Module 3 - Generative Modeling II

总结:

3.1 Linear Algebra I: Basic Notation and Dot Products

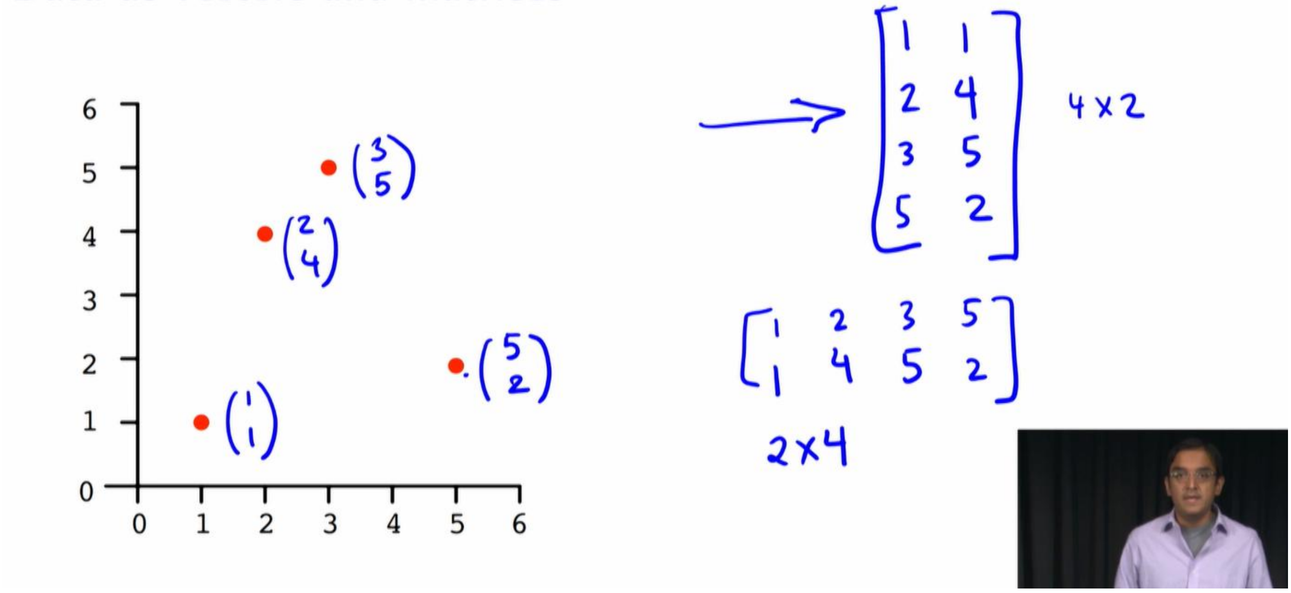
Topics we'll cover

1. Representing data using vectors and matrices
2. Vector and matrix notation
3. Taking the transpose
4. Dot products, angles, and orthogonality

Over the past few lectures, we've been seeing vectorsand matrices more and more. They've gradually been seeping in, and we're not going to be able to stave them off for very much longer. Now I know that many of you have seen linear algebra before at some stage, but just to be sure, I want to do a pretty thorough review of the concepts and skills that we're gonna be needing for this class. Linear algebra is really the most fundamental kind of math for machine learning, so it's something to understand very well.

What we'll talk about today is the basics of representing data using vectors and matrices and some basic notation and dot products.

Data as vectors and matrices



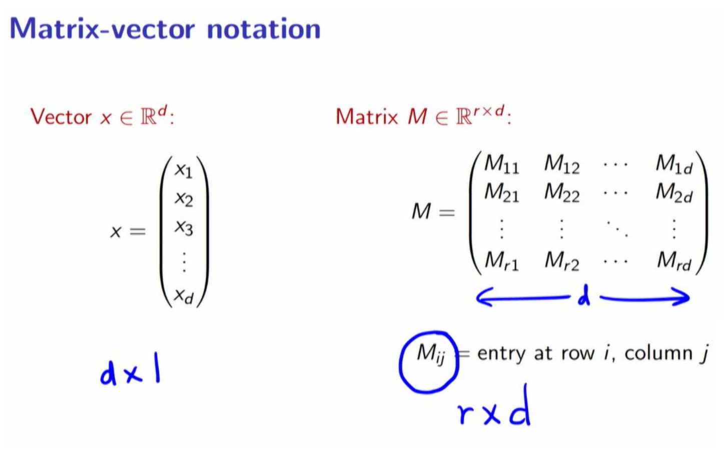
$$\begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 5 \\ 5 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 5 \\ 1 & 4 & 5 & 2 \end{bmatrix}$$

So here's a dataset consisting of four points.

- We can represent each of these points as **vectors**. So for example this point over here is the vector (5,2), so we can write it like this. And this one up here is (3,5) and the one near it is (2,4), two along the x1 axis, four along the x2 axis, and this one is (1,1).
- We can also represent the entire dataset in a single **matrix**. So for example, we could have a matrix in which each data point is a **row**. So something like this. 1-1, 2-4, 3-5, and 5-2. This could be our data matrix, and it's a matrix with four rows and two columns, so it's a four by two matrix. Alternatively, we could put the data points as **columns**. So we could have something like 1-1, 2-4, 3-5, and 5-2. This has two rows and four columns, so it's a two by four matrix. Either one is fine. In this course, we'll mostly be adopting the convention of doing it the first way. So putting the data points as rows.

总结：



Matrix-vector notation

Vector $x \in \mathbb{R}^d$:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix}$$

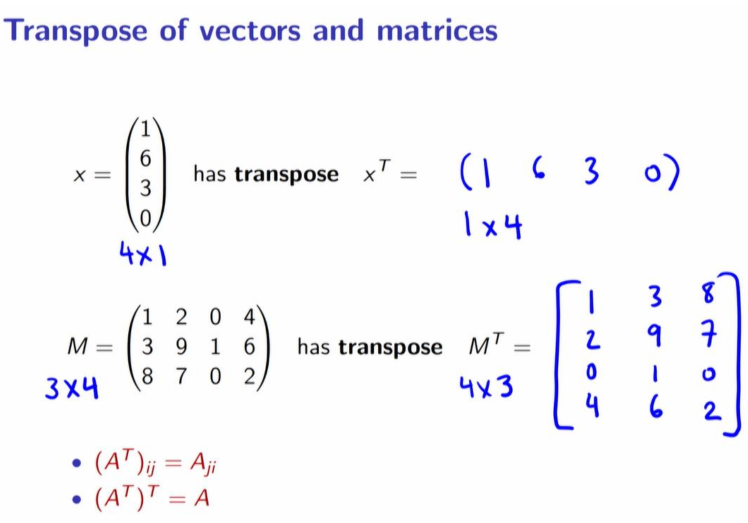
d x 1

Matrix $M \in \mathbb{R}^{r \times d}$:

$$M = \underbrace{\begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1d} \\ M_{21} & M_{22} & \cdots & M_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ M_{r1} & M_{r2} & \cdots & M_{rd} \end{pmatrix}}_{\longleftarrow d \longrightarrow} \updownarrow r$$

$\boxed{M_{ij}}$ = entry at row i, column j

r x d



Transpose of vectors and matrices

$$\underset{4\times 1}{x = \begin{pmatrix} 1 \\ 6 \\ 3 \\ 0 \end{pmatrix}} \quad \text{has } \textbf{transpose} \quad \underset{1\times 4}{x^T = (1 \;\; 6 \;\; 3 \;\; 0)}$$

$$\underset{3\times 4}{M = \begin{pmatrix} 1 & 2 & 0 & 4 \\ 3 & 9 & 1 & 6 \\ 8 & 7 & 0 & 2 \end{pmatrix}} \quad \text{has } \textbf{transpose} \quad \underset{4\times 3}{M^T = \begin{bmatrix} 1 & 3 & 8 \\ 2 & 9 & 7 \\ 0 & 1 & 0 \\ 4 & 6 & 2 \end{bmatrix}}$$

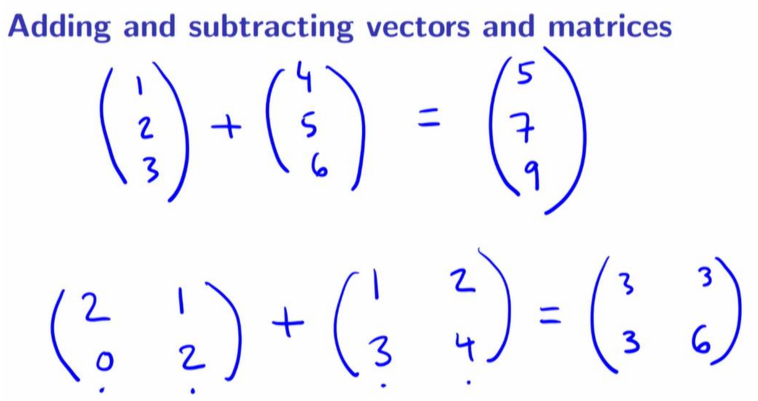
- $(A^T)_{ij} = A_{ji}$
- $(A^T)^T = A$

So let's go over some of the **indexing notation**.

- When we have a vector is d-dimensional space, so vector x in R^d, we'll write it as a column, and we'll index its entries as x1 through x3.
- When we have a matrix that's r by d, that means that it's a matrix with r rows and d columns, so d of these. So it's got a total of r times d entries, and the entry that's at row number i and column number j will be denoted Mij. And we'll say that its dimension is r by d; it's an r by d matrix. So likewise, we can describe the vector as being d by one, d rows and one column.

Now one of the simplest things that one can do with a vector or a matrix is simply to take the **transpose**, which means to switch the rows and columns. So for example, if we look at this vector over here, which is a single column. We take its transpose by just writing it as a row. So the transpose is (1, 6, 3, 0). So we started with something that was four by one and we ended up with something, x transpose, that's one by four. Likewise, we have a matrix over here. It has three rows and four columns, so it's a three by four matrix. And to take its transpose, what we can do is we can just look at the first row of the matrix and write it down as a column, so [1, 2, 0, 4]. And then the second row and write that down as the second column, and the third row and write that down as the third column. And so it's a four by three matrix.

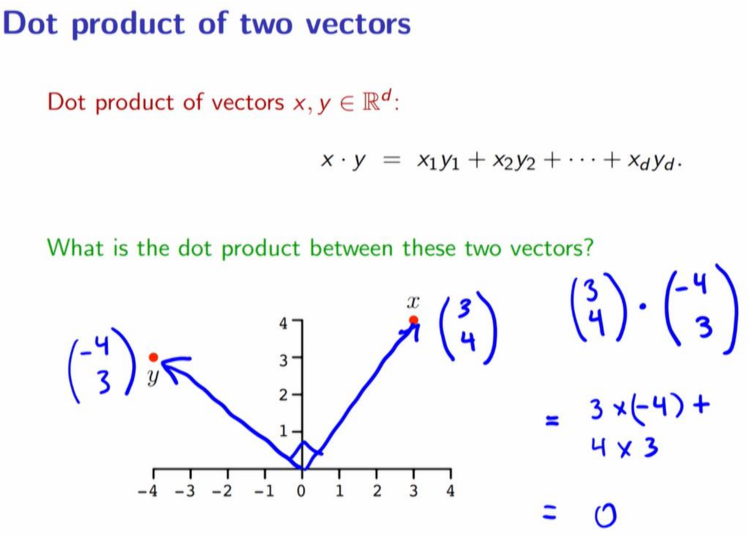
So formally, the transpose, A transpose, is given by the following equation. The ij entry of A transpose is simply the ji entry of the original matrix. Rows and columns are switched. And one thing that's not hard to see is that if you take the transpose and then you take the transpose again, then you end up with the original matrix.



Adding and subtracting vectors and matrices

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 3 & 6 \end{pmatrix}$$



Dot product of two vectors

Dot product of vectors $x, y \in \mathbb{R}^d$:

$$x \cdot y = x_1y_1 + x_2y_2 + \cdots + x_dy_d.$$

What is the dot product between these two vectors?

x
y
$\begin{pmatrix} 3 \\ 4 \end{pmatrix}$
$\begin{pmatrix} -4 \\ 3 \end{pmatrix}$
4
3
2
1
−4
−3
−2
−1
0
1
2
3
4

$$\begin{pmatrix} 3 \\ 4 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ 3 \end{pmatrix}$$

$$= 3 \times (-4) + 4 \times 3$$

$$= 0$$

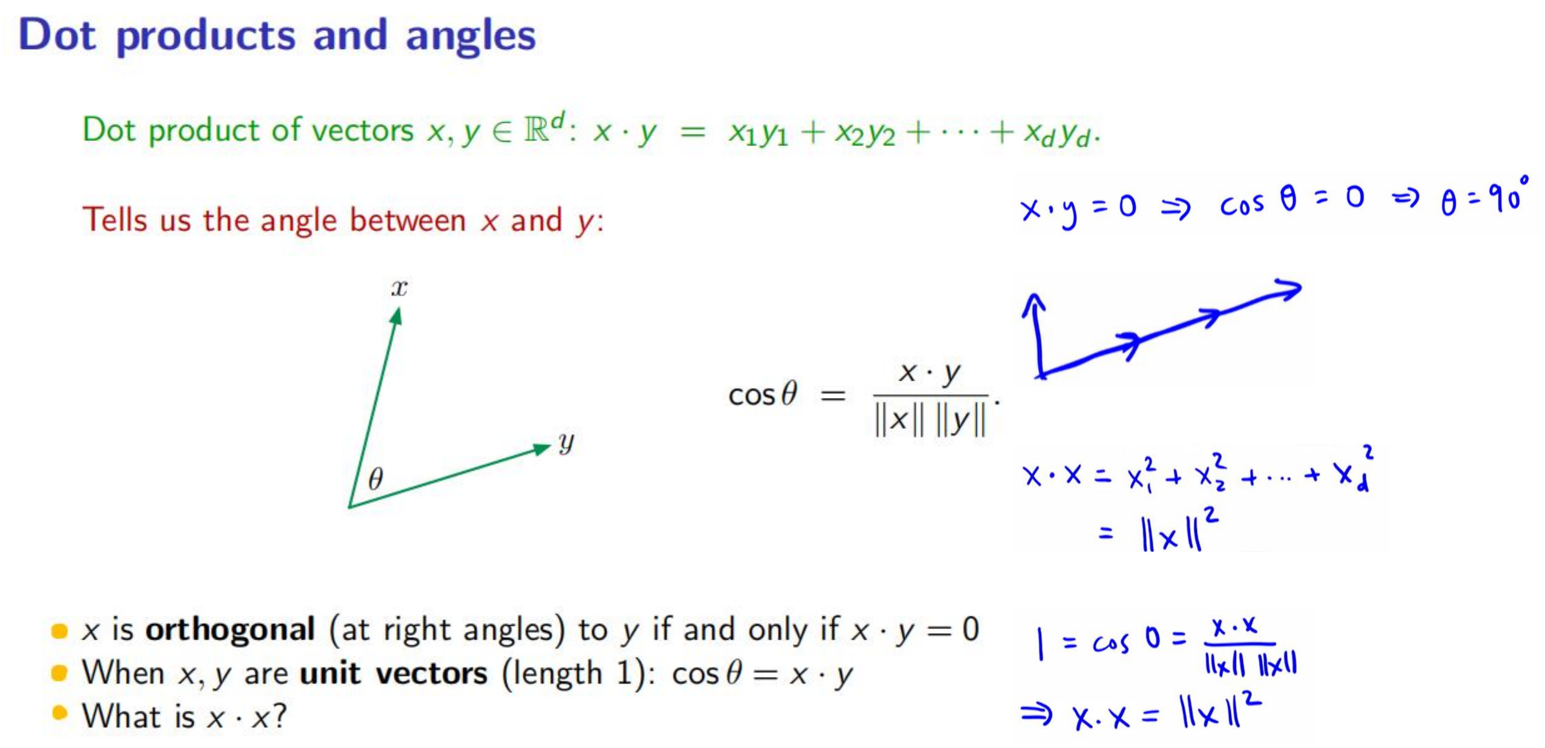
Now how do you add or subtract vectors and matrices? This is very simple. You just **do it element-wise**.

- So if you have two vectors like (1, 2, 3) and (4, 5, 6), the way you add them is just to add them up one element at a time. So one plus four five. Two plus five is seven. Three six plus six is nine.
- And it's similar with matrices. Let's say we have two matrices, [2-1, 0-2] and another one [1-2, 3-4]. We just add them up entry-wise. So two plus one is three. One plus two is three. Zero plus three is three. And two plus four is six.

So adding and subtracting is easy. What's much more interesting is multiplying, and all we'll talk about today is the most basic kind of product, the **dot product** between two vectors, the simplest sort of product. So let's say we have two vectors, x and y in d-dimensions, so they each have d-coordinates. What is the dot product between them? Okay, so we'll denote it x dot y, and to get the dot product, you just multiply the corresponding entries, so the first two numbers, x1 and y1, the second two numbers x2 and y2, and so on, all the way to xd times yd, and you add all of these up. So that's the dot product, and let's see an example.

- Here we have two vectors in the two-dimensional plane. So what are they? x is three to the right and four up, so it's (3,4), and y is minus four to the right and three up, so it's (-4,3). So this is x, and this is y. And what is their dot product? So what is (3,4) dot-producted with (-4,3)? First we multiply the first entries, so 3 times negative 4, and then we multiply the second entries, 4 times 3. So we get negative 12 plus 12, which is zero. So the dot product of these two vectors is zero. Now interestingly, looking at these two vectors, we see that they also seem to be at right angles to each other. So this looks like it's 90 degrees.

总结：



Dot products and angles

Dot product of vectors $x, y \in \mathbb{R}^d$: $x \cdot y = x_1 y_1 + x_2 y_2 + \cdots + x_d y_d$.

Tells us the angle between x and y:

$$\cos\theta = \frac{x \cdot y}{\|x\| \, \|y\|}.$$

- x is **orthogonal** (at right angles) to y if and only if $x \cdot y = 0$
- When x, y are **unit vectors** (length 1): $\cos\theta = x \cdot y$
- What is $x \cdot x$?

$x \cdot y = 0 \Rightarrow \cos\theta = 0 \Rightarrow \theta = 90^\circ$

$$x \cdot x = x_1^2 + x_2^2 + \cdots + x_d^2$$
$$= \|x\|^2$$

$$1 = \cos 0 = \frac{x \cdot x}{\|x\| \, \|x\|}$$
$$\Rightarrow x \cdot x = \|x\|^2$$

Is there some relation between them being at right angles and the dot product being zero? Well, there definitely is. It turns out that **the dot product between two vectors tells you the angle between them**. So here's the formula. So take any two vectors, x and y, so we have x and y, and the angle between them is some number theta. It turns out that the cosine of theta is exactly the dot product of x and y divided by the length of x and the length of y. So for instance, if the dot product is zero, if we have x dot y equals zero like we had before, that means that the cosine of theta is zero. And what angle has a cosine of zero? 90 degrees. We will call two vectors orthogonal if they are at right angles to each other, and this is gonna turn out to be a very important concept for us. We're gonna be using orthogonal vectors a lot. The nice thing is that we know have a simple test for orthogonality. To see if two things are at right angles, just take their dot product and check if it's equal to zero.

Another way in which we'll be using vectors is **to simply indicate directions**. So we might be interested in this direction or that direction. If we're merely indicating directions, the length of the vector doesn't really matter. So it doesn't matter if we used this direction, or this vector, or this vector. They all point in the same direction, and so a common convention when one is only interested in the direction is to normalize vectors to length one, and that's what we call **unit vectors**. So sometimes we'll be working with unit vectors, quite often actually, and when x and y are unit vectors, this formula for the angle becomes especially simple because the denominator is just one. And so we get that the cosine of the angle is simply x dot y.

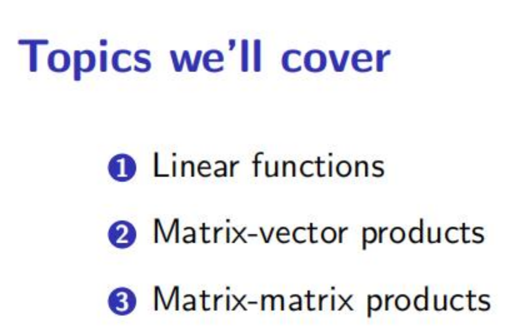
So one last thing that I want to look it in terms of dot products and angles is the following question. What is x dot-producted with itself? What is the dot product of a d-dimensional vector x with itself? So let's do this two different ways.

- So we can do it, we can do it by just plugging it into the formula above. Let's try that, okay? So x dot x is x1(x1), so x1 squared plus x2(x2), x2 squared, all the way to xd squared. Oh. So this is just the length of x squared. It's **the squared Euclidean length of x**. That's x dot x.
- We can also do it using the angle formula. So what is the angle between x and itself? It's zero, and what is the cosine of zero? The cosine of zero is one. So we have one equal to cosine of zero, which is equal to x dot x over the length of x times the length of x, and so this tells us that x dot x is just the length of x squared, which is the same answer we got before.

Okay, well that's it for this first portion of our linear algebra review. We'll be taking things slowly because we really want to get comfortable with this material.

