. So if the form of the distribution I want seems rather complex right and I want to have a simpler functional form for the represent for the distribution I can think of writing it as a combination of several simpler distributions right.

(Refer to slide time 10.21)



So likewise suppose my positive class look like this, so how will it look like in a 2D setting? So let us think of it. So my data looks like this is the positive class there's another class that comes to here okay so if you think about it this there is more data points here and then there are more data points here and there is a slight region of lesser density in between the two.

If I try to model this as a single Gaussian and if I use any kind of maximum likelihood estimate where will the peak go? It peak probability will be somewhere here who is obviously incorrect right so where's the peak probability will be somewhere here for the negative class which is obviously incorrect. I suppose to that if I say that okay they're both the positive class and the negative class are created by two Gaussians each then the mixture of this so I can have one Gaussian which has a peak somewhere here other Gaussian which has a peak somewhere here

likewise one for this and one for this and then I can combine them using some kind of weighting mechanism.

All right so this is what we mean by mixture distributions giving you class densities. So you can think about this I mean I can have more arbitrarily complex kind of distributions here and then instead of having two Gaussians I can say okay I am going to have ten and also in they need not be Gaussian say could be other functional forms but the more complex the forms I take the harder it is going to become solving this problem. So fk if you remember it's the probability of x given G right given that class is k . So this is a probability that given the classes case so likewise so if I have two mixture of two Gaussians here that'll be the probability of the data point given that the class is X and here they are given that the class is whatever. So that is what we are mapping here. So mixtures are fine if we still want to stay in a parametric space. Well then yeah so that is a hard problem right so usually you take some guess from whatever knowledge you have about the domain right or you can do some preliminary experiments you can try to run some kind of rough clustering by varying the number of clusters and trying to see whether you can decide on the number of mixture components alternatively. You could do nonparametric methods they are more complex but in the last 5, 6 years so lots of tools have been developed to be able to handle these kinds of nonparametric reasoning okay. So nonparametric is actually slightly misleading it does not mean that it does not have any parameters. It only means that it has an unbounded number of parameters. So it does not mean it doesn't have any parameters it is just that I do not fix it a prior like we are fixing the mixture component I'm saying the Gaussians per class. I fixed it when you fix these things we call them parametric and nonparametric methods typically can add parameters if the data needs it right you can start off with just one Gaussian okay.

And then figure out oh no I need more then I can add another Gaussian here I can add another Gaussian and so on so forth. So that is essentially what nonparametric methods bias. The ability to grow the number of parameters needed if it is supported by the data and if the data warrants it. So obviously we'll have to be very careful about doing things like over fitting the data but there are other ways of adjusting for it so I like I said in the last five years a lot of powerful techniques have come up for nonparametric reasoning but I'm not going to cover any of that in the class like

I keep reminding you people is “intro to ml” course. If at all we do an advanced topics in machine learning course then we will probably cover some of that right there hoping to hire a few more faculty members who can start taking all of these courses it was the most popular.

So I do not fix the bound a priori. I do not say that okay you can use only three Gaussians per class. So you can keep adding more Gaussians if the data warrants it that's what I meant unbounded. So I don't put the bound a priori and obviously I mean that is always a physical bound but in the modeling sense I don't bound it apriori I don't say that oh you have to use ten Gaussians.

So the most popular of assumptions that people typically make on f_k is sometimes called the Naive Bayes assumption. We will deal with this separately and just putting it out here to just to tell you that all of this come under the same class. So the Naïve Bayes assumption is essentially to factor my class condition density along each dimension assuming given the class one dimension does not influence the other dimension.

Right so if I have two dimensions here x_1 and x_2 . So I will say that I can write the probability of x given K as probability of x_1 given K times probability of x_2 given K that is a very strong assumption if you think about it and I'm saying given the class x_1 is independent of x_2 - if I don't know the class it look like there is some dependence between x_1 and x_2 but given k , I'll ask you x_1 is independent of x_2 . So this is essentially called the Naive Bayes assumption because looks like a very simplistic assumption or looks like a very naive assumption.

To make about the data so it's called Naive bayes and it turns out to be powerful in many settings we will come back to Naive bayes separately in one of the later classes and so right now I am going to go back to Gaussian. So I am going to assume that. Does it looks familiar? That's the Gaussian distribution.

(Refer to slide time 16.01)

The image shows a chalkboard with a handwritten mathematical formula. The formula is:

$$f_k(x) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}$$

The board also has some other faint writing and a piece of paper at the bottom left.

So that is Σ . So if I'm looking at a univariate Gaussian I will write the variance here looking at multivariate Gaussian this is the covariance. This is the covariance matrix this is multivariate Gaussian and here again this is what $(X - \mu)^2 / \sigma$. So people must be familiar by now when I say $(X - \mu)^T (X - \mu)$, that is actually the above in the vector sense and then the sigma becomes sigma inverse and this is the covariance matrix.

So this is the called the multivariate Gaussian. So the multivariate Gaussian will capture these kinds of scenarios where the input rate of dimensions itself is 2 and now I have to have a Gaussian that is actually jutting out of the window and the board.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

Copyrights Reserved

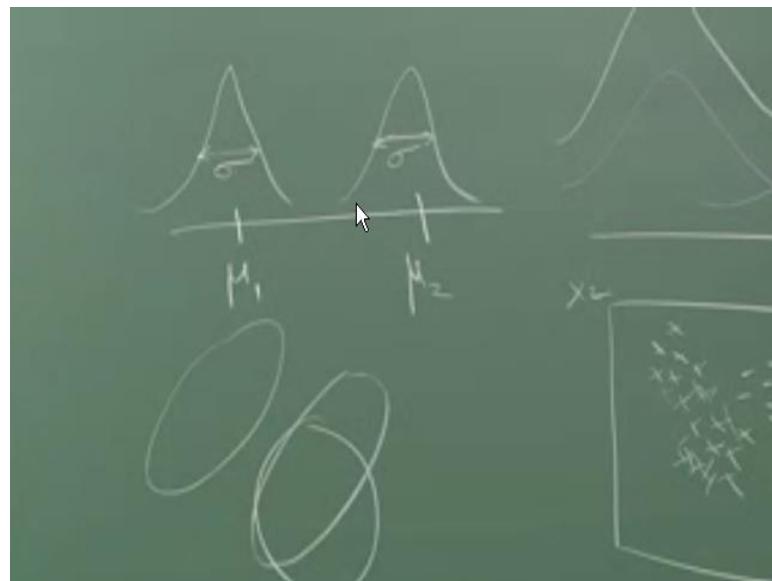
Introduction to Machine Learning

Lecture 22

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Linear Discriminant Analysis II

(Refer Slide Time: 00:17)



Okay so in LDA we make an further assumption than the fact that the class conditional density is Gaussian. So what else can you assume? So I am going to assume $\Sigma_k = \Sigma \forall k$. It is the same for all the classes k. So what does that mean? It means that I will do that in 1D , say, class one could have the mean of my Gaussian here class two could have the mean of the Gaussian somewhere else but when I look at this right so that is the same.

And all I can do is shift the Gaussian around but I cannot change the shape of the Gaussian. Is it clear, when I say \sum_k it is the same for all the classes? It essentially means that I can just shift the Gaussian around, I cannot change it so in 2D it will be like okay let us, let us assume that that is the equivalent of one \sum contour right if that is the case for one class right for other class also it has to be similar okay.

(Refer Slide Time: 02:53)

Linear in x

$$\delta_k(x) = \frac{1}{2} \log \left| \Sigma_k \right| + \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

$$P_k = N_k / N$$

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

$$G(x) = \arg \max_k \delta_k(x)$$

So I cannot have one class looking like this and the other class looking like that. Does it make sense? People are able to visualize what I mean? So it has to be that both the classes look similar in terms of the variance that is essentially the assumption we make here. So in looking at logistic regression we saw that we could look at the log odds, likewise I am going to look at $\log \frac{\Pr(G = k | X = x)}{\Pr(G = l | X = x)}$. So when I am at the class boundary what will this ratio be? One and log of that is going to be zero.

So I can actually solve for this ratio equal to zero and I can get my boundary, So now we know what is the form of $\Pr(G = k | X = x)$. Now I can put that in here and I can solve for what? What

are the parameters I should be solving for? μ and Σ and anything else we are talking about this we had it is all for π as well. Solving for π is rather straightforward.

I mean you just count the number of data points that belong to a class divided by the total number of data points you have that gives you π . So it is not like complex but you still have to solve for it as it is not that it is given to apriori. So you have to estimate from data so all these three parameters your estimate. So this gives you the boundary right so when the probability of it belonging to k is higher than the probability of it belonging to l then you will put it in k right.

So you will have to do this for every pair of classes to make sure that you have the right class so assuming that only two classes just you have to make one comparison but of course there are really k classes then will have to make $k-1$ comparisons to figuring out which class it belongs to. So this essentially will give you the boundary. So when the probabilities are equal then you know that well it could go either way so this is going to be 0. In the problem it actually belongs to class k the numerator will be higher when it belongs to class l the denominator will be higher so based on that you can decide which side it is going to go. What about now solving for this?

This is essentially log of

(Refer to slide time 5.55)

$$\log \frac{P_r(C=k|x=x)}{P_r(C=l|x=x)} = \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l}$$

So the denominator will get canceled out like you only worry about the numerators okay and the fact that we assume that the variances are the same is also going to allow us to cancel out a whole bunch of other terms.

So what other terms can cancel out that can go right so this is \sum_k it will become \sum so when I take the ratio of the two things this thing can go right so I do not have to worry about that and it is log and there say e right so all of that will go away right so.

(Refer to slide time 6.57)

$$\log \frac{P_r(G=k|x=x)}{P_r(G=l|x=x)} = \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l}$$

$$= \log \frac{\pi_k}{\pi_l} - \frac{1}{2} (\mu_k + \mu_l)^T \Sigma^{-1} (\mu_k - \mu_l)$$

Roughly the way to think of it is if I had taken out the product here. If you think of the term by term product here, so I will have some terms that will have an x^2 okay sometimes that is going to have $X\mu$ okay and some terms that will have μ^2 right.

So I am taking the ratio so I am going to get $\mu K^2 - \mu L^2$. So that is essentially what I am writing out here the first term here corresponds to $\mu K^2 - \mu L^2$ so you have to get familiar with doing this in the vector notation it makes life a lot easier. I am just giving you the intuition here if you can write it out and see that this is the right way to simplify it. So if you think of this essentially you are going to have a x^2 term μx term under μ^2 squared term.

And we take the ratio so you will have e power this divided by e power something else so that is going to become minus μ in the numerator so you are going to have $\mu K - \mu L^2$ right so that is essentially what I am writing out here. What about x^2 terms? x^2 will get cancelled out because \sum are the same right so \sum_K and \sum_L is the same so x^2 will get cancelled out I only have the X terms left. So I will have $X\mu K - X\mu L$ so I am going to get that term as well so turns out that this separating hyper plane that we have this is essentially the solving this for zero gives us the separating hyper plane the separating hyper plane turns out to be linear in X . It is a separating surface turns out to be a hyper plane.

So I should be saying it that to get the linearity we needed to make this assumption if you do not make this assumption so what will happen? The x^2 term will stay there and what we get this is

QDA. I told you about QDA. So if I do not make this assumption I will get you QDA. I said in discriminating function case we are we always have some function like this $\delta_k(x)$ right and if $\delta_k(x)$ is greater than any other $\delta_l(x)$ then we will classify the x into k right.

This is what we said was the idea we had discriminant functions in the very beginning so what would be the discriminant function version of LDA Please note that for most of this bar anyone had him Σ here on this side okay this covariance okay and I will make sure I write limits whenever I write this summation Σ this work whenever we use multivariate Gaussians okay so I knew I left out one somewhere.

So this is essentially the discriminant function. So you can just compare this and this is whichever has the highest discriminant value will become the class so $\hat{\pi}$ like I said earlier you count the number of data points in the training data there belongs to class k divided by the total number of data points you get $\hat{\pi}$ and $\hat{\mu}_k$. You pick out all those data points for which the class was k from the training data find the center of them that gives me $\hat{\mu}_k$ they make sense right.

So what about Σ ? What is that μ ? Like well presumably so because these are all data points that belong to one class, so when the training data comes I am assuming that I have sufficient number of data points of each class otherwise I will not be able to learn anything it will work to the extent possible with the small data set. Suppose I give you only ten data points for training then you are anyway in a soup like most of your parameter estimation algorithms will not work if you have very little data there are some class of algorithms is work with very little data right.

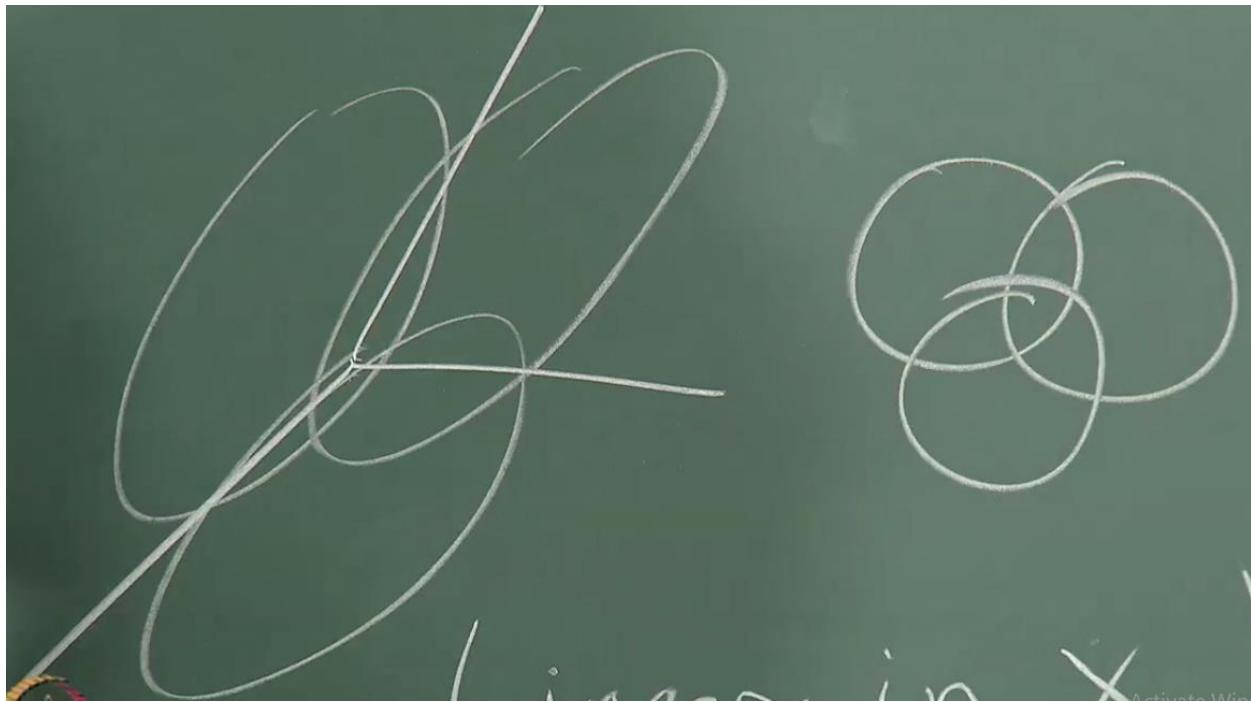
So one such thing which we will look at depending on time today or tomorrow it is support vector machine. So it works with very little data but most of the other parameter estimation methods require you to have some amount of data. So what we do with the variance? So if you remember that variance is not limited to the class okay it is across all the classes we really want the same variance so essentially we do what is called a pooled estimate.

So we essentially use all the data points for estimating the variance not just the data points belonging to one class. But then you know what is variance. So variance you add that up and divide by minus one. So remember that you always do a minus one for variance estimates it is cool stuff given, given a sample mean and sample variance you must have done all of that right sample mean is divided add up the data points divide by N and sample variance is data point minus μ^2 divided by N - 1 usually right to adjust for the fact that mean is a dependent variable on all the data points okay.

So the N-1 essentially gives you an unbiased estimate of the variance. But then what is the mean? You plug in here the mean corresponding so the mean of mean is a bad idea no, no, no so now I thanks for bringing it up because that is a natural confusion when they say you are going to estimate the variance across the entire population so the natural mean you are going to plug in is the mean of the mean of the whole data. But that is not correct. Why? Because I am only worried about the variance within the class so I should plug in the mean here of the class of x_i remember all of this is from the training data so I know what class x_i actually belongs to. So I take the class of x_i so I ask you I will take all those data points belonging to class k and then I will use μ_k here in computing this quantity. Then I will do this over all classes.

And then I will divide by k okay well divide by n-k. So this is a slightly different way of doing the variance as opposed to computing the variance of each class and taking some kind of a mean right this gives you a slightly more robust estimate of the variance okay so this is called a “pooled estimate”. So if you think about let us say I have three classes that look like that so this will become what the separating surfaces that I learn if this had been completely spherical if this had been like this then the separating hyper plane would have been perpendicular to the line joining the means.

(Refer to slide 17.57)



This is something which I just want you to note in fact I can ask you to show that it is fairly straightforward on univariate Gaussians. But if it had been spherical it would have been perpendicular to the line joining the means because I have now this is slanting so the separating hyper plane will also be at an angle to the line joining the means. If you look at many pattern recognition text books right they will talk about LDA as a feature selection mechanism right.

So you remember we looked at PLS when we did regression right so we did the principal component regression where we said we are looking at the directions in the input only taking into consideration the input right so that we are looking at the direction that maximizes the variance in the input and then when we did PLS we took into account the class labels as well.

So the equivalent of that in classification is LDA. So you can think of LDA as actually finding directions along which the variance between the classes is maximized at the same time minimizing the variance within the classes right so PCA just maximizes the variance of the data right LDA maximizes the variance between the classes how does it achieve that it tries to find a direction say set the means or mean of the data of each class is as spread apart as possible. So in

this case suppose this is the mean I am choosing this where I was using some direction where the mean such as spread apart as possible right so this is essentially the idea behind LDA right and we will just take that as our assumption.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

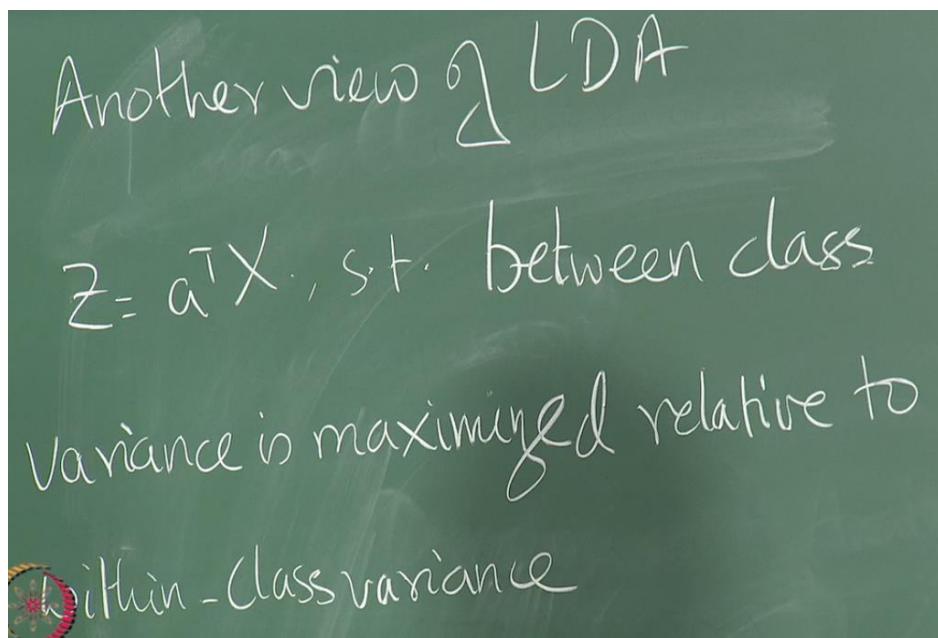
Introduction of Machine Learning

Lecture 23

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

Linear Discriminant Analysis III
- Another view of LDA

(Refer Slide Time: 01:14)



Okay so when I say between class variance, I say it is the variance of the class means. So I will take the classes look at the means of those classes and look at the projected means of those classes and compute the variance among the projected means. Suppose I have "k" classes I can compute the variance among those. If I have two classes, what will this amount to? Maximizing the distance between the projected means and if it is "k" classes, it will be maximizing the variance among the k centers relative to the within class variance and what would be the within class variance? For each class the variance with respect to the class mean and that is what we

already computed there but for each class. So within class variance that essentially what I'm looking at here. So let us just treat the first condition alone. So for simplicity's sake start off with a two class case and then we can think of the generalization to multiple classes. So I am going to have a surface defined by $w^T x$ so $y = w^T x$ if it's greater than some w_0 , I am going to classify it as class one just less than some w_0 or less than or equal to, I will classify it as class two.

I am going to say \bar{m}_1 and \bar{m}_2 are the means of c_1 and c_2 and well we know how to compute \bar{m}_1 just like you do a $\hat{\mu}$ there and I am going to assume that when I write the m_k without the bar, this the projected one.

(Refer to slide time 3.58)

$$\mathbf{m}_k = \mathbf{w}^T \bar{\mathbf{m}}_k$$

So $\bar{w}\bar{m}_k$ I should see the projection of the mean in the direction w^T . So that is essentially what this is. So the reason I am using this funny notation is in the textbook if this is bold it is m_1 if it is unbolted it is a projection but I cannot write bold every time on the board.

So I am just using the bar right then when you read the book you can translate back and for this you read this part alone from PRML (Pattern Recognition and Machine Learning) by Bishop the textbook reference is there. The rest you can get from so till that part it is from Hastie Tibshirani Friedman the ESL. So what is my goal when I say I want to maximize between class variance. It is essentially to maximize the quantity $w^T(\bar{m}_2 - \bar{m}_1)$. $w^T m_2$ is the projection of m_2 on w , $w^T m_1$ is a projection of m_1 on w . I'm trying to maximize this quantity so that is essentially my first criterion right.

The direction w that maximizes this right so there should be some alarm bells ringing for you what is the problem? If I do not have any bounds on w , I can just arbitrarily scale my w and get larger and larger values. So I will have to have some constraints assuming summation over.

