

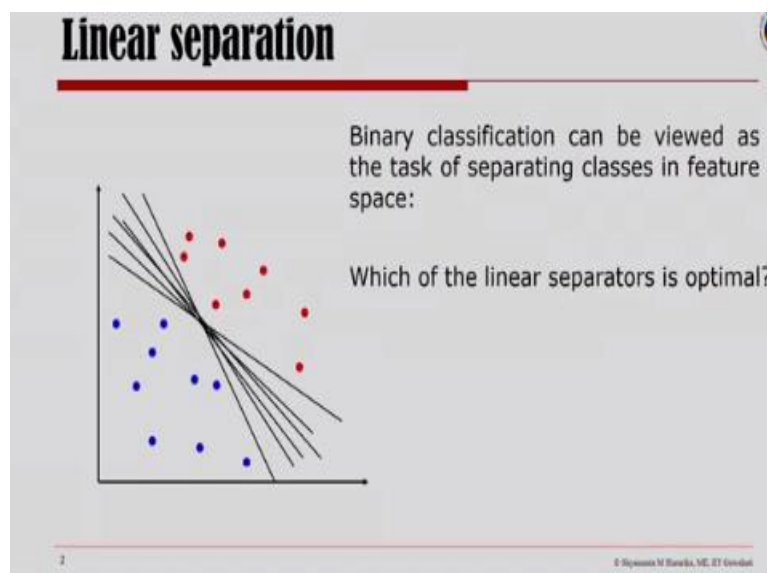
Fundamentals of Artificial Intelligence
Shyamanta M. Hazarika
Department of Mechanical Engineering
Indian Institute of Technology - Guwahati

Lecture – 30
Support Vector Machines

Welcome to fundamentals of artificial intelligence, we continue our discussion on machine learning. Today, we look at support vector machines within machine learning, support vector machine is a supervised machine learning algorithm that is used in classification and regression analysis. Support vector machines allow creation of hyper planes within a group of inputs.

Given a set of training examples which are labelled in one of 2 categories, a support vector machine learning algorithm is able to build a model, so as when new examples come, it could be categorized into 1 of the 2 categories. A support vector machine is a non-probabilistic binary, linear classifier. Support vector machines can also efficiently perform nonlinear classification by what is called the kernel trick that is by a mapping of the inputs to a higher dimensional space where it is linearly separable.

(Refer Slide Time: 02:38)



We look at the basic formulation of a linear support vector machine in this lecture. So binary classification can be viewed as the task of separating classes in feature space like what is shown on your screen here are 2 group of inputs, the blue dots and the red dots and we are

looking for some form of separation or a decision boundary that can be drawn between them. So we could have a number of decision boundaries as shown here.

So, we could have one those are very close to the blues, one those are very close to the reds or ones that lies somewhere in between. Question is which of the linear separators is optimal and this is what the support vector machine algorithm tries to figure out.

(Refer Slide Time: 03:42)

Classification Margin

Have some negative examples; positive examples.
Let the RED be the +ve and BLUE be the -ve.

Line with a view toward putting in the **widest street** separating the positive from the negative samples.

Two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible

Examples closest to the hyperplane are **support vectors**.

How do you divide the positive examples from the negative examples?

1 © Sreyas N. Suresh, MS, ET, Stanford

So, if we think of the red to be the positive examples and the blue to be the negative examples, question is how do we divide the positive and the negative examples and the idea in support vector machine is to have a separation using a hyper plane, where the width would be as maximum as possible. So, we are thinking of a line in between these examples with a view towards putting in the widest street separating the positives from the negative examples.

So, if we are thinking of putting in a line here which separates the positives and the negatives, we want that we should be able to put in as large or as wide a street as possible here. Now, we are thinking of having 2 parallel hyper planes that separate the 2 classes and the idea is that the distance between them is as large as possible and here the examples that are closest to the hyper planes like this one here and this one here are referred to as the support vectors and this is where it gets its name from.

The margin of the separator is the distance between the support vectors and our idea of getting a classification margin here is to get as wide a margin as possible.