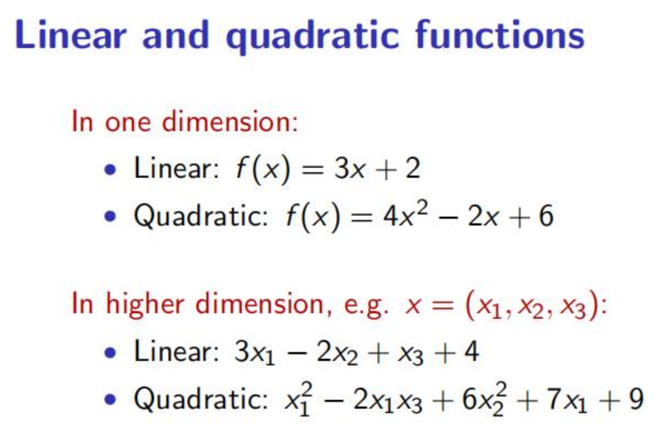
总结：

3.2 Linear Algebra II: Matrix Products and Linear Functions



Topics we'll cover

1. Linear functions
2. Matrix-vector products
3. Matrix-matrix products



Linear and quadratic functions

In one dimension:

- Linear: $f(x) = 3x + 2$
- Quadratic: $f(x) = 4x^2 - 2x + 6$

In higher dimension, e.g. $x = (x_1, x_2, x_3)$:

- Linear: $3x_1 - 2x_2 + x_3 + 4$
- Quadratic: $x_1^2 - 2x_1x_3 + 6x_2^2 + 7x_1 + 9$

We now continue our review of linear algebra. Last time, we looked at the dot product between two vectors. What we'll do today is to build upon this to get products between vectors and matrices and matrices and matrices. This is crucial to modeling linear functions, which are among the most common functions that we'll be using in this class.

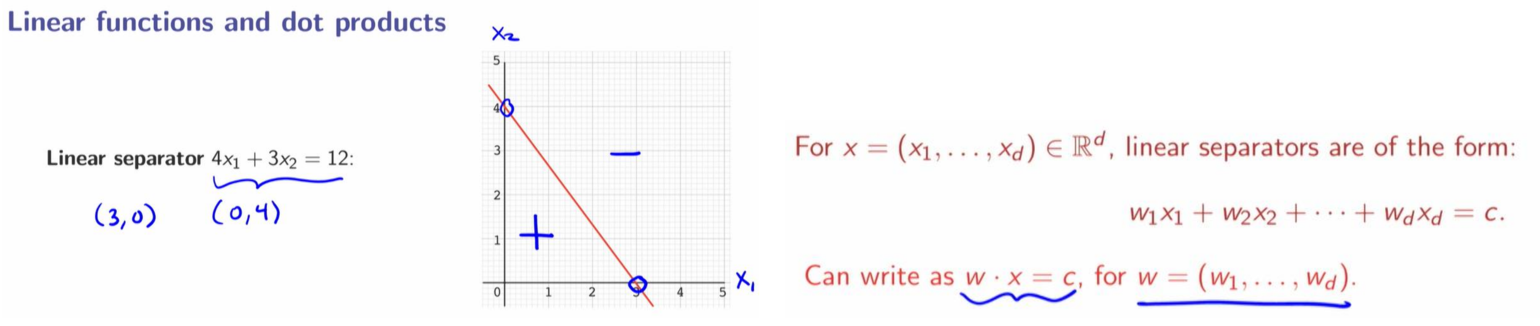
So what are **linear functions**? And while we're at it, what are **quadratic functions**?

- Well, in one dimension, a linear function of one variable X is something like three X plus two.
- And a quadratic function is similar, except that it's also allowed to have a squared term. So it can be something like four X squared minus two X plus six.

But what about in higher dimension? Let's say we're in three dimensions, for instance.

- In this case, a linear function is a sum of a bunch of terms, and each term is either a constant, like four, or it's just a multiple of one of the variables. So we end up with something like three X one minus two X two plus X three plus four.
- A quadratic function is also just a sum of terms, and each term is either a constant, or a linear term, or it could be the square of one of the variables like X one squared or six X two squared. Or it could use the pairwise product of two of the variables, like this term over here, minus two X one X three.

So this is what linear and quadratic functions are for higher dimensional data.



Linear functions and dot products

Linear separator $4x_1 + 3x_2 = 12$:

(3,0) (0,4)

For $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, linear separators are of the form:

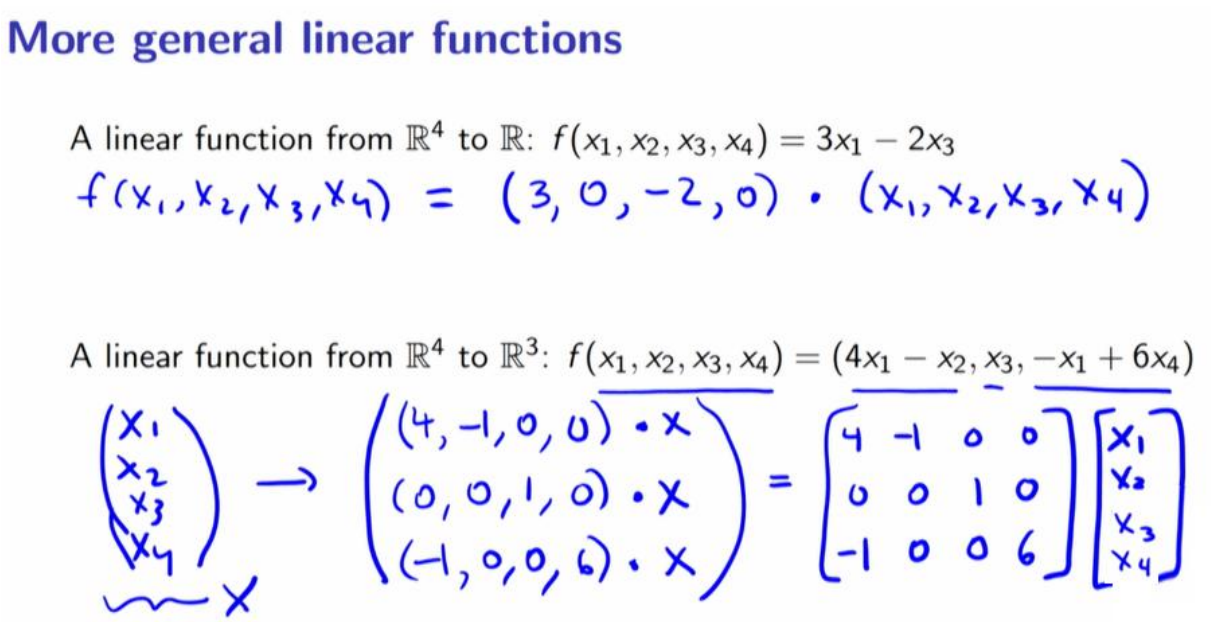
$$w_1x_1 + w_2x_2 + \cdots + w_dx_d = c.$$

Can write as $w \cdot x = c$, for $\underline{w = (w_1, \ldots, w_d)}$.

Now, we'll be making very heavy use of linear functions.For instance, a lot of the methods we study support vector machines, logistic regressions, and so on are based on linear decision boundaries, okay? What exactly are these? Well, suppose we have data in two dimensional space, then a linear decision boundary's just a line, something like the line showed over here. And what we would do with this decision boundary is one side of the boundary would be classified as plus and the other side would be classified as minus. Now, what is the equation of a linear boundary like this? Well, this is in two D so it's just a line and this particular line has the following equation. Four X one plus three X two equals 12. Let's check this, okay? So this is the X one direction here, this is the X two direction. Let's check this particular point. So this is the point three zero. Does it lie on this line? Well, four times X one plus three times X two equals 12, yes, it lies on the line. Now, let's check this point. Zero four. Does it satisfy the equation? Yes, it satisfies the equation, as well. So there's, indeed, the line, the red line shown in the figure.

So this is in two dimensions, what happens in D dimensional space? What does a linear boundary look like there? And it's a direction realization of the previous form. So when we're in D dimensional space, then X is a vector with D components, X one through X D, and a linear boundary, a linear separator is something of the from some constant W one times X one plus some constant W two times X two, plus all the way to some constant Wd times Xd equals some other constancy. This is the form of a linear separator in D dimensional space. Now, if you look at the left hand side of that equation, it looks a lot like a dot product. And in fact, it is... It can just be written as W dot X equals C. Where W is the D dimensional vector of coefficients W one trough Wd. So **dot products allow us to express linear boundaries**.

总结:



More general linear functions

A linear function from $\mathbb{R}^4$ to $\mathbb{R}$: $f(x_1, x_2, x_3, x_4) = 3x_1 - 2x_3$

$$f(x_1, x_2, x_3, x_4) = (3, 0, -2, 0) \cdot (x_1, x_2, x_3, x_4)$$

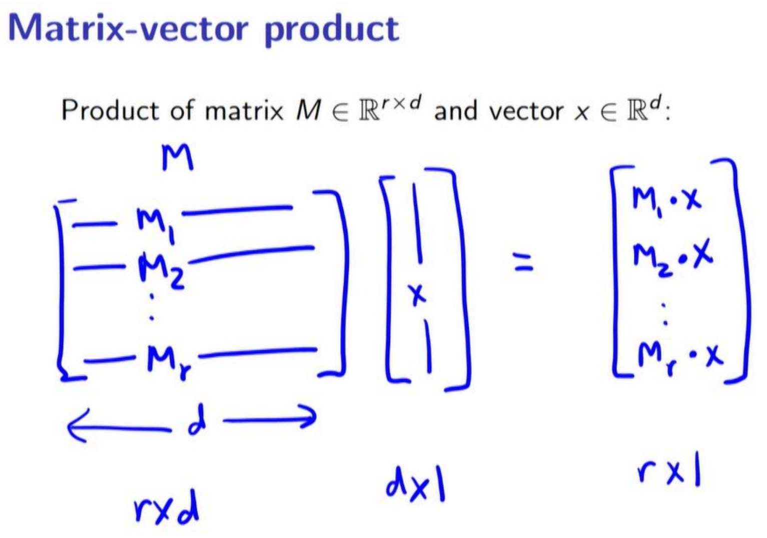
A linear function from $\mathbb{R}^4$ to $\mathbb{R}^3$: $f(x_1, x_2, x_3, x_4) = (4x_1 - x_2, x_3, -x_1 + 6x_4)$

$$\underbrace{\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}}_{x} \rightarrow \begin{pmatrix}(4,-1,0,0)\cdot x\\(0,0,1,0)\cdot x\\(-1,0,0,6)\cdot x\end{pmatrix} = \begin{bmatrix}4&-1&0&0\\0&0&1&0\\-1&0&0&6\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}$$

Let's see another example of this, and then generalize a little bit.

- So on top over here, we have a function that takes a four dimensional vector X one through X four, and returns a linear function of these four features. So let's see what we have here. We have f of X one, X two, X three, X four is... Let's go ahead and write it as a dot product. So the coefficient for X one is three. The coefficient for X two... There is no X two, so let's make that a zero. The coefficient for X three is negative two, and the coefficient for X four is zero. So it's this dot X one, X two, X three, X four. This is how we write down this linear function using a dot product.
- Now, let's look at the second example over here, which is a little bit of a generalization. So now, again, we take a four dimensional vector, but this time, instead of spitting out just one number, we are spitting out three separate numbers. **The output is a three dimensional vector, and each of the dimensions is just a linear function of X**. So let's see how we would write that. So we take...
    - This is a function that maps X one, X two, X three, X four to four, negative one, zero, zero dot X, this is X over here.
    - The second coordinate is just X three, so that's zero, zero, one, zero dot X.
    - And the third coordinate is minus one, zero, zero, six dot X.

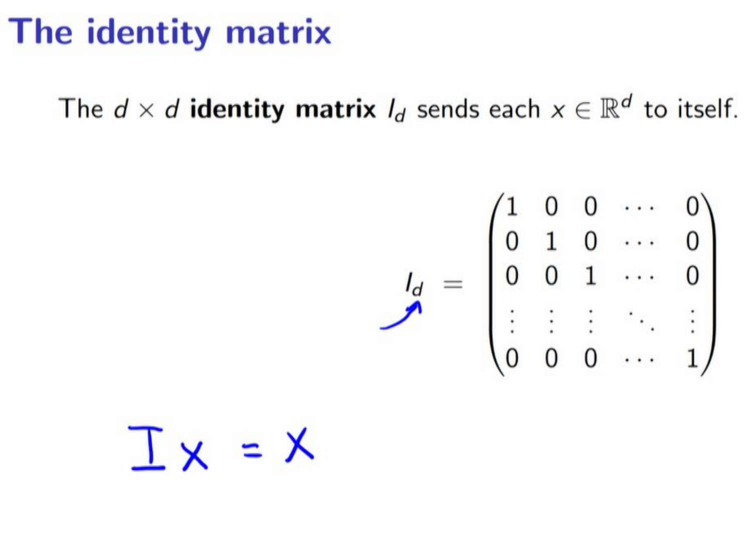
This is the function that we're talking about. So we can write this in a more convenient form using matrices. So our first row is four, negative one, zero, zero. The second row is zero, zero, one, zero. The third row is minus one, zero, zero, six times X, which is X one, X two, X three, X four. We have expressed it as a matrix times a vector.



Matrix-vector product

Product of matrix $M \in \mathbb{R}^{r\times d}$ and vector $x \in \mathbb{R}^d$:

$$\underbrace{\overset{M}{\begin{bmatrix}— M_1 —\\— M_2 —\\ \vdots\\ — M_r —\end{bmatrix}}}_{\substack{\longleftarrow d \longrightarrow\\ r\times d}} \underbrace{\begin{bmatrix}|\\x\\|\end{bmatrix}}_{d\times 1} = \underbrace{\begin{bmatrix}M_1\cdot x\\M_2\cdot x\\\vdots\\M_r\cdot x\end{bmatrix}}_{r\times 1}$$



The identity matrix

The $d \times d$ **identity matrix** I_d sends each $x \in \mathbb{R}^d$ to itself.

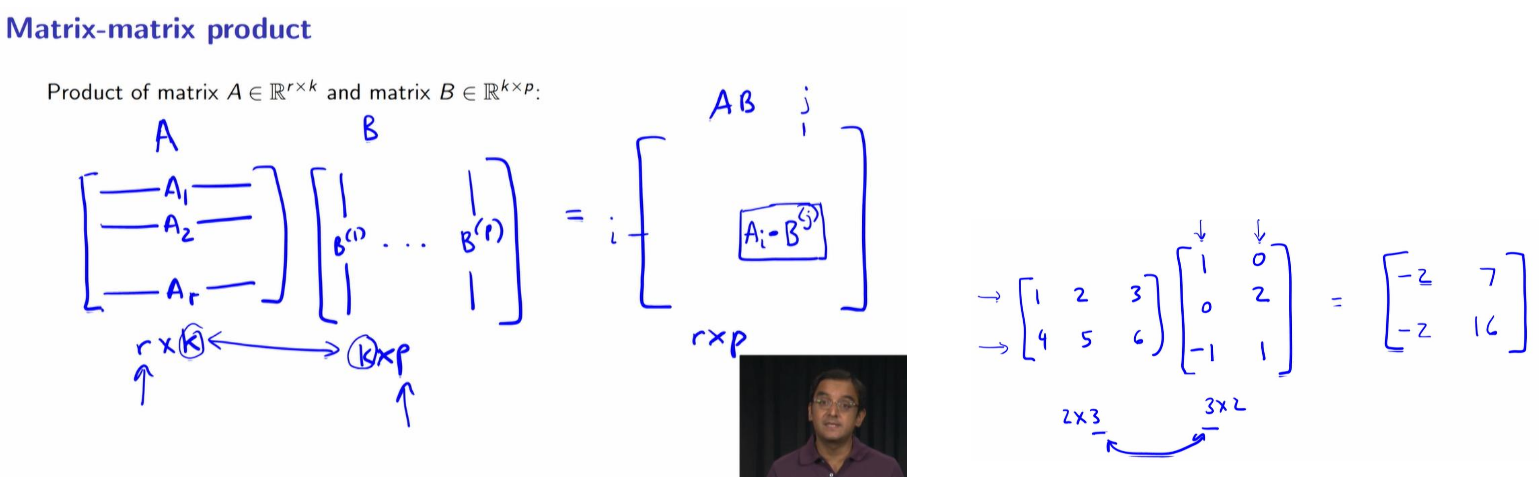
$$I_d = \begin{pmatrix}1&0&0&\cdots&0\\0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\end{pmatrix}$$

$$Ix = x$$

So let's just go ahead and talk in a little bit more...In a slightly more abstract way about this matrix vector product. How exactly does one multiply a matrix by a vector, so let's say we have a matrix that's R by D, that means that it has R rows and D columns. So we have this matrix and it has R rows, let's call them M one, M two, all the way to M of R. So we have R rows and each row is a D dimensional vector, so each row has length D. We want to multi it by a vector. The answer, in this case, is simply the first row dot producted with the vector, the second row dot product with the vector, all the way to the Rth row dot product with the vector. So we start with the matrix, that's R by D. We multiply it by a vector, which is D by one, it has D entries and the result is R by one. **That's the matrix vector product, it's just a series of dot products**.

Now, **a matrix vector product captures a linear function**, as we've seen, and the simplest possible linear function is just sending X to itself, the identity function. What matrix realizes that function. That's the **identity matrix**. It's a square matrix, if the vector X is D dimensional, it's a D by D matrix, and it has ones along the diagonal and zero everywhere else, okay? So if you take the identity matrix and you multiply it by any vector X, you get back the same vector. Very simple linear transformation. Now, we will sometimes be using identity matrices in several different dimensions. And so in those cases, we'll often use the subscript to denote what specific size identity matrix we're referring to, okay? So I sub D means a D by D identity matrix.

总结：



Matrix-matrix product

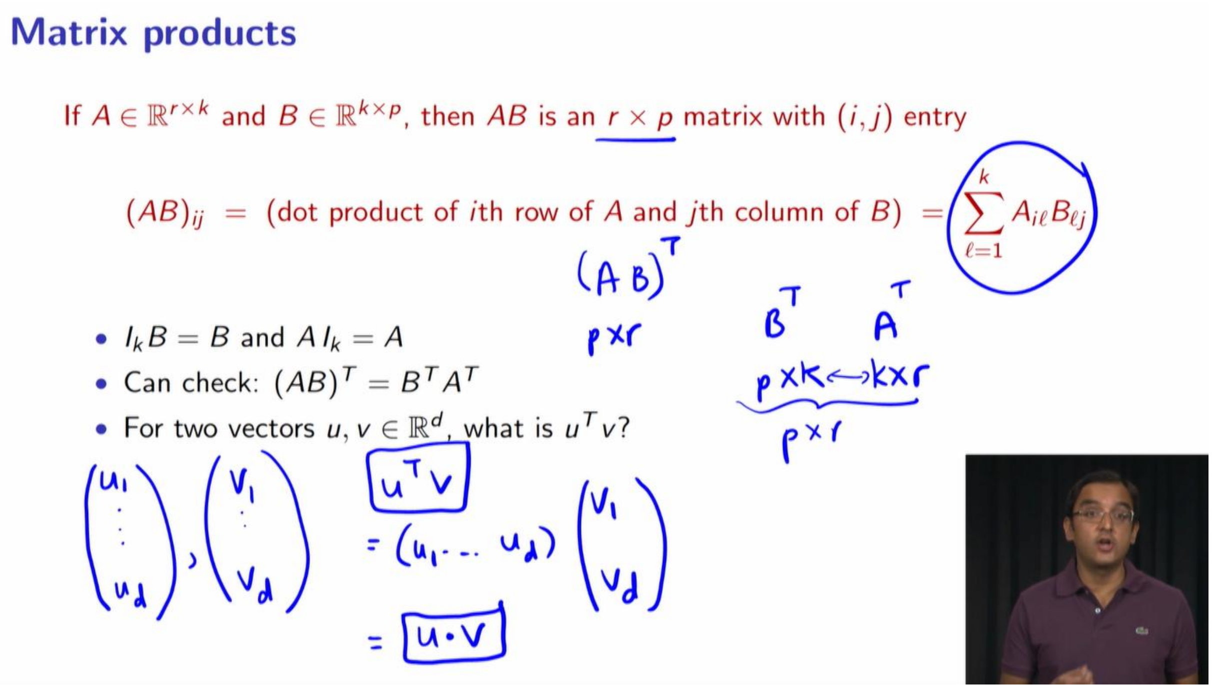
Product of matrix $A \in \mathbb{R}^{r \times k}$ and matrix $B \in \mathbb{R}^{k \times p}$:

A
B
A_1
A_2
A_r
$B^{(1)} \cdots B^{(p)}$
$r \times k$
$k \times p$
AB
j
i
$A_i \cdot B^{(j)}$
$r \times p$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 7 \\ -2 & 16 \end{bmatrix}$
2x3
3x2

Now let's move on to matrix-matrix products, so multiplying one matrix by another matrix. How do you do that? It's a very simple rule, and again, it's built out of dot products, okay? So we start with a matrix A, and in this case, we have one that is R by K, so that means it has R rows and K columns, so let's number the rows A one through Ar, R by K. Then, we have a matrix B, which is K by P. That means that it has P columns. Let's go ahead and number the columns. Let's call them B superscript one, all the way to B superscript P. So this is a K by P matrix. The result of these two is gonna be an R by P matrix. So the result, which is A times B is going to be an R by P matrix. And the IJ entry of that matrix, so if you look at the entry and row number I and column number J, that entry over there is A sub I dot product with B sub J. The IJ entry of AB is simply the dot product of the Ith row of A with the Jth column of B. So in order to compute the product, we simply take a bunch of dot products of every row of A with every column of B.