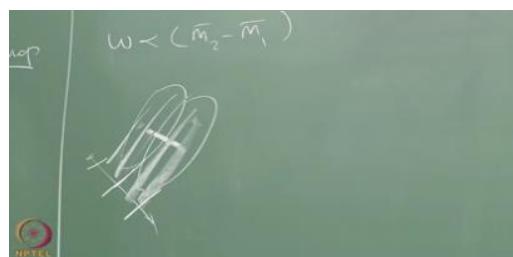
(Refer Slide Time: 05:58)

So essentially the norm w is one. That is an assumption we will make frequently to make sure that we do not get unbounded solutions. So this is numerically bounded.

Student question: Why can't we impose an inequality here?

Yeah good question so you could impose a inequality constraint saying that summation w square is less than 1 but what do you think will happen? You are maximizing the value and you can just scale it so essentially what will happen is you will scale it such that w_i hits 1 anyway. So even if you have the lesser than or equal to constraint because you are maximizing over w you will hit it you will essentially scale w till you hit 1. So you might as well leave it as equal to 1 right.

(Refer Slide Time: 07:39)

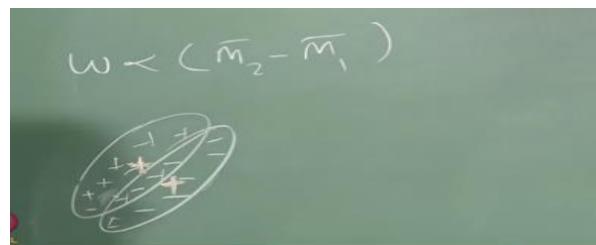


So you can solve this right but the take-home message is that your w is going to be $w \propto (\bar{m}_2 - \bar{m}_1)$. So w will be proportional and you add that here right you take the derivative ' w ' will go and that will become w , so there will be some constants here but essentially you are going to get w will be in the direction of $\bar{m}_2 - \bar{m}_1$. So what does this mean? Take the means and if again you can go back and show that if it is spherical then the constant will be half so it will be the midpoint of the line dividing that two means.

So let us do it again so I have two classes and I take the means. So this will be the direction of the projection. I predict everything on to this right this way this will become class one that will become class two. Does it make sense? So in this line and this line are actually parallel to each other I know you really did not want me to repeat the drawing but I think that will help. So I have class one I have class two, so I mean if you look at the data point that comes to me so the people understand when I say class one class two like this do you know the direction what I mean.

So this is the Gaussian corresponding to class one I am drawing the 1σ contour of that this is likewise the 1σ contour of the second Gaussian. So the data point that comes to me could be something like this. This could be the training data that I am getting it will be a mixture up of + and - in this region. That could be minuses here also. Already I drew one that could be minuses here that could be pluses here because the Gaussian still does extend beyond the contour I have drawn, the contour is only the most probable region for the data points to lie does not mean that outside this contour the probability is 0 okay. So this is essentially what it means. So I am going to get data like this and I am going to model it I am modeling the Gaussian by these contours.

(Refer Slide Time: 10:35)

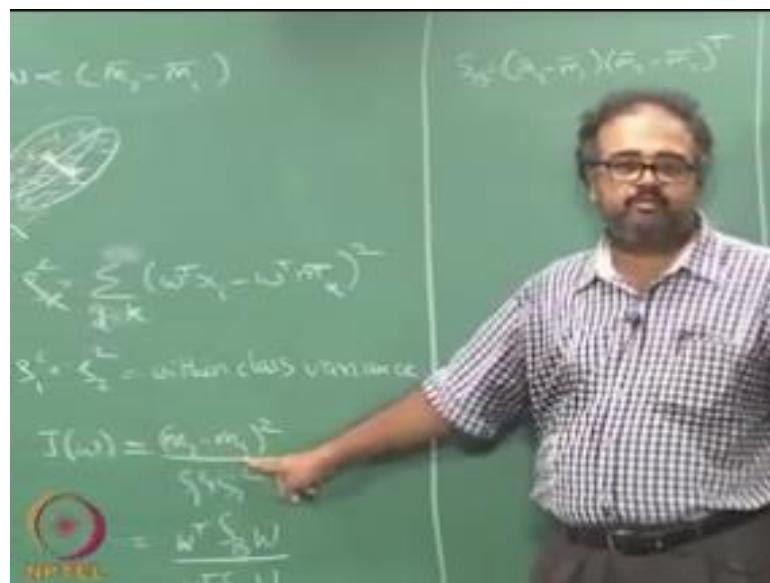


Roughly that these points are the centroids of the data that I get. So what this tells us is that can you join this by a straight line okay and essentially you take direction that is all right like this and project all the data points to that so you will get all the data points lying here now fix up threshold that what that is what I wrote here as w_0 pick a threshold such that above that it is class 1 below then it is class 2 right.

In this case in fact if this had been spherical you can show that the threshold would lie in the midpoint now we cannot because well you can I would guess I mean depending under special circumstances but now the point will be somewhere here and all the data points is projected above this I will say it is plus all the data points are projected below this I will say it is minus that makes sense right.

But then this is not what we are looking for. We are missing something important. What is that? The within-class variance. So this is the inter class variance within class variance is what we are missing. So what we will do now start looking at that right.

(Refer Slide Time: 12:34)



(Refer to slide time 14.33)

$$\begin{aligned} S_k^2 &= \sum_{g_i=k} (\omega^T x_i - \omega^T \bar{m}_k)^2 \\ S_1^2 + S_2^2 &= \text{within class variance} \\ J(\omega) &= \frac{(\bar{m}_2 - \bar{m}_1)^2}{S_1^2 + S_2^2} \end{aligned}$$

So that is a projected mean these are the projected data points belonging to class one. Keeping in with the terminology are using there so I'm picking on all the data points training data points which had class k and looking at the projected distance from the projected mean. This gives me the total within class variance. Why is there no n in the denominator? Where I'm going to maximize everything at the end. So I am just ignoring the things that do not affect the maximization of that. The squared term S_k^2 is essentially the projected data and that is a projected mean and just taking the variance of that is exactly what we did that except that I have not divided by the number of data points okay right.

So this criterion is called the official criterion it is called the "Fisher criterion" after Fisher who was a very famous statistician who came up with LDA several decades ago. So here I am going to do something confusing so I am going to rewrite it so this is the between class covariance matrix. So if you think about it so what I wanted was $\bar{m}_2 - \bar{m}_1$ what is \bar{m}_2 the projected right so the projected one, so \bar{m}_2 will actually be $w^T \bar{m}_2$ right so essentially I have $w^T \bar{m}_2 - w^T \bar{m}_1$ so I can take

out the w^T and just have the square of the $\bar{m}_2 - \bar{m}_1$ and I am adding the w^2 back in okay by doing $w^T S_B w$ okay. Now what about S_w ?

(Refer Slide Time: 17.06)

$$S_B = (\bar{m}_2 - \bar{m}_1)(\bar{m}_2 - \bar{m}_1)^T$$

$$S_w = \sum_{j=1}^k (x_i - \bar{m}_j)(x_i - \bar{m}_j)^T$$

$$+ \sum_{j=2}^k (x_i - \bar{m}_2)(x_i - \bar{m}_2)^T$$

So likewise so I have this as my S_k^2 so an $S_1^2 + S_2^2$ is essentially this is this S_1 right I had took take out the W from there and this is S_2 I take out the W from there so that gives me the $w^T S_w w$. So now what we want to do we want to maximize this. We want to maximize the between class variance relative to the within class variance that is what we said between class variance is maximized relative to the within class variance so that is between class variance is within class variance I have to take the ratio now I am maximizing this ratio.

So differentiate with respect to w and set it equal to zero. So this is what you use u/v method for differentiation. So people want to tell me what the differentiation will be? I will write it but you should recall all of this childhood memories and you should not forget whatever you studied to get in here. Like so the denominator in the thing will become zero because I equated to zero already so when you take the derivative of this you're going to get some term in the denominator.

So that will go to zero so I will just have to equate the two half's in the numerator and I will get this. So just refresh your derivatives the only thing that I am pretty sure putting everybody off is the fact that we are doing all of this in the matrix notation. Just practice it makes life a lot easier do it a couple of times. The best way to do it is try and write it out in matrix form in gory detail okay do the term by term the derivative of it and then look at how it simplifies after you do the derivative right then you will see the pattern and then you will know exactly what we are writing it it's a very simple things like there are quadratics so you should know how to differentiate quadratics that is the only thing that is throwing you off right $w^T w$ is actually a quadratic in w . So that is the only thing so it becomes a linear in w so that is all nothing more to it actually if you think about it $S_B w$, will always be in the direction of $\bar{m}_2 - \bar{m}_1$. You already saw that here when we had only the constraint on S_B . So here that the constraint was only on S_B only on the between class variance and when he had the constraint only on the between class variance we ended up finding out that the solution is going to be the direction of $m_2 - m_1$.

And a little bit of work you can show that always that $S_B w$ will be in the direction of $m_2 - m_1$. So I can actually drop that and replace that with a vector proportional to $m_2 - m_1$. So now it makes our life a lot easier right I only have one w left so what about these guys.

(Refer Slide Time: 21:22)

$$(w^T S_B w) S_w w = 0$$

$$(w^T S_w w) S_B w = 0$$

They are all simplified to some kind of scalar quantities right so finally what I will get this w is not equal to but proportional to so that is essentially what I will get so if I did not have the S_w constraint what I got was “ w ” as proportional time to $m_2 - m_1$ right. But now if I am taking into account the within class variance also then I will have to pay attention to the within class covariance matrix.

(Refer to slide time 21.56)

Diff wrt w set equal to 0

$$(w^T S_B w) S_w^{-1} = (w^T S_w w) S_B^{-1}$$

$$w \propto S_w^{-1} (\bar{m}_2 - \bar{m}_1)$$

So I will have to pay attention to the within class covariance so that is basically all there is to it. But how does this relate to this? I see any relation between this and that think about it that is basically what we are doing there right. So Σ^{-1} is S_w^{-1} just using different notation here. So S_w^{-1} is just taking the variance between the in the data right in the within class variance so Σ if you remember is the within class variance matrix right.

So that gives me Σ^{-1} here and this how I got Σ^{-1} here and then I have $m_2 - m_1$ and I have $\mu_k - \mu_l$ here. So essentially for modulo all of these other non X related terms so we are essentially finding the same direction right so whether you do it this way starting with that is your objective

function right between class variance and within class variance or you start off by saying that your class condition density is Gaussian and then you are trying to find out the separating hyper plane.

So in both cases you end up with the same direction modulo some scaling factors right, so you can use either motivation for deriving it but what is the nice thing about this motivation we did? It does not make any assumption about the class conditional distribution the Gaussian assumption is missing here and we worked only with sample means and sample variance and so on so forth.

So it just tells you that LDA does not work only when the distributions are Gaussian. They are fine even when the underlying distribution is not Gaussian that is actually well-defined semantics to doing LDA. People are with me on that so far. So any questions let us let them move on to the next thing what does $J(w)$ represent? So I want to look at the between class variance relative to the within class variance right so the numerator is the between class variance and the denominator is the within class variance so I'm trying to maximize the relative score.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Tutorial on Weka

<http://www.cs.wekato.ac.in/ccd/weka/>

Hello and welcome to this tutorial on Weka. Weka is an open source and freely available software package containing a collection of machine learning algorithms. The algorithms present in Weka are all coded in java and they can be used by calling them from your own java code. However, the algorithm also provides a graphical user interface from which the algorithms can directly be applied to data sets. For the programming assignments in the introduction to machine learning course, we will mostly be using Weka in its GUI form.

This will allow us to spend more time on understanding how the algorithms which we come across in the lectures actually work and how to use them in analyzing data. You can download different versions of Weka for different operating systems from the website of the University of Waikato. This tutorial is mainly aimed at people who have never used Weka before we will look at some of the basic features and options provided by the software and also do some linear regression experiments to help you in solving the questions in the third assignment.

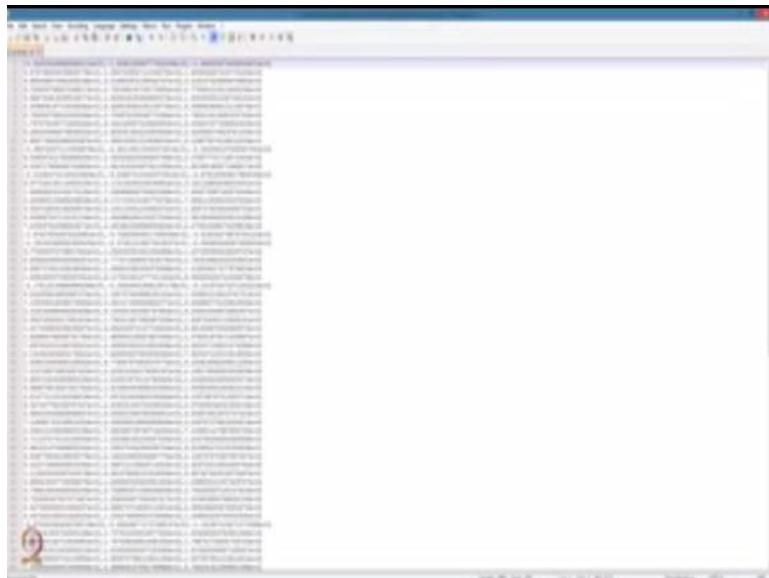
Before we start with Weka let us create a synthetic data set on which we can then apply linear regression. Since this is just for illustration purposes, we will create a simple one dimensional data set. We will create this dataset using a few lines of Python code. So here we have imported the numpy package. The statement creates the input data which ranges from -25 to 100 and consists of 100 data points. Now we will get the output data which will have a linear relation with the input.

Since we are creating a data set here, we know how the input and the output are related. However if this data was given to us, then our objective would be to try to run this relation that is the

parameters $\beta_0 = 12$ and $\beta_1=3$. As we will see when this input that is xy pair is provided to the linear regression algorithm. It will be able to learn a perfect model; this is because there is no noise in our data. So to make things a bit more challenging we will add some noise to the output.

The variable z is essentially the noise character output where we have used Gaussian noise with parameters 0 and 3. We will now save this data and apply linear regression on it using Weka. We have saved our data in a text file.

(Refer Slide Time: 04:09)



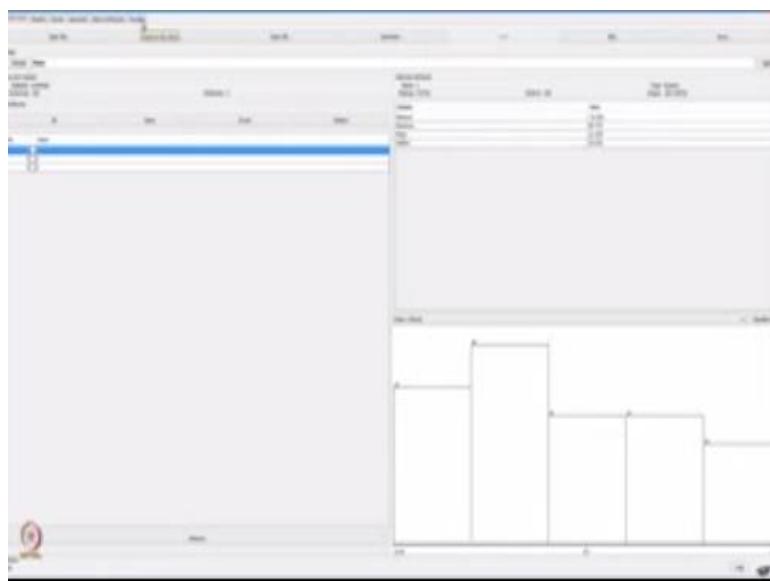
So let us have a look now Weka uses a specific format for its input it this format is known as a ARFF and we have to add a few bits of information to what is essentially a CSV file to make it suitable for use Weka. We essentially have to provide three pieces of information this first one just gives a name to the data. Eventually specifying what relation this data is showing, so since we have cooked up this data we are just given it the name synthetic. Next we provide the attribute information.

We have used x, y and z as the names of the three columns and specified the data type as numeric. There are other data types which will be seen a little later such as nominal and string

data type. The final piece that has to be provided is the data which we have already listed @data specify the start of the data. Now we should save this in the ARFF format that is with the extension .arff. Hopefully this gives you an idea of how to represent data in the ARFF format suitable for use with Weka.

Just to recap you have to provide the relation and attribute information and the data is listed row wise that is each row specifies one data point with the values being separated by commas. We will now open this data in Weka and apply linear regression on it.

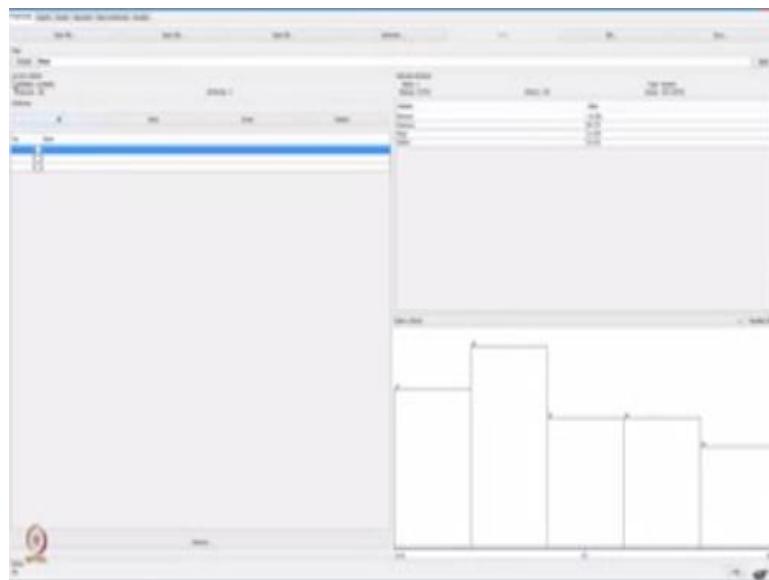
(Refer Slide Time: 06:17)



So this is the opening screen for the Weka application. We will be using explorer. So this is the start screen for the explorer application. As you can notice most of the options are grayed out this is because we do not have any data selected yet. So let us do that this is the synthetic data that we had just created, so let us open that. Right a lot of things to notice here. First of all at the top we have these tabs which allow us to specify different actions. So the first tab is preprocess where we can do different preprocessing activities such as normalizing the data, filling in missing values, in case the input as missing values and so on.

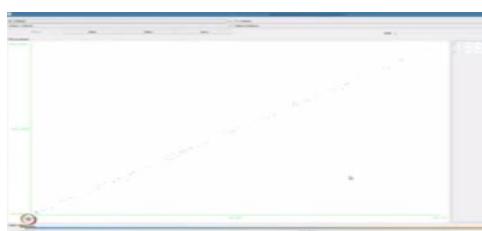
Under the classify tab are listed all the supervised learning algorithms, that is both classification and regression algorithms which will be looking at soon. Clustering algorithm was listed on the cluster tab. Association rule mining algorithms under the associate tab in. Under the select attributes tab we can perform attribute selection activities such as is subset selection PCA and soon and finally visualize allows us to visualize the data.

(Refer Slide Time: 07:42)



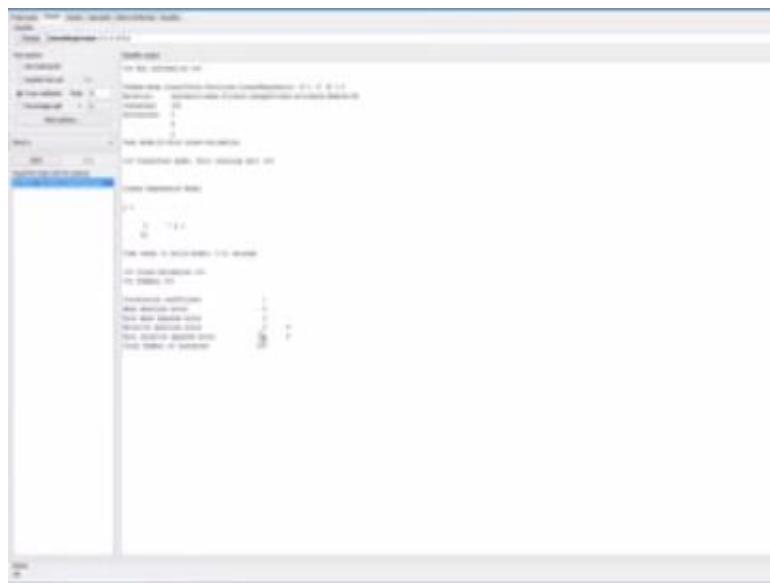
Let us have a look at that here we have the scatter plots between each pair of attributes in the data; this allows us to visualize the distribution of the data. For example, if we look at the scatter plot between y and x, we can observe the perfect linear relation between the two variables. Since that is how we created the data, however if we look at the scatter plot between z and x.

(Refer Slide Time: 08:10)



We can see the effect of adding noise to the output. Getting back to the preprocess tab here, we have the relation information, the name of the relation is synthetic and there are 100 instances and three attributes. The attribute window gives the list out the attributes. We have three attributes x, y and z, on the right-hand side for the selected attribute, we can use some information. So right now the attribute x is selected, so it has 0% missing data there are 100 distinct values with each value being unique. Some basic statistics which has mean maximum mean and standard deviation and at the bottom we have a histogram. Now let us go ahead and apply linear regression on the data. For our first attempt we will use the first two columns that is x and y and remove the noise character output set. So first we come to the classify tab.

(Refer Slide Time: 09:17)



Here we have to choose the algorithm, so which we will choose functions and linear regression. Note that there is a simple linear regression function which is actually suitable for this specific task because we have only one dimensional input but when the dimensionality of the input is more, we need the linear regression function. So let us just use this function these are the default parameters, for the linear regression function. We can change them by clicking here; the first parameter is the attribute selection method that is the method used to eliminate attributes which not contribute to the learning of the model. Since we would like to handle this ourselves, we will

select no attribute selection we will not be using the debug mode the third attribute the third parameter is eliminate collinear attributes which essentially allows Weka to identify and remove attributes which have a high correlation. We will set this too false for now. The final parameter is the value of the regularization parameter. Note that we are using Ridge regularization here.

