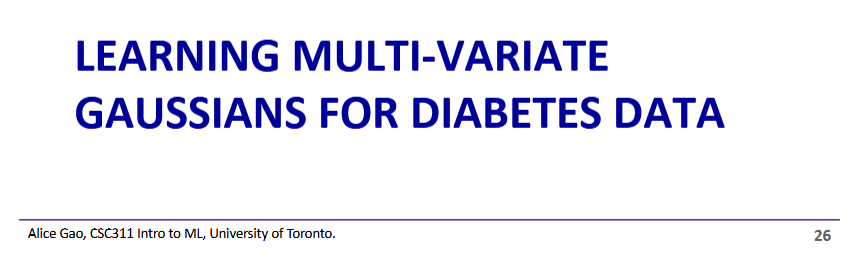
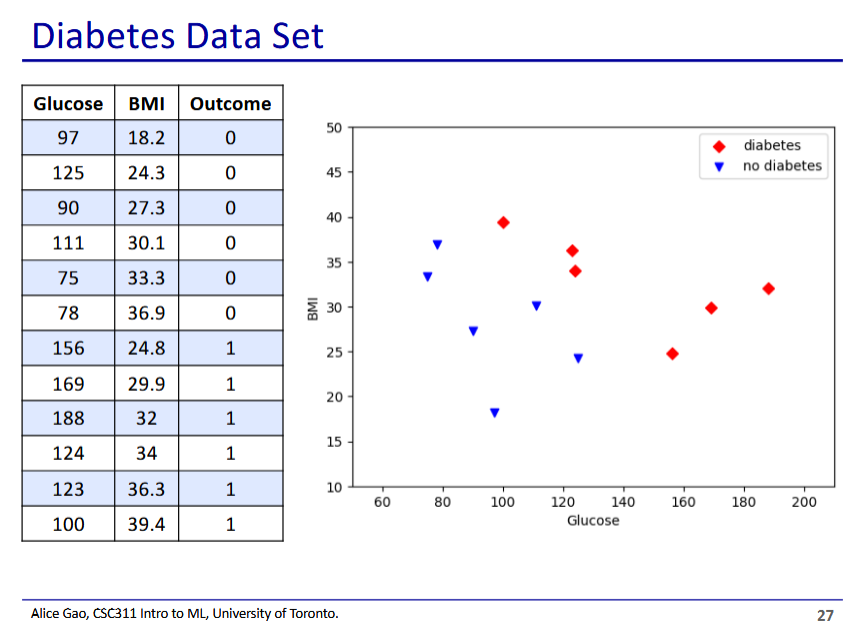
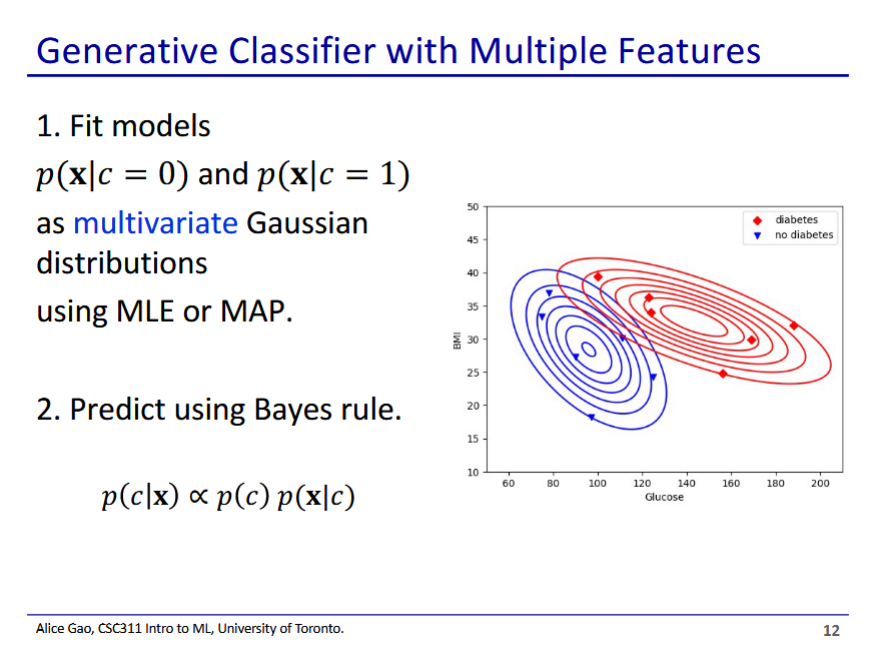
|  |
| --- |
| **Deriving Gaussian mean and covariance derivation (theory)**   1. Derive the log-likelihood of the data  * Then substitute in the univariate or multivariate Gaussian PDF for  1. Get the derivative of the log-likelihood with respect to and to  * Equations for multivariate on slides 23 and 24 * **Do the derivations for univariate as an exercise, may be on exam**  1. Set the derivatives to zero and solve for and  * **For the test, need to know how to do this process for univariate Gaussian only**    + **Multivariate Gaussian derivations will not be on the test** * Process for multivariate Gaussian on slides 19-24 * The best estimates for and are the mean and the covariance of the training data   **Learning multivariate Gaussian (in practice)**   * **The best estimates for and are the mean and the covariance of the training data**   + is an indicator variable for if training example i is in class k   + will be the mean of all training examples with class k   + will be the covariance of all training examples with class k * We can then plot out the distributions for each class   **Understanding covariance via spectral decomposition**   * Covariance affects the shape that the Gaussian distribution takes around the mean   + The exact effects can be understood through spectral decomposition * **Spectral decomposition breaks down the covariance matrix into eigenvalues and eigenvectors**   + - **The column vectors of Q are eigenvectors**     - **The diagonal values of are eigenvalues that correspond to an eigenvector**   + Eigenvectors form an alternative basis for representing the coordinate system   + Applying to a vector scales it along each eigenvector by it’s eigenvalue * **If is a diagonal matrix**   + Elliptical contours of the Gaussian distribution are axis aligned     - Eigenvectors aligned with axes   + Ellipse is scaled along axes according to diagonal values in * **If is not a diagonal matrix**   + Elliptical contours of the Gaussian distribution are aligned with the eigenvectors of   + Ellipse is scaled along the eigenvectors by the eigenvalues (diagonal values in )   + **Examples on slides 18-20** * **On the exam we will not be expected to do spectral decomposition, however we will need to know how the ellipse is scaled after being given the decomposed**   **GDA decision boundary**   * Decision boundaries for GDA is a conic section (circle, ellipse, parabola, hyperbola)   + Derivation on slide 26-28 * **In a unique case can be a linear decision boundary**   + Occurs when the covariance matrix of the different classes are identical   + Derivation on slide 28   **When to use GDA over other models?**   * GDA with a linear decision boundary is very similar to logistic regression, so which do we use? * **GDA makes a fairly strong assumption about the data**   + Assumes data has to be generated from a Gaussian distribution with a mean and covariance   + **GDA works very well when this assumption is satisfied** * If the GDA assumption is not reflected in the data, we may want to use a different model |