Now what this means is that if you want to multiply two matrices, the inner dimensions of the two matrices have to agree, we need a K in both these positions. And the product, the product matrix, the size of that matrix is given by the outer dimensions. So we take R by K and K by P and we get something that's R by P.

Let's do a little example of this. So let's take any matrix A, maybe one, two, three, four, five, six. Let's multiply it by some matrix B. How about one, zero, minus one, zero, two, one. So what are the dimensions here? Well, this is two by three. This is three by two. And indeed, the inner dimensions agree, as they must. What is the product? What's gonna be the size of the product? Well, it's just given by the outer dimensions. It's gonna be two by two, so let's make space for that. The answer is gonna be some two by two matrix, so we have four numbers to fill in. What are those four numbers? Well, we take the first row of A, which is this row, and we take the dot product with the first column of B, so let's see what that is. It's one minus three, so it's negative two. What number goes in the second position? We take the first row of A dot producted with the second column of B. So we get zero, four, three, we get seven. Two more numbers to fill in. We now move on to the second row of A. We take the dot product with the first column of B, so that's four minus six, which is negative two. And the final number is four times zero plus five times two plus six times one, which is 16. So that's the product of these two matrices, very simple.



Matrix products

If $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{k \times p}$, then AB is an $\underline{r \times p}$ matrix with (i, j) entry

$$(AB)_{ij} = (\text{dot product of } i\text{th row of } A \text{ and } j\text{th column of } B) = \sum_{\ell=1}^{k} A_{i\ell} B_{\ell j}$$

- $I_k B = B$ and $A I_k = A$
- Can check: $(AB)^T = B^T A^T$
- For two vectors $u, v \in \mathbb{R}^d$, what is $u^T v$?

$(AB)^T$
$p \times r$
$B^T \quad A^T$
$p \times k \leftrightarrow k \times r$
$p \times r$
$\begin{pmatrix} u_1 \\ \vdots \\ u_d \end{pmatrix}, \begin{pmatrix} v_1 \\ \vdots \\ v_d \end{pmatrix}$
$\boxed{u^T v} = (u_1 \ldots u_d) \begin{pmatrix} v_1 \\ v_d \end{pmatrix} = \boxed{u \cdot v}$

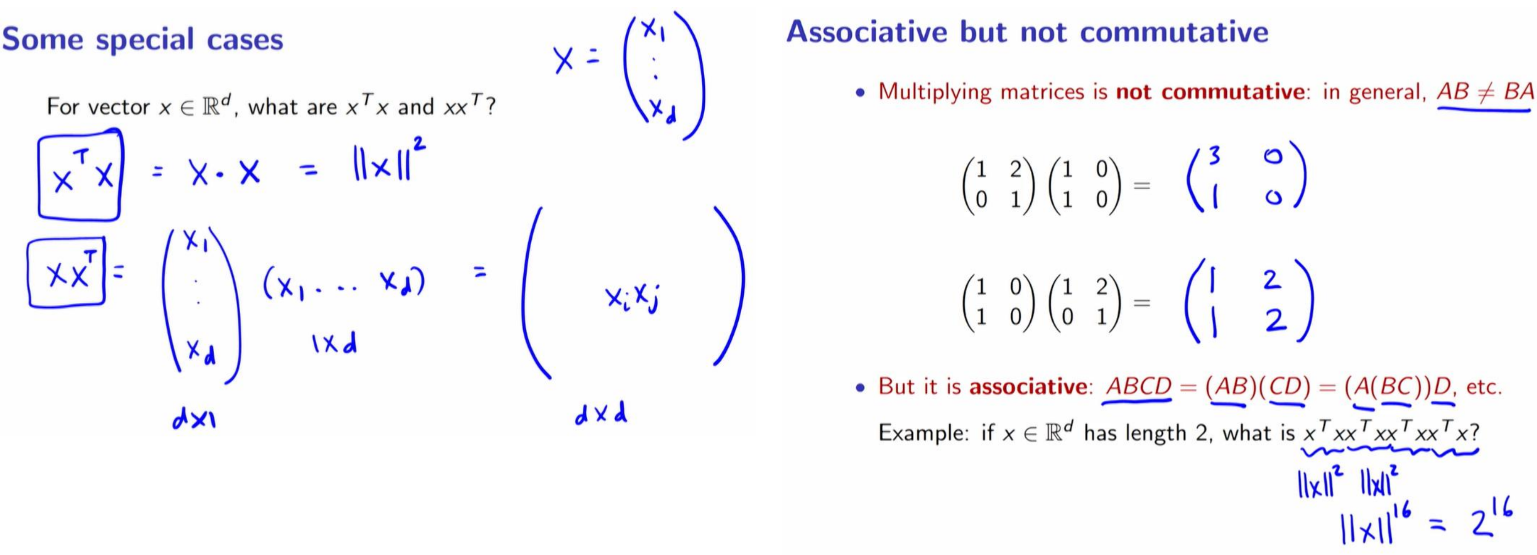
So here's the formula again, the formal expressionfor the product of two matrices. The IJ entry of the product is just a dot product between the Ith row of A and the Jth column of B. Now there are some important properties of these products.

- The first is that when you multiply something by the identity matrix, it does not change, that's to be expected.
- The second is a rather interesting one. So it says that if you take the transpose of a product, it's like taking the product of the transposes but with the flipped order. Now, this is something which you can check by just plugging in the definition of the product, by just plugging in the summation you see over here and just checking that left and right hand sides really work out correctly. But why don't we, at least, as a basic sanity check, just at least check that the dimensions of the two sides are the

总结：

same. So what is the dimension of AB transpose? Well, we've already seen that in this case, if A is R by K and B is K by P, then AB is R by P, so AB transpose is switching the rows and columns, so the dimension of it is gonna be P by R. Okay? Now, let's look at B transpose A transpose. B transpose has dimension B by K A transpose has dimension K by R. So good, you can actually multiply them, since these inner dimensions agree, and the result will have dimension P by R. So at least it's the right size. And it turns out that these two matrices will also be identical.

- Now, for this last one, what we're being asked is that we have these vectors U and V, so we have U, a D dimensional vector and we have V, a D dimensional vector. What is U transpose V, okay? So let's compute the transpose of U. So we just take that column vector and make it into a row. So U transpose V is equal to U one to Ud multiplied by V one to Vd. Okay, so we take each row of the first matrix and multiply it by each column with the second matrix. Oh, it's just the dot product of U and V. So this is interesting. We have two ways of writing exactly the same thing. The dot product of two vectors U dot V is exactly the same as U transpose V. So this a useful fact to remember. It's something that we'll be using to simplify a bunch of calculations later on. So **U transpose V is just a dot product between U and V**.



Some special cases

For vector $x \in \mathbb{R}^d$, what are x^Tx and xx^T?

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$

$$\boxed{x^Tx} = x \cdot x = \|x\|^2$$

$$\boxed{xx^T} = \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}}_{d\times 1} \underbrace{(x_1 \dots x_d)}_{1\times d} = \underbrace{\begin{pmatrix} x_ix_j \end{pmatrix}}_{d\times d}$$

Associative but not commutative

- Multiplying matrices is **not commutative**: in general, $AB \neq BA$

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 0 \\ 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

- But it is **associative**: $ABCD = (AB)(CD) = (A(BC))D$, etc.

Example: if $x \in \mathbb{R}^d$ has length 2, what is $x^Txx^Txx^Txx^Tx$?

$\|x\|^2 \ \|x\|^2$

$\|x\|^{16} = 2^{16}$

Let's look at a couple of examples. So now let's say we have a D dimensional vector X, so X has got D numbers.

- What is X transpose X? Well, we just saw this. It's the dot product of X with itself. And what's that? That's just a squared length of X. So X transpose X is just a number, it's the squared length of X.
- What about XX transpose? So XX transpose, so let's see. X is this vector. So X is D by one. X transpose looks like this. It's one by D. So the product is D by D. It's a big matrix. Okay. That's a little strange. What is the IJ entry of this matrix? It is XiXj. So it's the D by D matrix of all pairwise products or features of X.

So these are also two very useful things to bear in mind. **X transpose X, versus XX transpose. They look almost the same, but as you can see, they're completely different. One of them is just a simple number, the length of the vector squared, and the other is a full D by D matrix**.

So one of the things this really goes to show is that you cannot change the order in which things are multiplied, okay? And so this is a very general and important property of matrices. In general, matrix multiplication is not commutative. AB is not the same as BA. Unless you get lucky, okay? So let's look at these examples, for instance. So here we have two matrices, A and B. And let's multiply them one way and multiply them the other way and see if we get the same answer, okay? They're both square matrices, so we know they're both two by two, and so we know the result of the two by two.

- Let's see what we get in this case. So we get one plus two, that's three. Zero and zero is zero. Zero and one is one. Zero and zero is zero, okay? So that's the first product.
- The second product is one, two. One, two. They are completely different, they aren't even close. So this is something to be a little bit careful about, you cannot switch the order in matrix multiplication.

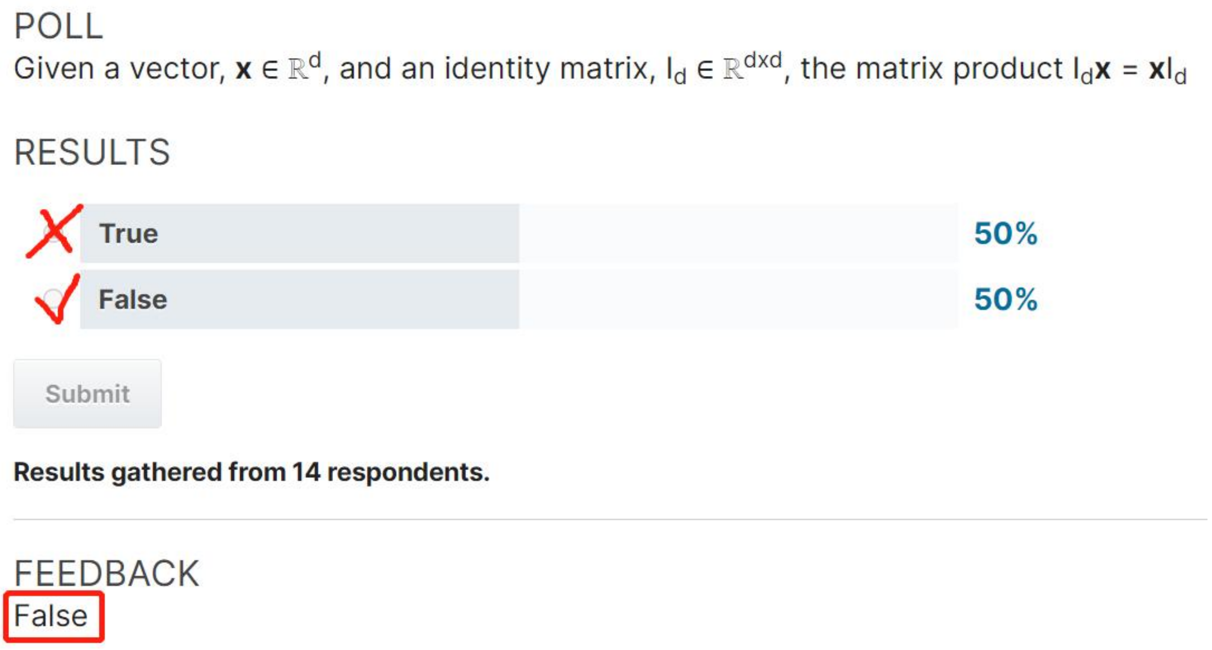
One thing you can do, however, and something that's very convenient is that when you're multiplying together a bunch of matrices, you can decide which pair of multiplications you're gonna do first, as long as you keep them in the original order. For example, you can decide, okay, you have to multiply together ABCD. Why don't do AB first and then do CD, and then multiply them together? That's allowed. Or you can do BC first, and then multiply the answer by A, and multiply that answer by D and so on, okay? So you can parenthesize the expression in whatever way you like, as long as you maintain the order of the matrices.

- Let's look at a little example of this. So here, let's say we have a vector X whose length is two. What is this huge expression over here? Okay, so clearly there are a lot of things to be multiplied together, but parts of it look familiar, we've seen these sort of things before, X transpose X, XX transpose. What are they? Okay, X transpose X, that's a D by D matrix, okay? XX transpose is a D by D matrix, X transpose X is just a number.Which should we prefer? The number, that seems a lot simpler. So let's do those first, okay? So we'll take X transpose X, we know that's just the length of X squared. Oh, here's another X transpose X, that's also the length of X squared. And here's another one and another one. And so what we end up with is the length of X to the power 16. And the total length of X is two, so this is two to the 16.

So **the associative rule can be very helpful in simplifying matrix products. We simply choose the products that turn out to be very simple and do those first**.

So that's it for our second installment of linear algebra. We now have all the notation and tools we need to be able to model linear functions, which is something we're gonna be using very heavily in this course.

总结:



POLL

Given a vector, $\mathbf{x} \in \mathbb{R}^d$, and an identity matrix, $I_d \in \mathbb{R}^{dxd}$, the matrix product $I_d\mathbf{x} = \mathbf{x}I_d$

RESULTS

✗	True	50%
✓	False	50%

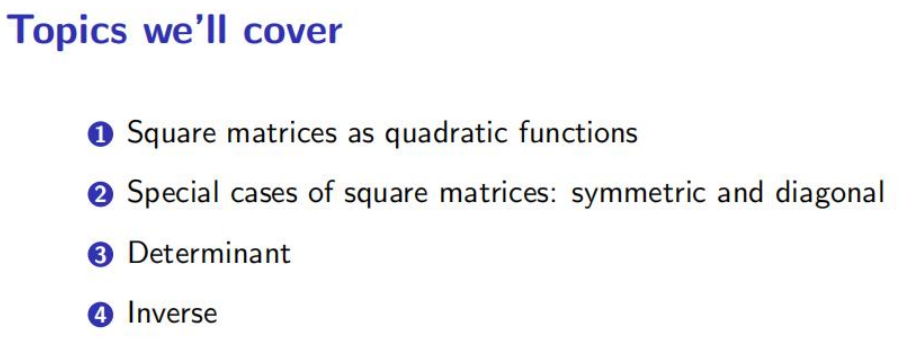
Results gathered from 14 respondents.

FEEDBACK

False

总结：

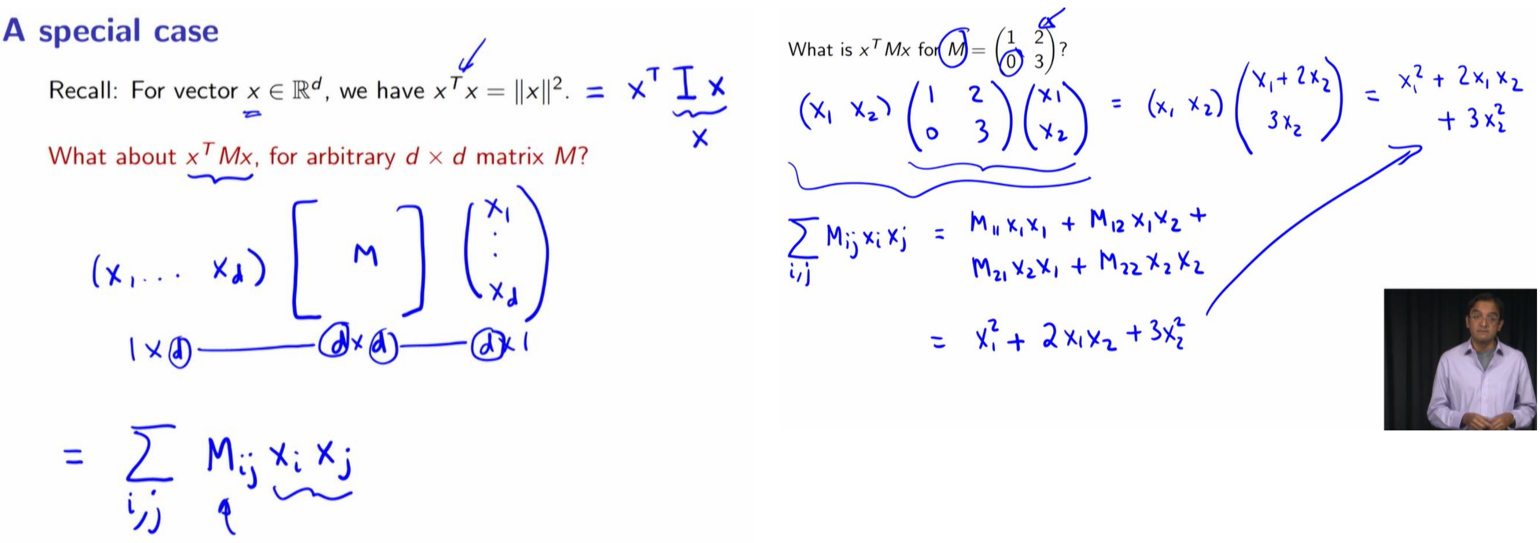
3.3 Linear Algebra III: Square Matrices as Quadratic Functions



Topics we'll cover

1. Square matrices as quadratic functions
2. Special cases of square matrices: symmetric and diagonal
3. Determinant
4. Inverse

In an earlier installment of our linear algebra primer,we saw that **any matrix can be thought of as representing a linear function**. What we'll see today is that **if the matrix is square, so if it has the same number of rows and columns, then it also represents a quadratic function**. Today is going to be all about square matrices and their properties.



A special case

Recall: For vector $x \in \mathbb{R}^d$, we have $x^T x = \|x\|^2$. $= x^T \underbrace{I x}_{x}$

What about $x^T Mx$, for arbitrary $d \times d$ matrix M?

$$(x_1 \dots x_d) \begin{bmatrix} M \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$

$1 \times d$ — $d \times d$ — $d \times 1$

$$= \sum_{i,j} M_{ij} x_i x_j$$

What is $x^T Mx$ for $M = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}$?

$$(x_1\ x_2) \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (x_1\ x_2) \begin{pmatrix} x_1 + 2x_2 \\ 3x_2 \end{pmatrix} = x_1^2 + 2x_1x_2 + 3x_2^2$$

$$\sum_{i,j} M_{ij} x_i x_j = M_{11}x_1x_1 + M_{12}x_1x_2 + M_{21}x_2x_1 + M_{22}x_2x_2$$

$$= x_1^2 + 2x_1x_2 + 3x_2^2$$

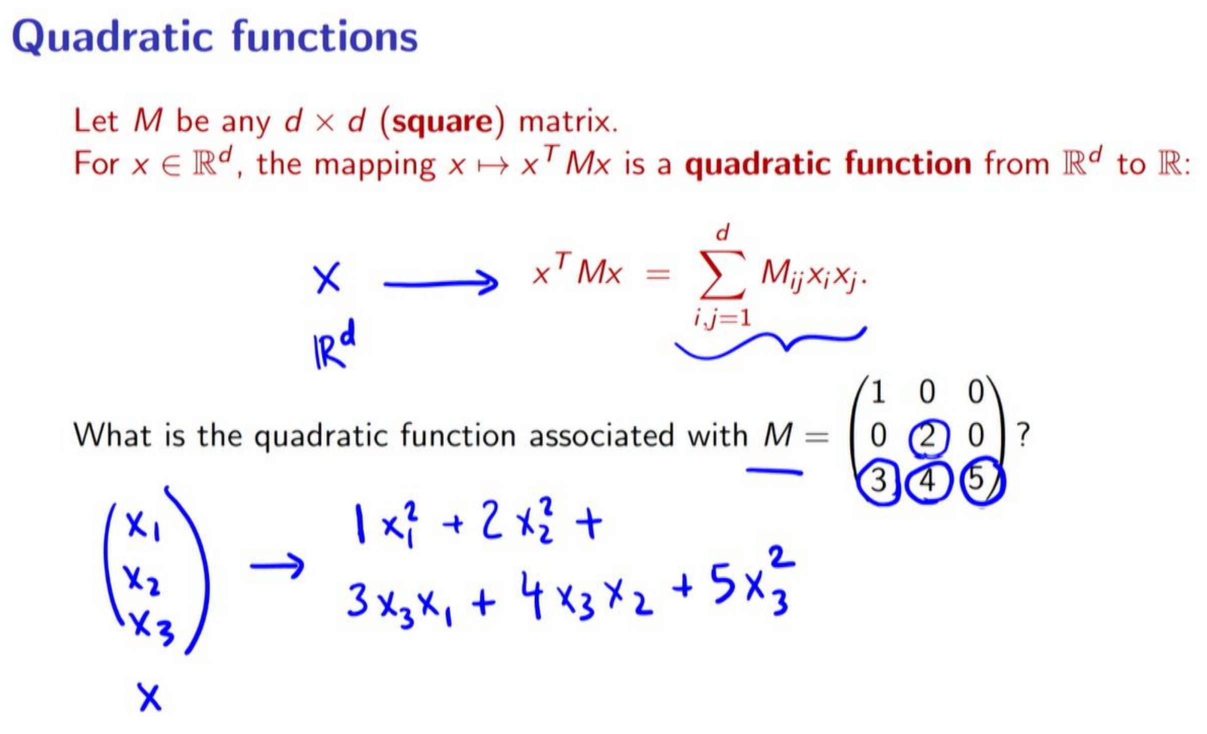
Let's start with a special case that we saw before. One thing we found is that if you have a vector x, then x transpose x is just another way of writing the dot product of x with itself, which is the length of x squared. **In fact you can also think of this as being x transpose times the identity times x because the identity times x is just x**. Now, the identity matrix is just a d x d square matrix. What if instead of having the identity there, we put some other d x d matrix in the middle? What if we use x transpose Mx, where M is just any arbitrary d x d matrix, what do we get?