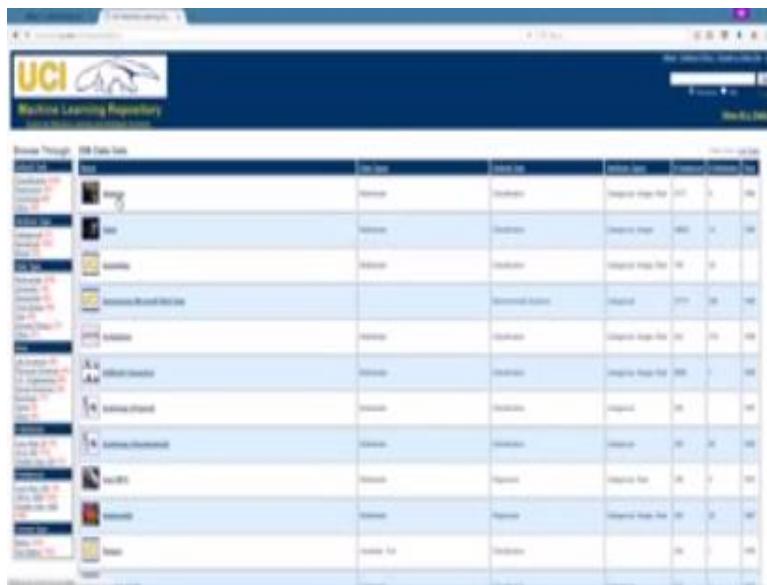
So initially we will take this to zero, that is we will not be using any regularization. Having set the parameters of the linear regression algorithm, we now look at the different evaluation options. The first option is to use the training data that is we use the training data to build a model and then use the same data to evaluate the model. In case we have a separate data set for testing that is a portion of the data which has not been used in training the model. We can supply that here.

More common you will be using cross-validation. In cross-validation, we will iteratively partition the training data into testing and training slits. In each iteration we will train on the training split and evaluate on the testing split. The partition will be done in such a way that each data point will appear in the train and the testing split at least once. The purpose of doing this is to get a robust estimation of the performance of the model. We will be discussing the concept of cross-validation in more detail in upcoming lectures. For now we can go ahead and use this evaluation method. Note that the number of folds simply indicates the number of your iterations and the sizes of each set. We have ten folds which mean we will partition, the data and build models ten times with each partition being a 90:10 split between training and testing. Only the percentage slit option allows us to split the training data and keep a portion for testing. Next this drop-down box allows us to select the output attribute. That is the attribute which we are trying to predict. With all settings in place we can execute.

The main window displays the results of the execution of the algorithm. Going through the output, we see the linear regression function used with the following parameters. The relation synthetic on which the filter applied is removing of the column 3. There are 100 instances two attributes namely x and y we use 10-fold cross-validation for evaluation and this is the linear regression model that we obtained. Note that we are able to recover the exact parameters values used in constructing the data.

The correlation coefficient is 1. The correlation coefficient specifies the correlation between the predicted output and the actual output. Here it is perfectly correlated and each error term is 0. Now for more meaningful evaluation we will use the noise corrupted output. So we can recover the column which we deleted by selecting the undo button and now we will remove the y column. Going back to classify we will have all the parameters will leave it same and just execute the algorithm. Note that Weka uses the z attribute for prediction. In general, Weka will use the last attribute listed for prediction or classification. Here we observe the effect of noise in the output variable. The parameters estimated parameters differ from the actual parameters, due to the noise. We also observe that the noise the error is no longer zero. Hopefully you will now be a little comfortable with the Weka interface and the steps involved in performing linear regression on the data set. Up to now we have worked with a very simple synthetic data, so next we will apply regression on the more realistic data set.

(Refer Slide Time: 15:08)

A screenshot of a web browser displaying the UCI Machine Learning Repository. The page has a dark blue header with the UCI logo and the text "Machine Learning Repository". Below the header is a search bar and a "Sort by relevance" dropdown menu. The main content area is a table with two columns: "Dataset Name" and "Description". The table lists various datasets, including "Abalone", "Adult", "Breast Cancer", "Cancer", "Churn", "Concrete", "Cover Type", "Echocardiogram", "Forest Cover Type", "Glass", "Heart Disease", "Iris", "Lung Cancer", "Pima Indians Diabetes", "Sonar", "Spambase", "Soybean", "Soybean Feature", "Vowel", "Wine", "Wine Quality", "Yeast", and "Zoo".

Dataset Name	Description
Abalone	Multivariate
Adult	Classification
Breast Cancer	Classification
Cancer	Classification
Churn	Classification
Concrete	Regression
Cover Type	Classification
Echocardiogram	Classification
Forest Cover Type	Classification
Glass	Classification
Heart Disease	Classification
Iris	Classification
Lung Cancer	Classification
Pima Indians Diabetes	Classification
Sonar	Classification
Spambase	Classification
Soybean	Classification
Soybean Feature	Classification
Vowel	Classification
Wine	Classification
Wine Quality	Classification
Yeast	Classification
Zoo	Classification

The UCI machine learning repository is home to a number of real-world datasets. Here you can see the different data sets characterized by the primary tasks they support the type of data they contain the area from which the data has been generated and so on. We will use the abalone data set for our next experiment.

(Refer Slide Time: 15:36)

The screenshot shows the UCI Machine Learning Repository website. The main header features the UCI logo and the text "Machine Learning Repository". Below the header, the title "Abalone Data Set" is displayed, along with a small thumbnail image of an abalone shell. A table provides basic statistics about the dataset:

Task	Classification	Number of Instances	Number of Attributes	Date
Classification	Regression	4177	9	1998
Identifier	None			

Below the table, there is a section titled "Details" containing links to "Dataset", "Data File", "Attribute Information", "Data Description", "Source", and "Citation". There is also a note about the dataset being part of the "UCI Machine Learning Repository" and a link to "View source".

Here you can see the basic information about the data set such as the associated tasks, the number of instances, the number of attributes whether the dataset contains any missing values and so on. The data description page contains more detailed information.

(Refer Slide Time: 15:56)

This screenshot shows the detailed description of the Abalone dataset. It includes sections for "Dataset", "Attribute Information", "Data Description", "Source", and "Citation".

Dataset:

- Dataset identifier: abalone
- Default Task(s): Classification
- Information: Abalone
- Source: Dua, D. and Graff, C. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Sciences.
- Format: ARFF
- Number of Instances: 4177
- Number of Attributes: 9
- Missing Values? No

Attribute Information:

- Sex: Male, Female
- Age: 0 to 40 years
- Length: 0 to 210 mm
- Diameter: 0 to 36 mm
- Hull Length: 0 to 130 mm
- Hull Width: 0 to 36 mm
- Rings: 0 to 21
- Shucked Weight: 0 to 1500 g
- Whole Weight: 0 to 2100 g

Data Description:

The Abalone dataset is a classification dataset from the UCI Machine Learning Repository. The task is to predict the sex of an abalone based on its physical measurements. The data consists of 4177 samples, each with 9 attributes: Sex, Length, Diameter, Hull Length, Hull Width, Rings, Shucked Weight, Whole Weight, and Age. The Sex attribute is categorical, while the other attributes are continuous. The Rings attribute represents the age of the abalone in years, which is used to determine its sex. The data is balanced, with approximately equal numbers of males and females. The attributes are measured in millimeters and grams, respectively. The data is used for classification tasks, such as predicting the sex of an abalone based on its physical characteristics.

Source:

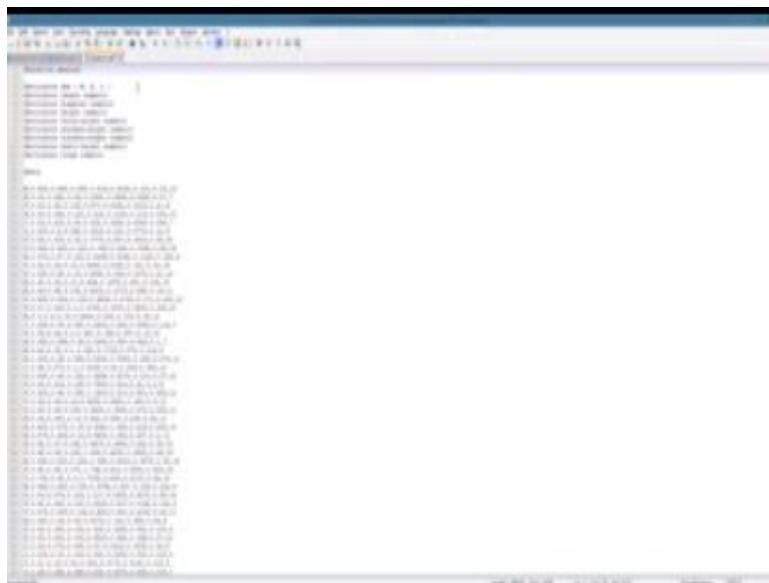
Dua, D., Graff, C. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Sciences.

Citation:

Dua, D., Graff, C. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Sciences.

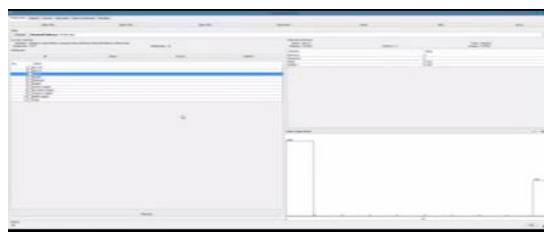
Most importantly lists out the attributes and their data types. This information will be needed for creating the ARFF input format.

(Refer Slide Time: 16:11)

A screenshot of a text editor window displaying a large amount of tabular data. The data consists of several columns of numerical values, likely representing attributes from a dataset. The window has a standard operating system look with a title bar and scroll bars.

This is the raw data and here we have added the information necessary for the ARFF representation. Note, that this data set contains categorical attributes. For such attributes we specify their data type by listing all the possible values they can take. In this case the first attribute can take one of three possible values (M,F or I). Note also that the last attribute is of integer type ranging from 1 to 29. Here we have specified it as a numeric attribute but in case, we wanted to consider this as the class level for a classification task we would have specified it as a categorical attribute. Let us get back to Weka and apply linear regression on this data set.

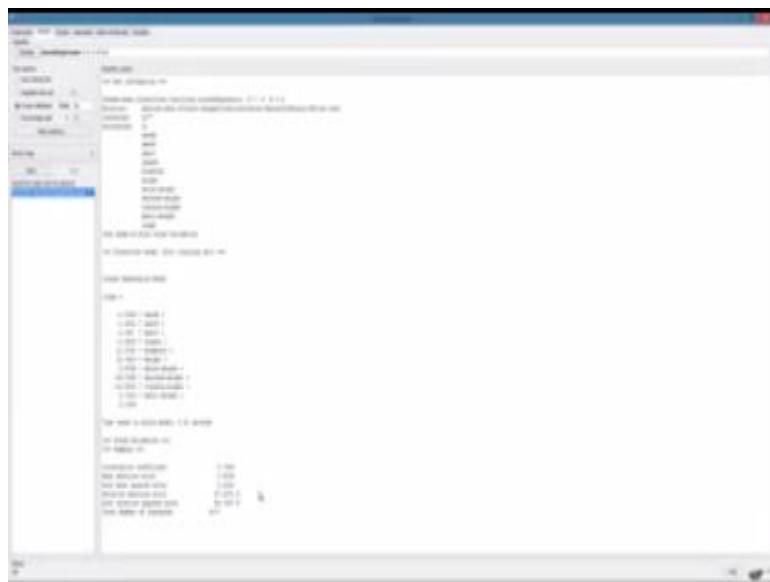
(Refer Slide Time: 17:07)



Here we see the nine attributes along with the associated information on the right. The first thing to do is to handle the categorical attribute. Recall from the lectures we learned about one hot

encoding this can be done in the preprocess tab using an appropriate filter. First of all we select the attribute then choose the appropriate filter. This filter comes under the unsupervised attribute folder and it is the nominal to binary filter. On applying this filter we observe that from one attribute we have now created three attributes. One corresponding to each of the possible values that the original attribute would have taken. Also from the histograms, we can see that each of the attribute is now 0 or 1 or that is binary variable, we can now apply linear regression on this data set.

(Refer Slide Time: 18:09)



We will start by using the same parameter values as used previously. We will stick with cross validation for evaluation and try and predict the Rings attribute. Here we observe the results. These are the estimated β parameters and below we see the error measures. The question here is can we improve this result. One idea is to apply regularization, however before applying regularization, we should normalize the data since ridge regression is not invariant to scale. This can be done in the preprocess tab by selecting the normalization filter. Note that except for the nominal attributes and the output attribute each of the other attributes has been normalized to a range within 0 & 1. Now we can try ridge regression. Let us start with a value of 0.5. However to compare this result we should apply linear regression on the normalized data. So we will run linear regression again with ridge parameter set to zero. Here we have the result of running linear

regression with regularization parameter set to zero on the normalized data. Looking at the β parameters as well as the error values we notice some slight changes. Especially if we look at the β parameters for some of the attributes we see that in the result for rigid regression the parameter values have shrunk this is understandable, since in ridge regression we are adding a penalty term to the error function to shrink the magnitude of the parameters. We can also observe a slight change in the error terms. By trying out different values of the regularization parameter, we can attempt to improve on the performance. However trying out parameters manually is not feasible for this. We will use a meta learning algorithm called CV parameter selection. Essentially CV parameter selection will take as input a learning algorithm, a parameter of that algorithm and a range of values to try out for that parameter. Let us specify this, so first we select the algorithm which is linear regression. We set its parameters. Notice that the regularization parameter is specified with the letter R. So this will allow us to specify the regularization parameter along with the range. So let us say we want to vary the regularization parameter between 0 and let us say 5 in 50 steps. We stick with the same cross-validation evaluation and execute. The result of the execution of the meta learning algorithm shows the optimal value of the regularization parameter as follows subject to the range constraints provided by us.

The β parameters and the corresponding error measures are shown here. Comparing this result with the results of the previous two executions, where the regularization parameter was set to 0 and 0.5, we see that there are small changes but not nothing drastic. This seems to suggest that regulation does not seem to have much effect on this model or perhaps that we have not found the right range of parameters. In case of the latter we can run the meta learning algorithm again and specify a larger range.

One very useful technique when searching for the optimal parameters for any learning algorithm is to start with a large range and the large step size. This initial step performs a coarse-grained search over the range of parameter values. Next we perform a fine-grained search in the vicinity of the value which gave the best results in the previous step. That is we restrict the range but decrease the step size. We can leave it to you to apply these two-stage parameters search on this data set this concludes the tutorial on Weka. We hope that people encountering Weka for the first time will feel a really comfortable with the basic features of the software. In this tutorial we covered most of the concepts which will be required for the first set of programming assignment

questions. In future assignments we will simply mention the algorithms that need to be used and expect you to apply them on the datasets supplied using Welka. We also encourage you to explore the algorithms provided in the package as and when we cover them and related one in class. For this the UCI machine learning repository is a very good source for data sets that can be used for all kinds of learning experiments.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

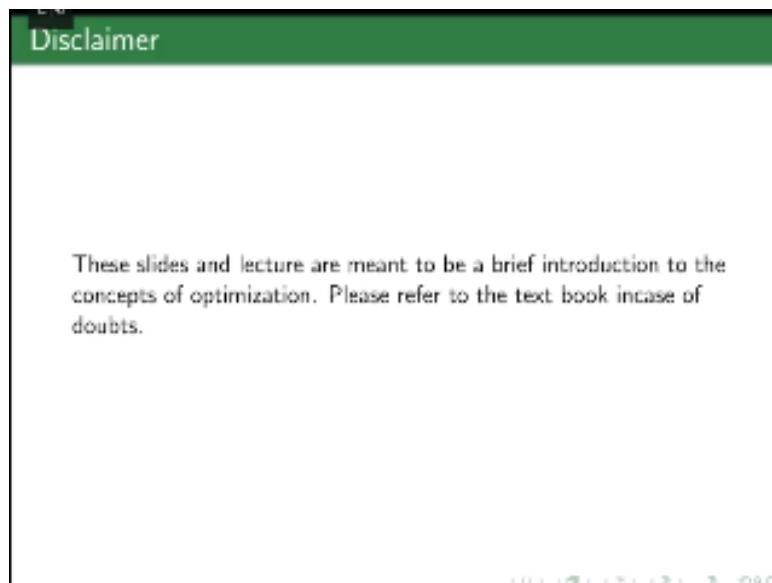
Introduction to Optimization

Abhinav Garlapati

**Introduction to Machine Learning
29th Jan 2016**

Hello everyone. I am Abhinav and in this unit we will be covering the basic concept of optimization, which should be useful in this course.

(Refer Slide Time: 00:26)



So before going in to details a small disclaimer this tutorial is meant to be a small introduction for a complete understanding of these concepts please refer to any standard text book.

(Refer Slide Time: 00:43)

Outline

- ① Introduction
- ② Some Definitions
- ③ Optimization
- ④ Duality
- ⑤ Algorithms

Navigation icons: back, forward, search, etc.

This tutorial is broken in to five chunks first let us start off with the introduction.

(Refer Slide Time: 00:53)

Mathematical Optimization

Definition

Mathematical optimization is the selection of a best element (with regard to some criteria) from some set of available alternatives.

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq b_i \quad i = 1, 2, 3, \dots, m, \end{aligned} \tag{1}$$

where, $x \in \mathbb{R}^n$ known as the optimization variable

$f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ defines the criteria. Also known as the objective function

$f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, 3, \dots, m$ are known as the constraints.

Navigation icons: back, forward, search, etc.

What is mathematical optimization? Mathematical optimization according to Wikipedia is a selection of a best element with regard to some criteria from some set of available alternatives. Now let us look at the mathematical formulation for the same. Here we are trying to minimize

$f_0(x)$ subject to m constraints of the form $f_i(x) \leq b_i$. f_0 is also known as the objective function f_i are the constraints and x is known as the optimization variable.

(Refer Slide Time: 01:37)

Optimal Solution

When do I know any $x \in \mathbb{R}^n$ is the solution for the problem?

- x satisfies all the constraints
- $f_0(x)$ is the minimum possible value in the feasible region.

Such a vector is generally represented by x^* , called as optimal solution.

Navigation icons: back, forward, search, etc.

x is known as the solution for the problem if it satisfies all the constraints and it minimizes $f_0(x)$. Such a solution is known as the optimal solution and it is represented by x^* so through all this tutorial whenever you see x^* it represents the optimal solution for the optimization problem.

(Refer Slide Time: 02:03)

Examples

- Data fitting:
 - **Variables:** Parameters of the model
 - **Constraints:** Parameter limits, prior information.
 - **Objective:** Measure of fit (Eg. Minimizing of error)
- Portfolio Optimization:
 - **Variables:** amounts invested in different assets
 - **Constraints:** budget, max./min. investment per asset, minimum return
 - **Objective:** overall risk or return variance

Navigation icons: back, forward, search, etc.

Now let us look at some examples where optimization is used. First data fitting, data fitting is a very common problem in the field of machine learning. What I mean by data fitting? Data fitting is fitting of a parametric model given some data. So one such example both linear regression in linear regression we are trying to fit a linear model whose parameters are β_i 's, so those translate to the optimization variables here and constraints, constraints in general encode something like parameter limits or prior information which you want to encode. So what is the specific example of linear regression? We do not have any constraints. And what would be the objective? You would try to fit get the best fit for the model. So one way of doing this would be minimizing this error and in linear regression we have seen how see to minimize this squared error.

So that forms the objective of the optimization problem. Another example of application of optimization is portfolio optimization. So by portfolio optimization we mean to optimize the amount of money I can invest in various assets. So these assets could be something like shares from different companies or any other investment options. So the variables would be the amount I invest in all the options available. The constraints would bead it overall budget the maximum or the minimum investment per asset. And the minimum return I expect from each asset, objective would be to minimize overall risk or minimize the return variants. You have seen what optimization problems are and you seen some examples. Now the next big question is how do we solve them.

(Refer Slide Time: 04:08)

Solving Optimization problems

- Optimizations are very tough problems to solve.
- Optimization problems are classified into various classes based on the properties of objectives and constraints.
- Some of these classes can be solved efficiently
 - Linear programs
 - Least Squares problems
 - Convex Optimization problems
- We will study Convex optimization problems, as we come across these problems very regularly.

Optimization problems very difficult problems to solve in general optimization problems are classified in to different types based on the properties of objectives and constraints. Some of the examples are linear programs, least square programs and convex optimization problems. These problems are well studied and can we solved efficiently. Not all class of problems can we solve very efficiently. In this tutorial we will be covering convex optimization problems in detail.

(Refer Slide Time: 04:45)

Targets for this tutorial session

- Convexity
- Properties of Convex functions
- Properties of Convex Optimization problems
- Numerical methods of solving optimization problems.

In this tutorial first we will be looking at convexity. What convexity means and how do we define it? Then we will look at properties of convex functions and then we will look at properties of convex optimization problems. And at the end we have briefly cover some numerical methods for solving optimization problems.

(Refer Slide Time: 05:11)

Convex Set

Definition

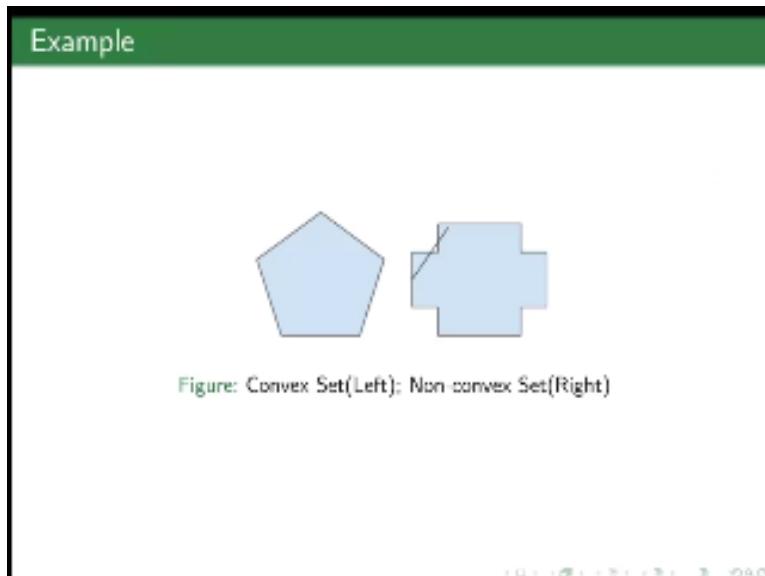
A set C is convex if for all points $a, b \in C$ then the line segment through the points a, b lies in the set C , i.e., $c = \theta a + (1 - \theta)b, c \in C, \forall \theta \in [0, 1]$

Convex Combination

A point of the form $\theta_1x_1 + \theta_2x_2 + \dots + \theta_kx_k$ such that $\sum_{i=1}^k \theta_i = 1$ and $\theta_i \geq 0$ is known as the convex combination of the k points $x_1, x_2, x_3, \dots, x_k$.