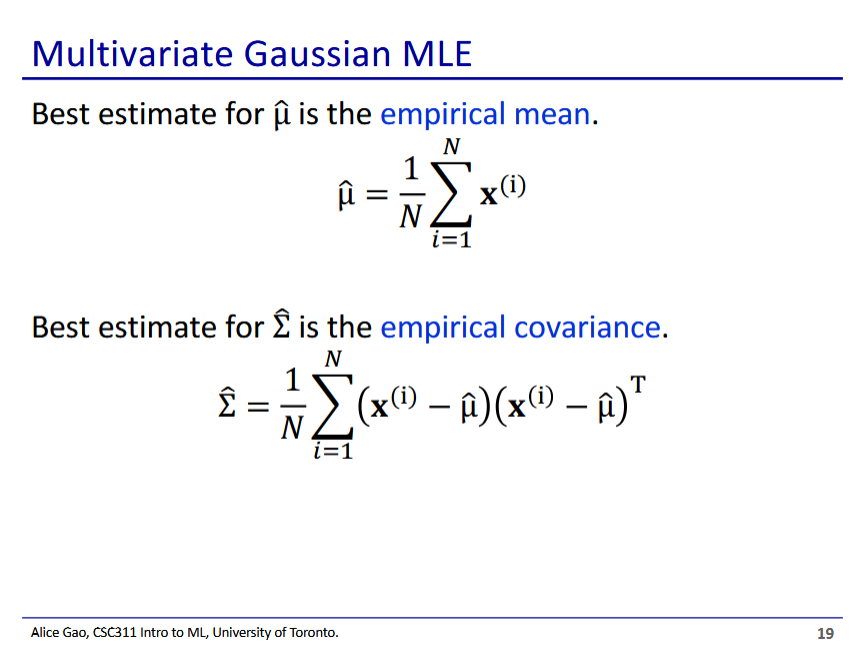




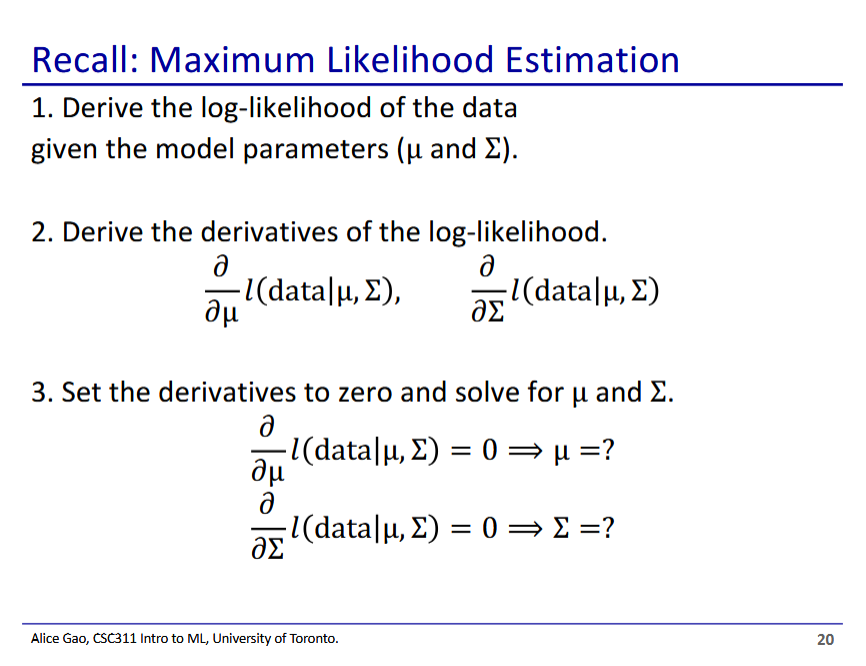
* This diabetes dataset has continuous features, which is why we are using the GDA model
  + Naive Bayes can only work with discrete features
* So how do we fit a model to this data?



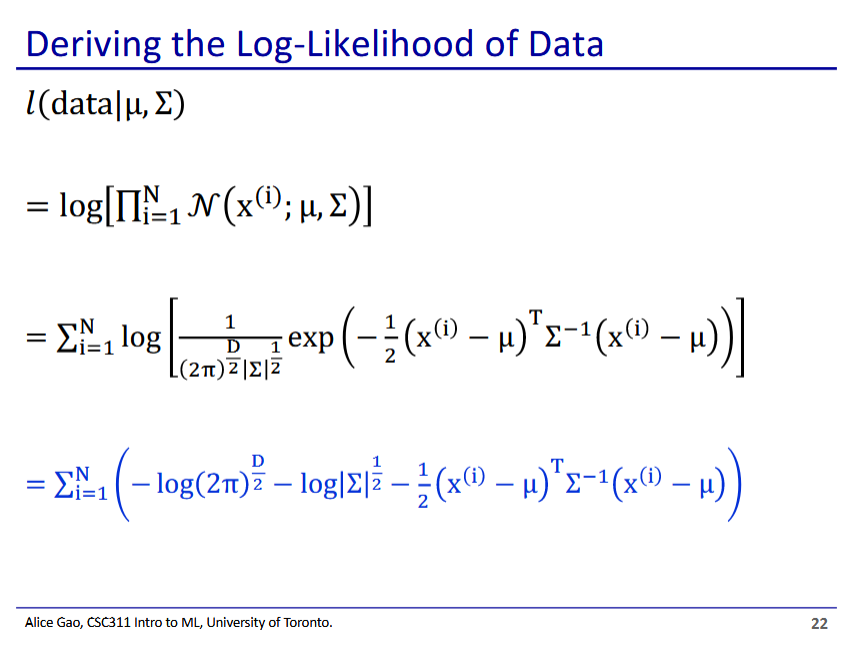
* This is the high-level description of the process for learning a GDA model



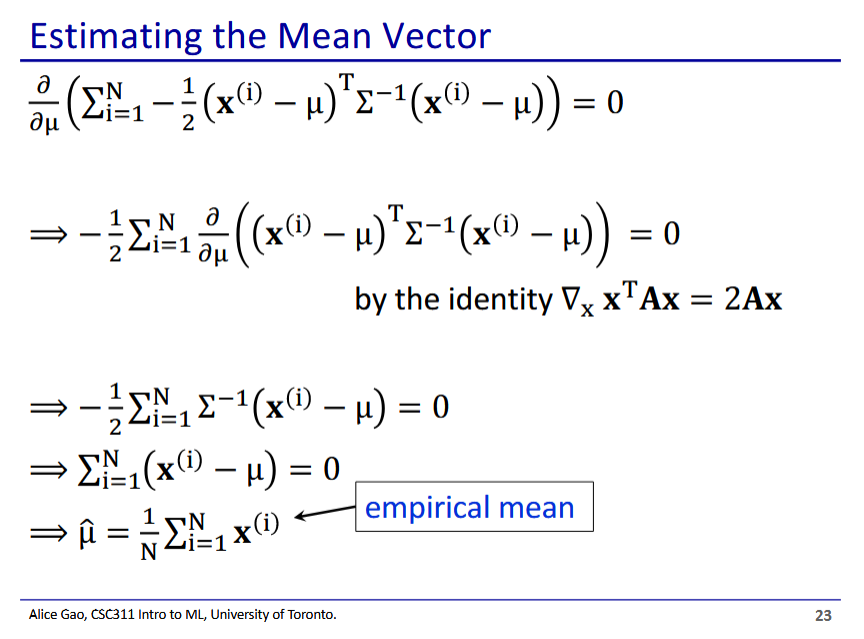
* In order to learn the model, we need to learn a Gaussian distribution for each class
  + Each Gaussian distribution is defined by a mean vector and a covariance matrix
* If we are using Gaussian MLE to learn the distributions, the ideal mean and variance are the empirical mean and variance of the data for each class



* Process for getting to the result on the previous slide: How do we know that the best mean and covariance is the empirical one?
* In order to figure out which mean and covariance matrix is the best, we take the maximum of the log-likelihood
  + To do this we find out when the derivatives is 0