(Refer Slide Time: 05:53)

Classification Margin

w – Vector constrained to be perpendicular to the ‘median’.

u – Vector that points to an unknown.

Whether the unknown is on the +ve side or on the -ve side?

- ✓ Project **u** on **w** to that is perpendicular to the separator. Reflects distance in this direction!
- ✓ Further out we go, the closer we'll get to the positive samples.

$$(\mathbf{w} \cdot \mathbf{u}) \geq c$$

© Stephen M. Draxler, MS, IT Overkill

So, while doing that we want to figure out what would be the widest street that we can manage in between these positives and negative examples and therefore, first we assume a vector which is constrained to be perpendicular to the median of the widest street possible. So, here is W which is the vector constrained to be perpendicular to the median and any unknown point, we will refer to by a vector u .

So, u is a vector that points to an unknown, whether the unknown is on the positive side or on the negative side, how does one make that conclusion now, in order to look at that what is to be done is; we project the unknown vector u onto w which is the vector constraint to be perpendicular to the median. So, we take u and let it fall on w , so we project u on w and then what actually that means is now we have some form of a measure of the distance of that unknown sample in the direction of w , which is perpendicular to the median of the street.

So, in a sense the larger we have that value farther out we go and the closer we will get to the positive samples, so even if we do not have a clear idea of whether u refers to a positive or a negative sample, taking the projection of u on w and having some idea on what is that distance in the direction of w , we will be at least having some notion of whether it is closer to the positives or it is closer to the negatives. And if we assume a particular constant c and somehow the dot product $w \cdot u$ is greater than or equal to c , then we can think of it to be nearly reaching the positives.

(Refer Slide Time: 08:54)

Classification Margin

Lot of w 's that are perpendicular to the median line; because it could be of any length.

What constant to use?

Classify unknown u as plus if

Decision Rule.

$$f(u) = w \cdot u + b \geq 0$$

Don't have enough constraint here to fix a particular b or a particular w .

w has to be perpendicular to the median line.

© Stephen H. Karickhoff, M.S., Ph.D.

So, without loss of generality what we can write is; we can write an equation saying instead of c now, we replace with a b and we say if I have $w \cdot u + b \geq 0$, then I am going to classify the unknown u as plus and this could be our decision rule. So, let me repeat what our decision rule is and how do we go about getting to that. First, we think of a vector w that is a perpendicular to the median of the street that we want to fit in between the positives and the negatives.

And u is a vector that is only pointing to an unknown now, we project u on w and we say $w \cdot u$ plus some constant b , if it is greater than or equal to 0, then we are on this side which means that the unknown u is a positive sample however, this decision rule that we have stated we do not have enough constraints here to fix a particular b or even a particular w for we could have many w 's perpendicular to the median line, for we have not committed to what could be the length.

So, we could have lot of w 's that are perpendicular to the median line because it could be of any length further, we have no idea of what constant b to use and therefore, we need to look for more constraints, so as to really make our decision rule work.

(Refer Slide Time: 11:08)

Classification Margin

✓ For a positive sample, insist that the decision function gives the value of one or greater.

✓ For a negative sample, insist that the decision function gives the value of equal to or less than minus 1.

Classify unknown \mathbf{u} as plus if

$$f(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u} + b \geq 0$$

✓ Constrain

For all plus sample vectors:

$$f(\mathbf{x}_+) = \mathbf{w} \cdot \mathbf{x}_+ + b \geq 1$$

For all minus sample vectors:

$$f(\mathbf{x}_-) = \mathbf{w} \cdot \mathbf{x}_- + b \leq -1$$

© Sreyas M Banerjee, MIT EE-6.034

What is done is we say that for a positive sample, the decision function gives us value of 1 or greater that is something we are going to insist and we say that for negative samples, the decision function gives us value of equal to or less than - 1, so all plus sample vectors now we have and then if all plus sample vectors refers to as x plus, then w dot x plus which is the positive sample vector, plus b is ≥ 1 . So, all positive samples we insist that the decision function gives a value which is 1 or greater and for all negative samples we write as w dot x of $- + b$ is ≤ -1 , so this is the constraint that we had. So, the constraint and forces that any vector that comes to the positive side here, if it comes here then I am going to have $+ 1$ or if it stays in the negative samples, then I am going to have less than or equal to $- 1$.