- Let's take a look, so we have x transpose that means take x and make it into a row. We have this matrix M, which is d x d. Then, we have just x, which is a column. What are the sizes? Here we have 1 x d. Here, we have d x 1. When we multiply things, we see that the inner dimensions agree and the result is simply going to be the outer dimensions, which is 1 x 1. The result is just going to be a single number as we expected from the case of x transpose x.
- The result is a single number and in fact, I'll give you a formula for the number. It turns out that if you just expand this out, it's quite simple to do. **It turns out to be the sum over all i and j of Mij xi xj. This is a quadratic function of x**. Each of these terms, it's a sum of a whole bunch of terms. Each of these terms has got some Mij. That's just a number from the matrix M, some number like two, three, four and so on, **but the dependence on x is quadratic. They're pairs of terms from x, xij, or also some terms like xi squared**.

Let's see a specific example of this just to drive the point home. What if we have a matrix that's two by two, what kind of quadratic function does this represent? **Well, if the matrix is two x two, these vectors x have to be two-dimensional**. The function we're talking about is x1 x2 times this matrix times x1 x2 and we can just expand this out. Let's start by doing this part. We'll copy over the x1 x2 and let's just simplify the matrix times the vector, what do we get? We get x1 plus 2x2 and 3x2. Now, we're multiplying together these two and this is just a single dot product. It is x1 squared plus 2x1 x2 plus 3x2 squared. This is certainly a quadratic function. All the terms in here are quadratic, x1 squared, x1 x2, x2 squared.

Let's actually go back and see how the squares with the formula I gave you earlier, this whole thing with the sum over ij of Mij, let's just copy that down and see that we actually get the same thing. I said that it should work out to Mij xi xj, the sum over all ij. What are i and J in this case, well it's just two-dimensional, so they're going to run over one and two. This is equal to M11 x1 x1. Let's just write out all the terms M12 x1 x2, M21 x2 x1 plus M22 x2 x2. Now, we can go in and plug in the entries from the matrix M. What's M11 that's 1, so it's x1 squared. What's M12 that's this entry over here. It's a 2 plus 2x1x2. What is M21 that's a zero and M22 was a three. You can see that it's exactly the formula we got simply by expanding out and in fact that's how you get this summation. If you just do what we did before, but in greater generality, you allow it to be d dimensional and you allow M to just consist of variables that haven't been defined yet.

总结:



Quadratic functions

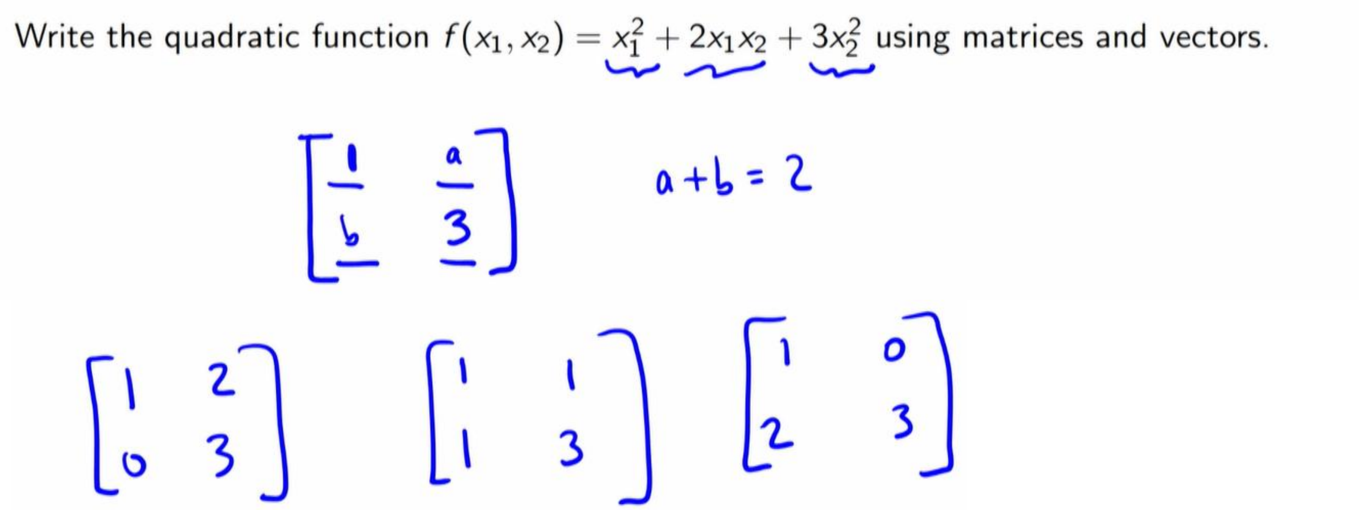
Let M be any $d \times d$ (**square**) matrix.
For $x \in \mathbb{R}^d$, the mapping $x \mapsto x^T M x$ is a **quadratic function** from $\mathbb{R}^d$ to $\mathbb{R}$:

$$\underset{\mathbb{R}^d}{x} \longrightarrow x^T M x = \underbrace{\sum_{i,j=1}^{d} M_{ij} x_i x_j.}$$

What is the quadratic function associated with $\underline{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \boxed{2} & 0 \\ \boxed{3} & \boxed{4} & \boxed{5} \end{pmatrix}$?

$$\underset{x}{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}} \to 1x_1^2 + 2x_2^2 + 3x_3x_1 + 4x_3x_2 + 5x_3^2$$

Here is a summary of the formula. If you start with any square matrix M, you can define a quadratic function by mapping x, a d dimensional vector to the number x transpose Mx and it works out to this summation over here. The summation actually lets us read out the quadratic function very easily. For example, we have this matrix, what if the quadratic function associated with it, it's something that sends three dimensional vectors to a single number, so the three dimensional vectors x1 x2 x3 and they get sent to what number? Well M11 is one times x1 squared. Then, we have M22 times x2 squared. Then, we have M31 times x3 x1. Then, we have M32 times x3 x2. Then, we have M33 times x3 x3. That is the quadratic function associated with this matrix.



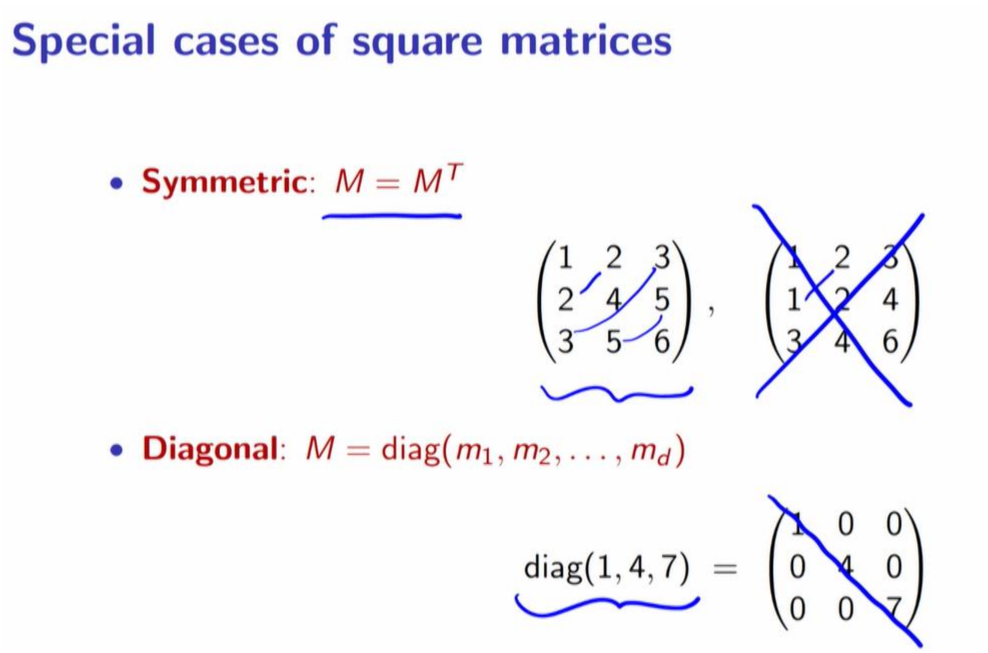
Write the quadratic function $f(x_1, x_2) = \underbrace{x_1^2} + \underbrace{2x_1x_2} + \underbrace{3x_2^2}$ using matrices and vectors.

$$\begin{bmatrix} \underline{1} & \underline{a} \\ \underline{b} & \underline{3} \end{bmatrix} \qquad a + b = 2$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}$$

We can also try to reverse engineering things. Here we have a quadratic function that takes a two dimensional vector and just returns a number, how do we write this function using a matrix? What is the matrix that corresponds to this? Well, these are two dimensional vectors, so we're going to need a two by two matrix. We have to fill in these four entries over here. Now, just from the fact that the leading coefficient here is a one, we know that this must be a one. From the fact that this coefficient for x2 squared is a three, we know that this must be a three, but it's not clear what these two numbers have to be. Let's just call them a and b. What we know is that a plus b has to be equal to two. There are actually several ways to achieve this function. We could use a matrix like 1 2 0 3 or we could use 1 1 1 3 or we could use 1 0 2 3 and so on. **There are actually infinitely many possibilities. There are many matrices that realize this particular quadratic function**.

总结:



Special cases of square matrices

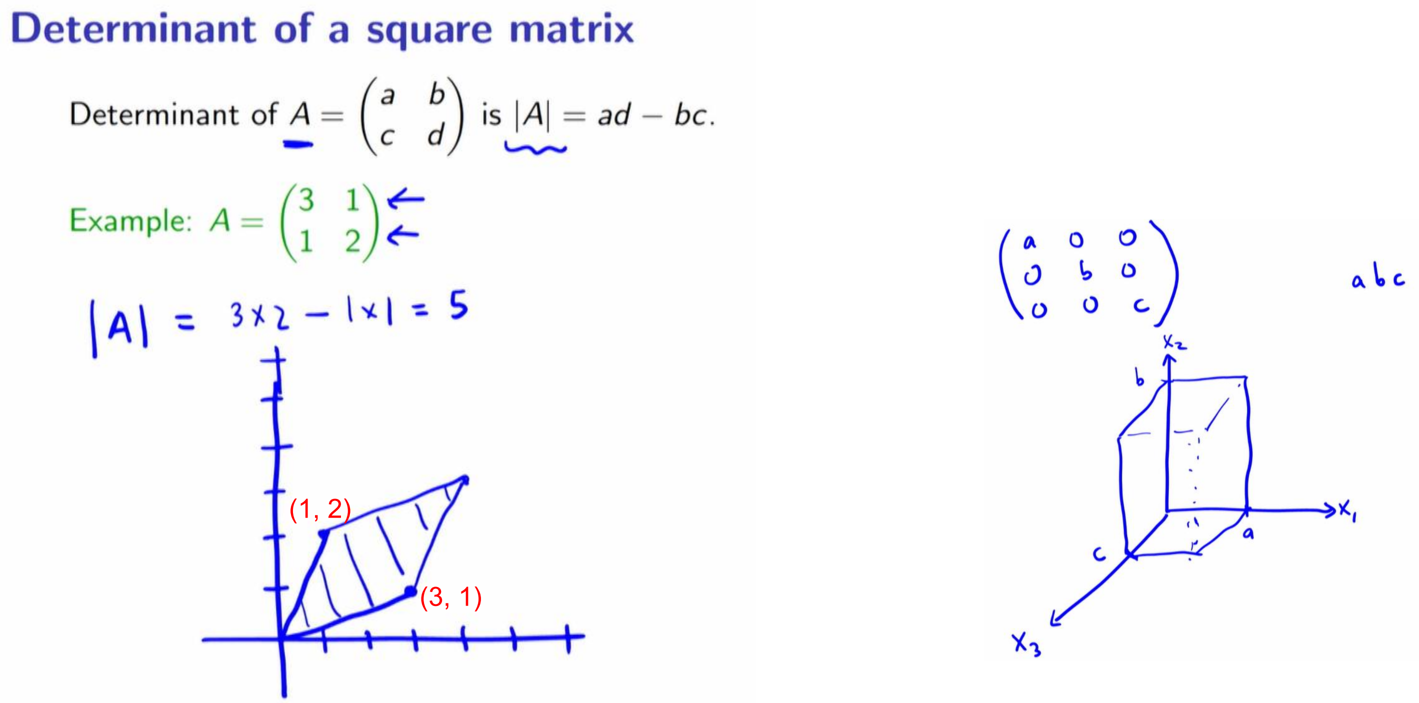
- **Symmetric**: $M = M^T$

$$\underbrace{\begin{pmatrix}1 & 2 & 3\\ 2 & 4 & 5\\ 3 & 5 & 6\end{pmatrix}}, \quad \cancel{\begin{pmatrix}1 & 2 & 3\\ 1 & 2 & 4\\ 3 & 4 & 6\end{pmatrix}}$$

- **Diagonal**: $M = \mathrm{diag}(m_1, m_2, \ldots, m_d)$

$$\underbrace{\mathrm{diag}(1,4,7)} = \begin{pmatrix}1 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 7\end{pmatrix}$$

We've been talking about square matrices, a special type of square matrix is one that is **symmetric**, one that is equal to its transpose. What this means is that the matrix remains the same if you reflect it in the diagonal. Here's an example of a **symmetric matrix**. If you reflect it in the diagonal, two comes back two, three comes back to three, five comes back to five, so it's symmetric. The second matrix over here is not symmetric. Look at this one and two for example. This is a non-symmetric matrix. It turns out that many of the matrices we'll be dealing with will just automatically turn out to be symmetric and this will simplify things quite a bit. Now, an extra special case of a symmetric matrix is one that is diagonal, a matrix whose only nonzero entries are along the principal diagonal, so one like this. **Diagonal matrix** is zero everywhere except along this diagonal over here and as a result, there's a very compact way of writing such matrices since there's no point using D squared entries since most of them are zero. We sometimes just use this shorthand over here.



Determinant of a square matrix

Determinant of $A = \begin{pmatrix}a & b\\ c & d\end{pmatrix}$ is $|A| = ad - bc$.

Example: $A = \begin{pmatrix}3 & 1\\ 1 & 2\end{pmatrix}$

$|A| = 3\times 2 - 1\times 1 = 5$

$$\begin{pmatrix}a & 0 & 0\\ 0 & b & 0\\ 0 & 0 & c\end{pmatrix}$$

abc

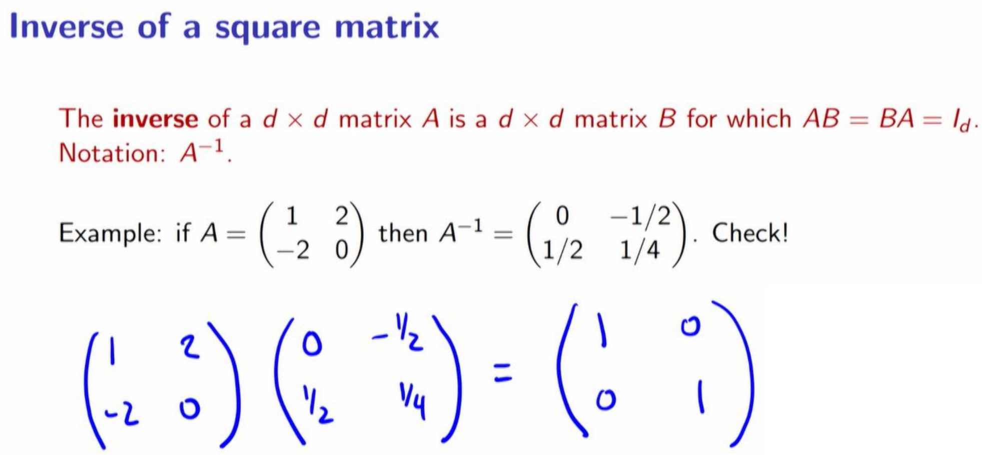
Now, one of the key quantities associated with the square matrix is the **determinant**. If we have a matrix A, the determinant is just some number and we represent it using these bars. We put these bars around A and you might remember from high school or early college that the determinant of a two by two matrix a b c d is ad minus bc. For example, if we have this two by two matrix over here, the determinant of A is 3 times 2 minus 1 times 1, which is 5. It seems a little bit random. Does the determinant actually mean anything? It turns out that it does. Let's look at this specific matrix.

Let's think of the rows of the matrix as being data points, so we have these two data points and let's just go ahead and plot these points. We have one, two, three, four, five, six one, two, three, four, five, six. The first point is three one. The second point is one, two. We've gone ahead and we've plotted those two points. Now, let's take these two points and extend them to get a full parallelogram. We had this three one and now we take another one two and we come out here. Now, we have a parallelogram. What is the area of this parallelogram? It turns out to be five. That's what the determinant is. If you think of each row of the matrix as being a data point and you plot those points and you create the parallelogram, the **determinant** is the **area** of the parallelogram. This works not just in two dimensions. It works in d dimensions as well. If you

总结：

have a d x d square matrix, you look at the rows of that matrix. You plot those points in d dimensional space. You extend them into a parallelogram. When it's in higher dimension, instead of being called a parallelogram, it's called a parallelepiped, so extend it into a parallelepiped. The **volume** of that object is the **determinant** of the matrix.

Let's actually go ahead and check that with an example that looked familiar to us, what if we take a three x three matrix? Like let's take this matrix over here. Let's go ahead and plot these points in 3D. Let's say a, b and c are positive. We have our three axes, x1, x2. I'll just draw the third axis over here x3. Along the x1 axis, our first point lies along the x1 axis, it's a. Our second point lies along the x2 axis, it's b over here and our third point lies along the x3 axis, so it's c over here. Now, we extend this into a three dimensional body. Not very good artwork I'll confess. What is the volume of this thing? Well, we know this, the volume of this is just abc. Now, let's go back to what we were saying about determinants. Is the determinant of this matrix abc? Indeed it is, the determinant of a diagonal matrix is just the product of the diagonal elements.



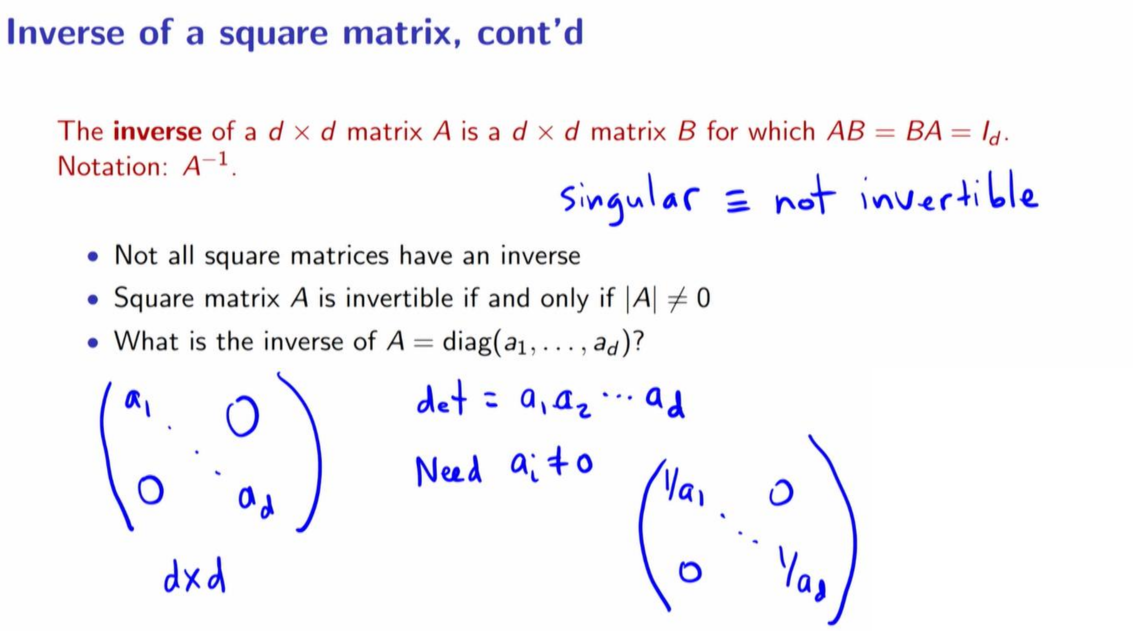
Inverse of a square matrix

The **inverse** of a $d \times d$ matrix A is a $d \times d$ matrix B for which $AB = BA = I_d$.
Notation: A^{-1}.