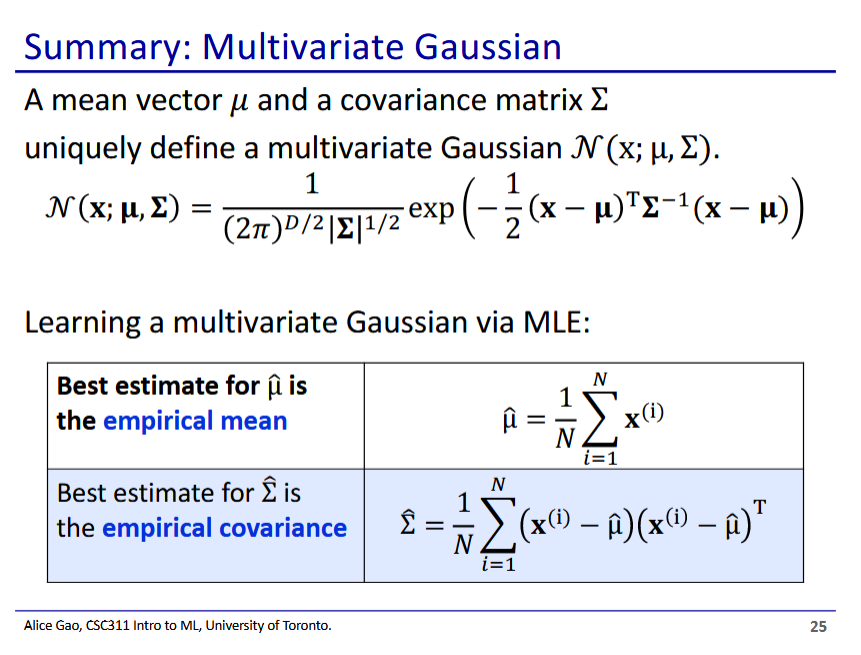


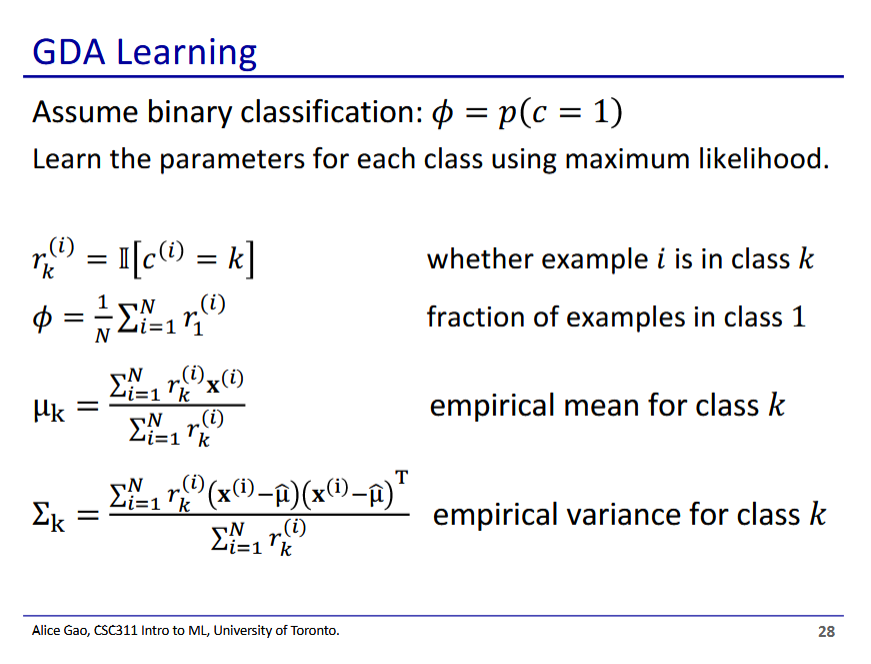
* is the probability of given the probability density function of the gaussian distribution defined by
  + To get the total likelihood of the entire input data we multiply these together for every
    - Definition on slide 16 of GDA part 1 lecture
* The first term of the final derivation has only constants, and thus is not useful when looking for the maximum distribution
* The second and third terms of the final distribution contain either the covariance or the mean and covariance, and thus are relevant

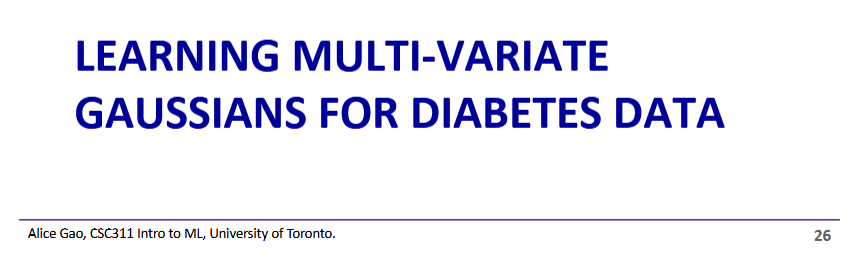


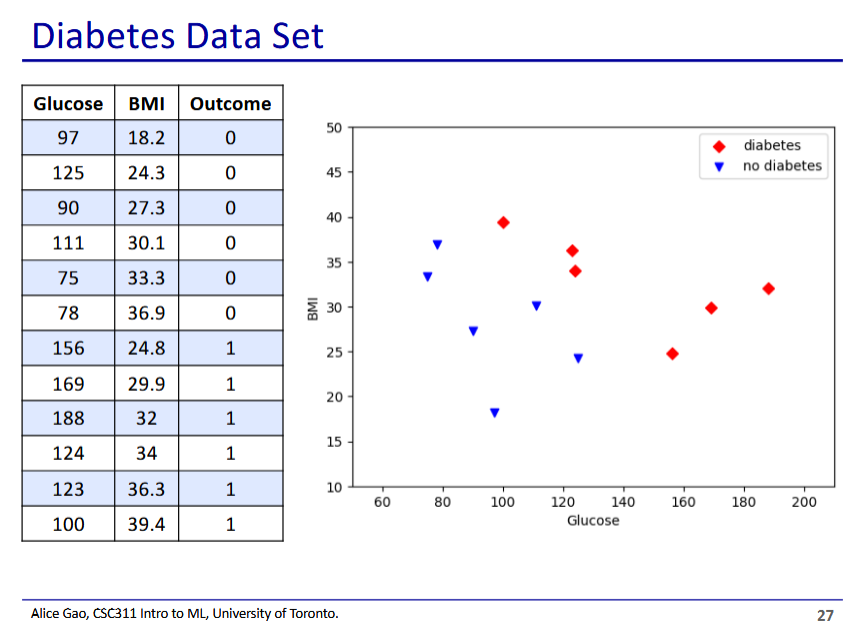
* First we try to find the ideal mean vectorA diagram of mathematical equations

  Description automatically generated
  + We do this by taking the partial derivative of the log-likelihood with respect to and then equating it to 0
* Only the third term of the log-likelihood contains , and so is the only relevant term
* Derivation is fairly straightforward if you have the linear algebra identity used
  + Shows that the ideal mean is the empirical mean
* Derivation not shown, its messy
* Nonvectorised version is on top, vectorised version is on the bottom
  + Take some time to figure out how the vectorised version works
* Shows that the ideal covariance is indeed the empirical covariance
* **On the test, we will not need to do the derivations for multivariate Gaussian**
  + **We may however need to derive for the univariate Gaussian distribution**