(Refer Slide Time: 12:57)

Classification Margin

Introduce a variable y_i

✓ $y_i = +1$ for pluses
-1 for minuses

Multiply each equation by the variable y_i .

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0$$

For all points on the boundary

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0$$

© Sreyas M Banerjee, MIT EE-6.034

So, these are the functions on the two sides of the hyper plane that I want to fit in or the widest street that I want in between the pluses and the minuses, so now we introduce a

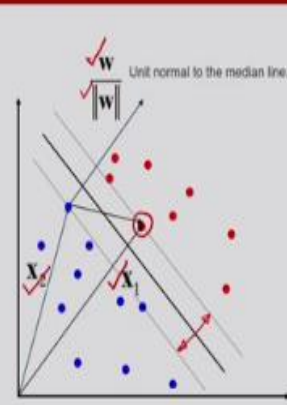
variable y_i , this variable is so defined that y_i is assumed to be + 1, for all the plus samples and - 1 for all minus samples. So, for anything this side of these streets, I consider y_i to be + 1 and on this side of the street y_i is considered to be - 1.

Now, once we have this y_i , the variable and we multiply each equation by the variable y_i , we have a very interesting thing happening here; let us look at what that is? If you multiply this side by + 1 because here we are talking of the plus samples and for the plus samples, we know y_i is + 1, so we try to multiply this with + 1 and once we multiply this with + 1, you will realize that this very fact that I multiply this by + 1, I have $x \cdot w \text{ dot } x_i + b \geq 1$. And if I multiply this by - 1, I actually have the same equation, for this portion that I multiply the variable with this becomes minus and this becomes plus, so I will end up having the same equation which is $y_i w \text{ dot } x_i + b \geq 1$ and if I take 1 this side, this is the equation that I will end up with. So, multiplying by the variable y_i which we have introduced +1 for pluses and - 1 for minuses, end up reducing the 2 equations that I have to 1.

And now as a special case, if I am talking of points only on the boundary, I would rather equate this to 0, so for all points on the boundary what I have is y_i multiplied by $w \text{ dot } x_i + b - 1 = 0$, so these are points on the boundary.

(Refer Slide Time: 16:13)

Classification Margin



The dot product of the unit normal and the difference vector, would be the margin.

Given the constraints

$$y_i(w \cdot x_i + b) - 1 = 0$$

Width?

$$w \cdot x_1 + b = +1$$

$$w \cdot x_2 + b = -1$$

Width of the margin

$$\frac{w}{\|w\|} \cdot (x_1 - x_2) = \frac{2}{\|w\|}$$

Now, given the constraint that I have now got by first projecting the unknown along w , introducing the fact that anything on the positives must be greater than 1, anything on the negatives would be less than - 1 and then introducing a variable y_i which is + 1 for pluses and - 1 for minuses, I finally arrived at this constraint equation. Now, given this constraint what is

the width that I can get on the classification margin? Now, in order to get this we again fall back to looking at the support vectors or the samples that lie on the boundary, so first we take a vector x_1 which refers to a point which is a positive sample on the boundary and then we look at x_2 , a vector that goes to a negative sample lying on the boundary and then we can do the difference of these 2 vectors. So, we have $w \cdot x_i + b$ which is $+1$ that is the equation because of the vector x_1 .

And $w \cdot x_2 + b = -1$ because of the vector x_2 and now when we look at their difference and then we project that along the unit normal to the median line that is you remember, I had w which was the normal to the median line and if I divided by the magnitude of w , I would have the unit normal to the median and now, if I take the difference and the dot product of the unit normal, so I have the difference $x_1 - x_2$.