Example: if $A = \begin{pmatrix} 1 & 2 \\ -2 & 0 \end{pmatrix}$ then $A^{-1} = \begin{pmatrix} 0 & -1/2 \\ 1/2 & 1/4 \end{pmatrix}$. Check!

$$\begin{pmatrix} 1 & 2 \\ -2 & 0 \end{pmatrix}\begin{pmatrix} 0 & -1/2 \\ 1/2 & 1/4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

A very important quantity associated with the square matrixis the **inverse**. If you have a square matrix, a d x d matrix A, the inverse is some other matrix B such that A times B is the identity and B times A is the identity. What this means among other things is that B has to also be a d x d matrix. The notation we'll use or that we'll call this matrix B, we'll just call it A to the minus 1, the inverse of A. Let's do a little bit of an example. Here's the two by two matrix A and I'm claiming that this particular matrix is the inverse of A. Let's check and see if this is really the case, so 1, 2, minus 2, zero that's A and the thing that I'm claiming is the inverse is zero minus half, 1/2 and 1/4. Let's see what this multiplies out to. The first row of A times the first column of B, so we get 1. The first row of A times the second column of B, we get negative 1/2, plus 1/2 that's zero. The second row of A time's the first column of B, we get zero. The second row of A times the second column of B, we get one plus zero one, which is indeed the identity matrix. This is the inverse.



Inverse of a square matrix, cont'd

The **inverse** of a $d \times d$ matrix A is a $d \times d$ matrix B for which $AB = BA = I_d$.
Notation: A^{-1}.

singular ≡ not invertible

- Not all square matrices have an inverse
- Square matrix A is invertible if and only if $|A| \neq 0$
- What is the inverse of $A = \text{diag}(a_1, \ldots, a_d)$?

$$\underbrace{\begin{pmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_d \end{pmatrix}}_{d \times d}$$

$\det = a_1 a_2 \cdots a_d$

Need $a_i \neq 0$

$$\begin{pmatrix} 1/a_1 & & 0 \\ & \ddots & \\ 0 & & 1/a_d \end{pmatrix}$$

Now, there are a few things about inverses that really should be emphasized.

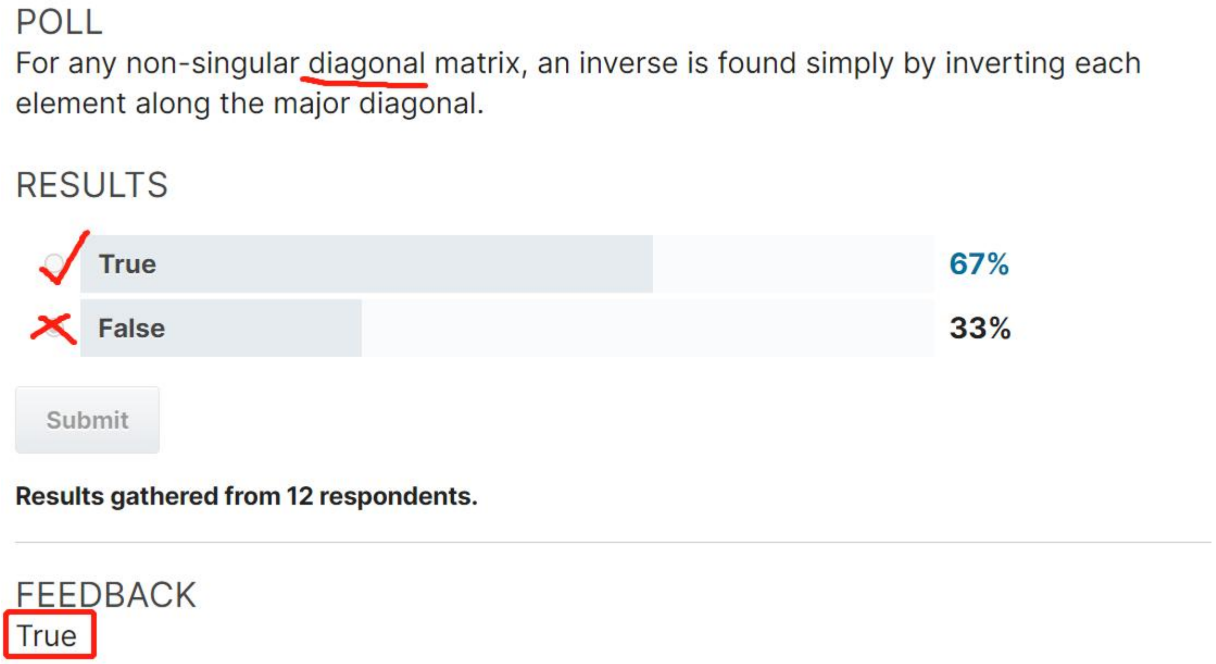
- First of all, they don't always exist. There are many square matrices that don't have an inverse and we call those matrices singular. A matrix is singular if it's not invertible. It's something fairly common.
- The second thing is that there's an easy way to assess singularity. A matrix is singular if it's determinant is zero. Another way to put it is a matrix is invertible if and only if its determinant is nonzero.
- Now, let's look at a special case. Suppose we have a diagonal matrix a1 to ad, what is its inverse? We have a diagonal matrix a1 to ad. If you remember the notation, it means that this is the matrix we're talking about. All the entries off the diagonal are

总结：

zero. It's a d by d matrix. First of all, can a matrix like this ever be singular, non-invertible? Yes, it can. If you recall the determinant of a matrix like this, it's just the product of all the diagonal entries. It's a1 times a2 all the way to ad. We want this product to not be zero. We need all the ai to be nonzero. If that's the case, then the determinant is not zero and the matrix is invertible.

What is the inverse? Well, it's very simple, you just invert each of the diagonal entries. That's the overall inverse of the matrix.

Well that's it for the third installment of our linear algebra primer. This is going to hold us for a little while, so we'll give it a week or two to sink in and then, we'll come back in a little while and do a little bit more linear algebra.



POLL

For any non-singular diagonal matrix, an inverse is found simply by inverting each element along the major diagonal.

RESULTS

✓ True 67%

✗ False 33%

Submit

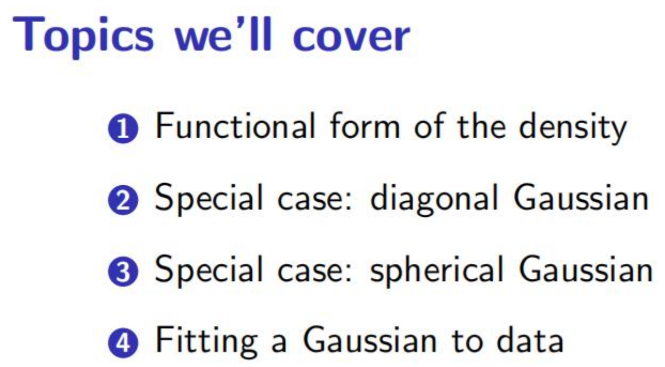
Results gathered from 12 respondents.

FEEDBACK

True

总结：

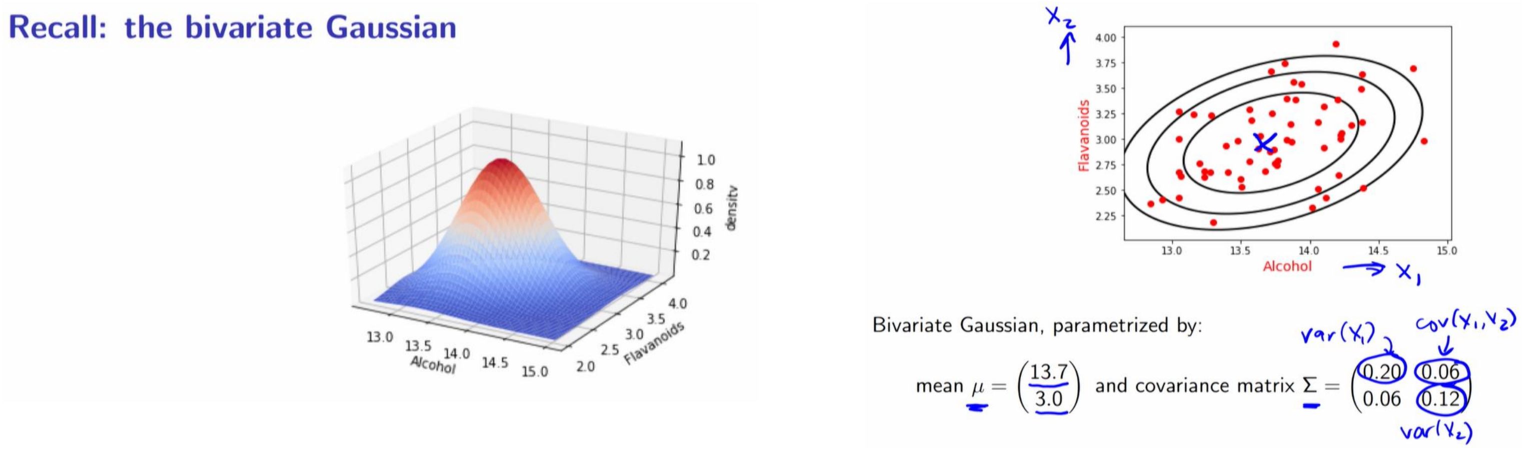
3.4 The Multivariate Gaussian



Topics we'll cover

1. Functional form of the density
2. Special case: diagonal Gaussian
3. Special case: spherical Gaussian
4. Fitting a Gaussian to data

Now that we have linear algebra somewhat under control, we can take a look at the Gaussian distribution in arbitrary dimension. So what we're gonna do today is to start by looking at the density of the high dimensional Gaussian, and getting some intuition for this, and then move on to some widely used special cases.



Recall: the bivariate Gaussian

Bivariate Gaussian, parametrized by:

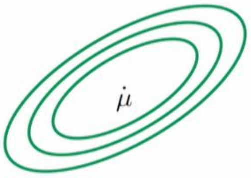
mean $\underline{\mu} = \begin{pmatrix} \underline{13.7} \\ \underline{3.0} \end{pmatrix}$ and covariance matrix $\underline{\Sigma} = \begin{pmatrix} \overset{var(X_1)}{\boxed{0.20}} & \overset{cov(X_1,X_2)}{\boxed{0.06}} \\ 0.06 & \underset{var(X_2)}{\boxed{0.12}} \end{pmatrix}$

So let's do a little bit of a rewind.If you recall, and we've been looking at the wine data set, we pulled out two features. Alcohol level, and flavanoids. And we fit a two-dimensional Gaussian to the data from one of the wineries, and we got something like this. Okay? So this is the 2-D Gaussian. So it's a distribution over the entire plane, over every possible pair x one x two. So that's the density shown here. So the density here is shown by the height above the surface.

I personally find pictures like this a little hard to understand. So this is another depiction, which just shows the contour lines drawn on the plane. So the red dots here are the actual data points. And the way we fit a Gaussian to them was to simply compute the mean of each of the features. So there's feature number one, alcohol level, feature number two, flavanoids. We computed the mean along each feature, and the variance of each feature, and then the covariance between the two features. And this gave us a 2D Gaussian. So the Gaussian parameters were the mean, which is the two dimensional vector, represented by the Greek letter mu, which consists of the mean of x one and the mean of x two. The other parameter of the Gaussian is a two by two covariance matrix, which we denote by the Greek letter sigma, capital sigma. And it has only three distinct numbers in it. It has the variance of x one, up here, it has the variance of x two, down here, and then it has the covariance between the two features. Which are the off-diagonal numbers. This is the mean, mu is the center point over here. It's the point of highest density. The ellipses show the contour lines of the density and the reason that tilted upwards is because there is a positive correlation between the two features. The two dimensional Gaussian is quite simple and it turns out that this generalizes in a straight forward way to d-dimensions.

总结：

The multivariate Gaussian



$N(\mu, \Sigma)$: Gaussian in $\boxed{\mathbb{R}^d}$

- mean: $\underline{\mu} \in \underline{\mathbb{R}^d}$
- covariance: $d \times d$ matrix $\underline{\Sigma}$

Generates points $X = (\underline{X_1}, \underline{X_2}, \ldots, \underline{X_d})$.

$$\Sigma = \begin{bmatrix} var(X_1) & cov(X_1,X_2) & cov(X_1,X_3) \\ & \ddots & \\ & & var(X_d) \end{bmatrix}$$

- μ is the vector of coordinate-wise means:

$$\mu_1 = \underline{\mathbb{E}X_1}, \ \mu_2 = \underline{\mathbb{E}X_2}, \ldots, \ \mu_d = \underline{\mathbb{E}X_d}.$$

- Σ is a matrix containing all pairwise covariances:

$$\Sigma_{ij} = \Sigma_{ji} = \text{cov}(X_i, X_j) \quad \text{if } i \neq j$$
$$\Sigma_{ii} = \text{var}(X_i)$$

Density $p(x) = \dfrac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\dfrac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$

So let's move to the higher dimensional case. So now we're in d-dimension so we want a distribution of Rd, and now instead of two features we have d features. x one, x two, all the way to xd.

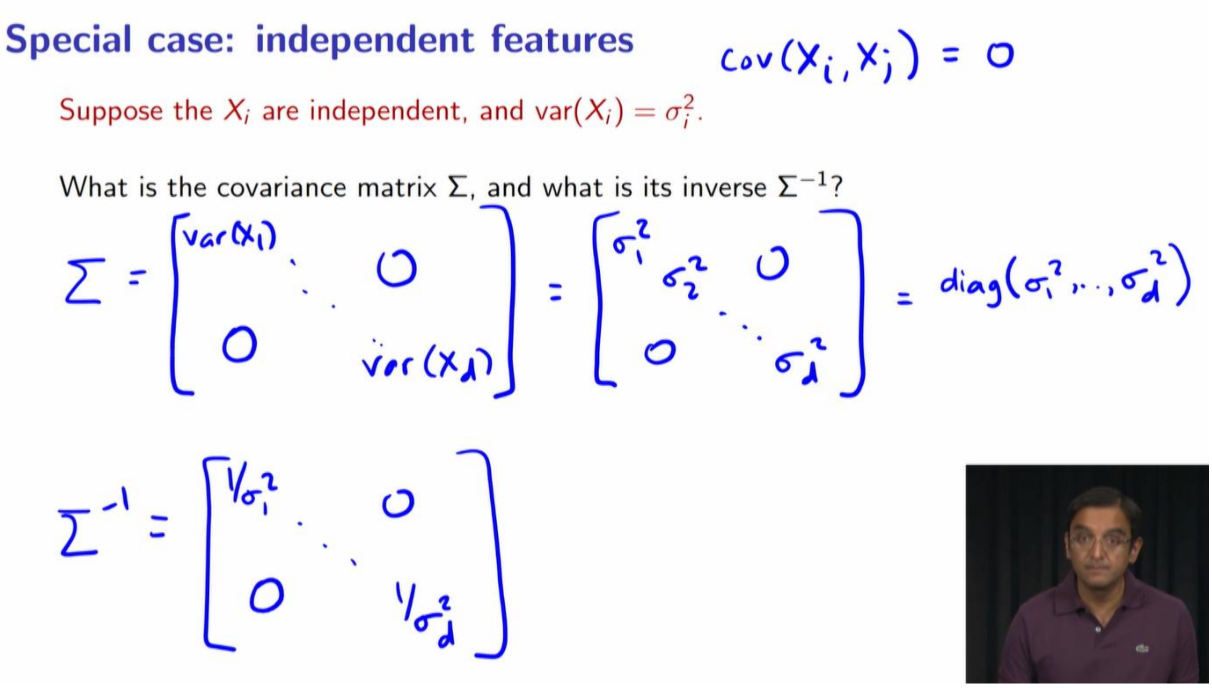
So in this case, the mean of the distribution, mu, will be a d-dimensional vector. It contains the mean along the first feature, the mean along the second feature, all the way to the mean along the d feature.

The covariance matrix, this time, is now a d by d matrix . What does it consist of? So it's a d by d matrix, and along the diagonal, it has the variances of each of the individual features. So these are d numbers, along the diagonal. The numbers off the diagonals are covariances between all possible pairs of features. So in the first row for example, we start with the variance of x one, and then we have the covariance between x one and x two. Followed by the covariance between x one and x three, and so on. In general, the i,j entry of this matrix is the covariance between Xi and Xj.

So these are the parameters of the Gaussium. And in terms of the density, it's highest at the mean, as always, and as you move away from the mean, the density folds off in these ellipsoidal contours. The shape of the ellipsoid is given entirely by the covariance matrix sigma. And here is the exact equation for the density. Let's take a brief look at this.

- So the first part of the density is just **a normalization term**. That's only there to make sure that the density integrates to one. And that's nothing unusual and perhaps you remember what this is. That is the determinant of the matrix sigma. So that's just a normalization term, it's a constant in front.
- The most important part of the density is the part that's inside the exponent, because that's the only part that depends on X. So what does this thing mean? What kind of function is this? X minus mu transpose sigma inverse X minus mu. Well to simplify it a little bit, let's just imagine that mu equals zero. So we have a Gaussian that is centered at the origin, instead of being centered at some arbitrary point mu. So mu is zero, let's see what we get. So the first term over there, X minus mu just becomes X. The second term also becomes X, and what is sigma inverse? Well sigma is some d by d matrix so its inverse is some other d by d matrix, so let's just call it M. So we get X transpose M X and that's something that looks familiar. That's a **quadratic function**. So the density depends only on a quadratic function of X. And indeed, **an ellipsoid is a kind of quadratic**.

总结：



Special case: independent features

$\text{cov}(X_i, X_j) = 0$

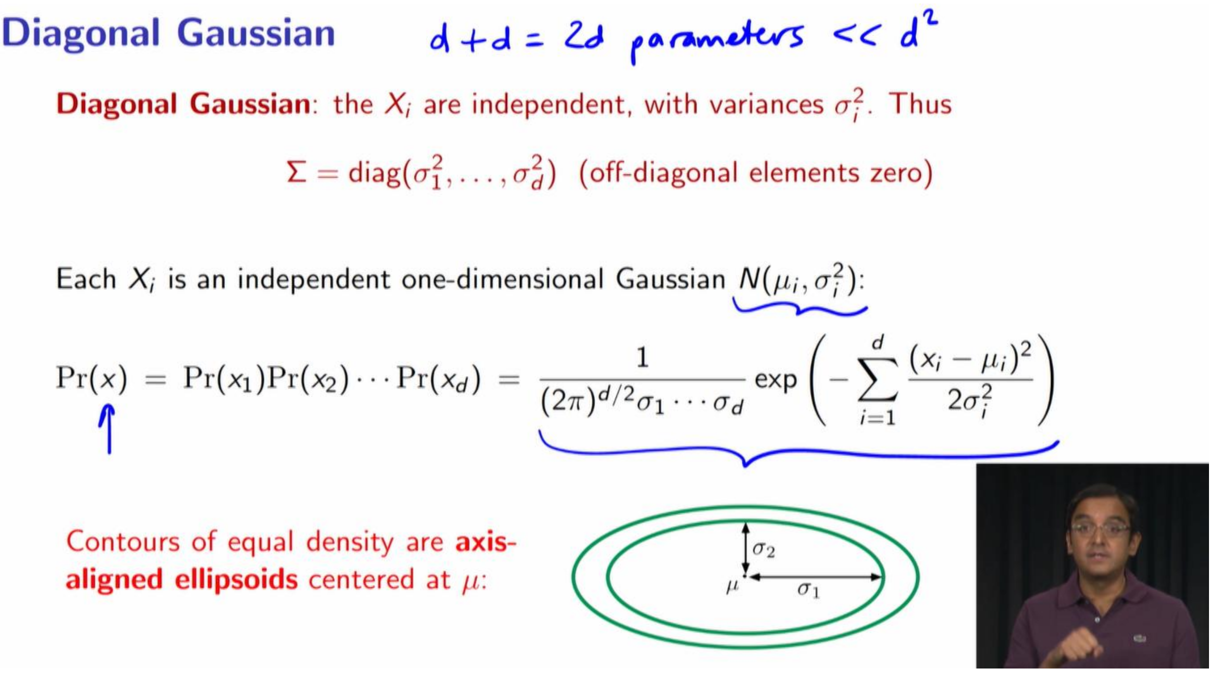
Suppose the X_i are independent, and $\text{var}(X_i) = \sigma_i^2$.

What is the covariance matrix Σ, and what is its inverse Σ^{-1}?

$$\Sigma = \begin{bmatrix} \text{var}(X_1) & & 0 \\ & \ddots & \\ 0 & & \text{var}(X_d) \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_d^2 \end{bmatrix} = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$$

$$\Sigma^{-1} = \begin{bmatrix} 1/\sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_d^2 \end{bmatrix}$$

So now let's go and look at a special case that's of particular interest. So suppose these different features are independent. What does the Gaussian look like then? So we have d features that are independent of each other and let's say that the variance of the I feature, xi, is sigma sub I squared. Let's figure out what the covariance matrix would be in this case. So we got the covariance matrix which is always a d by d matrix, and we know that its diagonal entries are just the variances of the individual features. And the off diagonal entries are covariances between features, okay? So let's see. What is the covariance between xi and xj? Well, in this case, the features are all independent of each other, so they're uncorrelated, which means that the covariance is zero. So the off diagonal elements are all zero. The covariance matrix is a diagonal matrix. And we can fill in the variances, the variance of x one is sigma one squared, then we have sigma two squared, all the way to sigma d squared. A nice short, form for this kind of matrix is just to say the diagonal matrix whose entries are sigma one squared all the say to sigma d squared. Now, in order to compute the density, we also need the inverse of this matrix. What is that? Well, it's easy to invert a diagonal matrix. You just invert each element along the diagonal. So the inverse is one over sigma one squared, all the way to one over sigma d squared. And that's it.