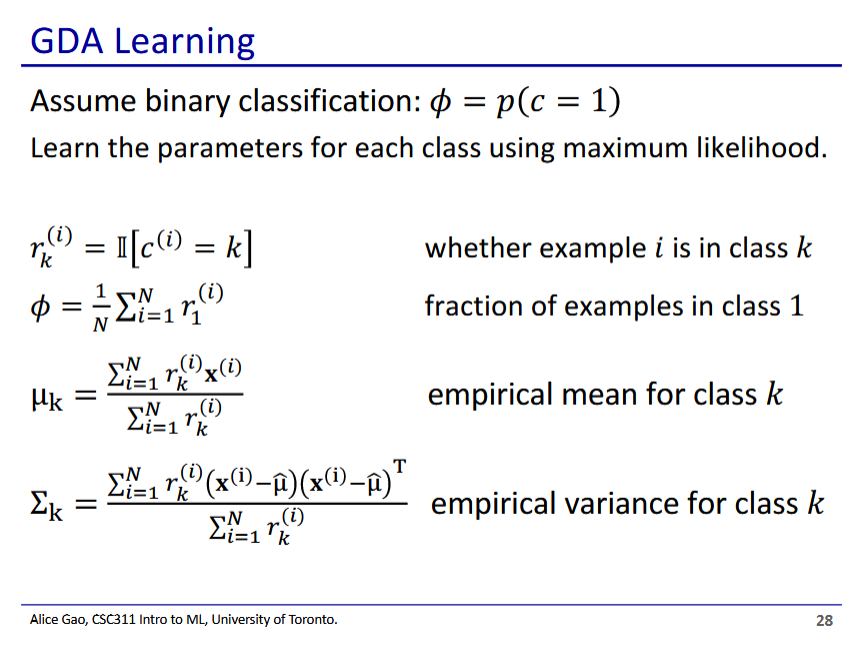




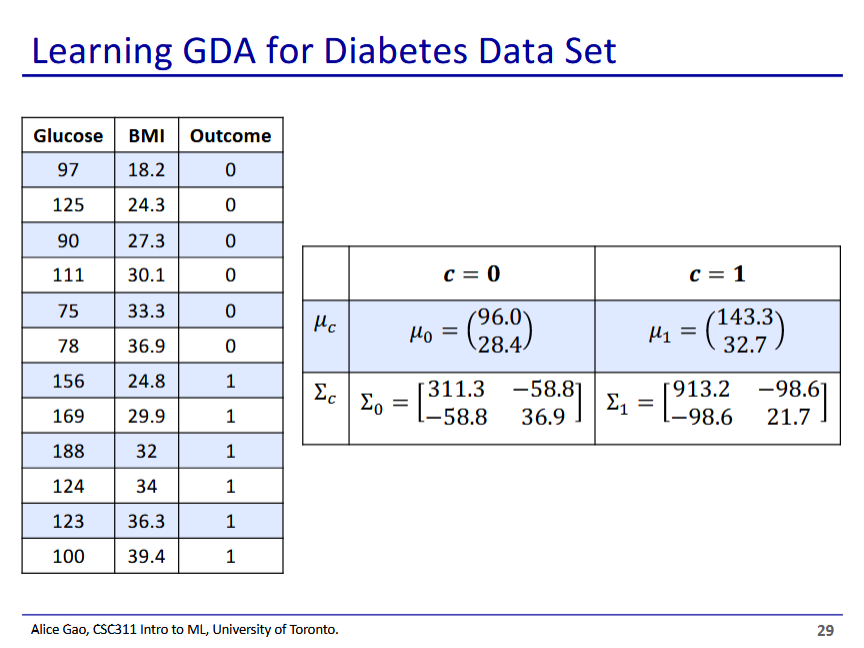




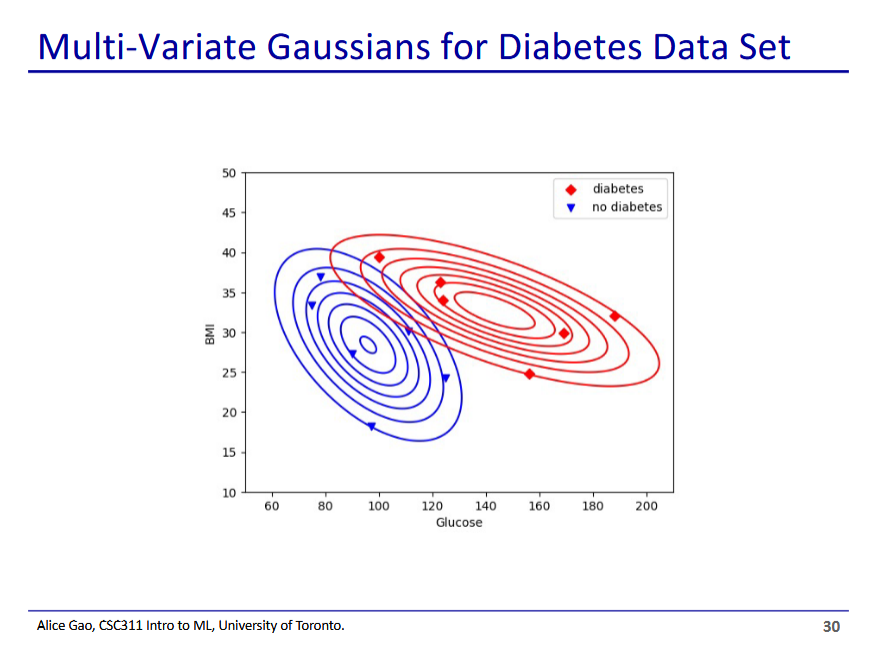
* Now back to the data, let’s learn the GDA model



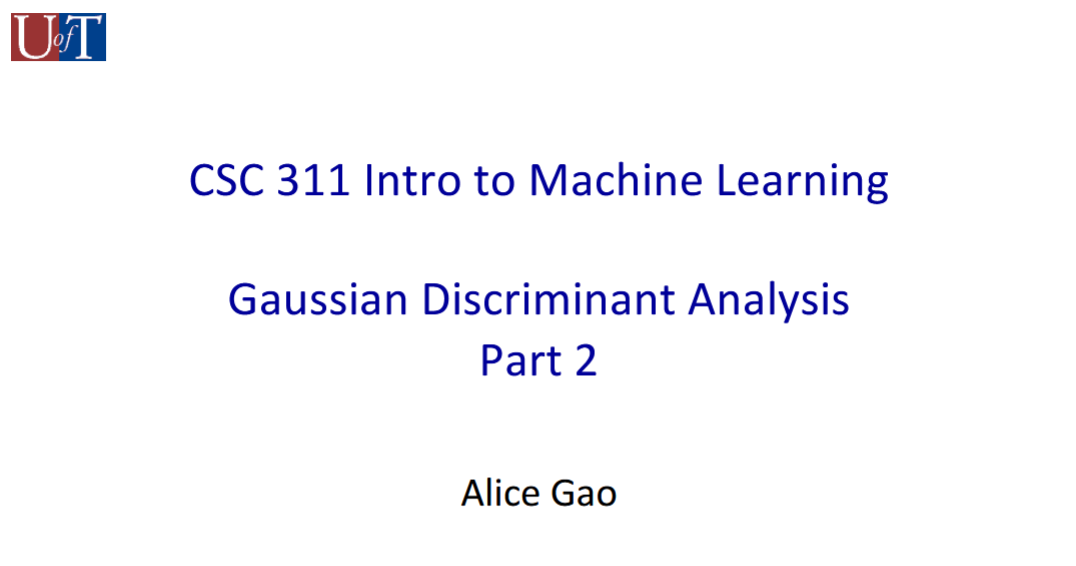
* Assuming binary classification to simplify things
* is an indicator variable showing if example i is of class k
* is class probabilities- without looking at features, what is the probability the example is in class 1
  + Basically counts for every i, then divides by N (total inputs)
* is calculated as the empirical mean for only elements with class k
* is calculated as the empirical covariance for only elements with class k

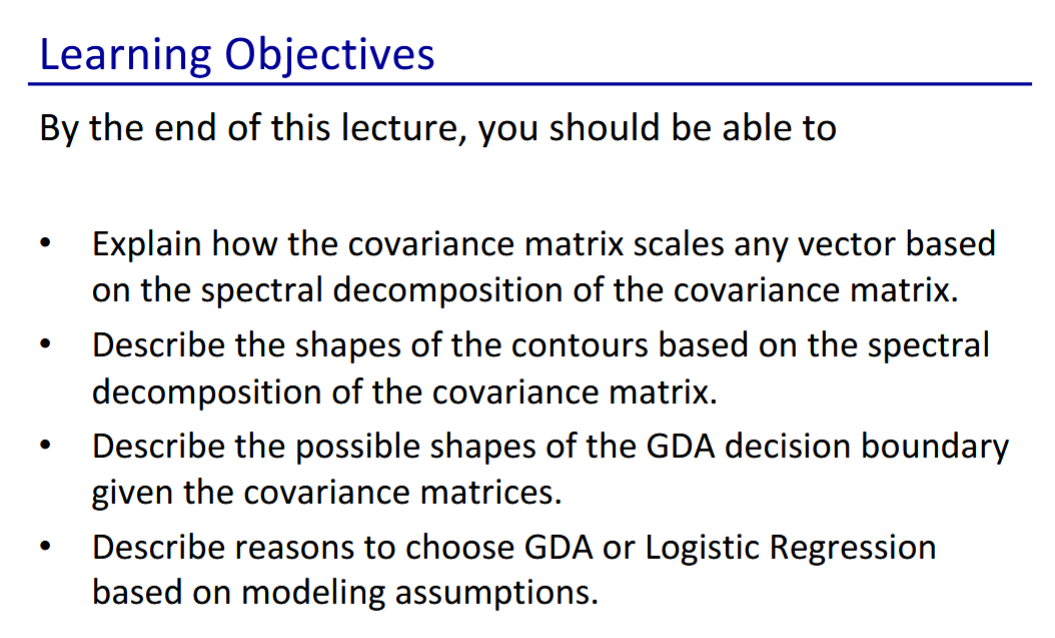


* We can then use the above formulas to calculate the mean and covariance for each class