And I take the dot product with the unit normal, what I have actually is what is this length in this direction, so, basically I have the width of the margin. So, this values here that I end up with which is the difference of these 2 vectors x_1 and x_2 of the two samples lying at the two boundaries, when I take its projection with the unit normal to the median, I end up having the width of the margin.

Now, if you look at it more closely, this difference here actually, $w \cdot x_1 - x_2$, if you go back and look at it here for the difference between the 2 vectors $x_1 - x_2$, then what I will have is actually, these b 's, b 's going out and I will have 2 here, when I do $x_1 - x_2$, so this is interesting that the width of the margin is 2 divided by the magnitude of the normal that I had first taken to be normal to the median line.

So, once I have this and now, if I want this width to be maximum so, all I need to look for is what would be the value of w , so that this becomes maximum for me.

(Refer Slide Time: 20:59)

Linear Support Vector Machine



- To maximize the width of the separation, one need to minimize w , while still honoring constraints with regards to the values on the edges of the separation.
- One possible approach to finding the minimum is to use the method devised by Lagrange.
- Maximizing the width is ensured if one minimize the following, while honoring constraints on edge values.

Lagrange multipliers: would give us a new expression, which we can maximize or minimize without thinking about the constraints anymore.

$$\sqrt{\frac{1}{2}} \|w\|^2$$

Translation of the previous formula into this one, with $\frac{1}{2}$ and squaring, is a mathematical convenience.

So, to maximize the width of the separation, one would need to minimize w , while still honouring the constraints with regards to the values on the edges of the separation and one possible approach to finding the minimum now is to use the method devised by Lagrange. So that Lagrange multipliers would give us a new expression which we can maximize or minimize without thinking about the constraints anymore.

So, maximizing the width is ensured if one minimizes w while honouring constraints on edge values, so all we look for minimizing is a term $\frac{1}{2} w$ square, so the translation of the previous formula into this one with $\frac{1}{2}$ and squaring is just a mathematical convenience, one needs to realize that from the previous discussion, we can maximize the width of the separation by minimizing w .

And because it comes with certain constraints, we follow the method devised by Lagrange and take help of Lagrange multipliers to get to a new expression, which we can maximize or minimize without thinking about the constraints anymore.

(Refer Slide Time: 22:43)

Linear Support Vector Machine

□ To minimize $\frac{1}{2}\|\mathbf{w}\|^2$ subject to the constraint
 $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0$

□ Find where the Lagrangian has zero derivatives

$$L = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^l a_i (y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1)$$

i.e., $\frac{\partial L}{\partial \mathbf{w}} = 0$ and $\frac{\partial L}{\partial b} = 0$

11

© Giuseppe N. Bascià, ME

So, now for the maximum margin for the hyper plane that we are thinking of putting in between the positives and the negatives, we have to minimize 1/2 w square subject to the constraint that $y_i \mathbf{w} \cdot \mathbf{x}_i + b - 1 \geq 0$. Now, the Lagrangian is written using Lagrangian multipliers and we arrive at new expression without constraints, so each of those constraint is going to have a multiplier alpha i.

Basically, we write the Lagrangian L as 1/2 of w square minus, now each of these sample points i which have to satisfy the constraint is multiplied by a Lagrangian multiplier which is alpha i and then we have this whole equation, which is the Lagrangian. Now, in order to realize the extremum's, we need to find where the Lagrangian has zero derivatives and therefore, we look at partial derivative of L with w equal to 0.

(Refer Slide Time: 24:30)

Linear Support Vector Machine

$$L = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^l a_i (y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1)$$
Decision vector is a linear sum of the samples.

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^l a_i y_i \mathbf{x}_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^l a_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^l a_i y_i = 0$$
Find a maximum of this expression.

$$L = \sum_{i=1}^l a_i - \frac{1}{2} \sum_{i,j=1}^l a_i a_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

Optimization depends only on the dot product of pairs of samples.