A set C is said to be convex if for all points a,b belong to the set the line segment passing through these points should also lie inside the set. So mathematically we can see the sets all the points of the form $\theta a + (1-\theta) b$, when θ lies in $[0,1]$ should also belong to the set C. Next let us look at the definition of convex combination. A point of the form $\theta_1 x_1 + \theta_2 x_2$ so what if $\theta_k x_k$ such that the coefficient sum up to 1 and the coefficients are non-negative is known as the convex combination of this k points.

(Refer Slide Time: 06:04)



Now let us look at examples of convex sets. This pentagon is a convex set because any line joining two points inside the set lies inside the set. Whereas this set is a non-convex set because this line going between two points here passes outside the set. Thus these points do not lie inside the set hence this does not satisfy the definition of convex set right it is not a convex set.

(Refer Slide Time: 06:36)

Convex Function

Definition

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be convex if,

- Domain of f is a convex set
- $\forall x, y \in \text{dom}(f)$, and $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

Let us look at the definition of convex function. A function f is set to be convex if the domain is a convex set and if for all x, y which belong to the domain f the convex combination of these two points is less than or equal to the convex combination of the values at these individual points. So what I mean is $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$. So geometrically you can see that the line joining $(x, f(x))$ and $(y, f(y))$ should lie above the curve. So if this happens we can see that the value $f(\theta x + (1-\theta)y)$ are the points along the curve $\theta f(x) + (1-\theta)f(y)$ are points along the line segment joining $(x, f(x))$ and $(y, f(y))$. So by ensuring that this always above the function we ensure that the inequality holds this making it a convex function.

(Refer Slide Time: 08:03)

Strictly Convex Functions

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y)$$

, when $x \neq y$.

Concave Function

A function f is said to be concave if $-f$ is convex.

Strictly Concave Function

A function f is said to be strictly concave if $-f$ is strictly convex.

Now let us define what a strictly convex functions is. In strictly convex functions the inequality becomes a strong inequality that is $f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y)$. And now let us define what is concave function is a function f is said to be concave if, $-f$ is convex and then similarly we define a strictly concave function a function f is said to be strictly concave function if, $-f$ is strictly convex.

(Refer Slide Time: 08:42)

Examples

- $f(x) = x^2$

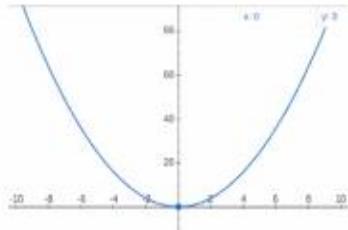


Figure: $y = x^2$

Now let us look some examples. First $f(x) = x^2$ is a convex function from the graph it is clearly evident that any line joining two points this will lie above the curve between these two points. This claim can also be verified by using the definition.

(Refer Slide Time: 09:07)

Examples

- $f(x) = e^x$

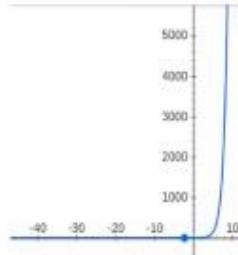


Figure: $y = e^x$

The next example that we see is graph of $f(x) = e^x$. Again graphically you can clearly see that this is the convex function. If you try to prove this according to the definition you can see that this is not trivial. So we would like to see if any other ways to check the convexity of function.

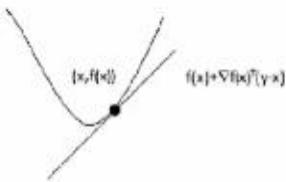
(Refer Slide Time: 09:36)

Conditions for Convexity

First Order Condition

Let f be differentiable, i.e., ∇f exists for each x in $\text{dom}(f)$. f is convex if and only if:

- $\text{dom}(f)$ is convex.
- $f(y) \geq f(x) + \nabla f(x)^T(y - x), \forall x, y \in \text{dom}(f)$



Navigation icons: back, forward, search, etc.

Let us look at the first order condition for the convexity. Let f be a differentiable function that is ∇f exists for all x in a domain of f . So a function f is convex if and only if the domain of f is convex and this inequality satisfies. This inequality states that function should always lie above all its tangents. If you look at the right hand side carefully it is nothing but the equation of the tangent at $(x, f(x))$ and we expect this value to be less than $f(y)$ this is nothing but the condition saying that the curve should be above the tangent.

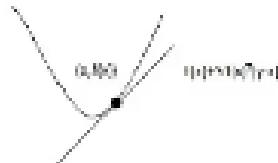
(Refer Slide Time: 10:24)

Conditions for Convexity

First Order Condition

Let f be differentiable, i.e., ∇f exists for each x in $\text{dom}(f)$. f is convex if and only if:

- $\text{dom}(f)$ is convex.
- $f(y) \geq f(x) + \nabla f(x)^T(y - x), \forall x, y \in \text{dom}(f)$



Navigation icons: back, forward, search, etc.

Now let us look at the second order condition for convexity. Let f be twice differentiable function and the f will be convex if and only if domain of the f is convex and its hessian is positive semidefinite. So if you look at the second example of the e^x the second derivative is always positive hence it can be proved that it is a convex function

(Refer Slide Time: 11:01)

Epigraph

Epigraph

$$\text{epi } f = \{(x, t) | x \in \text{dom}(f), t \geq f(x)\}$$

f is a convex function $\Leftrightarrow \text{epi } f$ is convex set.

The graph illustrates the epigraph of a convex function f . The horizontal axis is labeled x and the vertical axis is labeled $f(x)$. A wavy curve represents the function f . The region above and to the right of this curve is shaded gray and labeled "epi f ".

Now we defined a epigraph of function. For a given function f , the epigraph of f , is defined as the set of all pairs (x,t) such that x belong to the domain of f and t is greater than or equal to $f(x)$. So if you look at the graph you can see that the area above the curve is belongs to the epigraph of the function. One important property to know is for a convex function, the epigraph is always a convex set and the converse also holds, that is if for a function the epigraph is a convex set then the function is convex.

So we till now we have seen three ways of checking for convexity of a function. First you can do the first order test or the second order set or you can check for convexity of the epigraph of the function.

(Refer Slide Time: 12:07)

Sublevel sets

Sublevel sets

The (α -) sublevel set of f is

$$C(\alpha) \triangleq \{x \in \text{dom}(f) | f(x) \leq \alpha\}$$

f convex \implies sublevel sets are convex

Converse is not true.

Now let us look at what is sublevel sets of function are. An alpha sub level set of a function f is set of all pints x which belong to the domain of f such that the value of the function at these points is $\leq \alpha$. There is one important property that if the function is convex, the sublevel set sets of the function are also convex it is important to note the converse is not true.

(Refer Slide Time: 12:45)

Properties

- Convexity Preserving Operations

- Non-negative Weighted Sum

$\sum \alpha_i f_i$ is convex if $\alpha_i \geq 0$

- Composition with Affine Function

$f(Ax + b)$ is also convex if f is convex.

- Pointwise Maximum and Supremum

$$f_1, f_2 \text{ convex} \implies \max\{f_1(x), f_2(x)\} \text{ convex}$$

- Minimization

If $f(x, y)$ is convex in (x, y) and C is a convex set, then
 $g(x) = \inf_{y \in C} f(x, y)$ is convex

- Local minima is the global minima.

Now let us look at some other properties of convex functions. First we will look at the operations which preserve the convexity of the function. First non-negative weighted sum. That is a non

negative weighted sum of various convex functions which still remain a convex function. Consider f_i is the series of convex functions $\sum \alpha_i f_i$ where α_i 's are greater than 0, will also remain a convex function. Next, composition with affine function. Affine function is a linear transformation of x so $ax + b$ is an affine function. If f is convex then $f(ax + b)$ is also convex. Point wise maximum and supremum of two convex functions will also remain convex. Minimization if you look at as two variable function $f(x,y)$ which is convex, then if you try to minimize the function allowing any one variable in a convex set, the resultant function is also convex function. The most important property of convex functions is that the local minima is also the global minima. It is a very powerful result which can be proved easily. This result guarantees that the minima option while searching for the minimum of a convex function is the optimal solution. Another important property of convex function is that they satisfy the Jensen's inequality. This is a generalization of the inequality which we have seen in the definition of the convex function. Here instead of two points, we have n points. So the value of the convex combinations of n points is less than or equal to the convex combination of the values each of the function at each of the individual points. A colloquial way of saying this is that the value of the average is less than the average of the values. Here by the average I mean a weighted average.

(Refer Slide Time: 15:05)

Now let us look at a general optimization problem. Any optimization problem in generally can be reduced to this form. Of minimize an objective function subject to few inequality constraints and few equality constraints. So the optimal value p^* can also be written as

(Refer to slide time 15.36)

The optimal value p^* is given by

$$\inf\{f_0(x) | f_i(x) \leq 0, i = 1, 2, \dots, m \text{ and } h_i(x) = 0, i = 1, 2, 3, \dots, p\}$$

Now the next question is why did we use infimum instead of minimum? In some function the minimum might not be attainable; it might just tend to a minimum value but not actually attain it. Hence we write infimum instead of minimum.

(Refer Slide Time: 16:01)

Consider a general optimization problem,

$$\begin{aligned} & \min f_0(x) \\ \text{s.t. } & f_i(x) \leq 0 \quad i = 1, 2, 3, \dots, m, \\ & h_i(x) = 0 \quad i = 1, 2, 3, \dots, p. \end{aligned} \quad (2)$$

An optimization problem with satisfies the given three condition is not as a convex optimization problem. So first f_0 the objective function should be convex then the inequality constraints f_i should also be convex. And the equality constraints should be affine. When I say affine it should be at the form $a_i^T x = b_i$. So one can observe that the domain has become a convex set right now. So these inequality constraints represent a sublevel set of a convex function so it is a convex set. And a convex set intersection with affine function is a convex set. So why a convex

problems are so interesting. So convex representation problems are interesting because with the properties of a convex functions and the convex set. So first the most important property which is useful for us is that if there is local minima anywhere, it is guarantee that is the global minimum for the function. So it makes are like very simple and we do not have search a lot for global minimum.

(Refer Slide Time: 17:13)

The slide has a green header bar with the title "Duality". Below the header, there is a text area containing the following content:

Every problem can be seen in two perspectives, the *primal form* and *dual form*. Solving and understanding the dual helps us understand the behaviour of the primal form. Consider the standard form,

$$\begin{aligned} & \min f_0(x) \\ \text{s.t. } & f_i(x) \leq 0 \quad i = 1, 2, 3, \dots, m, \\ & a_i^T x = b_i \quad i = 1, 2, 3, \dots, p. \end{aligned} \quad (4)$$

Let \mathbb{D} denote the domain of the problem. This problem is called the *primal problem*, and its optimal value is denoted by p^* obtained at x^* .

Every optimization problem can be seen in two perspectives. One the primal form and the dual form, so whatever we seen till now is generally known as the primal form. And we will now develop the dual form. So why do we need another view of the problem? So sometimes the primal form might be very difficult to solve it. So the dual form might be easier to solve and also cases some understanding on how the solution of the primal form may be. So before going ahead let me just recap the notation which we going to use. So this is the standard optimal convex optimization problem and when I said p^* it denotes that the optimal value of this problem and the value of p^* is attained at x^* which is the solution of a solution. Now let us consider the alternative relaxed problem. Instead of minimizing the f_0 will may the weighted some of the objective functions and the constraints.

(Refer Slide Time: 18:19)

Let us consider an alternative relaxed problem,

$$\begin{array}{ll} \min & f_0(x) + \sum \lambda_i f_i(x) + \sum \nu_i h_i(x) \\ \text{s.t.} & \lambda_i \geq 0 \\ & x \in D \end{array} \quad i = 1, 2, 3, \dots, m \quad (5)$$

$$L(x, \lambda, \nu) = \{f_0(x) + \sum_{i=0}^m \lambda_i f_i(x) + \sum_{i=0}^p \nu_i h_i(x)\}$$

$$\inf_x \{f_0(x) + \sum_{i=0}^m \lambda_i f_i(x) + \sum_{i=0}^p \nu_i h_i(x)\} \leq L(x*, \lambda, \nu) \leq p^*$$

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$$

So we will be minimizing

(Refer to slide time 18.29)

$$\begin{array}{ll} \min & f_0(x) + \sum \lambda_i f_i(x) + \sum \nu_i h_i(x) \\ \text{s.t.} & \lambda_i \geq 0 \\ & x \in D \end{array} \quad i = 1, 2, 3, \dots, m \quad (5)$$

Here we also have an addition constraint that λ should be greater than or equal to 0. And as usual x should belong to the domain we call the object of this optimization of this optimization problem as the Lagrangian. So $L(x, \lambda, \nu)$ is defined

(Refer to slide time 18.45)

$$L(x, \lambda, \nu) = (f_0(x) + \sum_{i=0}^m \lambda_i f_i(x) + \sum_{i=0}^p \nu_i h_i(x))$$

(Refer to slide time 18.58)

$$\inf_x (f_0(x) + \sum_{i=0}^m \lambda_i f_i(x) + \sum_{i=0}^p \nu_i h_i(x)) \leq L(x^*, \lambda, \nu) \leq p^*$$

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$$

Infimum of the Lagrangian over x is less than equal to p^* this can be seen very easily. But thing to be noted as in equality is valid only when x is feasible. So now we define g as a function of λ and ν as the infimum of the Lagrangian over x .

(Refer Slide Time: 19:14)

The slide has a green header bar with navigation links: Introduction, Save Definition, Optimization, **Duality**, and Algorithms.

Lagrangian Dual problem

$$g(\lambda, \nu) \leq p^*$$

g forms a lower bound on the optimal value of the primal problem.

Dual problem

$$\begin{aligned} & \max \quad g(\lambda, \nu) \\ & \text{s.t. } \lambda_i \geq 0 \quad i = 1, 2, 3, \dots, m. \end{aligned}$$

The optimal value of this problem is attained at λ^*, ν^* . We can see that the dual is concave irrespective of the form of the primal problem and can be solved. The optimal solution of the dual problem is denoted by d^* .

$p^* - d^*$ is known as the duality gap.

So we have seen of a function g which cases lower bound of the optimal value of the primal problem. So if you try to maximize the function g we will achieve a very good lower bound of the optimal value. So this is what is may known as the dual problem. So maximizing $g(\lambda, \nu)$ such that $\lambda_i \geq 0$.

(Refer to slide time 19.42)

$$g(\lambda, \nu) \leq p^*$$

g forms a lower bound on the optimal value of the primal problem.

Dual problem

$$\begin{aligned} \max \quad & g(\lambda, \nu) \\ \text{s.t.} \quad & \lambda_i \geq 0 \quad i = 1, 2, 3, \dots, m. \end{aligned}$$

The optimal value of this problem is attained at λ^*, ν^* .

We can see that the dual is concave irrespective of the form of the primal problem and can be solved. The optimal solution of the dual problem is denoted by d^* .

The optimal value of the problem is λ^* and ν^* , we can see that this function g is concave irregardless of the form of the primal problem. So if you go back and see we started with the general form of primal problem and we achieve when reached with g which is concave. So g can always be solved the optimum value of the dual problem is denoted by d^* so now we would like to see how far is this d^* from the actual value p^* . So $p^* - d^*$ is known duality gap.

(Refer Slide Time: 20:25)

Introduction Some Definitions Optimization **Duality** Algorithms

Strong and Weak Duality

If the duality gap is 0, then it is known as Strong Duality.
The primal problem can also be written as

$$p^* = \inf_x \sup_{\lambda \geq 0, \nu} L(x, \lambda, \nu)$$

The dual problem can be written as

$$d^* = \sup_{\lambda \geq 0, \nu} \inf_x L(x, \lambda, \nu)$$

If strong duality holds, we can see that the order of inf and sup don't matter. The optimal variables occur at a saddle point of the Lagrangian.

Navigation icons: back, forward, search, etc.

The next obvious question is to find out when this $p^* - d^*$ will be 0 and when it is not so whenever it is 0 it is known as the strong duality and when it is not it is known as the weak duality. So next we will try to further characterize when what can occur. So first decide we can see that p^* can be written as

The primal problem can also be written as

$$p^* = \inf_x \sup_{\lambda \geq 0, \nu} L(x, \lambda, \nu)$$

d^* can be written as

The dual problem can be written as

$$d^* = \sup_{\lambda \geq 0, \nu} \inf_x L(x, \lambda, \nu)$$

So when this strong duality holds we know that $p^* = d^*$. So you can see that the order of the infimum supremum can be interchanged and it is equivalent. So this means that at the same point we have maxima in one direction and the minima in another direction. So it is a saddle point, so we have one good result here, that is whenever that is strong duality optimal variables occur at the saddle point of the Lagrangian.

(Refer Slide Time: 21:34)

Sufficiency condition for Strong Duality

In a convex optimization problem, slater's condition implies that if $\exists x \in \text{reint}D$ such that $f_i(x) < 0$ and $h_i(x) = 0$ then strong duality holds.

In other words, slater condition states that, strong duality holds if there exists a point x in the interior of feasible region of the problem.

Now let us look at sufficiency conditions for strong duality. So we look at slater's conditions which gives us conditions for a convex optimization problem to be strongly dual. So Slater's conditions states that for a convex optimization problem, there exists an x , such that it belongs to the relative interior of the domain such that $f_i(x) < 0$ and $h_i(x) = 0$, then strong duality holds. So here we require the inequality constrains to be strongly strictly unequal and the function should be the point should belong to the relative interior and not the boundary.

So Slater's conditions state that for any convex optimization problem if that exits a point inside the feasible region then strong duality surely holds. So note that this is only for convex optimization problems and not a general result.

(Refer Slide Time: 10:22)

Sufficiency condition for Strong Duality

In a convex optimization problem, slater's condition implies that if $\exists x \in \text{reint}(\Omega)$ such that $f_i(x) < 0$ and $h_i(x) = 0$ then strong duality holds.

In other words, slater condition states that, strong duality holds if there exists a point x in the interior of feasible region of the problem.

So now we look at complementary slackness. Assume strong duality holds and x^* is the primal variable and λ^* also dual variables. So when I say strong duality holds, we know that
(Refer to slide time 22.56)

$$\begin{aligned}
f_0(x^*) = g(\lambda^*, \nu^*) &= \inf_x (f_0(x) + \sum_{i=1}^{i=m} \lambda_i^* f_i(x) + \sum_{i=1}^{i=p} \nu_i^* h_i(x)) \\
&\leq f_0(x^*) + \sum_{i=1}^{i=m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{i=p} \nu_i^* h_i(x^*) \\
&\leq f_0(x^*)
\end{aligned}$$

So by expanding g by this definition and looking at some simple inequities, we can reach to a conclusion that for all i , $\lambda_i^* f_i(x^*)$ should be 0. Okay so basically we know that $f_i(x^*)$ is ≤ 0 because x^* is a feasible value. So whenever $f_i(x^*)$ is not equal to 0 we know that $\lambda_i = 0$. So this is known as complementary slackness. That is either $\lambda x^* = 0$ or $f_i(x^*) = 0$ then strong duality holds.
(Refer Slide Time: 23:46)

Now we will look at Karush Kuhn Tucker conditions, also known as KKT conditions. So these provide us the necessary conditions for a point $x^* \lambda^* \nu^*$ to be optimal. So consider any point $x^* \lambda^* \nu^*$ if it has to be a optimization these things have to be satisfied. So first stationarity so since you already seen that at the optimal point 1 has a saddle point so the gradient at that point should

be 0 so that is trivial to c and then primal feasibility and dual feasibility should hold and then you have seen complementary slackness as you seen previously we also be valid at this point.

(Refer Slide Time: 24:39)

The slide title is "KKT conditions". The content is as follows:

- If x, λ, ν satisfy strong duality then KKT conditions hold.
They are necessary conditions for a solution to be optimal.
- For a problem where Slater's conditions are satisfied, KKT conditions become sufficient too.

So just to reiterate what you already seen if x, λ, ν satisfy strong duality then KKT conditions hold. So these are just necessary conditions and sufficient but for our optimization problems when Slater's conditions are satisfied then KKT became sufficient also.

(Refer Slide Time: 25:03)

The slide title is "KKT conditions". The content is as follows:

- If x, λ, ν satisfy strong duality then KKT conditions hold.
They are necessary conditions for a solution to be optimal.
- For a problem where Slater's conditions are satisfied, KKT conditions become sufficient too.

Now we look at some examples first is the most popular example of least squares, so we are trying to minimize a least square function so we are trying to minimize a least square function so we are trying to minimize this $\min \|Ax - b\|_2^2$ with no constraints so we can clearly say that this a convex function and there is no constraints and we solved in this thing in while solving linear regression to give x^* as $(A^T A)^{-1} A^T b$. So this is a very trivial convex optimization problem which you are able to solve but just by differentiating.

(Refer Slide Time: 25:46)

The slide has a navigation bar at the top with tabs: Introduction, Some Definitions, Optimization, Convexity, and Algorithms. The main content area has a green header bar with the title "Example 2". Below the header, the problem is stated:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ \text{s.t.} \quad & (x_1 - 1)^2 + (x_2 - 1)^2 \leq 1 \\ & (x_1 - 1)^2 + (x_2 + 1)^2 \leq 1 \end{aligned}$$

where $x \in \mathbb{R}^2$.

- We can see analytically that the each of the constraints define circular regions with centers at $(1, 1)$ and $(1, -1)$ of radius 1. There is only one point in common which is $(1, 0)$. $p^* = 1$.
- Lagrangian,

$$L(\bar{x}, \bar{\lambda}) = x_1^2 + x_2^2 + \lambda_1((x_1 - 1)^2 + (x_2 - 1)^2 - 1) + \lambda_2((x_1 - 1)^2 + (x_2 + 1)^2 - 1)$$

Navigation icons: back, forward, search, etc.

Now let us look at another example, so here we are trying to minimize $x_1^2 + x_2^2$ subject to these two linear quadratic constraints. So you look at these constraints carefully both of them are circular regions one centered at $(1, 1)$ and the other centered at $(1, -1)$. Each of which is radius at one. So if you just plot them and see that you can see that there is only one feasible point that is $(1, 0)$ so trivially the optimal value will become one. But now let us do analysis which we have learnt and how to do and then try to arrive at the same answer, so first when you have a convex of machine value or for that matter optimization problem, the first thing you do is write that like Lagrangian so here the lagrngian will be

(Refer to slide 26.39)

- Lagrangian,

$$L(\bar{x}, \bar{\lambda}) = x_1^2 + x_2^2 + \lambda_1((x_1 - 1)^2 + (x_2 - 1)^2 - 1) + \lambda_2((x_1 - 1)^2 + (x_2 + 1)^2 - 1)$$

(Refer Slide Time: 26:55)

The slide has a green header bar with the title "Example 2 (Contd..)". Below the header, there is a bulleted list of KKT conditions:

- Let us now list the KKT conditions,

Below the list are six equations:

$$\begin{aligned} (x_1 - 1)^2 + (x_2 - 1)^2 &\leq 1 \\ (x_1 - 1)^2 + (x_2 + 1)^2 &\leq 1 \\ \lambda_1 &\geq 0 \\ \lambda_2 &\geq 0 \\ 2x_1 + 2\lambda_1(x_1 - 1) + 2\lambda_2(x_1 - 1) &= 0 \\ 2x_2 + 2\lambda_1(x_2 - 1) + 2\lambda_2(x_2 + 1) &= 0 \\ \lambda_1[(x_1 - 1)^2 + (x_2 - 1)^2 - 1] &= 0 \\ \lambda_2[(x_1 - 1)^2 + (x_2 + 1)^2 - 1] &= 0 \end{aligned}$$

At the bottom of the list, there is another bullet point:

- At (1, 0) the equations are not valid.