Diagonal Gaussian

$d + d = 2d$ parameters $<< d^2$

Diagonal Gaussian: the X_i are independent, with variances σ_i^2. Thus

$$\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2) \quad \text{(off-diagonal elements zero)}$$

Each X_i is an independent one-dimensional Gaussian $\underbrace{N(\mu_i, \sigma_i^2)}$:

$$\underset{\uparrow}{\Pr(x)} = \Pr(x_1)\Pr(x_2)\cdots\Pr(x_d) = \underbrace{\frac{1}{(2\pi)^{d/2}\sigma_1\cdots\sigma_d} \exp\left(-\sum_{i=1}^{d} \frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)}$$

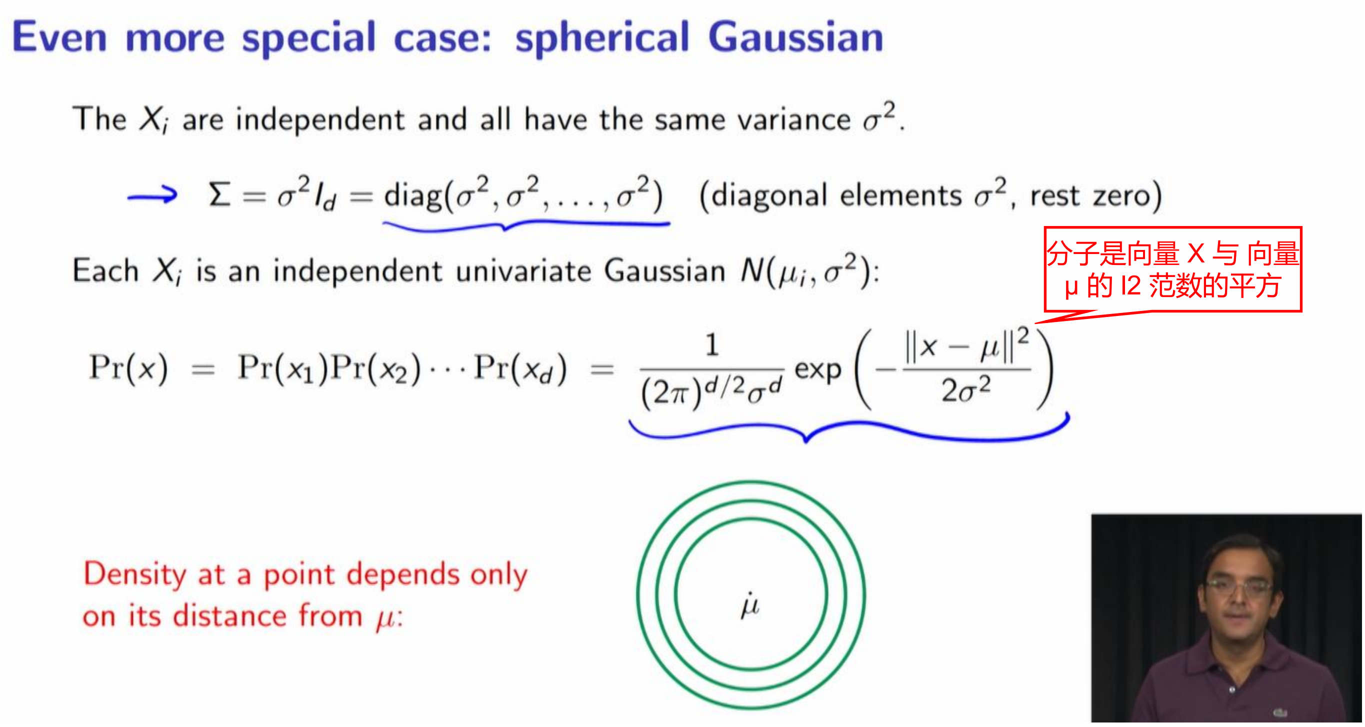
Contours of equal density are **axis-aligned ellipsoids** centered at μ:

So because this covariance matrix is diagonal, we sometimes call this a **diagonal Gaussian**. And covariance matrix being simple makes it rather easy to simplify the density. We end up with something like this. It turns out that each feature, if you look at it just by itself, is a Gaussian, it's the I feature, x sub i, taken just by itself, is a Gaussian with mu sub i and variant sigma i squared. So this is what x sub i by itself looks like, a one dimensional Gaussian. And the density at a d dimensional point x is just the one dimensional density at x one, times the one dimensional density at x two, all the way to the one dimensional density at xd. The d features are all independent. Now as always, the density if highest at the mean, and folds off in these ellipsoidal contours, what do the ellipsoids look like in this case? Well, there are no correlations. The covariance matrix is a diagonal matrix. So because there is no correlation, positive or negative, the ellipsoids are not tilted up or down. **The ellipsoids**

总结:

are axis aligned, that is to say that they are parallel to the coordinate directions. They look something like this. Now they are potentially stretched out differently, along different directions. The stretch along the x one direction is just the standard deviation of x one, which is sigma one. So that's the stretch in this direction. The stretch in the x two direction is the standard deviation of x two, which is sigma two, and so on. So it's a fairly simple distribution.

One of the very nice things about it is that **it can be specified using very few parameters**. So how many parameters do you need for a diagonal Gaussian? Well, the mean is the d numbers because it's a d-dimensional vector. What about the covariance matrix? Well, it's a diagonal matrix, so you only need d numbers. So the total number of parameters is **2d**. This is a huge savings over the general Gaussian, which has a full covariance matrix, and therefore needs something on the order of d squared parameters. **This can be a big convenience, and as a result, the diagonal Gausian is often used even when the features are not independent of each other**.



Even more special case: spherical Gaussian

The X_i are independent and all have the same variance σ^2.

$$\longrightarrow \quad \Sigma = \sigma^2 I_d = \underbrace{\mathrm{diag}(\sigma^2, \sigma^2, \ldots, \sigma^2)} \quad \text{(diagonal elements } \sigma^2\text{, rest zero)}$$

Each X_i is an independent univariate Gaussian $N(\mu_i, \sigma^2)$:

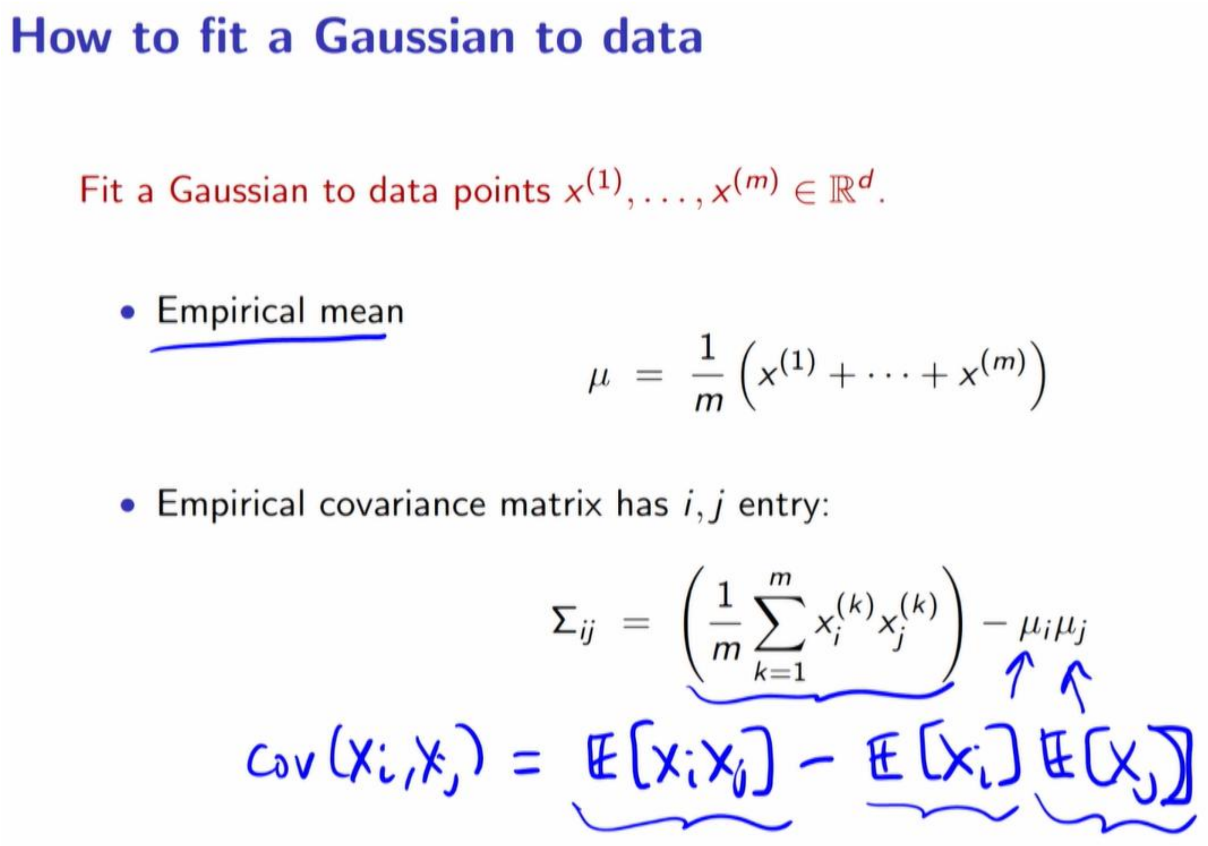
分子是向量 X 与 向量 μ 的 l2 范数的平方

$$\Pr(x) = \Pr(x_1)\Pr(x_2)\cdots\Pr(x_d) = \underbrace{\frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{\|x-\mu\|^2}{2\sigma^2}\right)}$$

Density at a point depends only on its distance from μ:

Now let's look at an even more special case that's also very popular. This is when the features are independent as before, but in addition, they all have the same variance. So now the covariance matrix is a diagonal matrix and in addition, the numbers along the diagonal are all exactly the same. In other words, the covariance matrix is a multiple of the identity matrix. So the covariance matrix looks like this, a multiple of the identity matrix. **In this case, the density can be simplified even further.** This is the density function, and as you can see, the density at point x depends only on the distance from x to mu. **It depends only on the distance to the center of the Gaussian. So the points of equal density are points that lie on a sphere, centered at mu**. So as always, the density is highest at mu, and as you move away from mu, the shells of equal density are these concentric spheres (同心圆). So another popular Gaussian. Okay, so we have a little bit of insight now into multivariant Gaussians.

总结:



How to fit a Gaussian to data

Fit a Gaussian to data points $x^{(1)}, \ldots, x^{(m)} \in \mathbb{R}^d$.

- Empirical mean
$$\mu = \frac{1}{m}\left(x^{(1)} + \cdots + x^{(m)}\right)$$
- Empirical covariance matrix has i, j entry:
$$\Sigma_{ij} = \underbrace{\left(\frac{1}{m}\sum_{k=1}^{m} x_i^{(k)} x_j^{(k)}\right)}_{} - \mu_i \mu_j$$

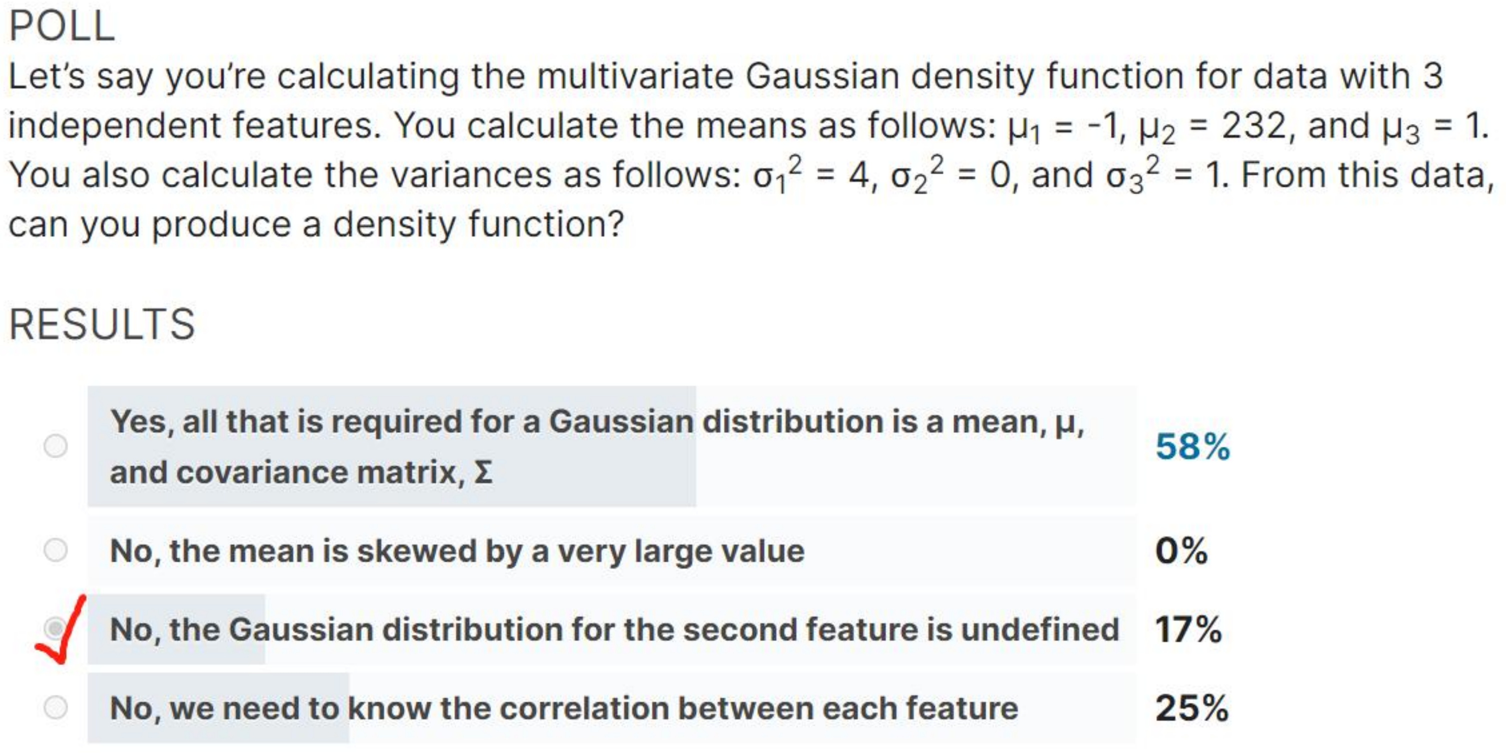
$$\mathrm{cov}(x_i, x_j) = \underbrace{\mathbb{E}[x_i x_j]}_{} - \underbrace{\mathbb{E}[x_i]}_{}\,\underbrace{\mathbb{E}[x_j]}_{}$$

Let's turn to a somewhat more concrete and practical question, which is how do you fit a Gaussian into data? So we have a bunch of data points in d dimensional space, let's call them x one through x m. How would you go about fitting a Gaussian to them? Well, it's quite simple. Just compute the mean and covariance of these points. And those are the parameters, mu and sigma of the Gaussian.

- So, the mean of a bunch of data points is often called an **empirical mean**. So you take the points, these m data points, and you just add them up, and you divide by m. And this is some d dimensional vector that's the mean mu.
- The covariance matrix, finding the empirical version of this is also very simple. You just apply the formula for covariance to the data cell. So how exactly does that work out? Well, we now have this d to t by d matrix. What is the ij entry of the matrix? It's the covariance between xi and xj. And what was the formula for covariance again? Well, the covariance between xi and xj is the expected value of xi times xj minus the expected of xi times the expected value of xj. What is the expected value of xi? Well, we've already computed that, that's mu sub i. What is the expected value of xj? That's mu sub j, So all we need to to do is compute the average value of xi times xj in the data set, and that's exactly what this term is over here.

So that's it for the multivariant Gaussian. This is one of the absolutely most popular distributions for modeling high dimensional data and it is now part of your repertoire.

总结：



POLL

Let’s say you’re calculating the multivariate Gaussian density function for data with 3 independent features. You calculate the means as follows: $\mu_1 = -1$, $\mu_2 = 232$, and $\mu_3 = 1$. You also calculate the variances as follows: $\sigma_1^2 = 4$, $\sigma_2^2 = 0$, and $\sigma_3^2 = 1$. From this data, can you produce a density function?

RESULTS

○ **Yes, all that is required for a Gaussian distribution is a mean, μ, and covariance matrix, Σ** 58%

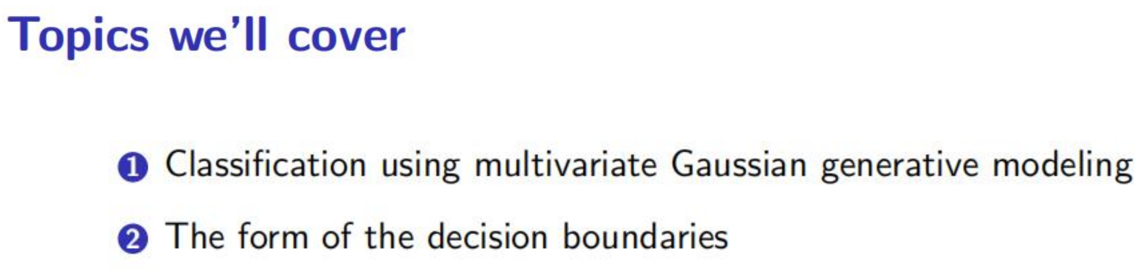
○ **No, the mean is skewed by a very large value** 0%

✓ ● **No, the Gaussian distribution for the second feature is undefined** 17%

○ **No, we need to know the correlation between each feature** 25%

总结：

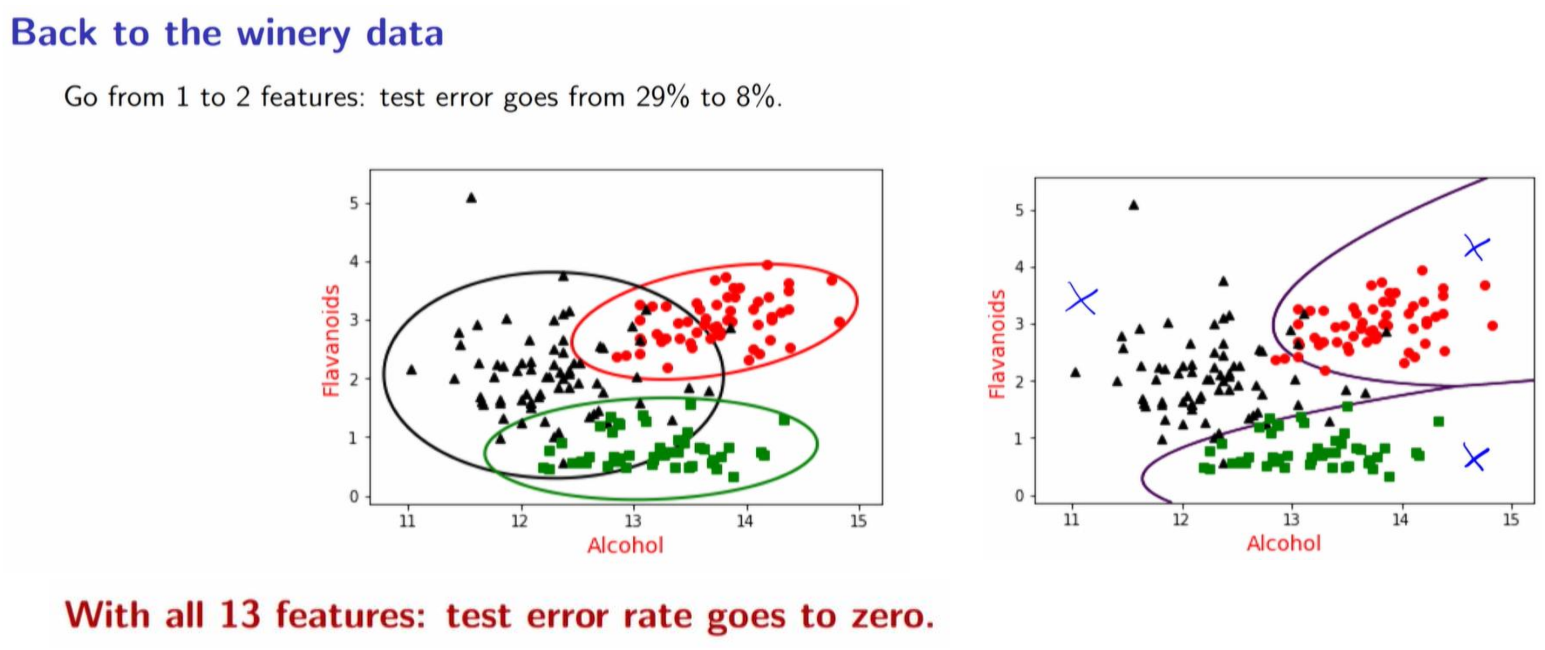
3.5 Gaussian Generative Models



Topics we'll cover

1. Classification using multivariate Gaussian generative modeling
2. The form of the decision boundaries

So we have just seen the multivariate Gaussian, a popular and powerful probability distribution for data of arbitrary dimension. Today, we'll look at using these distributions to build classifiers. Now, we've gone over the generative approach to classification, in which we fit a probability distribution to each class individually. What we'll look at today is the kinds of decision boundaries that result from doing this.