12 © Giuseppe N. Bascià, ME, IT Graduate

And partial derivative of L with b equal to 0. So if you derive the Lagrangian with w, so partial derivative of L with w, we have w here because of the first term and here w is only here, the rest of them are constant, so all will be left, it is summation alpha i y_i x_i, so this is that term and this term will vanish. So, will this final term here and now interestingly, what we have is what w is, so we are now in a position to get to w being the linear sum of the samples.

For, if you remember alpha i is a Lagrangian multiplier, y_i is a variable that we have introduced based on whether we are doing plus or minus and x_i is the sample, so this decision vector w is a linear sum of the samples and when I do partial derivation of L with b, this goes away all that I have is here, b there and therefore, I have summation of a_i y_i = 0. Now, once I have this y_i can go and plug in this w here as also here.

And this I have not covered in this lecture but once you plug in there and take care of not to mess up the subscripts, we will finally arrive at the expression for L as shown on your screens, so plugging in w into this L, will finally end up giving us this equation for L here that is what we will arrive at. So, finally we have to find a maximum of this expression in order to get the maximum width for the boundaries.

Now, this optimization is interesting to note for the very fact that the optimization only depends on the dot product of a pair of samples and this is a very, very interesting realization.

(Refer Slide Time: 27:29)

Linear Support Vector Machine

- Recall that the decision rule was to
 - Classify unknown \mathbf{u} as plus if

$$f(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u} + b \geq 0$$
- With $\mathbf{w} = \sum_{i=1}^l a_i y_i \mathbf{x}_i$ the decision rule is
 - If $\sum_{i=1}^l a_i y_i \mathbf{x}_i \cdot \mathbf{u} + b \geq 0$ then classify \mathbf{u} as plus.

✓ The decision rule, also, depends only on the dot product of the sample vectors and the unknown.

13 © Sreyas N Banik, M.E. IT Guwahati

Now, let us see what the decision rule is; recall that the decision rule was to classify unknown u as plus, if we had $f(u) = w \cdot u + b \geq 0$, we are going to plug in the value of w that we have got into this equation. So with w equal to the summation of the samples now, the decision rule is here so, we see that the decision rule to classify any unknown u as plus again depends only on the dot product.

So, the decision rule also depends on the dot product of the sample vector and the unknown. Now, as I was referring to in the beginning of this lecture that support vector machines are actually binary linear classifiers.

(Refer Slide Time: 28:58)

Nonlinear Classification

□ General idea: the **original feature space** can always be mapped to some **higher-dimensional feature space** where the training set is separable:

Impossible to draw a line in the 2D plot which could separate the blue from the red samples. Is it still possible for us to apply SVM algorithm?

Datasets that are NOT linearly separable.

14 © Arjun Gupte, MIT OpenCourseWare

But then it can also deal with nonlinear classification like here is a data set, which is not linearly separable that is it is impossible to draw a line in the 2D plot which could separate the blue from the red sample. Now, the question is under such scenario is it possible for us to apply a support vector machine algorithm and this is where the kernel trick plays a very important role.

The general idea here is that even if the original feature space is not linearly separable, you can always map that feature space to some higher dimensional feature space, where the training set then is separable, so given this feature set which is not linearly separable, I can think of having a function ϕ of x_i , which will try to map every point in this data set to a high dimensional space.

(Refer Slide Time: 30:40)

Nonlinear Classification

- Optimization and also the decision rule, require inner product.
 - $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ for optimization.
 - $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{u})$ for decision rule.
- All that is required is a way to compute dot products in high-dimensional space as a function of vectors in original space!
- A **kernel function** is a function that is equivalent to an inner product in some feature space.
 - $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$

We just need to know K and not the mapping function itself. Doesn't need to know what Φ is: having K is enough!

15

© Vijayasarathi Sivasankar, MIT, EE-6.035

And then it would be possible to have linear separation as shown in this diagram. Now, one thing that we need to take note of at this point is that when we were talking of the linear support vector machine, the optimization as also the decision rule required only the inner product, if you remember all it required was the inner product of the samples for the optimization. And it required the inner product of a sample and the unknown in the decision rule that is if we are even somehow mapping them using phi, then all I will need is phi xi dot phi xj for optimization and I will need phi xi dot phi u for the decision rule. So there is a requirement of only the inner product, so all that is required is a way to compute the inner product or the dot product in the high dimensional space as a function of vectors in original space.