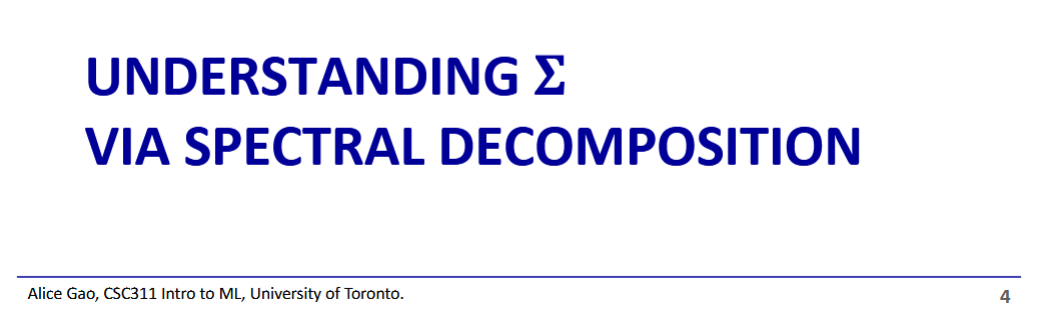


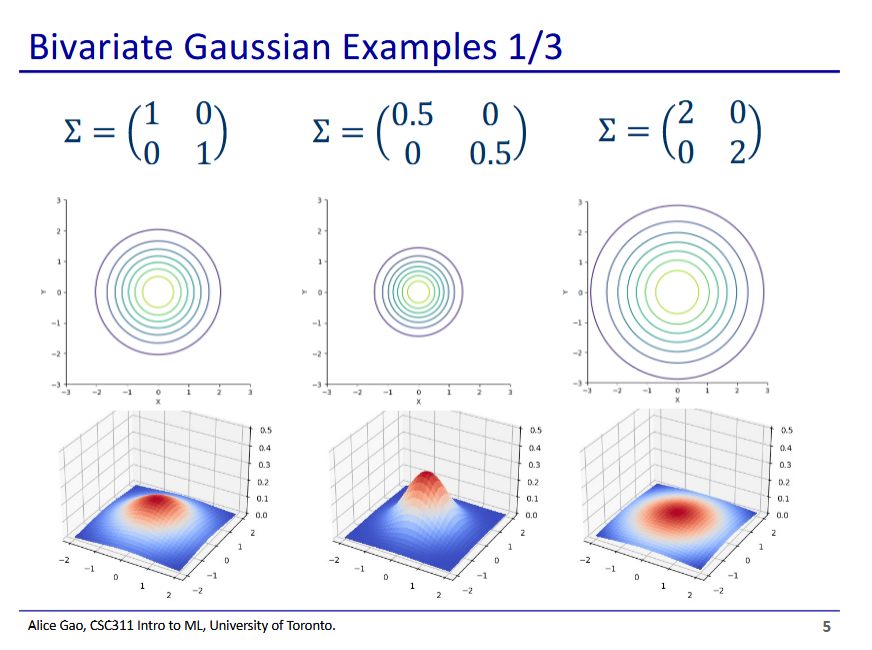
* We can then plot these out to get our 2 Gaussian distributions
* We assume that the data is generated using these distributions



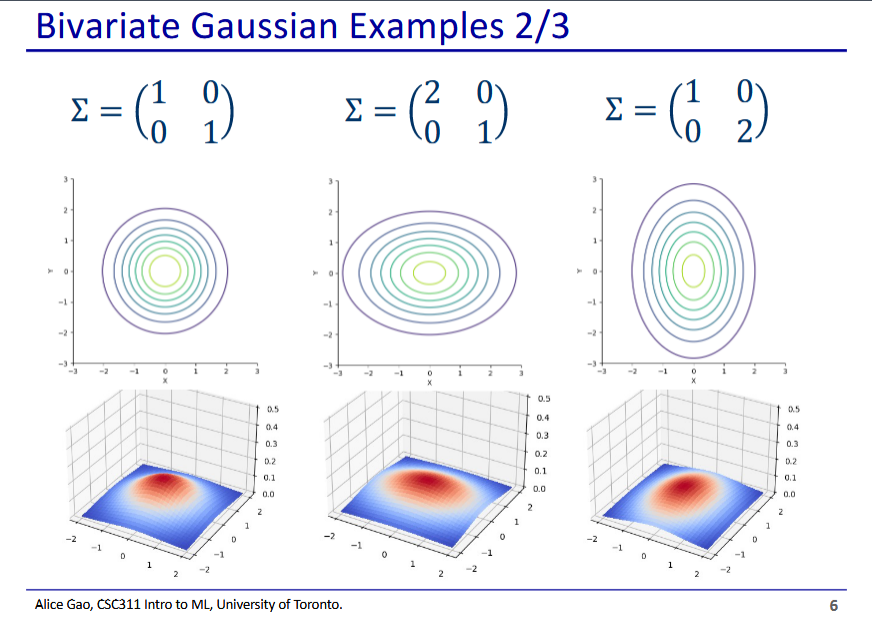




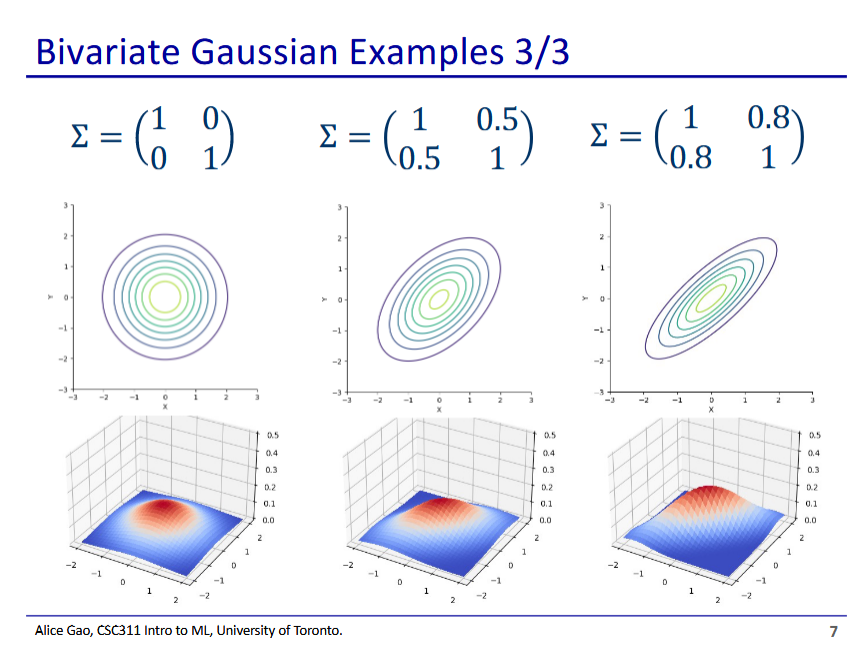




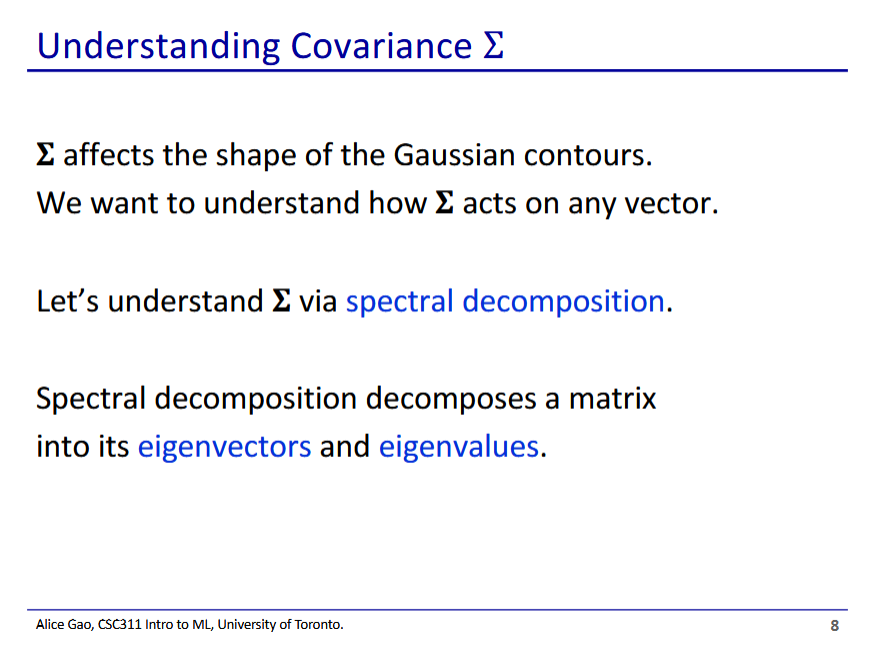
* When the covariance matrix numbers are smaller, it results in a thinner and taller distribution
  + The distribution gets more concentrated



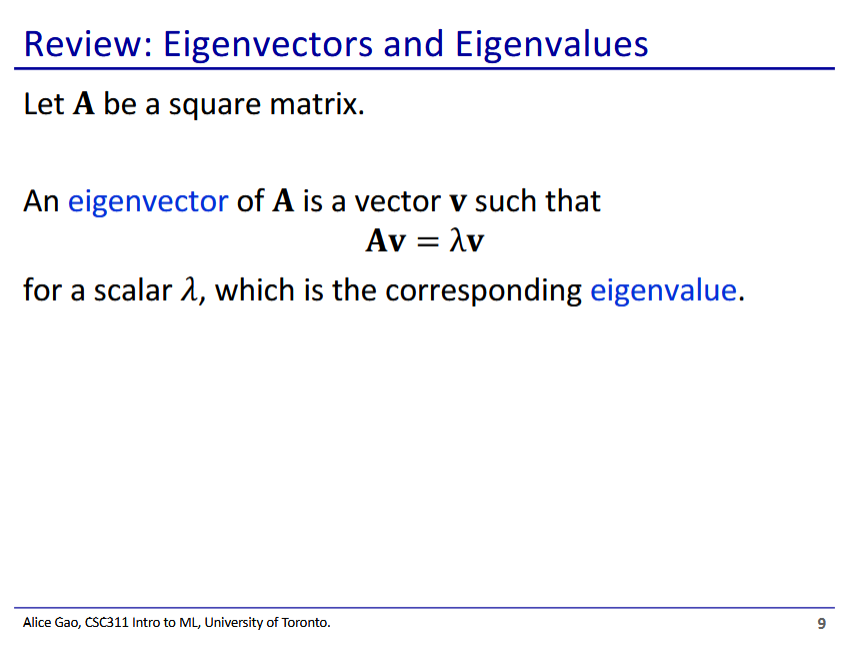
* If we make the first number larger, stretches along x axis
* If we make the second number larger stretches along y axis