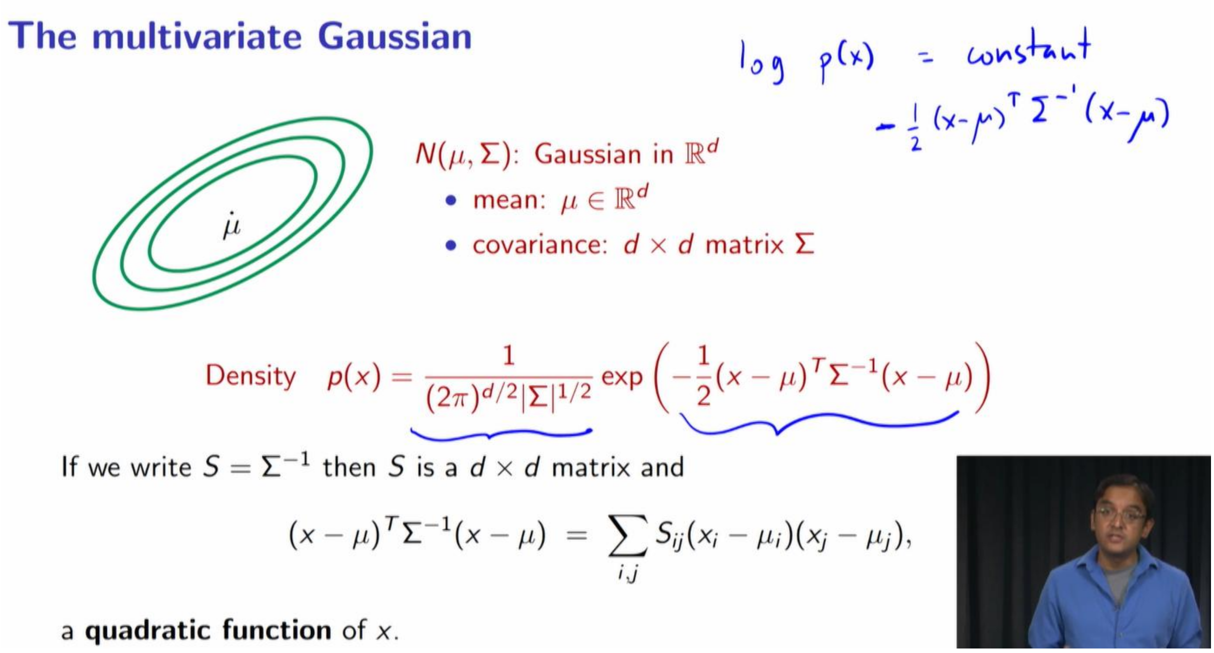
Back to the winery data

Go from 1 to 2 features: test error goes from 29% to 8%.

With all 13 features: test error rate goes to zero.

So let's start by going back to the winery data. If you remember, this was a dataset with three classes, representing three different wineries and 13 features. We started by just picking one of the features, alcohol level, and we've fit a one-dimensional Gaussian to this feature for each of the three classes. The resulting classifier had an error rate of 29% on the test set, not very good at all. Then we added a second feature, flavonoids, and we've fit a bivariate Gaussian to each class, and that's what you see over here, the three ellipsoids that resulted. And the classification boundary, the decision boundary turned out to be this. So, the points over here get classified as red, the points over here get classified as green, the points over here get classified as black, and adding the second feature made the test error drop from 29% to just 8%. But now that we have the multivariate Gaussian under control, we can use all 13 of the features. So we can fit a multivariate Gaussian to the 13 features for winery one and an other one for winery two, and another one for winery three. And then apply Bayes Rule for Classification. If we do that, we see that the test error actually goes down to zero. Does that mean that it's a perfect classifier? No, it doesn't really mean that, but one thing the test said, it's not that large, it consists just of 48 points. But **it does mean that it's a much better classifier than what we were able to obtain using just one or two features**, and **it's a sign that it can be very beneficial to include multiple relevant features**. So now, let's look at the decision boundaries that you get in general from Gaussian generative modeling.

总结：



The multivariate Gaussian

$$\log p(x) = \text{constant} - \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)$$

$N(\mu, \Sigma)$: Gaussian in $\mathbb{R}^d$

- mean: $\mu \in \mathbb{R}^d$
- covariance: $d \times d$ matrix Σ

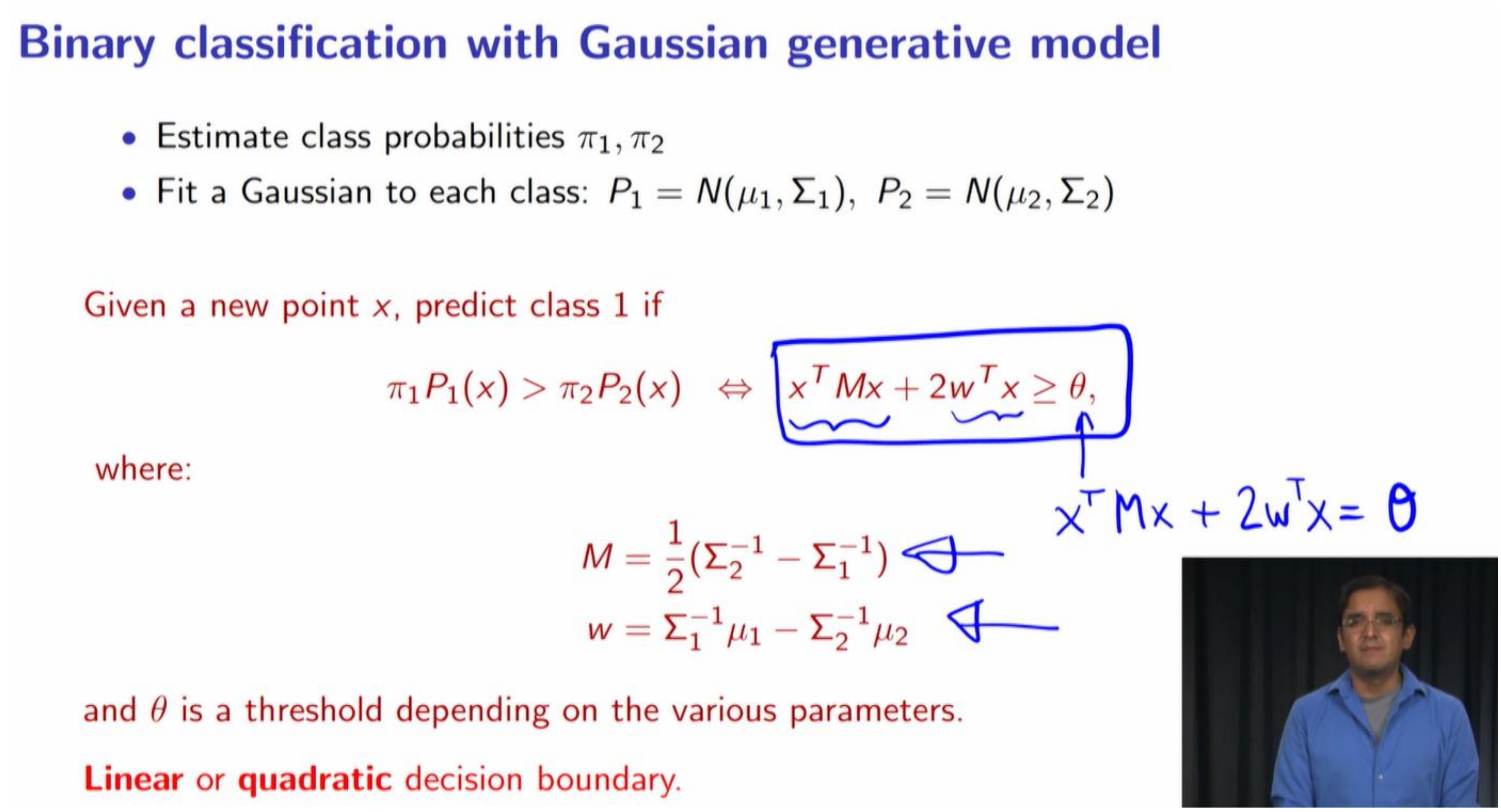
$$\text{Density} \quad p(x) = \underbrace{\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}} \exp\left(\underbrace{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}\right)$$

If we write $S = \Sigma^{-1}$ then S is a $d \times d$ matrix and

$$(x-\mu)^T\Sigma^{-1}(x-\mu) = \sum_{i,j} S_{ij}(x_i-\mu_i)(x_j-\mu_j),$$

a **quadratic function** of x.

The multivariate Gaussian has this density that we've gone over. As we saw, the first part of the density is really just a normalizing factor. The important part is the stuff inside the exponent, which is a quadratic function. Now, that quadratic is in the exponent. Is there a way to bring it down? There is, if you want to bring it down, what you need to do is to **take the logarithm of the density**. Let's see what happens if we do that. Let's take log of p of x. The first part just becomes some constant. The part in the exponent then becomes negative 1/2 x minus mu transpose sigma inverse x minus mu. Log of px is a bona fide quadratic function. As we'll see, what this means is if you use Gaussian distributions in the generative approach to classification, the decision boundaries you get will, in general, be quadratic.



Binary classification with Gaussian generative model

- Estimate class probabilities π_1, π_2
- Fit a Gaussian to each class: $P_1 = N(\mu_1, \Sigma_1)$, $P_2 = N(\mu_2, \Sigma_2)$

Given a new point x, predict class 1 if

$$\pi_1 P_1(x) > \pi_2 P_2(x) \quad \Leftrightarrow \quad \boxed{\underbrace{x^T M x} + 2\underbrace{w^T x} \geq \theta,}$$

$$x^T M x + 2w^T x = \theta$$

where:

$$M = \frac{1}{2}(\Sigma_2^{-1} - \Sigma_1^{-1})$$

$$w = \Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2$$

and θ is a threshold depending on the various parameters.

Linear or **quadratic** decision boundary.

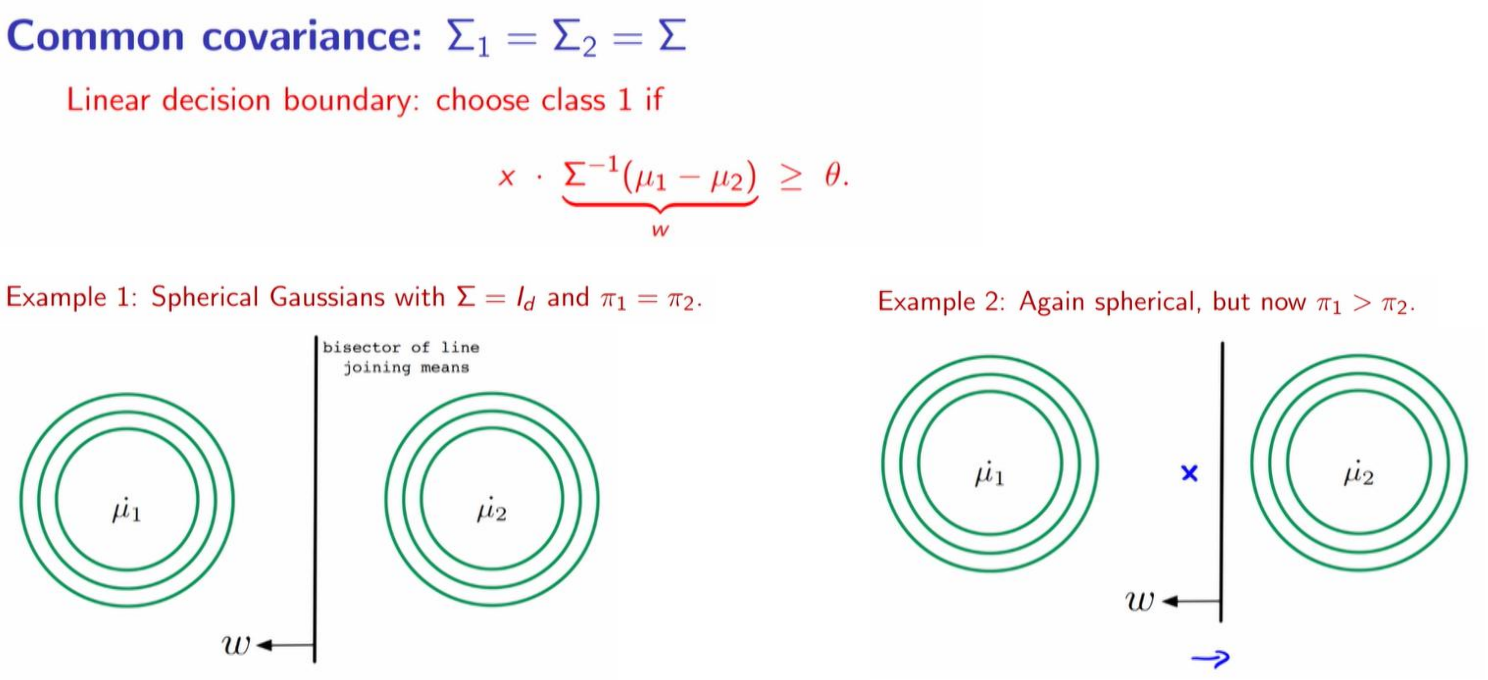
Let's see a little bit more closely why that's the case. Let's look at a binary situation for simplicity. There are just two classes, and the first thing we do in the generative approach is to estimate the class probabilities, pi one and pi two. So maybe 60% of the data is from class one, so pi one is .6, and pi two is .4. Then we fit a Gaussian to each class, pi one and pi two, and now, we're letting them be arbitrary multivariate Gaussians. So these are the parameters of the model. Now, when a new point x comes along, in order to classify x, we compute pi one times p one of x, and we compute pi two times p two of x, and we look at which of these two is larger. So, that's our classification rule. Now, what we can do is we can plug in the formula for the Gaussian density into p one and into p two, and then we can take the log of both sides, which brings the quadratic functions down, and if we do some algebraic simplifications, it turns out that this is what we get. This is the decision rule. So let's see what this is. (后续)

总结：

So first, we have x transpose Mx, and as we've seen, this generically a quadratic function.

- So M is a matrix and of that, we can say precisely what it is. It's just the difference of the inverse covariance matrices.
- The second term is twice x transpose x, so that's the same as w dot x, and so that's a linear function. And w at the end can just be read off (读出) from the means and covariances.
- And then, there's a threshold theta, which is just a number like .2 or negative .6. And that again can be read off from the parameters, including pi one and pi two.

So what happens is, when you use this rule, some points get classified as class one, some points as class two, and the decision boundary is the separating region between these two zones. The decision boundary corresponds exactly to this equation, X transpose MX plus two w transpose X equals theta. And this is a quadratic boundary. Now, it turns out that, sometimes, the matrix M is zero. When will this happen? Well, it would happen, looking at the equation form, **it** would happen if the two covariance matrices are identical. In that case, M drops out, the quadratic term drops out, and we **just get a linear decision boundary**. Now, **a linear boundary is a special case, a degenerate case of a quadratic boundary**, and let's start by looking at that linear case.



Common covariance: $\Sigma_1 = \Sigma_2 = \Sigma$

Linear decision boundary: choose class 1 if

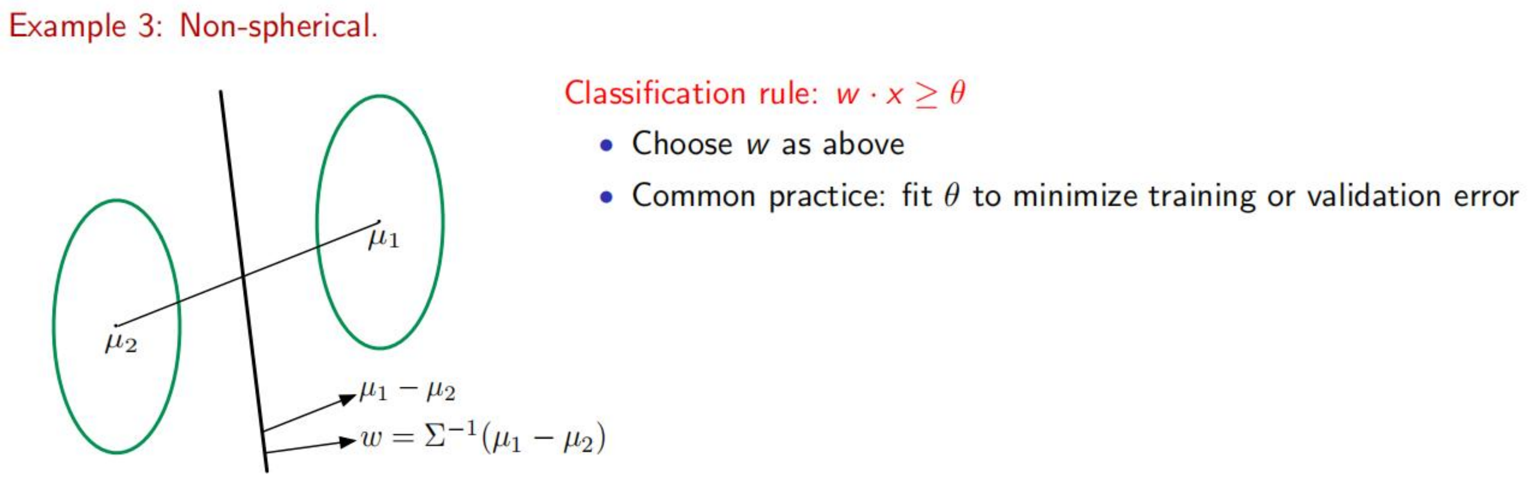
$$x \cdot \underbrace{\Sigma^{-1}(\mu_1 - \mu_2)}_{w} \geq \theta.$$

Example 1: Spherical Gaussians with $\Sigma = I_d$ and $\pi_1 = \pi_2$.

Example 2: Again spherical, but now $\pi_1 > \pi_2$.

So, the linear boundary arises when the two classes have exactly the same covariance matrix, sigma one equals sigma two. Let's see an example of that. Let's say that each class is a spherical Gaussian, and they had exactly the same covariance matrix, say, the identity matrix. And two make the situation even more symmetric, let's say that the two classes had exactly the same class probability. So pi one equals pi two equals .5. What is the decision boundary in that case? Well, it's just the perpendicular bisector between the centers of the two classes, between the two Gaussian means. So the center of class one is mu of one, the center of class two is mu of two, and the perpendicular bisector between them is the decision boundary, and is exactly what one would expect, given the extreme symmetry of this case. But now, let's if we were to vary it a little bit. Let say that we change this one thing. Currently, we're saying that the class probabilities are equal, they're both .5. What if class one is a little bit more likely, say, pi one equals .6 and p two equals .4, what happens then? So here's what happens. The decision boundary just moves slightly to the right, and that's in pi this is. So if you look at the points smack between the two centers, this point over here, this point would now get classified as class one just because class one occurs more frequently than class two. And so, the decision boundary just get shifted slightly to the right to take care of this discrepancy in prior class probabilities.

总结：



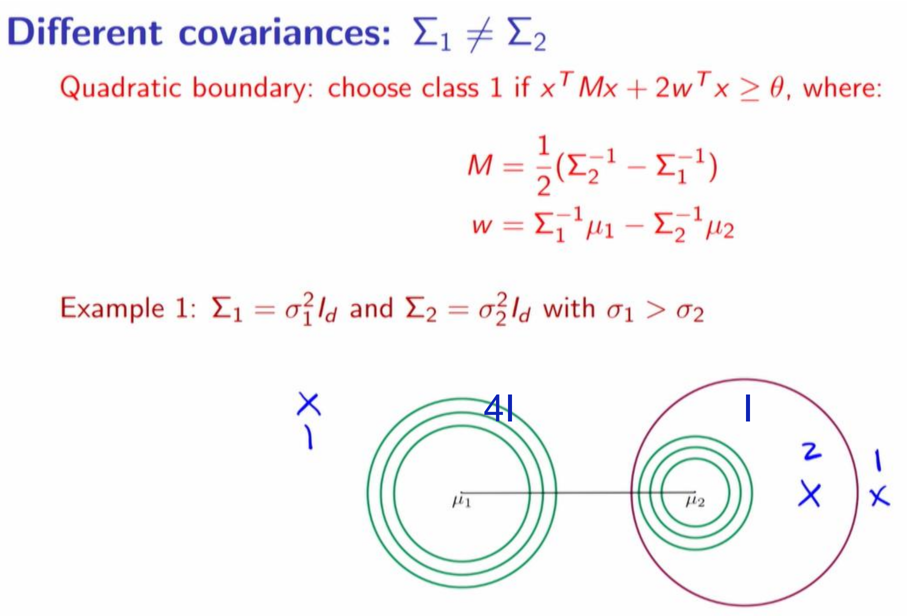
Example 3: Non-spherical.