So now that you have seen the lagrangian let us try to list out the KKT conditions. So here the first two are the primary feasibility conditions second two are the dual feasibility conditions and the next two are obtained by differentiating the lagrangian with x_1 and x_2 respectively and the next two are obtained by writing the complimentary slackness equations. And we have seen that there is only one feasible point (1, 0) and at that point these conditions are not valid. You get contradictory answers for λ_1 and λ_2 and you try to solve. See that is KKT conditions are not valid. But this is tricky so we have already seen that we have an optimal value but KKT conditions are not satisfied. We will try to see why this is happening here. Now let us try to investigate what exactly is happening so we will try to solve that your problem now.

(Refer Slide Time: 28:10)

The slide has a navigation bar at the top with links to 'Introduction', 'Some Definitions', 'Optimization', 'Duality', and 'Algorithms'. The main content area is titled 'Example 2 (Contd..)'. It contains the following text and equations:

- Taking derivatives of L with respect to x_1, x_2 gives the following equations.
$$x_1 = \frac{\lambda_1 + \lambda_2}{1 + \lambda_1 + \lambda_2}$$
$$x_2 = \frac{\lambda_1 - \lambda_2}{1 + \lambda_1 + \lambda_2}$$
- Substituting them in the Lagrangian we get
$$g(\lambda_1, \lambda_2) = \frac{\lambda_1 + \lambda_2 + (\lambda_1 - \lambda_2)^2}{1 + \lambda_1 + \lambda_2}$$

We see it is symmetric and substitute $\lambda_1 = \lambda_2$ to get,

$$g(\lambda_1, \lambda_2) = \frac{2\lambda_1}{2\lambda_1 + 1}$$

- $g(\lambda_1, \lambda_2) \rightarrow 1$ as $\lambda_1 \rightarrow \infty$. $p^* = d^* = 1$. KKT not satisfied, Slater's condition not satisfied.

So for solving the dual problem we have find the maxima for g . So first let us find out what the function g is. g is in few form of the lagrangian over x . So we will substitute we will try to take the derivative of l with respect to x solve it and then if we arrive at this g function which is the function of λ_1 and λ_2 now you can see that this is a concave function which is symmetric λ_1 and λ_2 so we can substitute this $\lambda_1 = \lambda_2 = \lambda_1$ and the go ahead. So when we do that we get this $2\lambda_1 / (2\lambda_1 + 1)$ as that g function. So if you see that under the limit λ_1 tending into ∞ , g tends to 1 but otherwise there is no maxima achieved. So under a simple conditions $p^* = d^* = 1$ and because this is these points are not been attained at point KKT conditions and Slater's conditions are not satisfied. So this example is just to show you that just solving KKT conditions or checking first latest condition is not sufficient we might have to solve sometimes the dual problem and see what exactly is happening.

(Refer Slide Time: 29:28)

There exists standard algorithms which can be used to solve optimization problems once in standard form. Some of them are

- Simplex Method for Linear Programs
- Interior Point methods

Optimization under no constraints

Various methods exist to solve this class of problems

- Gradient based methods
- Genetic Algorithms
- Simulated Annealing

So we have seen the mathematical characterization for optimization problems, now we will try to see how to solve them. So there exists very many standard algorithms to solve optimization problems once you taken them to standard form. So for linear programs there is this well known simplex method and the most popular methods for solving general optimization problems right now are interior point methods. We will not be covering in these methods in detail at all will be looking at simpler class problems that is, optimization under no constraints. So that is given an objective function under no constraints, how can we solve this? We will look at algorithms, so to do this there exist a lot of algorithms; gradient based methods, genetic algorithms and simulated annealing. First we will look at gradient based methods which are very popular used in machine learning.

(Refer Slide Time: 30:35)

Consider a convex, twice differentiable function f then

$$\min f(x)$$

Assume, the minimum ρ^* is finite and is attained by f . These algorithms produce a sequence of points x_i starting from a given point, such that

$$f(x_k) \rightarrow \rho^*$$

These algorithms require that the sublevel set at x_0 be closed.

So first let us look at proper mathematical definition of unconstrained minimization. Consider a convex which is twice differentiable function is and we want to find minimum of this function. So assume p^* is minimum and it is finite and it is attained by f , so we want algorithms, start from some point and give a series of x_i 's, such that value of $f(x_k)$ tends to this optimal minimum. So these algorithms required one condition that is the sublevel set should be closed.

So what exactly this condition means is, so when I start from x_0 and I go to some other point is which is $<$ then so basically each time I am trying to reduce the value of f . So x_1, x_0 to x_1 where $f(x_1) < f(x_0)$. So that is, this belongs to the subset of $f(x_0)$ and this point x_1 should be inside the set. So we just need this condition, so that we get a chain of points, which are in the domain of the function.

(Refer Slide Time: 32:03)



So now we will look at what are the most popular algorithms gradient descent. So this works in convex problems where there exist in minimum and you start from one point from the top and go down according to the gradient, so if you see this visualization gives you the 3-dimensional surface, which is basically $f(x_1, x_2)$ say. So if we start of $f(x_1, x_2)$ at the top point and we take the gradient there and move along the negative direction slowly.

As we keep going down we reach the bottom of this, so and the bottom is where the minima exist, at the last point the gradient become 0. So this is the motivation for gradient descent

algorithms that is by going along the negative direction of the gradient, we reach the minima in convex functions.

(Refer Slide Time: 33:09)

The slide is titled "Gradient Descent". It has a navigation bar at the top with tabs: Introduction, Some Definitions, Optimisation, Duality, and Algorithms. The "Algorithms" tab is highlighted. The main content area has a green header "Gradient Descent". Below it, the text says "Move in the opposite direction of the gradient." followed by the formula $\Delta x = -\nabla f(x)$. A horizontal line separates this from the algorithm. The algorithm is titled "Algorithm 1 Gradient Descent" and consists of the following steps:

- 1: Given x_0 in $\text{dom}(f)$
- 2: repeat
- 3: $\Delta x = -\nabla f(x)$
- 4: Update $x = x + t\Delta x$
- 5: until stopping criteria is satisfied

Another horizontal line separates this from the final question. The question asks "How do we choose t ? Is t constant?". At the bottom right of the slide are standard presentation control icons.

So let us formally look at the gradient descent, we intuitively seen that if you move in the direction opposite to the gradient we will reach the minima, so will state that as an algorithm,. So if we start x_0 in the domain of f , you can update in every iteration x as $x = x + t\Delta x$. so essentially what we are doing is, we are moving along the negative direction of the function in some step size of t , basically this t is the multiplicative factor, which will magnify or minimize step size that you are taking in the direction.

So the next question is how do we choose t ? Should t be constant? So in the ideal case t should be depended on the curvature of the functions? So if you look at the graph in the previous slide carefully, so where ever there is low curvature you could afford to take larger steps, where ever there is high curvature at the bottom especially where there are minima, you should take small steps, so you do not jump over the minima value.

Methods which choose t according this are out of the scope of this tutorial, so but we will just answer this question, is t constant is enough for us. In most conditions a small t if you take a

small enough step size it is fine and you will very reasonably very close to the minima. So in practice the constant t works. So we will end this tutorial session with this. So the main take home message from this tutorial session should be, what optimization problems are? What is the generic form? What are convex optimization problems? What is duality? What is strong duality? What is KKT and Slater's conditions?

So knowing these will be enough for you to navigate, whatever the optimization that come across this course but ideally we can look up other resource online if you are not clear with these basics still.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

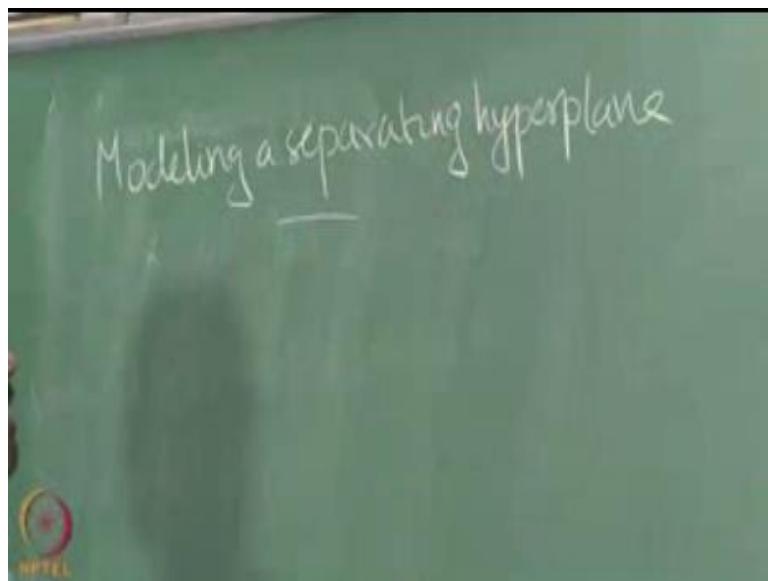
Lecture 26

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

**Separating Hyperplane Approaches
Perceptron Learning**

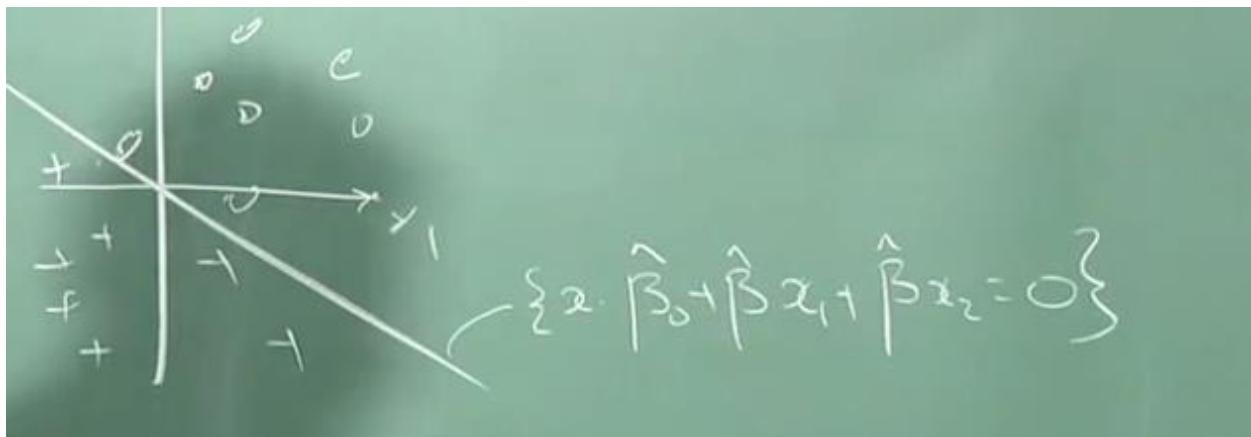
So I said there are two ways in which we can build linear classifiers. So one was the discriminant function approach, where by comparing discriminant functions we decide what is the separating hyperplane and what is other approach? Modeling the separating hyperplane directly.

(Refer Slide Time: 00:38)



So there are many ways in which this can be done and we look at two in particular one because it leads us into neural networks later and the other because it is the most popular way of building classifiers nowadays.

(Refer Slide Time: 02.11)



Before we go on let us just have a little fundamentals again. So that is a separating hyperplane so this constitutes of all points x such that this equation is satisfied okay so this is the definition of a separating hyperplane. So I am going to call that defines our hyperplane if we equate $f(x)$ to 0 that defines our hyperplane and if $f(x)$ is whatever if it is greater than 0 it implies in this case $g(x)$ is +.

(Refer to slide time: 02.42)

$$f(x) < 0 \Rightarrow g(x) = +$$

$$f(x) = \beta_0 + \beta^T x \quad (= 0)$$

$\{x_0 \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 = 0\}$

So some properties are listed here. If x_1 and x_2 both belong to the line and I am going to call this L here if x_1 and x_2 both belong to L then we know that $\beta^T(x_1 - x_2)$ will be 0. So what does it mean? The β is actually a normal to L but actually let me rephrase this: β^* is normal to L , β is perpendicular to L . β^* is a normal is unit direction.

(Refer Slide Time: 04:00)

$$\text{(i)} \quad \beta^*(x_1 - x_2) = 0$$
$$\Rightarrow \beta^* \cdot \frac{\beta}{\|\beta\|}$$
$$\text{(ii)} \quad \beta^T x_0 = -\beta_0$$

This also we know so $\beta^T x_0$ equal to $-\beta_0$, so I mean that essentially gives you this is $1 - \beta_0$ as $+\beta_0$ so it will be 0. This is β remember the β_0 is left out here okay so $\beta^T x_1$ will be $-\beta_0$ and $\beta^T x_2$ will be $-\beta_0$ again so this expression will become 0 that is why x_1 and x_2 belong to L . If people have not thought about that. So what is that this $\beta^{*T} (x - x_0)$ where x_0 belongs to L , x does not belong to L . β_0^* is a direction that is perpendicular to L , so this is essentially the distance of x from L . There actually the signed distance of x from L depending on which side of L it is this is going to be different. But because I am multiplying with β^* it gives me the projection on the perpendicular, so this gives me the distance to some x and this will be the direction of β^* . So this is essentially projecting that on to this direction so this will give me a β^* . We know is beta by norm β , so I replace that so I get $\beta^T x + \beta_0$ because x_0 times β is $-\beta_0$ so I get $+\beta_0$ so and this expression is actually $f(x)$. So this is $f(x)$ divided by norm β and β is what? $f'(x)$ if it take the derivative of $f(x)$ with respect to x that gives me β right.

(Refer to slide time 07.10)

$$(1) \beta^T(x_1 - x_2) = 0$$

$$\Rightarrow \beta^T - \frac{\beta}{\|\beta\|}$$

$$(2) \beta^T x_0 = -\beta_0 \quad \forall x_0 \in L$$

$$\beta^T(x - x_0) = \frac{1}{\|\beta\|} (\beta^T x + \beta_0) = \frac{1}{\|f(x)\|} f(x)$$

NPTEL

So it is norm of $f'(x)$ right so this is $f'(x)$ divided by the norm of $f'(x)$ so what did we say this expression was the signed distance to the hyperplane so $f'(x)$ gives me a quantity that is proportional to the signed distance to the hyperplane. And if I find a hyperplane such that I normalize my β to be one $f'(x)$ gives me the sign distance to the hyperplane it is not proportional it actually gives me the sign distance to the hyperplane. Because whenever I say I am going to minimize $f'(x)$ it essentially means I am going to try and minimize the distance of the data point to the hyperplane all right I am going to say were maximizing $f'(x)$ I am going to maximize the distance of the data point to the hyperplane, so just to keep that clear I am just introducing these notations beginning.

So the first thing we look at it is a perceptron learning algorithm so it is got a hoary tradition. People are familiar with have heard the term neural networks artificial neural networks. So you know that we are going through the third boom of artificial neural networks, so in the back in the 50s and 60s there was this initial boom of artificial neural networks right everybody was so taken

up with artificial neural network they said oh! here is something that can solve the human learning problem.

And we have talked about this already right, so that was started by the perceptron learning algorithm and what about the second boom? People know what the second boom was? Okay now we will come to that later, so the second boom was started by the back propagation algorithm which took care of some of the problems with perceptrons and then that also died away and the third boom has been shattered by what people call deep learning now and the hype is scary but apparently not as scary as it was I mean I recently read some newspaper clippings from the 60s it is really scary.

So the idea behind the perceptron learning algorithm is very simple. So I have this decision boundary and I want to make sure that any point that I actually misclassify at any point that I misclassify is as close as possible to the decision boundary. In fact if I put it on the other side I am happy. I am doing the signed distances, so if I put it on the right side I am happy put it on the wrong side and I try to keep it as close to the hyperplane as possible so that early there is some kind of a satisfaction that we are not getting something very egregiously. So I am going to minimize, so we will assume that for data points which is such that $x_i \beta + \beta_0$ is greater than 0 and output as class plus 1 right, whenever it is less than 0, I will output the class as -1.

(Refer Slide Time: 14.23)

Perceptron Learning Algo

Min distance of misclassified points to decision boundary

$$y_i = 1 \text{ is misclassified} \Rightarrow x_i^T \beta + \beta_0 < 0$$
$$y_i = -1 \text{ is } " \text{ --- } \Rightarrow x_i^T \beta + \beta_0 > 0$$

So if the true class level is + 1 but if the $f(x)$ is < 0 , I mean if $f(x) < 0$ that means I will be outputting minus 1 right but the true class is + 1 so that means this has been misclassified right.

Does that make sense? People have understand this point now. So if the true class was +1 but $f(x)$ is < 0 then x will get misclassified the true class was -1 and $f(x) > 0$ then x will get misclassified, so these are the two conditions under which you will have a point being misclassified. So if I take this quantity right so what it will be for misclassified points it will be negative for misclassified points, it will be positive for correctly classified points right take that and I minimize it. So take this for all the misclassified points and I try to minimize this, so D is sometimes called the perceptron criterion of the perceptron objective function. Does it make sense?

(Refer to slide time 15.23)

$$D(\beta, \beta_0) = -\sum_{i \in M} y_i (x_i^\top \beta + \beta_0) \quad M = \text{set of misclassified points}$$

Point here is that if you take the misclassified data points so this is going to be a negative quantity and I want this to be as close to 0 as possible. In fact I want M to be empty. I want M to be empty, so that is that is the goal this is the way to minimize this is to make sure that M is empty as long as M is non-empty I still not achieved the optimum. When can M become empty? So if my data is actually linearly separable. If you remember I drew some data points sampling from Gaussians there and I drew a minus somewhere here and then pluses there and so on so for that kind of a data will never be linearly separable. I can never draw a straight line separating those data points so here things are nicely linearly separable. All the x 's are one side all the zeros are on the other side. Here the data was nicely separable so the perceptron objective function works well if the data is linearly separable if it is not linearly separable.

(Refer Slide Time: 17:28)

$$D(\beta, \beta_0) = -\sum_{i \in M} y_i (x_i^\top \beta + \beta_0) \quad M = \text{set of misclassified points}$$

$$\frac{\partial D}{\partial \beta} = -\sum_{i \in M} y_i x_i, \quad \frac{\partial D}{\partial \beta_0} = -\sum_{i \in M} y_i$$

we will run into problems. So now what we do is just use gradient descent. So I will differentiate D wrt β what will that be? So this is the derivatives and then so what I do now is essentially the technique that is very popular nowadays but it was not really called by that name in the olden days it is called stochastic gradient descent.

(Refer to slide time 17.25)

$$\frac{\partial D}{\partial \beta} = - \sum_{i \in M} y_i x_i, \quad \frac{\partial D}{\partial \beta} = - \sum_{i \in M} y_i$$

So people know what gradient descent is. What is gradient descent? Usually find the direction of steepest ascent and go in the opposite direction. How far do you go in the opposite direction? Proportional to the gradient. Gradient is very large in what you do? So it depends , so if you know the gradient properly I can compute the gradient set it equal to zero solve for it. And then just jump there. I do not have to go in small steps if I actually know where the gradient becomes zero I can just set my solution to that point.

So the problem comes when things are little iffy why as soon as I move my β in some direction what will happen in the perceptron case? If I move my β say in the direction of the gradient what will happen? Set M changes. So immediately my definition of gradient changes. So they move in a change β slightly I have to recompute the gradient. So the gradient I am computing is valid for wherever I am sitting in the β space when I move from there then I have to re compute the gradient.

So I cannot just move in the direction all the way. I can't really move a long distance pointed to by that particular point in space a particular gradient so I have to iteratively re compute the gradient so in such cases what we do is usually take a step in the direction indicated with a gradient and hope that you are moving in the generally expected direction. So if you take the expectation of the gradient you are moving in the right direction. So the stochastic gradient

descent has a lot of nuances and other things to it so we will not get into that whenever we used to stochastic gradient descent I will point out to you what are the things to be concerned about but will not get into the details of that. Possibly anyone doing the optimization course and look yeah you might I am not sure whether they are going to get to stochastic gradient descent but they might covered it in the course.

(Refer Slide Time: 20:36)

$$\beta \leftarrow \beta + \rho \begin{pmatrix} x_i y_i \\ y_i \end{pmatrix}$$

So what we are going to do is essentially take β so the gradient is $x_i y_i$ but since I am anyway doing stochastic gradient, as soon as I find one misclassified data point I am going to find the estimate of the gradient and move in the direction I am not going to wait till I find all the misclassified data points under that particular β for the current β , I find one misclassified data point and then I will change my β in the direction of the gradient. So what does it look like? So I just take the misclassified data point multiplied by the decided output and add it to the weight vector. I have written this in a really funny spacing right that is because I can usually what should go here now. I can convert this into matrix notation. So every time I encounter a misclassified data point I change my β . So we are doing it in this particular form because this is the perceptron learning algorithm. So if you want to find all the misclassified data points and then compute the cumulative gradient and then change your β within that direction you are welcome to do that and that is a perfectly valid way of doing it. The reason we are doing it once to one data point at a time is saying this is exactly how the perceptron learning algorithm was

derived So the ρ has to be really small if you are operating in a very, very stochastic environment. Because so in effect what you are doing when you are doing stochastic gradient descent is that you are making many, many different estimates of the gradient in a local region and you are trying to move in the expected direction of the average direction of the gradient so if the ρ is very large it turns out that you just make one estimate of the gradient and then move out of the region. So you will take a large step so we will go somewhere else the gradient will be completely different or this could have been the wrong estimate for the gradient direction. So to make sure that you are following a reasonable expected gradient ρ has to be really small. Having said that in the perceptron learning algorithm setting ρ to one actually works and that is how the original perceptron algorithm was also stated ρ was one.