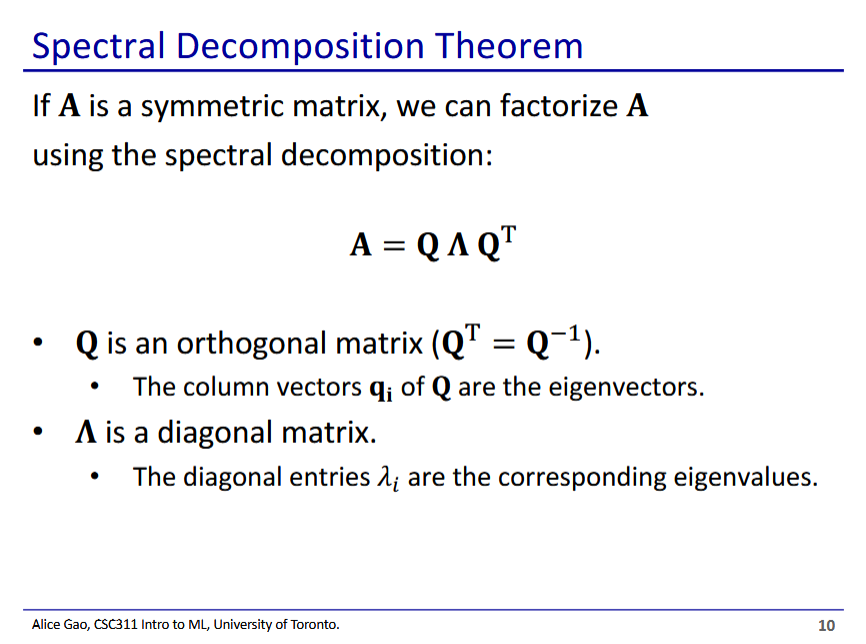
* If we make the covariance matrix not a diagonal matrix, we end up with an ellipse that are not axis aligned



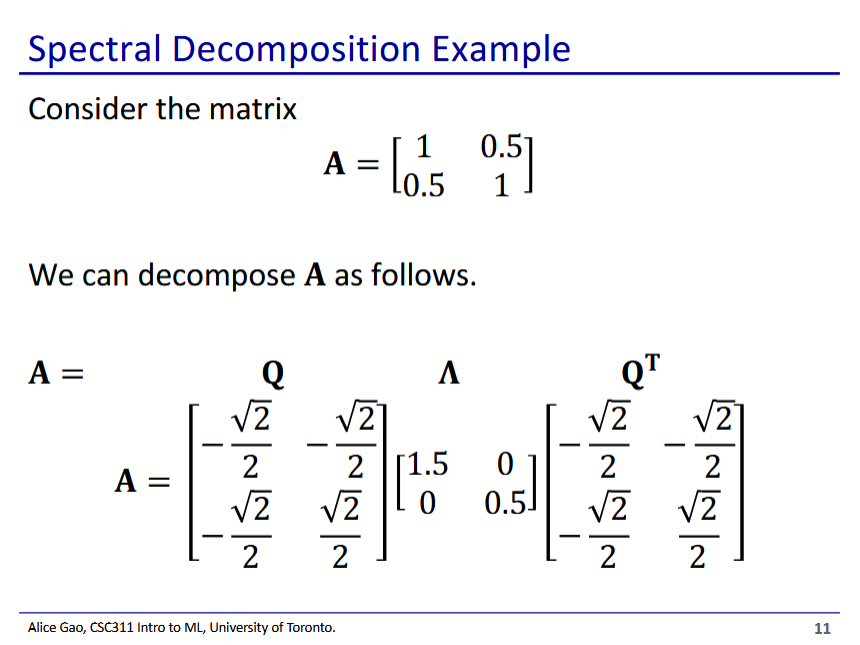
* Covariance affects the shape of the gaussian distribution
  + We can understand how through spectral decomposition



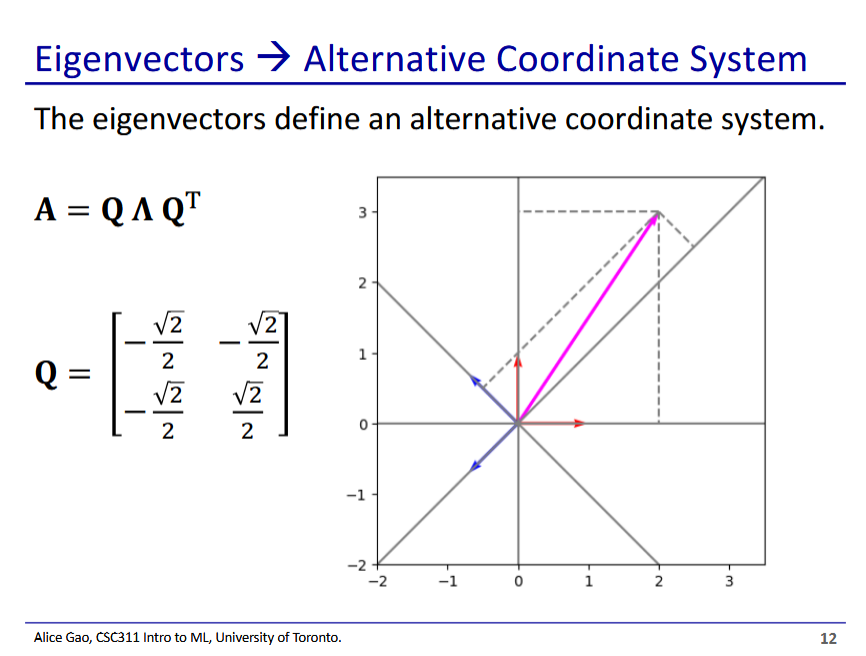
* If v is a non-trivial solution, then we call v an eigenvector of A
  + Each eigenvalue has a corresponding eigenvector



* Spectral decomposition involves factorising A into the product of 3 matrices: , , and
  + Q contains the eigenvectors and contains the eigenvalues



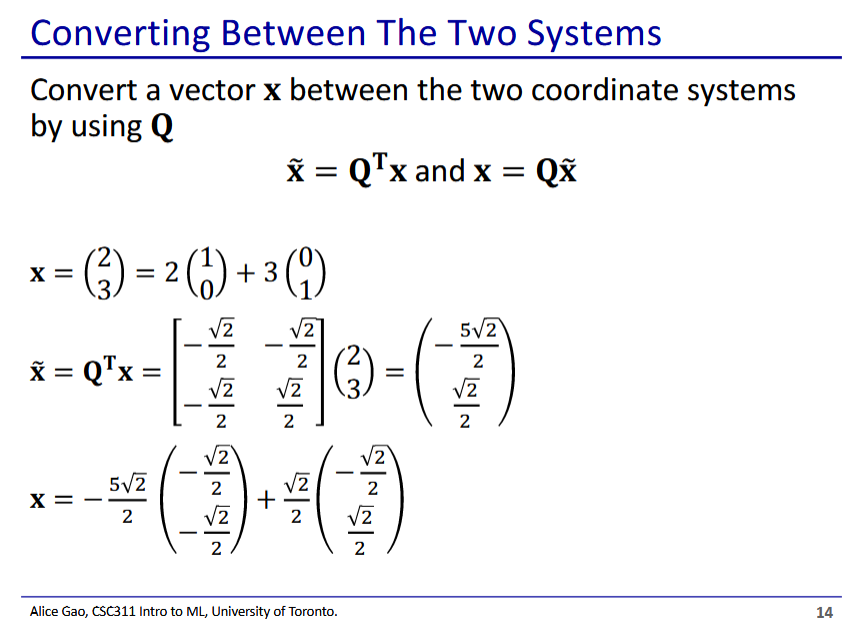
* We decompose the covariance matrix A using spectral decomposition
* In this case Q is a symmetrical matrix so
* The 2 column vectors of Q are the eigenvectors
* The 2 diagonal values in are the eigenvalues