Classification rule: $w \cdot x \geq \theta$

- Choose w as above
- Common practice: fit θ to minimize training or validation error

Now, what if the two classes have the same covariance matrix, but they're not spherical? That's what we see in this picture here. Because the covariance matrices are the same, the boundary's again linear, but now, it's not the perpendicular bisector any longer, it's slightly skewed to accommodate the different shape. So in all of these cases, the classification rule is a simple linear rule of the form w dot x greater or equal to theta, and **the vector w and the number theta can just be read off from the parameters of the model, from the means, the covariances, and the weights, pi one and pi two**. Now, it turns out that the way in which these models are often used is to set w in this way, but to then allow theta, just that number, that threshold, to vary a little, to tweak it a little to maximize performance on the training set or on some validation set. So, in other words, you choose exactly this boundary, The one shown over here, but you allow it to shift parallel to itself slightly to the left or right if that helps performance.

Okay, so this is the case where the covariances are equal. What is they aren't equal?

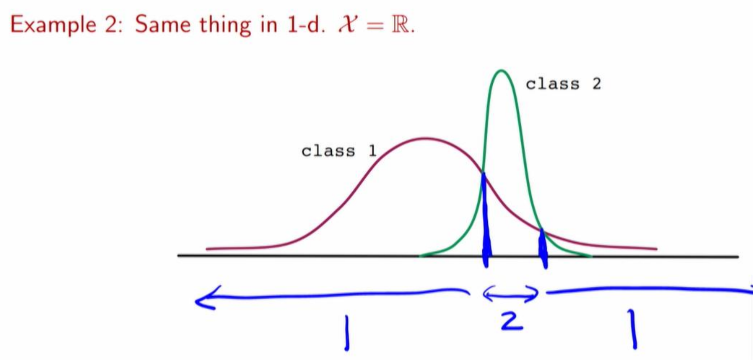


Different covariances: $\Sigma_1 \neq \Sigma_2$

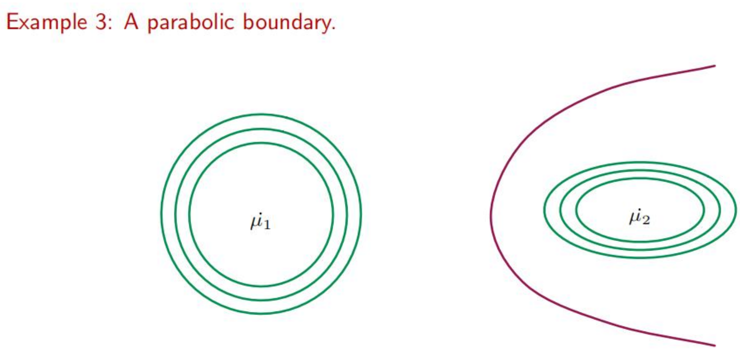
Quadratic boundary: choose class 1 if $x^T M x + 2w^T x \geq \theta$, where:

$$M = \frac{1}{2}(\Sigma_2^{-1} - \Sigma_1^{-1})$$
$$w = \Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2$$

Example 1: $\Sigma_1 = \sigma_1^2 I_d$ and $\Sigma_2 = \sigma_2^2 I_d$ with $\sigma_1 > \sigma_2$



Example 2: Same thing in 1-d. $\mathcal{X} = \mathbb{R}$.



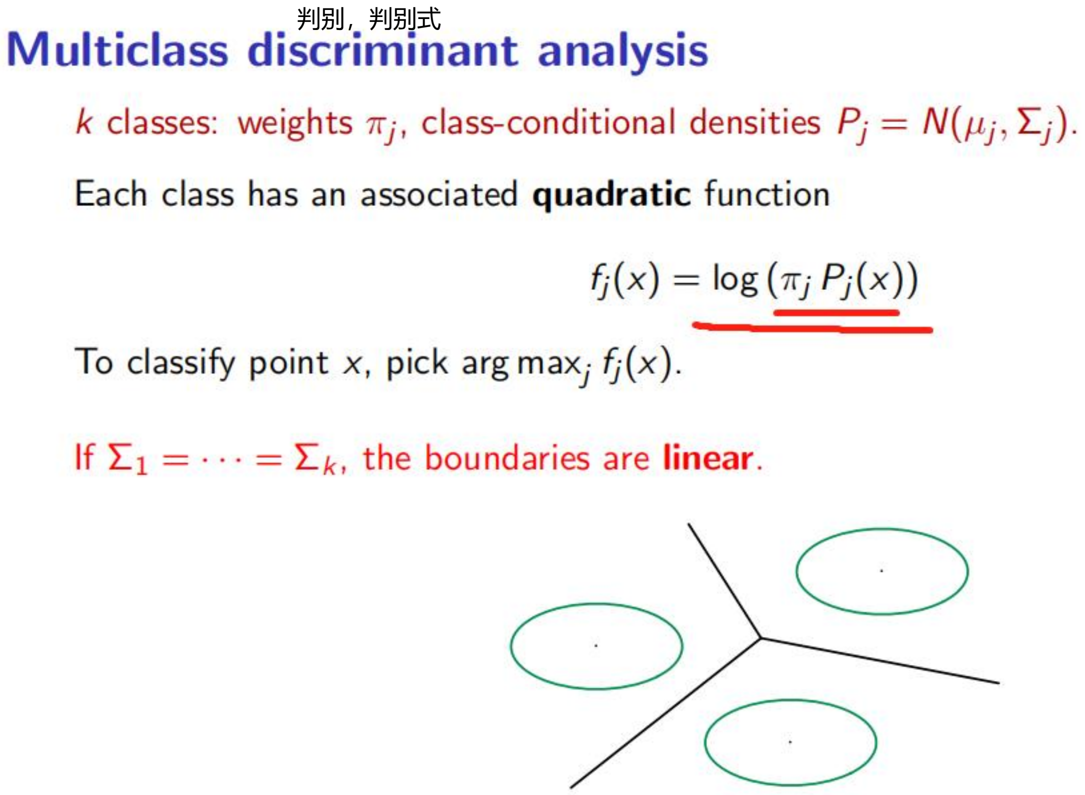
Example 3: A parabolic boundary.

If the covariances aren't equal,then the boundary is a general quadratic. And let's see some examples of that.

- So, in this first example over here, the two classes are both spherical Gaussians, so that's nice and simple, but they don't have the same covariance matrix. So the one on the left has got a larger variance, so maybe it's covariance matrix is four times the identity, whereas the one on the right might just be the identity. So the variance is higher on the left. What kind of decision boundary do we get in this case? Let's take a look. So here's what results. The decision boundary is a sphere. Points inside the purple sphere get classified as being class two, whereas points outside the sphere are class one. So this gets classified as two, this gets classified as one, and this point over here also gets classified as one. Seems a little strange, right? Well, perhaps it will be a little clearer if we look at a one-dimensional analog of this. (后续)

总结:

- So let's look at roughly the same picture in 1-d. So, here we have two Gaussians and one dimension, and one of them has got a very large variance, the one for class one, whereas the other for class two has a very small variance. What is the decision boundary? The data is one-dimensional. Let's say **the weights are the same, pi one equals pi two equals .5**. In that case, this would be the boundary. You look at where the densities overlap. This is the boundary. Everything over here is class one, everything over here also gets classified as one, and the stuff in the middle over here gets classified as two.
- Here's another example. So one of the classes is a spherical Gaussian, and the other one is a diagonal Gaussian. So you have an ellipsoidal shape because it's diagonal. It's ellipsoidal but parallel to the axes. And, in this case, the decision boundary turns out to be a **parabola**. **And there are lots of other ways the decision boundary could look, but they're all quadratic forms of one kind or another**.



判别，判别式

Multiclass discriminant analysis

k classes: weights π_j, class-conditional densities $P_j = N(\mu_j, \Sigma_j)$.

Each class has an associated **quadratic** function

$$f_j(x) = \underline{\log(\underline{\pi_j P_j(x)})}$$

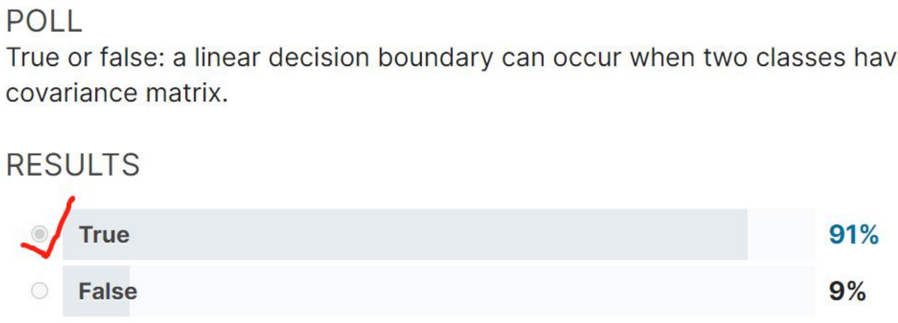
To classify point x, pick $\arg\max_j f_j(x)$.

If $\Sigma_1 = \cdots = \Sigma_k$, the boundaries are **linear**.

So we've talked mostly about the binary case, and if the case where there are just two labels, and this is because it's particularly simple to then talk about the boundary between these two regions, class one and class two. **But when there are multiple labels, the overall picture is roughly the same**. So let's say that there are k classes. So what we do is that, for each class, we fit a weight, Pi of j, that's just a number, and we fit a Gaussian density, P of j. So what's our classification rule in this case? When a new point x comes along, we compute pi j times P j of x, and we pick the class j for each this is largest. We can equivalently pick the class j for which the **log of this is the largest. Doesn't change the outcome, and it can be a little bit easier to think of** because, as we saw a little earlier, **log of the Gaussian density is a quadratic function**. So what happens essentially is that each of the k classes has got its own quadratic function. And when a new point comes along, each class evaluates its quadratic function. They each get a number, and the one with the largest number is the winner. That's the class that gets predicted.

So that's how classification works when there are k classes with general Gaussians. Now, in the situation where **all the covariance matrices of the k classes are the same, then once again, the boundaries become linear**.

So we've talked a lot about generative modeling, about building a classifier by fitting a Gaussian model to each class. This is a very simple approach to classification, and it works extremely well in many settings. But the Gaussian isn't the only distribution that can be used, and next time, we'll look at some other choices.



POLL

True or false: a linear decision boundary can occur when two classes have the same covariance matrix.

RESULTS

✓ ● True 91%

○ False 9%

总结：

3.6 More Generative Modeling

Topics we'll cover

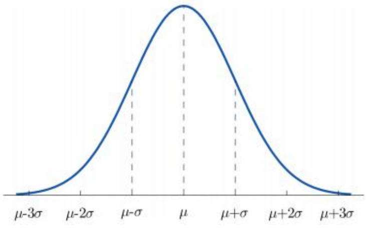
1. Beyond Gaussians
2. A variety of univariate distributions
3. Moving to higher dimension

We've been talking a lot about building classifiersby fitting a gaussian distribution to each class.
What we'll do today is to look beyond gaussians at more general kinds of generative modeling.
We'll start by looking at a variety of other popular one dimensional distributions and then we'll see how we can combine these to deal with higher dimensional data.

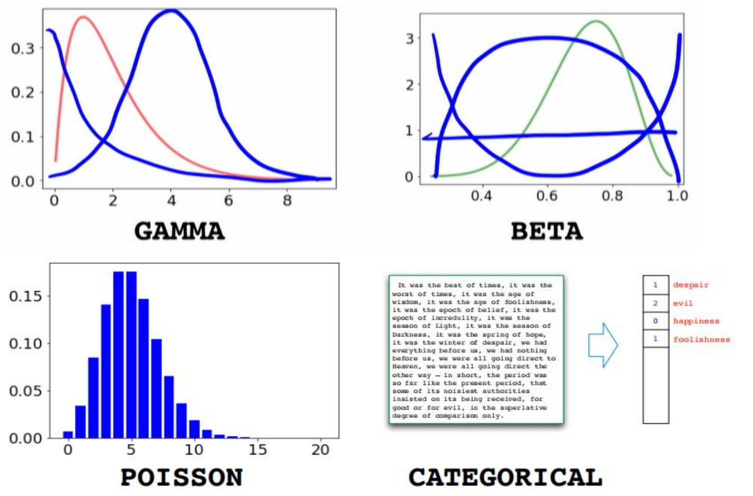
Classification with generative models

- Fit a **distribution** to each class separately
- Use Bayes' rule to classify new data

What distribution to use? Are Gaussians enough?



Exponential families of distributions



It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way – in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

In the generative approach to classification, **the main idea is to fit a distribution to each class separately. When a new data point arises, we classify it by just using Bayes' rule**. Now so far, we focused exclusively on using gaussian distributions for each class. Indeed, this is the single most important case but other situations in which a gaussian distribution would not be suitable, and indeed there are many such situations.

- Suppose that our data is real valued so it's numeric but it's always positive. In such situations, it can sometimes be a little bit strange to use a gaussian just because a gaussian is a distribution over the entire real line, positive and negative. Other distributions that are just over the positive reals, there are. A prime example is **the gamma distribution**, which is shown over here on the upper left. The gamma is specified by two parameters, just like the gaussian and as you vary the parameters you can get a lot of different shapes. You can get something like the red curve shown over here or you can get something like this. You can get something like this and so on. You might have heard of **the exponential distribution** or **the chi squared distribution**. **These are all just special cases of the gamma**.

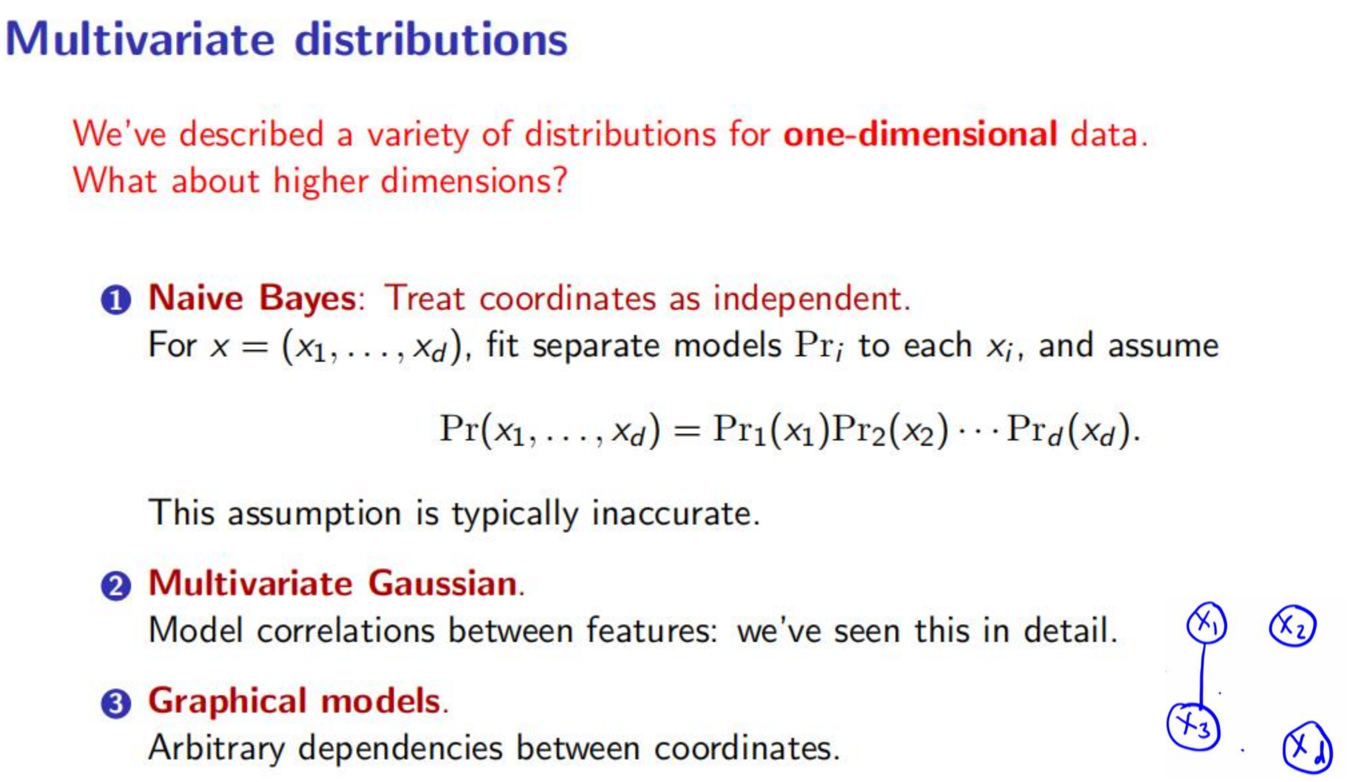
- Here's another situation. What if the data is real-valued but it always lies in a specific interval, like between zero and one? In this case, it can be a little strange to use the gaussian because that's a distribution over all the reals and it would also be strange to use the gamma distribution because that's over all positive reals. Is there a distribution just over an interval? In this case, a very common choice would be **the beta distribution** which is shown over here on the upper right. Like the gaussian and the gamma, it's specified by two parameters and it also takes on a variety of different shapes so it can look like this or it can look like an upside down bowl, or you can get something like a **uniform distribution** or you can even get something like this. Very useful.

- Here's yet another situation. Suppose you have data that's real-valued, it's numeric but all the data points turn out to be integers. This would happen, for instance, if the data consists of counts. How many people went to the emergency room at night? How many earthquakes struck this particular region of California? If the data area positive integers, then the gaussian, the gamma, the beta are all not entirely suitable. Their distributions are of arbitrary real numbers. In this case, the most common choice would be **the poisson distribution**, which is shown over here on the lower left. It's specified by a single parameter, the mean of the distribution and it assigns a probability to every non-negative integer. Zero, one, two, three, and so on. This distribution turns out to be a good fit to all sorts of data.

- Finally, what if we have data that isn't even numericbut there are finitely many possible outcomes? For instance, suppose we want to fit a distribution to the words that appear in a particular corpus of documents so we want the probability of each word. Here there are finitely many possibilities, finitely many words and the distribution we would use is **the categorical distribution**. It's very simple, it just assigns a probability for each specific outcome for each specific word. (后续)

总结：

All these distributions, the gamma, the beta, the poisson, the categorical, these all are useful in situations where the gaussian would not be entirely appropriate and the four of these distributions along with the gaussian turn out to actually be part of a broader class of probability distributions called **exponential families**. The gammas are a particular exponential family. The betas are another exponential family. The gaussians are an exponential family. What are these exponential families? Sadly, this is not something that we're gonna have time to really go into, but briefly exponential families are distributions that have a particular function or form. This form makes them easy to work with, quite simple to estimate for example.



Multivariate distributions

We've described a variety of distributions for **one-dimensional** data.
What about higher dimensions?

1. **Naive Bayes**: Treat coordinates as independent.
   For $x = (x_1, \ldots, x_d)$, fit separate models $\Pr_i$ to each x_i, and assume

   $$\Pr(x_1, \ldots, x_d) = \Pr_1(x_1)\Pr_2(x_2)\cdots\Pr_d(x_d).$$

   This assumption is typically inaccurate.
2. **Multivariate Gaussian**.
   Model correlations between features: we've seen this in detail.
3. **Graphical models**.
   Arbitrary dependencies between coordinates.

We talked about a variety of distributions for one dimensional data. The gamma, the beta, and so on. How do we deal with higher dimensional data? Is there a way to combine these distributions somehow? It turns out there are **basically three standard options**.

- The first is to **pretend that the different coordinates are independent**. This is called **Naive Bayes**. Let's say you have D dimensional data, so with coordinates, x1, x2 all the way to xd.
  - If x1 is arbitrary real values, we can fit a gaussian into it.
  - If x2 consists of integer counts, we can fit a poisson into it.
  - If x3 happens to lie in a fixed interval, we can fit a beta distribution into it.
  - **In this way, we fit a distribution to each individual coordinate** and now, when we want the probability of the entire vector X, we just take the one dimensional probability of x1 times the one dimensional probability of x2 times the one dimensional probability of x3 all the way to the one dimensional probability of the last coordinate, x of d.
  - Now what this is assuming is that these coordinates, x1 through xd, are all independent of each other and that is something that is rarely true. However, despite this false assumption, this particular methodology turns out to often be quite effective in practice so that's Naive Bayes.
- Option number two is to use the **Multivariate Gaussian** which we've talked about a lot. Now **the Multivariate Gaussian does not pretend that the coordinates are independent. It allows us to model pair wise correlations between the coordinates** and as we've seen, it's really quite simple to use.
- The third option is to use a general **Graphical Model**. Now this is sadly outside the scope of this class, but briefly a Graphical Model represents a probability distribution over many coordinates using a graph so if they're d coordinates, there is a node from each coordinate. A node for x1, a node for x2, all the way to a node for xd and the graph has edges between two nodes if those two coordinates have some sort of dependence between them. Now **in full generality, a graphical model allows us to use any distribution for the individual coordinates. X1, x2, et cetera and it also allows us to model arbitrary dependencies between these coordinates.** It's a very powerful methodology.

So that's it for generative modeling. This is a very popular approach to classification. It's simple, efficient, and very easy to understand.

总结：