And normally for stochastic gradient descent one is very, very, very large. You typically think of the order of 10^{-3} to 10^{-4} and things like that for step sizes okay but it turns out in this case it is fine ρ equal to 1 is fine and it will work it will converge you can show convergence with ρ equal to 1 but if in typically in stochastic gradient descent you want ρ to be small because the idea is what I told you just like we talked about taking a replacing expectation with an average over a region and into nearest neighbors. So something like that at a very, very gross level so you want to be able to take an average of the gradient in the local region but if you move very fast then you might be in a completely wrong direction. So therefore you go slowly okay so this is the perceptron learning algorithm

(Refer Slide Time: 25:07)

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \frac{\lambda}{n} \sum_{i=1}^n \begin{pmatrix} x_i y_i \\ y_i \end{pmatrix}$$

- Linearly Separable

So if the data is linearly separable so what you mean by linearly separable. That there exists a linear separating hyper plane that will make no mistakes. So if the data is linearly separable then the perceptron algorithm will converge to some solution. So what do I mean by some solution? So let us say these are the data points that are given to you.

(Refer Slide Time: 25:52)

- Linearly Separable then converge to some soln
- Can take a long time
- If not linearly sep, then loops

There are many, many solutions that work. I mean assuming all of those are straight lines so you have many, many different straight lines that you can draw that separates these two classes.

infinite number of them actually correct so the perceptron learning algorithm will converge to one of these three or one of these many infinitely many number of separating hyper planes which one would it converge to depends on the starting point. When it is hard to say apriori given a starting point can you tell me what it is going to converge to? You can just have to run the perceptron algorithm what is the problem. We will come to that. That is a way of defining one of these infinite number of hyper planes as the most desired hyper plane. So if you think there is a way to pick one of these yes we will come to that the second problem is it can take a long time and take a longtime especially if the gap between the two classes is very small and if the gap between the two classes is very small then it can take a long time. So partly it is a function of ρ to 1 but then it is hard I mean so see setting ρ to 1 essentially makes your thing oscillate a little bit right so. So you will have something that makes a mistake here then you will go back you will make a mistake here and then you have to keep going back and forth multiple times before you converge to the right answer but then setting ρ to something small okay will also make you move only small steps at a time so that also may take a long time so there is always a trade-off between how large you make ρ and how small you make ρ . And so one way of fixing this is what ρ is one thing but can increase the gap between the data points using transformations. Use basis expansions that we talked about earlier so instead of doing it in the original space say do it in the x^2 space or three x space or whatever I mean you can think of some way of scaling the data or transforming the data. So that the gap between the classes widen and therefore it converges quicker right.

But the third problem is the harder one. The data is not linearly separable what will happen to the perceptron algorithm? There will be no hyper plane where M is empty so as long as M is not empty I am going to keep adding something to the β again and again right as long as M is not empty I will keep adding it up was every iteration I will be changing the β it is not linearly separable. Then it enters then it loops basically then it loops but the problem is sometimes the loops can be very, very large and if you have a very large data sets it might actually be setting up a very large loop so it may not be easy to detect also. So normally it takes a long time to converge that in that if it is looping so you have no way you do not know if that the algorithm is taking a long time to converge or whether it is simply looping so how could you indirect loops.

So you think that every time it has to be monotonically decreasing. If it is going to converge yes right you think so towards convergence yes but when it is taking this whole very long time to actually work through the thing. So if there is no guarantee that cardinality of M will monotonically decrease also, so it is not very easy its cardinality of M is not the only thing that will give you so it is yeah has no efficient way to get around that okay so the problem.

So people discovered a lot of drawbacks to perceptrons and the biggest drawback was they discovered that some very, very simple problems that you want to solve are not linearly separable. So XOR is not linearly separable. I simply cannot even solve simple problems like XOR on what a useless thing is I do not care how well you make the baby speak you cannot solve XOR.

So I am going to forget about people remember I talked even the example how they trained a perceptron to reproduce speech and as the training progress it just sounded like a baby learning to speak and people just went gaga over it. Any way so we'll stop here it looks like I am actually out of time so the next class we will try to look up a I will try to explore a way of fixing this. So what is the problem there? Not that it is linearly separable but the fact that it could converge to any solution that is linearly separable we start by trying to define a single optimal solution.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

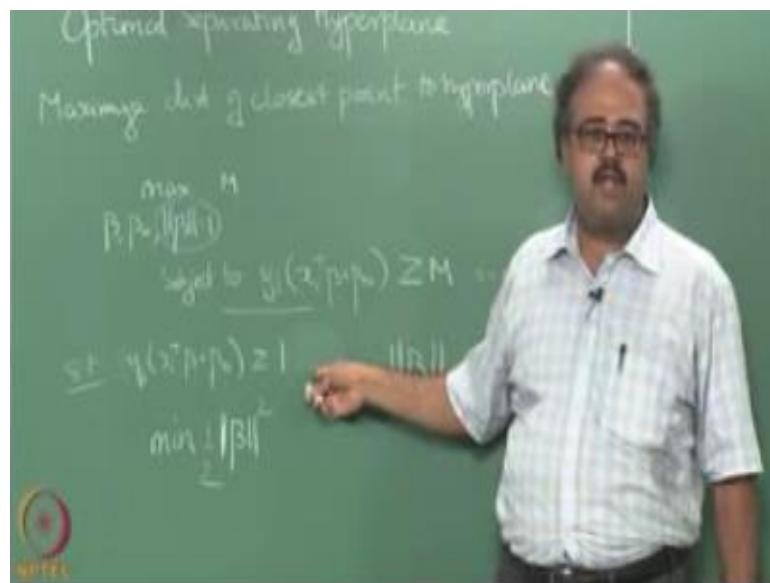
Lecture 27

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Support vector machine 1
Formulation

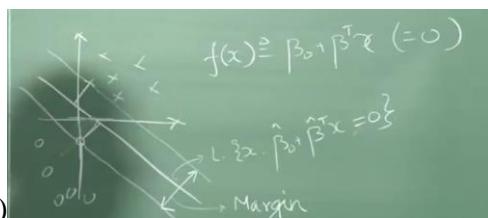
Also we are looking at linear classification and I will quickly remind you of the properties of hyper planes that we wrote down in the last class.

(Refer Slide Time: 00:27)



Ok we will denote that by $f(x)$. So the hyper plane is essentially given by solving the $f(x)=0$ and what I really want is, so I do not care about the other properties what I really want is for you to recall that the signed distance to the hyper plane is given by $f(x)$. So then we looked at the perception learning algorithm so does a problem with the perception learning algorithm?

Convergence yeah. So if it is linearly separable it will converge but it might converge slowly if it is not linearly separable it will cycle. But if it is linearly separable what can you say about the convergence apart from the fact it is slow? It depends on the starting point and it is not definite as to where it will actually converge. There is no particular solution to which it will converge so now what we're going to try and do is try to characterize a specific optimal solution. We will first start by considering the case of linearly separable data. So just like whatever I have drawn here so the data point is given to you is actually perfectly separable by a hyperplane. So we will start with this case now I am going to try and characterize what I mean by an optimal separating hyperplane. Give me some options for what could be optimal. The modulo sum of distances is maximum between the points and separating hyperplane. Sum of the distance of all the data points to separating hyperplane or maybe this point is that is closest. Exactly right so that makes a lot of sense so that is exactly what we are going to use so we are going to maximize distance of closest point to the hyperplane. So essentially so if you think of this data so that is close but this is closer so if I want to maximize the distance of the closest point what should I do? Move it like that or like that whatever some are moved further away from this so what how much further away can I move it until other side also I say the closest point from both sides should be at the same distance. For both classes the closest point should be at the same distance from the hyperplane and I have to choose an appropriate orientation for the hyperplane so that this distance is maximized. So that is essentially what we are going to try and do so instead of erasing the hyperplane and redrawing it again I just move the data point so that this is closer now. I'm essentially going to have a slab in some sense around the separating hyperplane which will have no points. So on the thickness of the slab will be the same on either side of the hyperplane. So that is essentially what am looking for so this is called the margin. Whatever you cleaned up around the hyperplane is called the margin so that is why these kinds of optimal hyperplane classifiers or sometimes known as max margin classifiers, because they are trying to maximize the margin is the distance of the closest point to the hyperplane is the margin.



(Refer to slide time 07.06)

So in fact so the margin would be that. So we know what $x_i^T(\beta + \beta_0)$ this lets the distance signed distance from the hyper plane so I multiply by y_i so that I always get it positive. So essentially what I am saying is I look at the quantity this is essentially the distance a data point is away from the hyper plane and I am saying that every data point has to be at least M away from the hyper plane that is my constraint. So go through all my training data points 1 to n and I am saying that every data point should be at least a distance M away from the hyper plane and under that condition maximize M . So I cannot make M arbitrarily large because I might not be able to find a β which will satisfy for every data point. So this is how I will write down the optimization function what we wanted was maximize the distance of the closest point to the hyper plane.

So now what I am going to say every point should be at least M away. Now maximize M this will automatically maximize the distance of the closest point right so what do you think will be the distance of the closest point? Whatever M you end up with so whatever is optimal M for this will be the distance of the closest point and that will be the margin so that is essentially what you are doing is you are directly maximizing the margin.

(Refer to slide time 09.59)

$$\begin{aligned}
 & \text{Optimal Separating Hyperplane} \\
 & \text{Maximizing dist. of closest point to hyperplane} \\
 & \max_{\beta, \beta_0, \|\beta\|=1} M \\
 & \text{Subject to: } y_i(x_i^T \beta + \beta_0) \geq M, i=1, \dots, N
 \end{aligned}$$

So we have this constraint that $\|\beta\|=1$ because we do not want the solution to blow up arbitrarily. So instead of that we will get rid of that by that if I did not have if you didn't have the constraint of $\|\beta\|=1$, I can arbitrarily make β large here and make things larger than M . So now I am normalizing by β so that I do not have to worry about that. I can remove this constraint here that instead of that I put the $\|\beta\|$ here in the denominator in the constraint. It make sense right but then

I can do something more interesting now can I do that right great now let us step back and think about it for a minute so if something satisfies this constraint right.

$$\frac{1}{\|\beta\|} (y_i(x_i^T \beta + \beta_0)) \geq M$$

$$y_i(x_i^T \beta + \beta_0) \geq M \|\beta\|$$

I can arbitrarily scale it right so it will still satisfy the constraint you think of it because I have $\|\beta\|$ on this side so if β already satisfy this constraint okay I can just scale the β and then that will satisfy the constraint as well correct so I can arbitrarily set and I still have to find out the direction of orientation of the β . I'm just saying that whatever orientation of the β you pick you normalize it so that the $\|\beta\|$ is $1/M$.

(Refer to slide time 12.26)

$$y_i(x_i^T \beta + \beta_0) \geq 1 \quad , \quad \|\beta\| = \frac{1}{M}$$

So people with me so far so kind of I started with this constraint started with that optimization problem made the assumption that well $\|\beta\| = 1 / M$ and I came up with this problem so then nothing just little geometric I mean algebraic manipulation in fact it is geometric as well. I am just not drawing the geometry here right but then the objective function also now can change it is a maximizing M again minimize norm β because now β is 1 by M right.

Now we do this instead

$$\min \frac{1}{2} \|\beta\|^2$$

So I can do that subject to the constraints here right I am going to do all of that so that it makes it easy for me to take derivatives minimize and things like that so I just made it a squared function so that can manipulate it more easily does not matter minimizing $\|\beta\|$ is the same as minimizing β square right so norms or anyway positive or non-negative.

Now I can go and say that this margin is actually going to be $1/\|\beta\|$. So another way of thinking about it is what I am trying to find is a minimum norm solution such that all the data points are what are correctly classified. So what does this mean when $y_i(x_i^T \beta + \beta_0) \geq 1$, that essentially means that x_i is in the right side of the hyper plane right remember that so y_i is +1 then this is also positive right I mean at least +1 right so not only it should this be positive there has one and then you will get a 1 here and similarly if y_i is -1 which is the other side of the hyper plane so this product this term has to be at least -1 right so that you will get a +1 it will be greater than or equal to +1 so you know that all the data points are correctly classified correct and minimizing β essentially it is the smallest possible β . So finding the smallest both β set data points are correctly classified and not only correctly classified they are at least certain distance away from the hyper plane.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 28

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

**Support Vector Machines II
Interpretation & Analysis**

So this is the optimization problem which is actually a simple optimization problem is a quadratic objective and a set of linear constraints. We already saw how to solve this. You guys had a convex optimization tutorial. So one of the things that we are looking for from the convex optimization tutorial is that you will know how to solve this problem. So what we do after this, write the Lagrangian right.

(Refer Slide Time: 01:46)

$$\text{st } y_i(x_i^T \beta + b) \geq 1, \quad \|\beta\| = 1$$
$$\min \frac{1}{2} \|\beta\|^2$$
$$\text{Lagrangian } L_P = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i(x_i^T \beta + b) - 1]$$

(Refer to slide time 01.46)

A handwritten equation on a chalkboard. It starts with the word "Lagrangian" followed by a crossed-out "f". To the right is the formula $L_p = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i(x_i^T \beta + \beta_0) - 1]$.

$$L_p = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i(x_i^T \beta + \beta_0) - 1]$$

I have to apply this for every data point so I run runs from $i=1$ to n okay. And I put a p there so there is a primal so we will have to form the dual of this the dual looks a lot easier to solve. Dual is actually a lot easier to solve. So we will go ahead and do the dual. So for first I will take the derivatives. So derivative with respect to β and you can do that and solve it we will get that derivative with respect to β_0 so now is where I'm going to do some hand waving but you can go through this computation so take that substitute into this okay. And do a lot of simplification rights, so remember we have this β squared here therefore I am going to get a $\alpha_i \alpha_j y_i y_j$ kind of terms.

(Refer to slide time 03.26)

Handwritten text "Setting derivatives to 0". Below it are two equations:

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i$$
$$0 = \sum_{i=1}^N \alpha_i y_i$$

A small timer icon in the bottom right corner shows "03:59".

(Refer to slide time: 04.38)

$$\text{Dual: } L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N \alpha_i \alpha_k y_i y_k x_i^T x_k$$

Subject to $\alpha_i \geq 0 \quad \forall i$

So the dual will be so the dual is going to be a slightly simpler form why is it a slightly simpler form so I have to only consider my constraints have become of lot simpler here right it just going to be α_i 's should be non-negative that solves my constraints are so it turns out that there are efficient ways of solving optimization problems of this form. You do not have to worry about it. Here are lots of packages that solve SVMs for you.

But then you just need to know what kind of optimization problem we are solving. I do not want you to use it as a black box. Essentially what you are going to be solving is this. So when you have a solution, when you have something that is both primal and dual so we can actually show that the duality gap is 0 in this case so it is not going to that. But the point is when I have a solution to the problem right it has to satisfy certain conditions.

It is already looked at that the KKT conditions. If people do not remember it please go back and revise that right. So there are a whole bunch of things so you need to for you need to have the solution to be primal feasible right. You need to have the solution to be dual feasible and so that essentially have a bunch of things. Primal feasible would mean that well your α_i 's have to be great that will be dual feasible way that will be one condition these need to whole because it is a solution for the primal and there you are you have your complimentary slackness right. So that in this case becomes

(Refer to slide time 06.37)

$$\alpha_i [y_i(x_i^T \beta + \beta_0) - 1] = 0$$

So I know if in the notes I think you saw it as $\lambda_i f_i$ right. So this essentially that is it so this is $\alpha_i f_i$. So this is this may affect so that is the fourth these are the KKT conditions, that need to be satisfied okay. And so what does this tell us?

Tells us a couple of things one so we know what the form of β should be what is the form of β it has to be $\alpha_i y_i x_i$ right. So it is essentially what you are going to do is your β will be taking out certain data points from your training data right and adding them up. So suitably multiplying it by the output the desired output, so if x_i 's output was positive then this will be +1 if x_i 's output was negative this will be -1.

So it is going to take a few of those and they are going to add them up right. So this should remind you of perceptrons. So if you remember what we did in perceptions is we took whatever was misclassified we just kept adding it to the weight vector. So in some sense you are doing something very similar to that but instead of having some kind of a heuristic approach to optimizing things. We did do a gradient descent but then we just said ok we will arbitrarily pick the set of misclassified points and we will do the gradient descent and so on so forth. But here we started off by saying okay we will minimize the distance to the closest point and from there we derive something and it looks very suspiciously like the perceptron update rule. In fact nowadays when people say I am going to train a perceptron they are actually doing this more often than using the perceptrons learning rule right away. So now something else that you can observe so this condition has to be satisfied. This condition has to be satisfied. So let us look at it there are two terms here so when will $\alpha_i [y_i(x_i^T \beta + \beta_0) - 1]$ be 0? When either α_i is 0 or $y_i(x_i^T \beta + \beta_0) - 1$ is 0. These are some condition when this has to be 0. Yeah! You are right but for geometrically can you give me an answer. So α_i has to be zero is when the other term is not 0. So when will the other term be not 0? When it is not the closest point. If x_i is the closest point it will be bang on the margin and for a point here that term will be 0. For a point here that term will be greater than 1 or a point here the term will be greater than 1.

You see that so since the term will be greater than 1 the term in the square brackets will be non 0 so α_i 's have to be 0. Correct, so what does this mean it means that points that are further away from the hyper plane do not contribute to finding β . Because the α 's will be zero points that are

far away from the hyper plane are not going to contribute in finding β . In fact the points that will contribute to β are exactly those points that are on the margin.

So in fact for this, this data set that they drew here right there are only two important points at that one and this one because only two points are on the margin right. Such points which lie on the margin are known as support points or support vectors right. And your β is going to depend only on the support points. What about β_0 ? So we can plug in any data point here, and we can solve for β_0 right.

One of these support points you can plug it in here and you can solve for β_0 . Which support point do you pick? Ideally all of them should give you the same answer but usually does not happen because of numerical reasons. So what typically people do is they plug in all the support points okay. Solve for β_0 and take the average right. So each one in turn it for every support point you are going to get slightly different β_0 you just take the average okay.

So that is how you compute the hyper plane at the end of it is basically how here when would α be 0 if your data is on the hyper plane. So that will be one case when that happens. Essentially you have two points which are on the same. It is not collinear but repeated things; I give you two data points that are on the same point right. So by definition most of the support vectors will lie on the same line so it cannot be collinear okay. So right in such cases that could be the case but, yeah! These are generally degenerate cases yeah! So sure call them support vectors if you want. So one thing to note is my \hat{f} so how this going to look like now that I given the form for β here. This is essentially going to look like I can flip these things around anyway that plus β_0 right.

(Refer to slide time 15.24)

$$\hat{c}(x) = \text{Sgn}(\hat{f}(x))$$

$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}^T \hat{x}$$

$$= \sum_{i=1}^N \alpha_i y_i \underline{x_i^T x} + \hat{\beta}_0$$

So, so if you think about it I will come back to this point later so if you look at the dual I only have $X^T X$ right and if you look at the final classifier I am going to use I am going to have $X^T X$. So if I have a very efficient way of computing $X^T X$ right I can do some tricks with this whole thing we will come back to that okay. I will just I want you to remember this so any questions on this, any questions on this?

So before we move on I just wanted to point out something so if you think about, how LDA works right. So LDA tries to do density estimation eventually right, if you if you think about it you make some assumptions about the probability distribution the form of the probability distribution. What assumption will you make it is Gaussian with equal covariance across all the classes' right.

Though, that essentially means that every data point in your training set is going to contribute towards the parameters that you are estimating right. So the β will estimate there will depend on all the data points that were given to you, whether they are here right close to the hyper plane or whether they are very far away from the hyper plane. Let us all the data points will determine your class boundary so that means that it becomes little susceptible to noise.

And if I have one or two data points that are generated through noise right even that will contribute to determining the separating plane hyper plane right. On the other hand we test with

this kind of optimal hyper plane we are only worried about points that are close to the boundary right. So I can do whatever I want here right I can change move a few points over here and things like that it does not really matter.

What matters is if any noise enters close to the boundary? So that so in some sense if my noise is uniform, the LDA will get more affected. Because even if noise insert some points there right LDA classifier will change. Well my optimal hyper plane classifier will not move it will be affected only by that fraction of the noise that changes the actual decision surface. Having said that I should point out that if, if your data is truly Gaussian with equal covariance LDA is actually optimal. It is probably optimal. While this one will depend on the actual data that you get but in general would say this is more preferable because this is more stable. People remember what stability is right; small changes in the data will not cause the classifier to change significantly right.

So here small changes in the data will not cause it to change significantly in an expected sense right. If I go and take the support vector and move it somewhere else okay. The class boundary will change. But then I have whole bunch of other vectors which I can move around nothing will happen to the class boundary unless I move it closer to the hyper plane than the existing support vectors. If I take a point from here and move it here of course the class boundary will change. As long as I do not modify which are the support vectors, I will get back the same classification surface again and again all right. So in that sense SVM or will come to SVM little bit, this kind of optimal hyper plane are very stable.

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Higher Education

Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 29

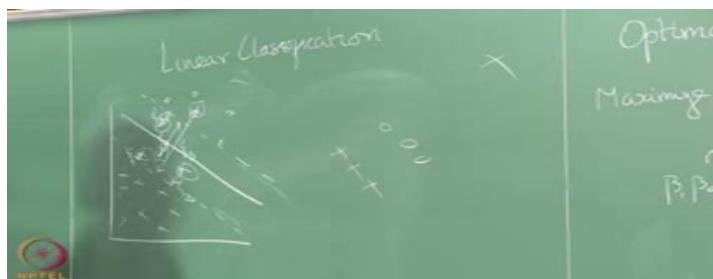
Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

SVMs for Linearly Non Separable Data

Suppose I have some data which is not linearly separable. So that is the problem that we have seen with perceptions? So what happens if the data is not linearly separable? Perceptrons do not converge. So can we tweak our objective function that we have here to make sure that we can handle non linearly separable data. We are saying it is okay to say non linearly separable data was my question. It should be linearly inseparable data right, so you have to be careful where you put the not the negation there.