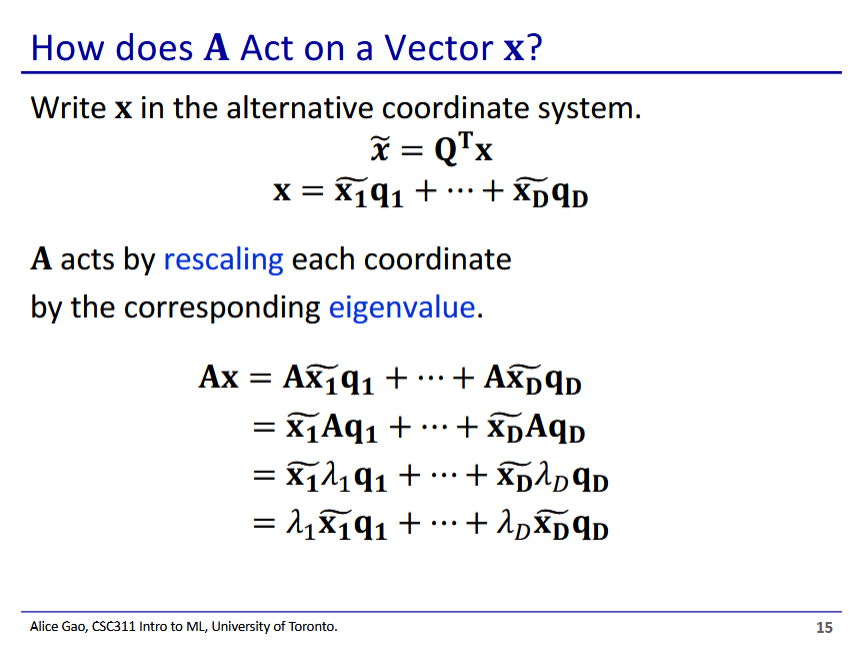
* The eigenvectors in Q form a different orthogonal basis for our space
  + We can use our eigenvectors to define an alternative coordinate system



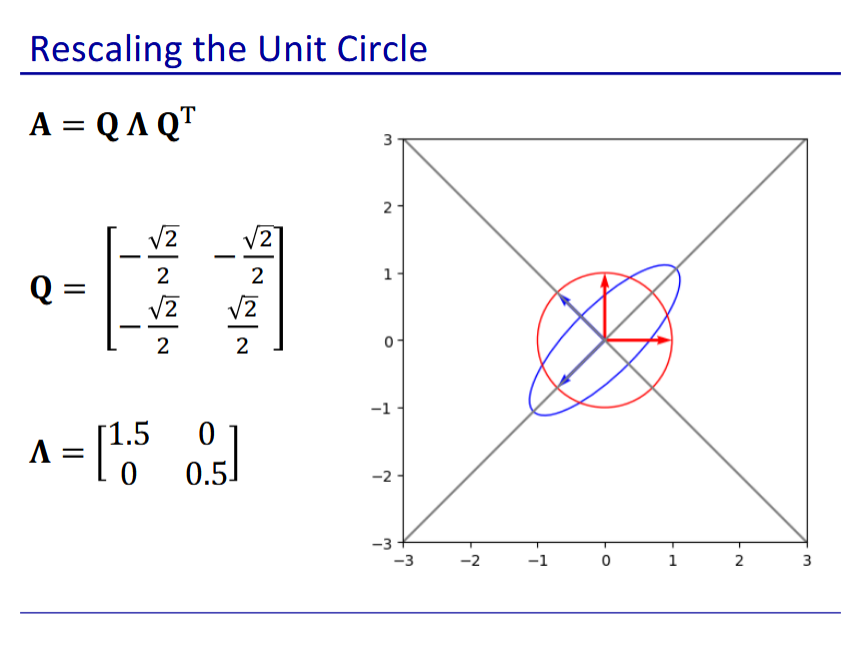
* We can represent any vector in the space as a combination of our eigenvectors as well



* We use Q to convert vectors between the two coordinate systems



* How does the covariance matrix (A) act on a vector x?
  + We first write x in the new eigenvector coordinate system
  + A rescales each coordinate in this eigenvector representation by the corresponding eigenvalue
* We use the definition of eigenvector and eigenvalue to go from lines 2 to 3

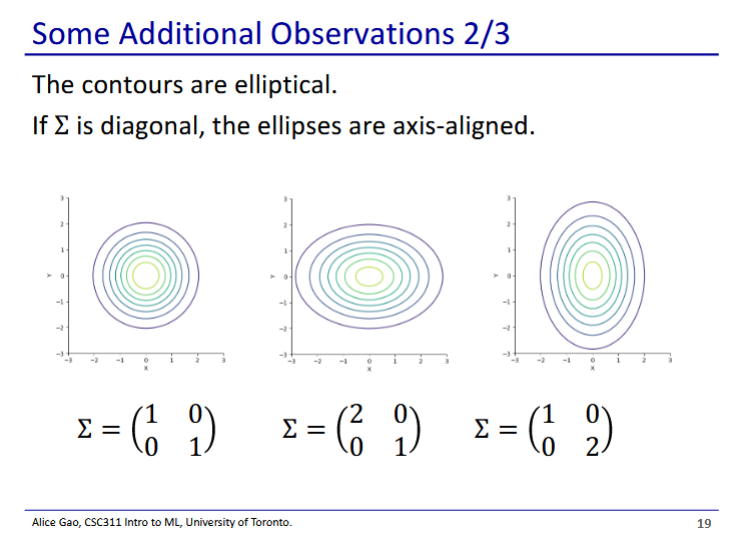


* This is the result of applying the covariance matrix to a gaussian distribution
* We stretch in the direction of each eigenvector by the eigenvalue
  + We stretch the contour in the direction of q1 by 1.5
  + We then stretch the contour in the direction of q2 by 0.5
    - Stretch by 0.5 is a compression
* On a test, may give decomposed eigenvectors and eigenvalues and ask how the covariance matrix transforms the distribution

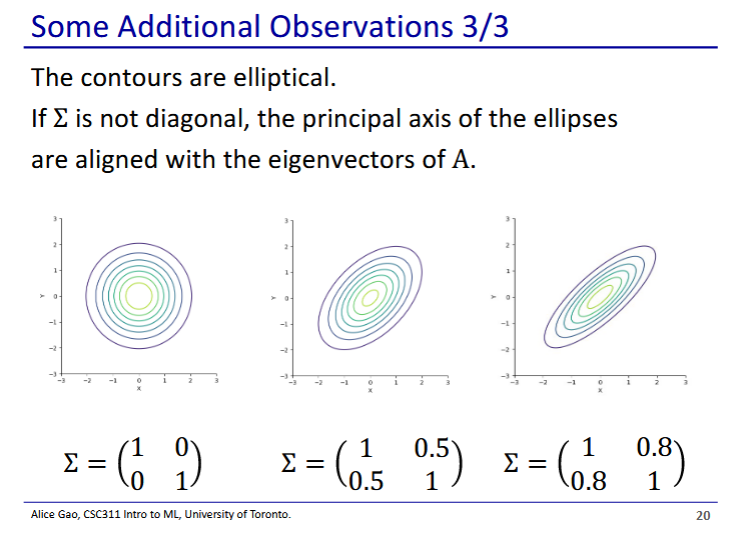


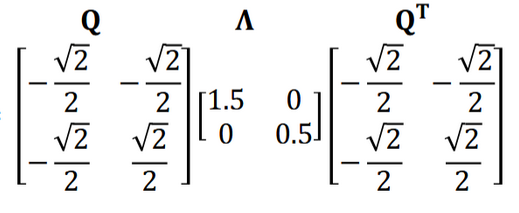


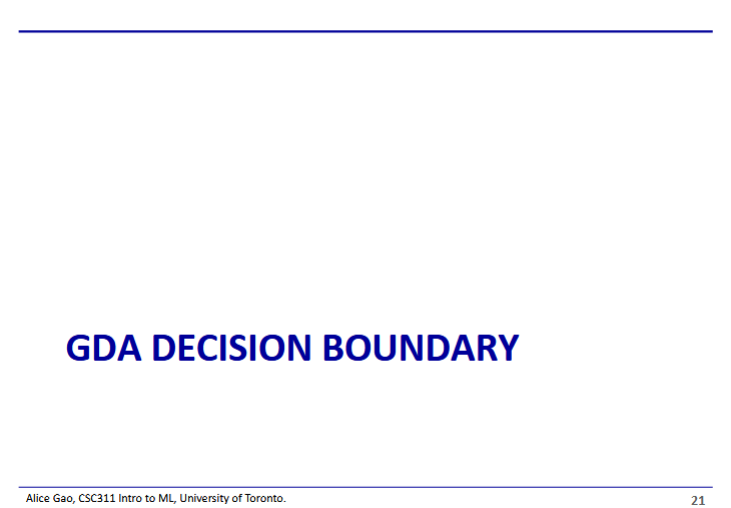
* Don’t need to prove this, but it is always the case
* If is a diagonal matrix, then the resulting ellipse formed by the distribution will always be axis-aligned
* Middle distribution is more concentrated since it is being shrunk by 2 along both axes
* Right distribution is more spread out since it is being stretched by 2 along both axes