This is very interesting for we have something called a kernel function that is a function that is equivalent to an inner product in some feature space, so basically we could be given a $K(\mathbf{x}_i, \mathbf{x}_j)$, which would be equal to $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, in that case we just need to know K and we need not bother about the mapping function itself. So, we do not need to know what phi is given K to us and this is what the kernel trick is.

You are given the kernel and once you are given the kernel, it means that you are given the inner product in that feature space, once you have the inner product in the feature space we know we do not need the explicit function that has map from one feature space to this feature space that we are currently dealing with.

(Refer Slide Time: 32:52)

Nonlinear Classification

The **kernel trick** avoids the explicit mapping to get linear learning algorithms to learn a nonlinear function or decision boundary.

1. Linear Kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$

Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single line. It is one of the most common kernels to be used. Used when there are a large number of features in a particular data set.

2. Polynomial Kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^n$

Represents the similarity of vectors in a feature space over polynomials of the original variables. Intuitively, the kernel looks not only at the given features of input samples, but also combinations of these.

3. Gaussian Kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$

Gaussian kernel computed with a support vector is an exponentially decaying function in the input feature space, the maximum value of which is attained at the support vector and which decays uniformly in all directions around the support vector, leading to hyper-spherical contours of the kernel function.

So, the kernel trick avoids the explicit mapping to get the linear learning algorithms to learn a nonlinear function or decision boundary and this is very, very important. For now, you can think of taking something in a nonlinear space, a space which is not linearly separable and you can get to a feature space where you could apply SVM but then you would not need the explicit mapping, all you would need is the kernel function. Now, we have a number of kernels I would just mention 3 here for the sake of introduction to the idea of having kernel functions and how they look or what is their behaviour. So we have something called a linear kernel; the linear kernel is the same as what we have discussed, it is just the inner product of x_i and x_j . Now, linear kernel is used when the data is linearly separable that is the data is something like the example that we saw in the lecture today.

We can separate the plus and the minus using a single line now, the linear kernel is one of the most common kernels to be used, linear kernels are also something that can be used when there are a large number of features in a particular data set. The second kernel is called polynomial kernel; so the polynomial kernel is what is shown here, it is 1 plus the inner product of the i th and j th sample whole to the power m .

Now, this polynomial kernel actually represents similarity of vectors in a feature space over polynomials of the original variables that is the kernel actually does not look only at the given features of input samples but also it looks at combinations of this, one of the most popular support vector machine kernels is the Gaussian kernel referred to as the radial basis function.

So, the Gaussian kernel is an exponentially decaying function in the input feature space and the maximum value of this is attained at the support vector and they decay uniformly in all directions around the support vector leading to some hyper spherical contours of the kernel function and the Gaussian function or the radial basis function is one of the most popular kernel functions used in support vector machines.

(Refer Slide Time: 36:25)

Main Ideas

- **Maximum-Margin Classifier**
 - Formalize notion of the best linear separator
 - Best hyperplane is the **one that represents the largest separation, or margin**, between the two classes - distance from it to the nearest data point on each side is maximized.
 - If such a hyperplane exists, it is known as the **maximum-margin hyperplane** and the linear classifier it defines is known as a **maximum-margin classifier**
- **Lagrangian Multipliers**
 - Way to **convert a constrained optimization problem** to one that is easier to solve
- **Kernels**
 - Projecting data into higher-dimensional space makes it **linearly separable**.