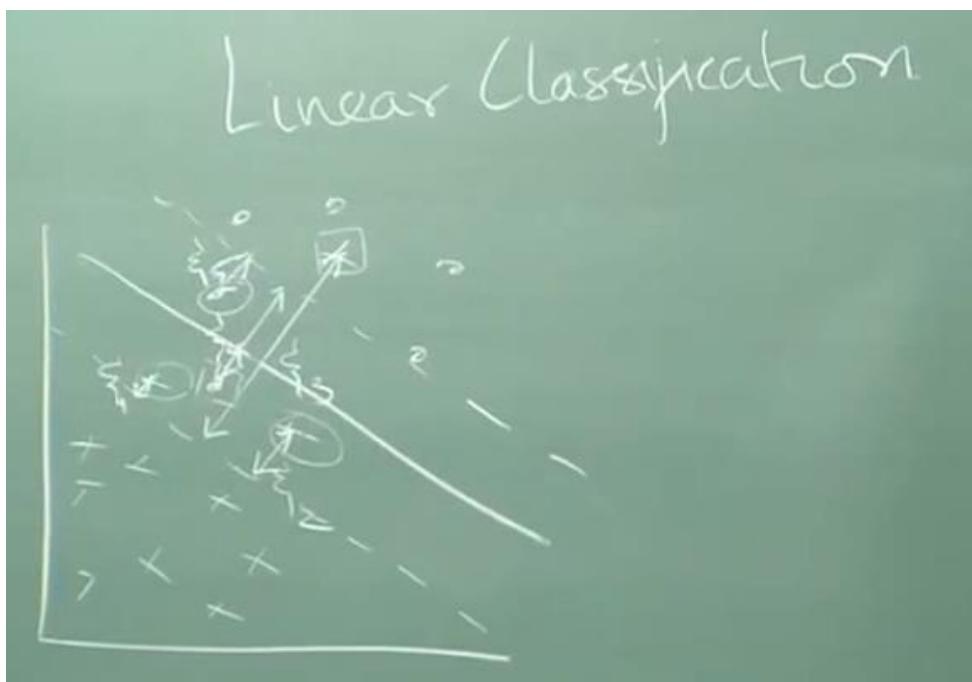
So what we do in this case? Somebody had a suggestion. So there are many ways there are many choices you can make right let me not play around with it are many choices you could make but there is one particular choice which is seems to yield a very nice optimization formulation. So what is a choice I am going to say that I would really like to maximize the margin and I would like to get as many data points correct as possible right.

(Refer Slide Time: 01:42)



So if you think about it so there are a couple of things. So this is the margin that I want so what are the problems here? Well these data points are within the margin so I have some data points that are within the margin so I would like to minimize such cases there are some data points that are within the margin and erroneous. I would like to minimize such cases as well. If you think of what if I had tried to get this correct, there is a gap here and there seems to be a gap here between the points if I try to get this correct and move my classification surface below then the margin would have been reduced even further. So it is okay to get this wrong but then what about this case is it within the margin or outside the margin? Within. So the margin for that class is defined on the other side right so the margin for that class is this side so anything to this side and x is within the margin. This will be y_i times this right, so this will actually be negative so it is within the margin if we want things to be greater than one y_i times $f(x)$ we want it to be greater than one right ≥ 1 this is going to be negative.

(Refer to slide time: 04.25)



So obviously this is within the margin. So essentially what I want to do is minimize these distances so you can see the distance that I marked here so these distances I would like to minimize, so this is a certain small distance inside the margin, this is a large distance inside the margin is a very large distance inside the margin likewise so I can mark each one of these and I

want to minimize these, so let us denote them say ξ_1 to ξ_5 and I want to minimize those right essentially.

So if I minimize the sum of these deviations I make along with my original objective function. Why do not they minimize the maximum here? Because that would essentially mean that I will try to get as many things character as possible so in this case I do not mind getting something wrong as long as the overall deviation is not does not exceed a certain limit. See that the difference between minimizing the maximum and minimizing a sum is that I might as well give up all of the sum to a single data point it might be something that is very hard to classify.

And I might have one single outlier somewhere here right let us let us draw so this data might be perfectly separable and I might have an outlier then okay so now if I say okay minimize the sum of the things it is fine. But if I say minimize the max okay then it is going to actually give me a hyper plane somewhere there but like I said many different formulations are possible this one actually yields a very nice computation that is one of the reasons people use this.

(Refer Slide Time: 07:19)

$$y_i(x_i^\top \beta + \beta_0) \geq M(1 - \xi_i)$$

So what I am going to do is write it here. So I am going to say that this has to be that we had already found out and I am going to introduce a slack variable so it does not have to be greater than M . It can be some fraction lesser also M is what I would really like but I allow it to have a slack. Ideally I would want most of these ξ_i 's to be 0 and I force ξ_i 's to be zero I am back here but I really like some leeway right.

So I am allowing myself that leeway by introducing ξ_i here. This is a very standard technique for relaxing constraints in optimization. That is one of the reasons people adopt this is a standard constraint. So another thing which I could have chosen is that in fact this is a little bit more common constraint but it turns out that in this particular case if I choose $M - \xi_i$ instead of $M(1 - \xi_i)$ I end up getting a non convex optimization problem.

So we do not want that so we end up doing this. So I drew this figure first because I wanted to get an idea of what these slack variables actually mean. So the slack variables essentially tell you by what fraction right you are violating the margin. So is ξ_1 is essentially what fraction of distance you are coming in here from the margin ξ_2 is what fraction of the distance you are coming in from the margin.

So the margin is M so I have moved some fraction of the distance inside. So they essentially that is what the ξ tells me. So what are the constraints we have? So the first constraint I have is okay all ξ_i have to be ≥ 0 . I do not care about points going to that side of the margin. So all $\xi_i \geq 0$ and the second thing is whatever we have been talking about. I do not want the ξ_i is to be very large taken in total so I want to upper bound them by a constant.

So because I am talking about going that side of the margin, if ξ_i are negative so essentially I will be imposing a tighter constraint than what I was looking for so this will be like it will larger than M . This is well I will be having a thing that is larger than M and it is a relative distance, so essentially this becomes $M - M\xi_i$. So the original should be M , so it is now $M - \xi_i$ away from there. So ξ_i is essentially a relative distance and if I make ξ_i is negative so this will become plus so that will essentially mean that not only do I want the data points to be away than M actually asking it to be further away than so it just imposes a tighter constraint. So I do not want that to happen so and here we are essentially giving it a budget that we do not want it to be greater than the budget right fine. So we saw such a constraint earlier where it we see such a constraint earlier we had a budget we did not want it to be greater than a budget.

(Refer to slide time: 10.43)

$$y_i(x_i^\top \beta + \beta_0) \geq M(1 - \xi_i)$$

$$\forall i, \xi_i \geq 0$$

$$\sum_{i=1}^N \xi_i \leq \text{Constant}$$

So yeah ridge regression and LASSO and other things we had this thing. So wherever we are looking at this regularized regression, so we had this greater than or lesser than a constant and what did we do in those cases? We push it into the objective function and then add a multiplier there and then we said okay it has to be. So then there is a relationship between this constant and the multiplier that we put in the objective function. So likewise will do the same thing here I will do all the other transformations that we need to do to normalize β and things like this.

So essentially I will end up with the same objective function I had there and you want \geq to because they have gotten rid of the M. Why how do we get rid of M, because M is $1/\beta$. So $1/\|\beta\|$ so we got rid of that just anything else we need here. So now that we have this objective function what should be the value of C? If I want a linearly separable case we want to solve the linearly separable problem right or I want to ensure that all ξ_i are 0 what should the value of C be this is the simple question infinity right C should be infinity.

(Refer to slide time: 13.53)

$$\min \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

Subject to:

$$y_i(\mathbf{x}_i^\top \beta + \beta_0) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

So the larger the value of C the more you are penalizing the violations. So the smaller the ξ_i should be. So the larger the value of C the smaller the ξ_i should be so there is a trade-off. So the larger you make C, the smaller will the margin be. But we will be getting more of the training data correct right so for large values of C you are allowing a little bit more leeway. So C is very small then you are allowing lot more errors to happen if C is very large then you are forcing the classifier to classify as much of the training data is correct as possible okay.

The data is truly linearly separable and you make C very large what will happen? You will find the correct linear separator. But if the data is truly linearly separable but you keep C small what might happen? You might trade off errors in the training data for a larger margin even if the data is linearly separable. Is that a desirable thing? When exactly? So if the data is noisy such that there are some data points that are closer to the margin it may be one or two data points that are closer to the margin. So if you are trying to find the perfect linear separation you will pay attention to them as well right and therefore you will end up having a low margin, but then if you are willing to ignore a few noisy data points, even if the training data looks perfectly separable you might end up making a few errors on it but you will get a more robust classifier. So can people visualize a situation I am going to try and do something here let us see that works it has looks perfectly separable. That is noise is it still separable. There you go there it is still separable and if you try to solve it as a perfectly separable problem that is the separator that you are going to get but if you allow errors right so that will probably be the separating hyper plane you get and that is probably a more appropriate hyper plane right.

(Refer to slide time: 16.11)



Apart from being robust it is also correct in an expected sense. We will move on to the primal. So I just wanted to leave this on board till I wrote this note so that you can compare it.

(Refer Slide Time: 16:46)

Setting derivatives to 0

$$\beta = \sum_{i=1}^N \alpha_i y_i \beta_i$$

$$0 = \sum_{i=1}^N \alpha_i y_i$$

$$\alpha_i = C - \mu_i - \gamma_i$$

So that is the primal value also having have α and ξ , α, μ has to be ≥ 0 .

(Refer to slide time: 18.33)

$$\text{Primal } L_p = \frac{1}{2} \|\beta\|^2 - C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(x_i^\top \beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^N \mu_i \xi_i$$

Setting derivatives to 0

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i$$

$$0 = \sum_{i=1}^N \alpha_i y_i$$

$$\alpha_i = C - \mu_i \quad \forall i$$

(Refer Slide Time: 18:59)

$$\alpha_i = C - \mu_i \quad \forall i$$

$$D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

$$\text{Subject to } 0 \leq \alpha_i \leq C \quad \sum_{i=1}^N \alpha_i y_i = 0$$

Yeah I do not have to do this. It is not a single condition is there for each i . Do we need the

$\sum_{i=1}^n \xi_i \leq \text{const}$. No right, so that is why we consider constructed put that into the optimization

objective function itself right, so by minimizing this right we are ensuring that $\sum_{i=1}^n \xi_i$ will be less

than some limit right and like I was mentioning in the ridge regression discussion, so you can find a relationship between this constant and this C right.

It is also a function of the range of the objective function but you can always find so basically they are equivalent ways of writing the optimization problem except that you have to this constant and the C will not be the same there will be different values so this constraint is gone this is no longer present here that went into the objective function.

So putting all of this back in and doing some algebra can be surprised at the algebra outcome of this. Anyone has already solved it? Looks familiar right? It is essentially the same dual you will get but your constraints are different. This is already there so it is just added for completeness sake but what is important here is earlier while I had only a non negativity constraint on α now I have an upper bound on the value of α . So why is that because α is only $C - \mu$ and since α is only $C - \mu$ so there has to be an upper bound on α okay good so what about the other KKT conditions.

(Refer Slide Time: 22:28)

$$\alpha_i = C - \mu_i \quad \forall i$$

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

Subject to: $0 \leq \alpha_i \leq C \quad \sum_{i=1}^N \alpha_i y_i = 0$

So 1 to 7 are the KKT conditions.

(Refer to slide time: 23.31)

$$\begin{aligned} \text{KKT} \\ \alpha_i [y_i(x_i^\top \beta + \beta_0) - (1 - \xi_i)] &= 0 \quad (5) \\ \mu_i \xi_i &= 0 \quad (6) \\ y_i(x_i^\top \beta + \beta_0) - (1 - \xi_i) &\geq 0 \quad (7) \end{aligned}$$

So what do you notice here again? Well you notice again that your β is determined by your $\alpha_i y_i x_i$ just like you had earlier. β is given by those x_i for which α will be nonzero. So like we had earlier those x_i 's for which α is non zero or called support points. Our support vectors depending on how we want to look at it. Now let us go look at when it will be nonzero. So when will α be 0? When will this whole thing be nonzero? Well it lies at a large enough distance on the right side of the margin right what about ξ_i is a will be 0. Then ready somewhere here the ξ_i will be 0. So in the $\xi_i = 0$ so it will be left with this term alone -1 that just takes exactly the same condition that we had earlier. So if this is far enough away from the margin then this will be nonzero so α is have to be 0. So we know for sure okay the same thing. Things that are on the right side of the margin means correct side. Things that are on the correct side of the margin then α will be 0 so they would not contribute anything.

(Refer Slide Time: 29:21)

$$\begin{aligned} \text{If } y_i(x_i^\top \beta + \beta_0) > 1 &\Rightarrow \alpha = 0 \quad \left\{ \begin{array}{l} \xi_i = 0 \\ \mu_i = 0 \end{array} \right. \\ \text{If } y_i(x_i^\top \beta + \beta_0) = 1 &\Rightarrow 0 < \alpha_i < C \quad \left\{ \begin{array}{l} \xi_i = 0 \\ \mu_i = 0 \end{array} \right. \\ \text{If } y_i(x_i^\top \beta + \beta_0) < 1 &\Rightarrow \xi_i > 0 \Rightarrow \mu_i = 0 \Rightarrow \alpha_i = C \end{aligned}$$

So now what about things that are on the margin? Is that a third case? We have to consider third case now right. We have to consider the third case in which case what happened as ξ_i will start increasing right ξ_i 's will become non zero. If ξ_i is nonzero what does it imply? Because my α is $C - \mu_i$, so if ξ_i is nonzero then μ_i will become 0 that for my α is will become C . So now how will this term go to 0 by suitably makings ξ_i a large enough.

So I will make ξ_i a large enough. So that this term will go to 0. Because this will be negative will be less than 1. So I will make this I will just ξ_i so that this term in the square bracket goes to 0 because my α_i will be C what is that in case this is because I do not want to penalize this case. This case also ξ_i will be 0. So this case ξ_i is 0 this case also ξ_i will be 0 because what I really need is that is my condition $\geq 1 - \xi_i$ so if is equal to 1, so I can set ξ_i to 0. In both these cases $\xi_i = 0$.

So what are all the support vectors? Everything on the margin and everything on the wrong side of the margin as well right. Everything for which alpha is nonzero now becomes support vectors so at the end of the day I am going to say that you are just going to use a package to solve all of these things right but it is like saying yeah anyway you are going to use Microsoft Windows or I mean Mac OS X or something why do you learn operating system right. So you need to know what the internals are it is not the fact that you just use the tools that matters it when you can just use the tools well yeah we can do a tool course right how to use the tools right how will you start up limbsvm it is not trivial. So many people I know actually run experiments with SVM's by just using the default parameter settings that the package will give. The thing is you need to understand what is it that you are tuning right. So now I told you about the C parameter right, so you understand have some idea of what a large C means versus what a small C means hey instead of blindly saying that okay I am going to queue C from some number to some other number right, so to have an appreciation of what these things are doing actually helps you even use the tools better right so that is the whole idea behind doing all of this is not that I am going to expect you to come and derive a large margin classifier tomorrow when ideally.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

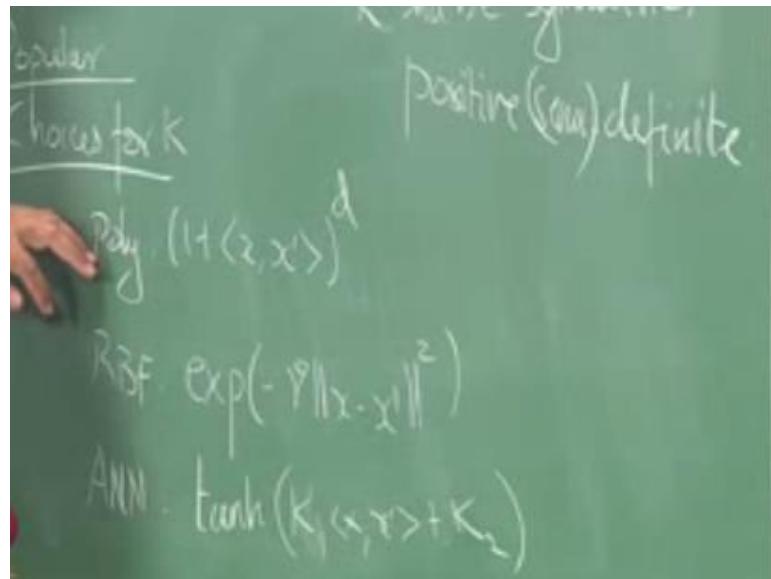
Lecture 30

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras**

SVM Kernels

So if you remember I asked you to note the fact that I am using a inner product there right $x_i^T x$ as the inner product of two vectors and the way I wrote the dual also I had only inner products in there. So in fact if I want to evaluate the dual I need to only know the inner products of the two vectors. Likewise if I want to finally evaluate and use the classifier that I learn I still need to only find inner products right.

(Refer Slide Time: 01:24)



So if I can come up with a way of efficiently computing this inner products, I can do something interesting. So what is that so what do we normally do to make linear classifiers more powerful?

Basis transformations. So I can just take my x and replace it with some function $h(x)$ that gives me a larger basis. It could be just replace it with the square.

I take x and replace it with x^2 and then I will get a larger basis and now it turns out that I can do a lot of math but I can get a dual that looks like this. So that is the inner product notation and so if I can compute the inner product, I can just solve the same kind of optimization problem but I can do this in some other transformed space.

(Refer to slide time 02.11)

$$x \rightarrow h(x)$$

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle$$

So likewise our $f(x)$ is going to be so essentially what I need to know is $h(x)$ for whatever pair x and x' that I would like to consider. So in the training it is the pairs of training points right while I am actually using it is one of the support point and the input data that I am looking at any point I just take this pairs of data points and I need to compute the inner product.

(Refer to slide time 03.13)

$$f(x) = \sum_{i=1}^N \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0$$

$$\langle h(x), h(x') \rangle$$

So I am going to call this as some function which is a kind of a distance function or a similarity measure between $h(x)$ and $h(x')$. Such similarity measures are also called as kernels. So we have been hearing about kernels in the context of support vector machines we have been trying to use this libsvm or any of the other tools for some projects over the summer you have heard of kernels. Kernels are nothing but similarity functions. So the nice thing about the kernels that we

use is that they actually operate on x and x' . They operate on x and x' but they are computing the inner product of $h(x)$ and $h(x')$. Did you see that? They are going to work with x and x' but they will be computing the inner product of $h(x)$ and $h(x')$.

(Refer to slide time 04.27)

$$\begin{array}{l} i=1 \\ \hline \text{Kernels} \\ \langle h(x), h(x') \rangle = K(x, x') \end{array}$$

So I will give you an example so the kernel function k should be symmetric and positive semi-definite. Positive definite and semi definite is fine in some cases positive definite. People remember what positive definite is right? $x^T A x > 0$ if it is definite and $x^T A x \geq 0$ if it is semi definite. It essentially we want the quadratic forms to be to be positive.

We do not want to take $x^T A x$ and suddenly find it is negative so it is in fact you remember I told the $x^T A x$ is usually the quadratic form that we are trying and that will actually mess up big time in the computation if the quadratic form becomes negative. Then we will have problems in all the optimization thing going through okay so that is the mechanistic reason for wanting it to be positive semi-definite. There is a much more fundamental reason for it which I have not developed the math or the intuition for you to understand. So it has to come at a later course. So hopefully in the kernel methods course if you are taking it you will figure out why that is needed. So there are many choices which you can use for the kernels.

(Refer to slide time: 07.26)

$$\begin{array}{ll} \text{Popular Choices for } K & \text{Positive } K \\ \text{Poly: } (1 + \langle x, x' \rangle)^d & \\ \text{RBF: } \exp(-\gamma \|x - x'\|^2) & \\ \text{ANN: } \tanh(K_1 \langle x, x' \rangle + K_2) & \end{array}$$

So there is something called the polynomial kernel which is essentially $(1 + \langle x, x' \rangle)^d$. So d is a parameter you can have. d of two three four you can even have d of one is essentially here whatever we have solved. The next one is called the Gaussian kernel or the RBF kernel right so where the distance is given by $\exp(-\gamma \|x - x'\|^2)$ and is essentially the Gaussian without your normalizing factor. So that is why it is called the RBF kernel so if you want to call it the Gaussian kernel you actually have to make it Gaussian otherwise call it the RBF kernel.

And then this is called the neural network kernel or the sigmoidal kernel sometimes. This is just the hyperbolic tangent $\tanh(K_1 \langle x, x' \rangle + K_2)$. Some arbitrary constants k_1 and k_2 which are your parameters that you choose and this is x, x' inner product. So these are some of the popular kernels which can be used for any generic data but then depending on the kind of data that you are looking at where the data comes from people do develop speech the specialized kernels they for examples for string data people have come up with a lot of kernels.

When you want to compare strings how do I look at similarity between strings so the nice thing about whatever we have done so far is that you can apply this not just to data that comes from \mathbb{R}^p right you been assuming so far that your x comes from some p dimensional real space as long as you can define a proper kernel right you can apply this max margin classification.

That we have done to any kind of data does not have to come from a real-valued space. Which is not true of many of the other things you are looked at right all those inherently depend on the fact that the data is real valued. Because of this nice property of what is called the kernel trick you could do all of this nice things so as long as you can define appropriate kernel that you can actually apply this to any kind of data. So that is one very powerful idea.

(Refer Slide Time: 09:28)

$$\begin{aligned}
 (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^2 &= (1 + \mathbf{x}_1 \mathbf{x}'_1 + \mathbf{x}_2 \mathbf{x}'_2)^2 = (\mathbf{x}_1 \mathbf{x}'_1)^2 \\
 &= 1 + 2 \mathbf{x}_1 \mathbf{x}'_1 + 2 \mathbf{x}_2 \mathbf{x}'_2 + (\mathbf{x}_1 \mathbf{x}'_1)^2 - (\mathbf{x}_2 \mathbf{x}'_2)^2 \\
 h_1(\mathbf{x}) &= 1, h_2(\mathbf{x}) = \sqrt{2} \mathbf{x}_1, h_3(\mathbf{x}) = \sqrt{2} \mathbf{x}_2 \\
 h_4(\mathbf{x}) &= \mathbf{x}_1^2, h_5(\mathbf{x}) = \mathbf{x}_2^2, h_6(\mathbf{x}) = \sqrt{2} \mathbf{x}_1 \mathbf{x}_2
 \end{aligned}$$

So just to convince you so let us look at the polynomial kernel of degree two operating on vectors of two dimensions. There are two 2's here so the degree is two the d is two and the p is also two but they need not necessarily be the same that I could have had a much larger thing but it was easy for me to write something so this is what

(Refer to slide time: 10.33)

$$\begin{aligned}
 (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^2 &= (1 + \mathbf{x}_1 \mathbf{x}'_1 + \mathbf{x}_2 \mathbf{x}'_2)^2 \\
 &= 1 + 2 \mathbf{x}_1 \mathbf{x}'_1 + 2 \mathbf{x}_2 \mathbf{x}'_2 + (\mathbf{x}_1 \mathbf{x}'_1)^2 - (\mathbf{x}_2 \mathbf{x}'_2)^2
 \end{aligned}$$

Now just squared it now if you think of h we get the following.

(Refer to slide time: 11.18)

$$\begin{aligned}
 h_1(\mathbf{x}) &= 1, h_2(\mathbf{x}) = \sqrt{2} \mathbf{x}_1, h_3(\mathbf{x}) = \sqrt{2} \mathbf{x}_2 \\
 h_4(\mathbf{x}) &= \mathbf{x}_1^2, h_5(\mathbf{x}) = \mathbf{x}_2^2, h_6(\mathbf{x}) = \sqrt{2} \mathbf{x}_1 \mathbf{x}_2
 \end{aligned}$$

So what is this function h ? It is essentially the quadratic basis expansion. So I have two features $x_1 x_2$. So if I give so remember that x, x' is $x_1 x_2$. So this is essentially the quadratic expansion the first thing is one the second coordinate is x_1 third coordinate is x_2 so it keeps it as it is then fourth coordinate is x_1^2 fifth coordinate is x_2^2 the sixth coordinate is $x_1 x_2$ it is all the quadratic basis expansion. Now if I make this operate on x and x' and take the inner product so what will be the terms? $1, 2x_1 x_1', 2x_2 x_2', x_1^2 x_2 x_1'^2, x_2 x_2'^2, 2x_1 x_1' x_2 x_2'$ is exactly what we have here right so what is the nice thing about it is I can essentially compute the inner product of x and x' first add 1 and square it so numerically what I will end up with is the same as what I would have ended up with if I had done the basis expansion right and then taken the inner product

(Refer Slide Time: 13:05)

$$\begin{aligned}
 (1 + \langle x, x' \rangle)^2 &= (1 + x_1 x_1' + x_2 x_2')^2 \\
 &= 1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 + (x_2 x_2')^2 \\
 x = (x_1, x_2) &\quad + 2x_1 x_1' x_2 x_2' \\
 h_1(x) = 1 & \quad h_2(x) = \sqrt{2} x_1, \quad h_3(x) = \sqrt{2} x_2 \\
 h_4(x) = x_1^2 & \quad h_5(x) = x_2^2, \quad h_6(x) = \sqrt{2} x_1 x_2 \\
 (2, 3) \\
 (4, 5)
 \end{aligned}$$

If I had just taken whatever is the original vectors let us say I have some 2, 3 and 4, 5 so instead of doing this basis expansion and then computing the inner product I can just take the inner product right away. This is essentially what the answer would be so this well for degree 2 it might not seem great what about degree 15 polynomial? I have essentially doing similar amounts of computation except that I have to rise something to the power of 15.

That is basis expansion if you thought something else about basis expansion please correct it this is basis expansion. So I take the original data and then since I said you could have a new component set or $\sin x \cos x$ mean does not matter right you could think of variety of different ways of expanding the bases in this case I am just doing the quadratic basis expansion.

So whatever we have done so far and so this whole idea for kernel and other things are arriving rather straightforward so what I cannot write now for you is what is the basis expansion for the RBF kernel it turns out that the computation is doing is actually in an infinite dimensional vector space okay so here the computation is a six dimensional space and I took some data point from a two dimensional space computation in a six dimensional space right. And I gave you back the answer but all the time doing computation only in a two-dimensional space and I only took the inner product of these two and then added 1^2 so I am essentially doing computations only R^2 . Well the actual number I am returning to you is the result of computation done in R^6 that is why it is called the kernel trick. So likewise the RBF kernel I will do something in whatever is the original dimensional space you give me but the resulting computation has an interpretation in some infinite dimensional vector space case it is not even easy to write it down so that is why the RBF kernel powerful they work on a variety of data right but they are not all powerful this have to be careful about it right so, so that is all there is to support vector machines so we have done this support vector machines as well.

So I don't know if people who have used libsvm or one such tool for that for most RBF kernels you would have to tune two parameters one is C which we already saw that is essentially how much penalty you are giving to the thing other one you will tune is γ essentially this right it is some kind of a width parameter for your Gaussian this how wide you are Gaussian is it just it is controls that so that is γ so those are the two parameters you tune and for polynomial kernels you have a d and you have your C right and for sigmoidal kernels you have constants k_1 and k_2 and you have C and this form of defining a support vector machine is called as C-SVM okay.

There are other ways other constraints that you can impose on it not just the penalty on the ξ 's you can impose penalty on the number of support vectors you consider right you want to say so suppose I run the data and it comes back and says okay everything is a support vector right so that is not something interesting how can everything be a support vector can all the data points

be equal distance from the separating hyper plane not if you are considering linear but when I am considering RBF kernels right the separating hyper plane can be very very complex right.

So in which case you might end up with a lot of support vectors typically I do not know if you have not thought too much about it and you are setting some very high values for C and trying to run this thing you will end up with like sixty percent of your data as being support vectors so instead of trying to do that empirically second on why only 20 support vector so let me try different see differential γ and so on so forth you can use something called the nu-SVM not new but nu-SVM which gives you a upper bound on the number of support vectors you are going to get you can say do the best you can but do not give me more than 30 support vectors something like that to that effect.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

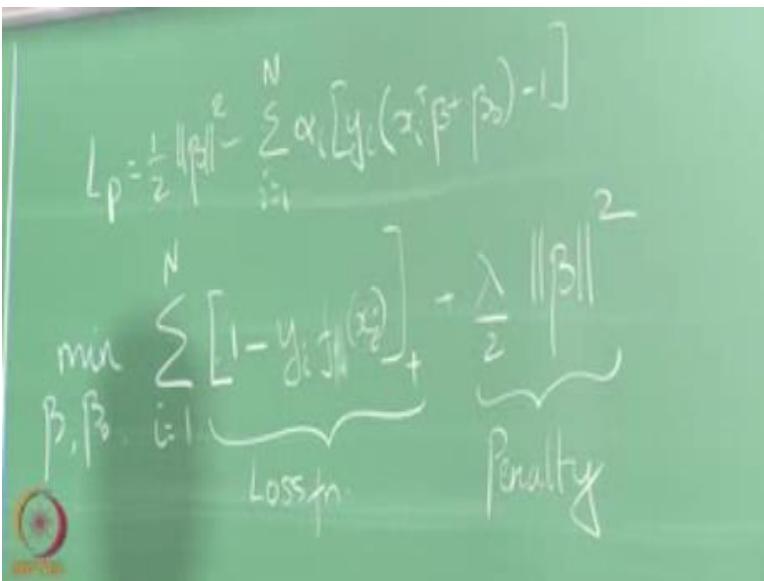
Introduction to Machine Learning

Lecture 31

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

**Hinge Loss Formulation Of the
SVM Objective Function**

(Refer Slide Time: 00:19)



The image shows a handwritten derivation of the SVM objective function. It starts with the formula for the primal loss function:

$$L_p = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i(\beta^T \beta_0) - 1]$$

Below this, the full SVM objective function is shown:

$$\min_{\beta, \beta_0} \sum_{i=1}^N [1 - y_i(\beta^T \beta_0)]_+ + \frac{\lambda}{2} \|\beta\|^2$$

The term $[1 - y_i(\beta^T \beta_0)]_+$ is labeled "Loss fn" and the term $\frac{\lambda}{2} \|\beta\|^2$ is labeled "Penalty".

Okay so people remember the primal objective function that we had for SVM's. So this is a primal objective function we had for SVM's. So one way of thinking about it is to say that I am going to write it the following way maybe some jugglery so the α 's I have replaced it with a λ here okay and well you know $x_i^T \beta + \beta_0$ is actually $f(x_i)$ and then essentially the same objective function except for this plus thing here.

So what is a plus thing?

(Refer to slide time: 01.00)

$$L_p = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i (\mathbf{x}_i^\top \beta + \beta_0) - 1] \\ \sum_{i=1}^N [1 - y_i f_i(\mathbf{x})]_+ + \frac{\lambda}{2} \|\beta\|^2$$



It means that I will call this only whenever this is positive , whenever it is negative I will read it as 0. Does it make sense? I will count this only whenever it is positive, whenever it is negative I will make it I will consider it as 0. So that is what the plus term here indicates they went into λ , I mean I am kind of redid this thing, so I divided everything by some factor of α and moved to λ . So if you stop a minute this should look familiar to you. What does it look like? Ridge regression. In ridge regression so you have a loss function and you have a penalty term and doesn't it look like that? So far we have been talking about $\|\beta\|^2$ as being the objective function that you are trying to minimize and the other thing is constraints and then we then wrote the Lagrangian and then we got the constraints into the objective function. So now I am saying you can think of another way of writing the objective function which is to say that there is this loss function right which is accounted whenever it is negative, so now your goal is to minimize this.

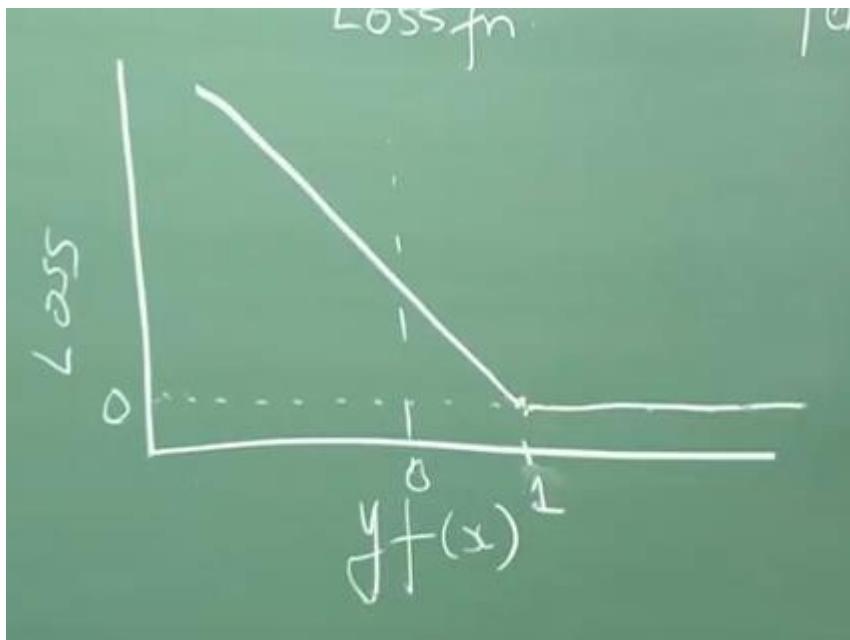
(Refer Slide Time: 03:28)

$$L_p = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i (\beta^\top \beta_0) - 1]$$

$$\min_{\beta, \beta_0} \sum_{i=1}^N \underbrace{[1 - y_i f(x_i)]_+}_{\text{Loss fn.}} + \underbrace{\frac{\lambda}{2} \|\beta\|^2}_{\text{Penalty}}$$

So how will this loss function look like? So when $yf(x)$ is one, after that it will be 0. We talked about loss functions and not about the penalty term but till $yf(x)$ becomes one it is going to be a linear function. You can see that it is just $1 - yf(x)$ so it is going to be a linear function of $yf(x)$. This kind of a loss function where this is like a door or a book opening on a hinge right if you think about it this is like two flaps of a book or a door right and it is opening on the hinge which is here right. So it is also called hinge loss. So sometimes if you have read about SVM's elsewhere you might have heard that the SVM's minimize hinge loss right so this is exactly what we are doing here so the hinge loss actually arises from the constraints, that we are imposing on the SVM right but if you think about it whether the constraints come from why were the constraints imposed? What is the semantics of the constraint? What was that we wanted to make sure? That they are correct and a certain distance away right that is the reason for this.

(Refer to slide time 05.20)

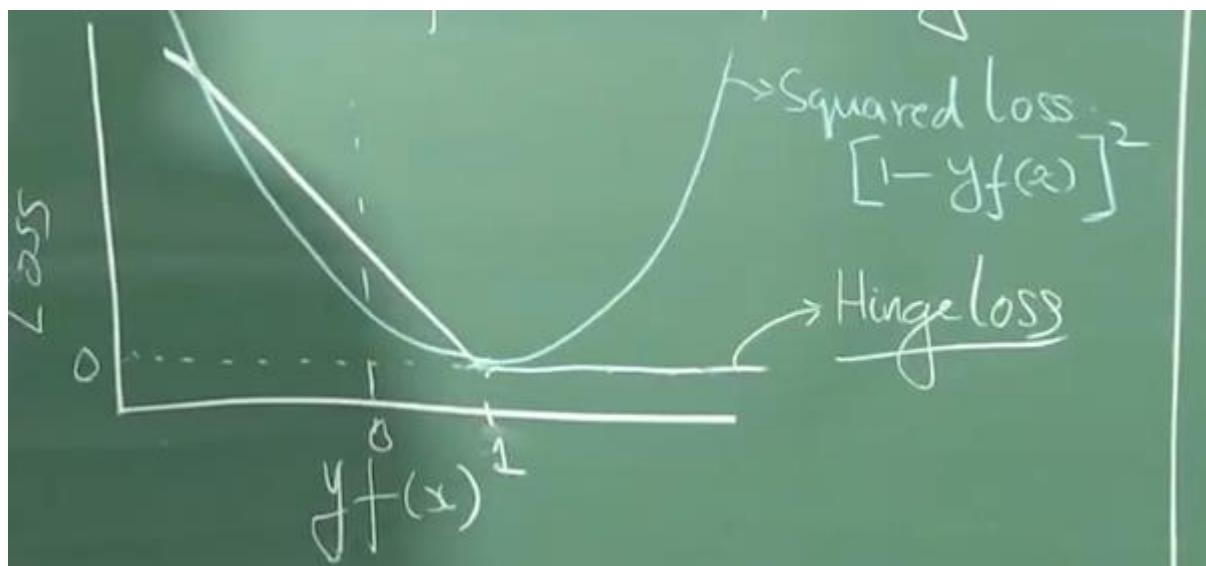


So in effect the constraints are enforcing the correctness of the solution and what the objective function originally was enforcing was essentially the robustness of the solution how far away are you from the hyperplane. The constraints were making sure that you are on the right side of the hyperplane and if you think about it so in effect the constraints are an important part of what you are trying to optimize it is just not the distance from the hyperplane that matters but it is also matters that you should be on the right side of the hyperplane right.

So the putting it is a hinge loss makes it explicit and I am saying okay this is the loss function I am interested in right so that essentially tells me I am interested in the correctness I want to make sure that all my data points are correctly classified and the penalty tells me okay make sure it is a small norm solution it essentially becomes like Ridge regression. You make sure that the squared loss is as little as possible at the same time make sure that the norm of the solution is also small. So that is what we did. We enforce the L2 norm in the ridge aggression case and we are doing the same thing in the SVM case okay does it make sense now? We can ask interesting questions like okay if I replace this with some other norm penalty what will happen can you do L1 regularized SVM's no that was regression so L1 regularised regression was LASSO so can you do like loss so like regularization for SVM's since the β^2 if you put β what happens what do you think will happen?

Well you will have a much harder optimization problem on your hand but it is actually a valid thing, so what it will try to do if you remember we talked about this in LASSO I did in a admittedly a little hand wavy fashion but we talked about how it will enforce sparsity. We said it will try to make as many coefficient 0 as possible, so in this case what do you think will happen? If I put norm can attend for sparsity will it reduce the number of support vectors does that enforce sparsity think about it. What is that? Now the squared loss is actually like this if you think about it is little weird, so if you are to this side you are actually correct but, the further away you are from the hyperplane on the right-hand side also you still contribute to the loss because of this squared error function whether you are on the right side of the wrong side of the hyperplane you still contribute to the loss. So that is why sometimes the squared error function is not the ideal thing to minimize. So the hinge loss more often than not gives you a much better solution than optimizing squared error. So what will the squared error be?

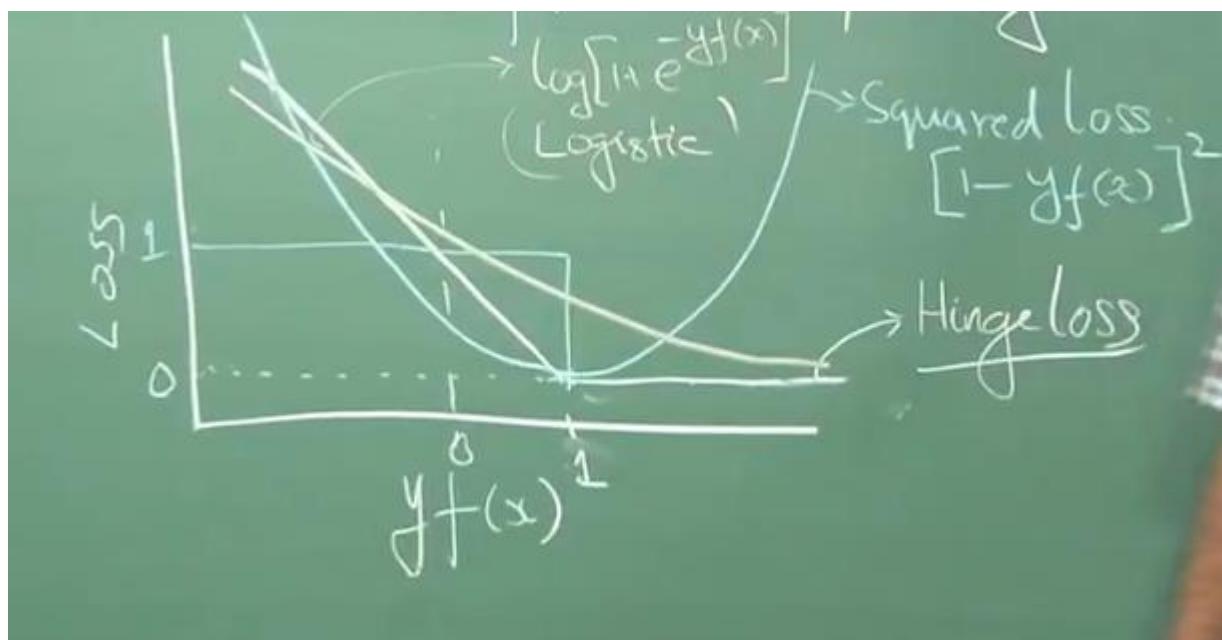
(Refer to slide time 10.38)



Right that is what the square loss function is so normally you are used to seeing this as $(y - f(x))^2$ but I have written it as $1 - yf(x)$ that is also fine because if it is correct $yf(x)$ will be one all the time. So what is the actual loss function that you want? This is fine that is the actual loss function you want. What is the loss function called? 0-1 is what you really want it should be 0 if it is correct and it should be one if it is incorrect at 0-1 is what you really want and a lot of this just like a segue I am not really not going to test you or anything on it just for your interest a lot of work in theory in machine learning goes into showing that if you optimize some other loss

function will end up with the same solution as if you optimize the 0-1 loss. So if you take the 0-1 loss I try to find a solution for it, I am trying to find the β that gives me the smallest possible 0-1 loss. It is as small as possible 0-1 loss 0 depends on the data and you say linearly separable. But why because you chose to use a linear classifier. So depending on what family of classifiers you choose and the and the data. The minimum 0-1 loss could be 0 or it could be something higher, so you say minimizes 0-1 loss I mean whatever is the minimum possible achievable given the data distribution and the class of I mean the class of classifiers of the family of classifiers your chosen given that what is the minimum achievable will you be somewhere close to that. If I minimize a different loss function. So that is interesting question to ask right so I can arbitrarily come up with other loss functions I can come up with hinge loss or a squared loss so if you minimize hinge loss or squared loss will I get the same solution as I would have gotten if you had minimized 0-1 loss? So that is something people do think about right so we did look at one other loss function which is the I guess it goes something like that so that is what we minimize actually in the logistic regression case.

(Refer to slide time 13.48)



Even though we did not write it out explicitly as a loss function right so if you think about it this is what we actually minimize in the logistic regression case. Also you are trying to what were we trying to do? To estimate parameters it is maximum likelihood, so we made some assumptions

about the distribution and then we try to maximize the likelihood and so on so forth right so if you work through that you can write it out as a loss function. It turns out that this is what you are trying to so you can see that this never goes to 0. This is going to go like this okay but then you can still think of minimizing that so we will just an aside you do not have to worry about the logistic loss function right now will come back to that later.

IIT Madras Production

Funded by

Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

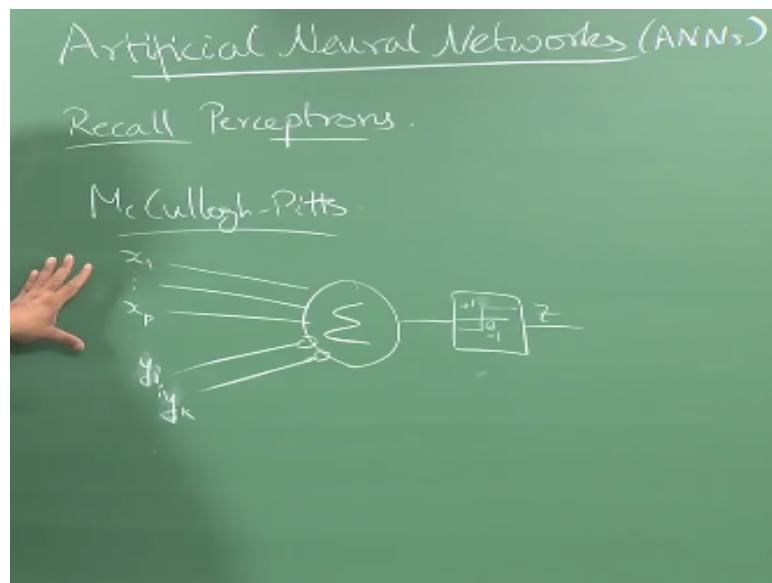
Introduction to Machine Learning

Lecture 32

**Prof. Balaraman Ravindran
Computer Science and Engineering
Indian institute of technology**

**Artificial Neural Networks I –
Early Models**

(Refer Slide Time: 00:14)



Okay so we have been having discussions about neural networks off and on right, the last time we had a discussion about ANNs was went and we did perceptions right. So, basically so the whole class of solution methods which are lumped under ANNs are artificial neural networks were primarily inspired by trying to emulate the brain architecture right yeah, so then after a while the field split into two right one class of researchers who are looking at neural networks as just computing elements right trying to interpret it in terms of linear algebra right and partial analysis and other mathematical tools and then trying to understand what computing these artificial elements were doing.