17 © Siddhant N. Shrivastava, SE, IIT Bombay

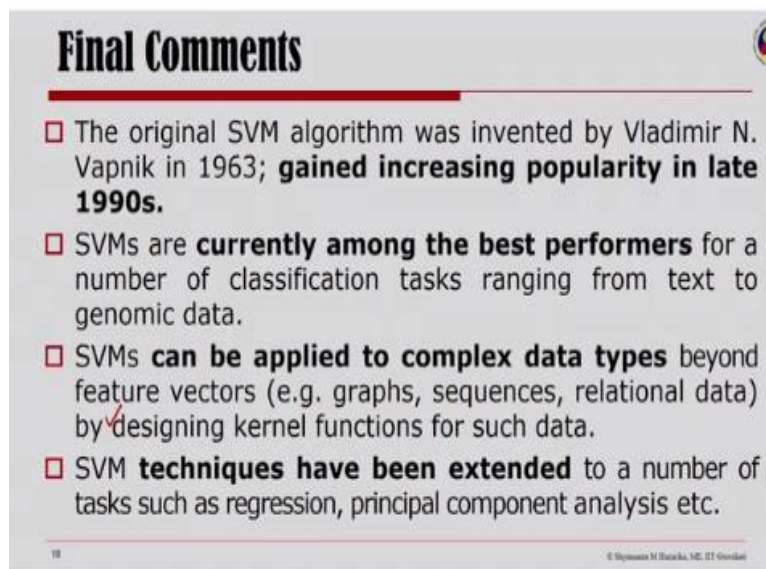
Now, let us quickly recall what is the main idea of the support vector machine that we have covered in today's lecture, we had actually looked at 3 very fundamental ideas today, the first of the idea is about the maximum margin classifier, so we have formalize the notion of the best linear separator. Now, the best linear separator is the one that represents the largest separation or the margin between the 2 classes.

That is when we were looking at the plus and the minus samples, our interest was to get to the distance between the 2 classes and make that distance as large as possible. So we were looking at the distance to the nearest data points on each side to be maximized. In a problem if such a hyper plane exists, then it is known as the maximum margin hyper plane and the linear classifier it defines is the maximum margin classifier.

So, in a way the support vector machine that we were trying to arrive at today is a maximum margin classifier. The second idea, that is very vital and have changed the way we look at optimization is the concept of the Lagrangian multipliers, where we had a way to convert a constrained optimization problem to one that is easier to solve bring in a new expression using the multipliers without having any constraints any longer.

So, the constraints are factored into the new expression. The third of the ideas that we have explored today is the idea of kernels. Now, one needs to take note that the very idea of kernels have changed the way support vector machines have been put to use. As I have explained earlier support vector machines are non-probabilistic binary linear classifiers therefore, when I have problems that are with data which are not linearly separable application of a support vector machine is made possible only because of the kernel trick, which is about projecting data into higher dimensional space and makes it linearly separable. We have looked at the linear kernel, we have mentioned what is the polynomial kernel and also the Gaussian or the radial basis function kernel.

(Refer Slide Time: 39:56)



Final Comments

- The original SVM algorithm was invented by Vladimir N. Vapnik in 1963; **gained increasing popularity in late 1990s.**
- SVMs are **currently among the best performers** for a number of classification tasks ranging from text to genomic data.
- SVMs **can be applied to complex data types** beyond feature vectors (e.g. graphs, sequences, relational data) by designing kernel functions for such data.
- SVM **techniques have been extended** to a number of tasks such as regression, principal component analysis etc.

Now, finally to conclude the original support vector machine algorithm was invented by Vladimir Vapnik in 1963. However for almost 30 years, nobody had noticed this for until someone in the Bell Laboratories found Vapnik's work and Vapnik immigrated to the US and thereafter, he also worked on the kernel trick further. The kernel trick had a mention in his thesis but with the use of kernel trick in big way, the support vector machines gained increasing popularity in the 1990's.

The support vector machines are possibly currently the best or among the best performers for a number of classification tasks which range from text to genomic data. In fact, for anyone who claims to have worked in machine learning or any literate in machine learning, the support vector machine needs to be in his toolbox, support vector machines can be applied to complex data types beyond the feature vectors such as graphs, sequences, relational data by

designing kernel functions for such data and kernel tricks hold huge promise. Support vector machine techniques have also been extended to a number of tasks such as regression, principal component analysis which we have not covered in this lecture, thank you very much.