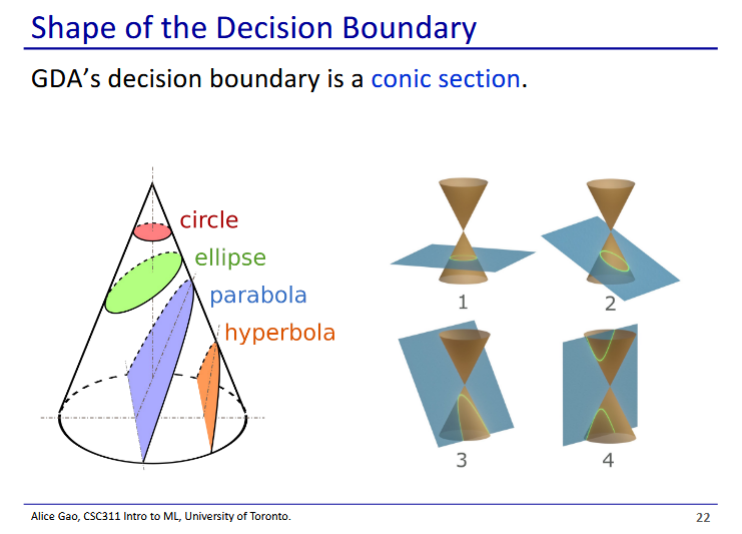


* is a diagonal matrix, ellipses are still axis-aligned
* Top left is x axis, bottom right is y
* Middle ellipse is stretched by 2 along the x axis
* Right ellipse is stretched by 2 along the y axis

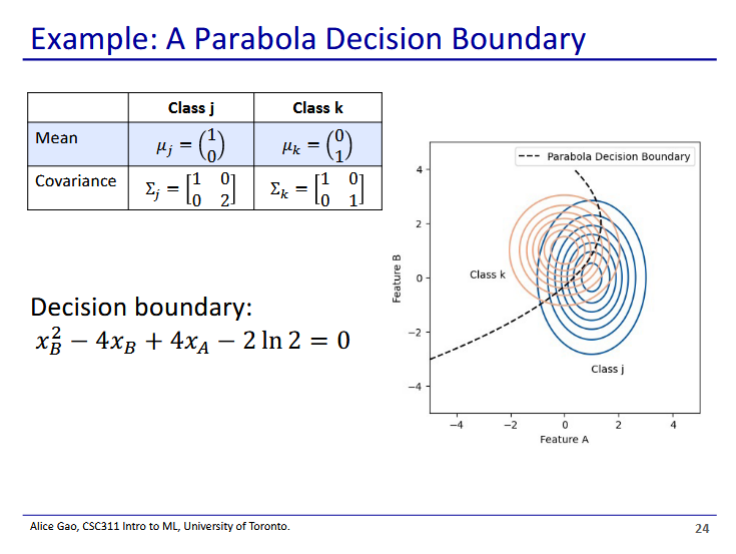


* This time is not a diagonal matrix - is not axis aligned
* In the middle ellipse we stretch along one eigenvector and shrink along the other eigenvector
* We scale by 1.5 in the axis of and by 0.5 in the axis of

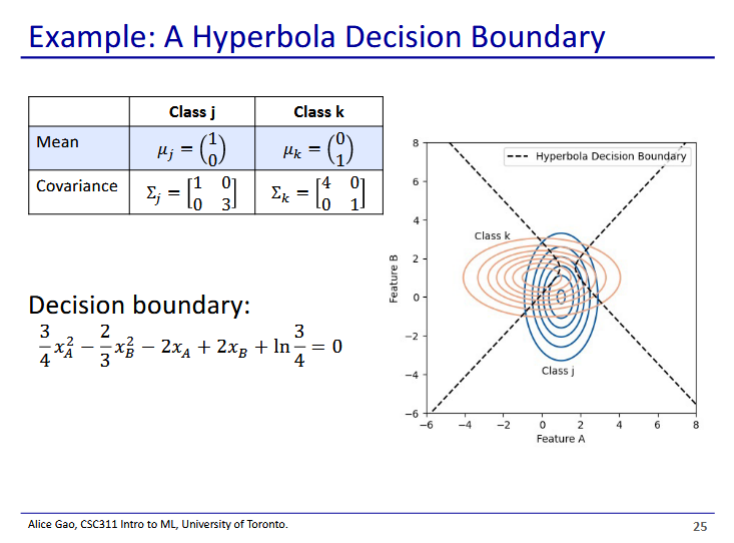




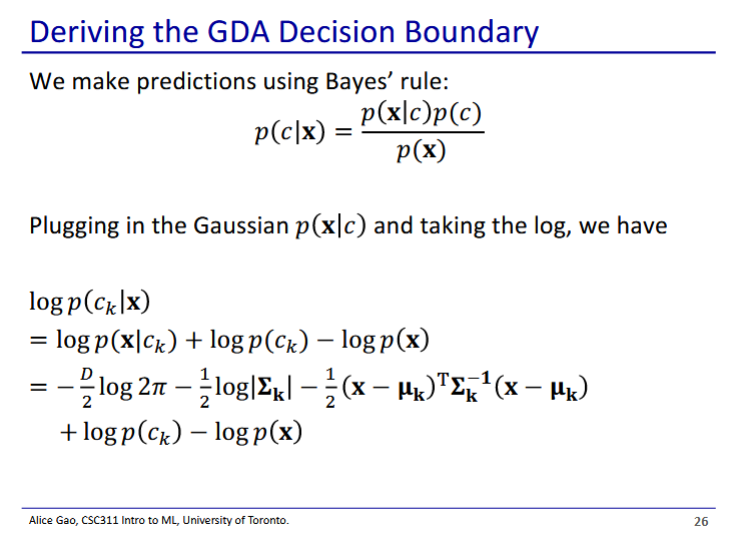
* The decision boundary for GDA is a conic section
  + We intersect a cone with a plane
  + Depending on the angle of the plane we can get a circle, ellipse, parabola, or hyperbola
* <https://mathformachines.com/posts/discriminant-analysis/#three-questionssix-kinds>
  + Website with animations of the different kinds of decision boundaries



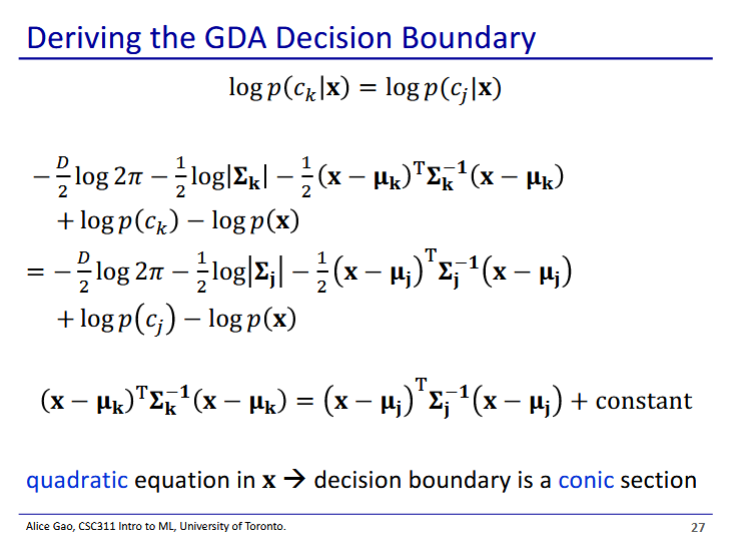
* Example of parabolic decision boundary



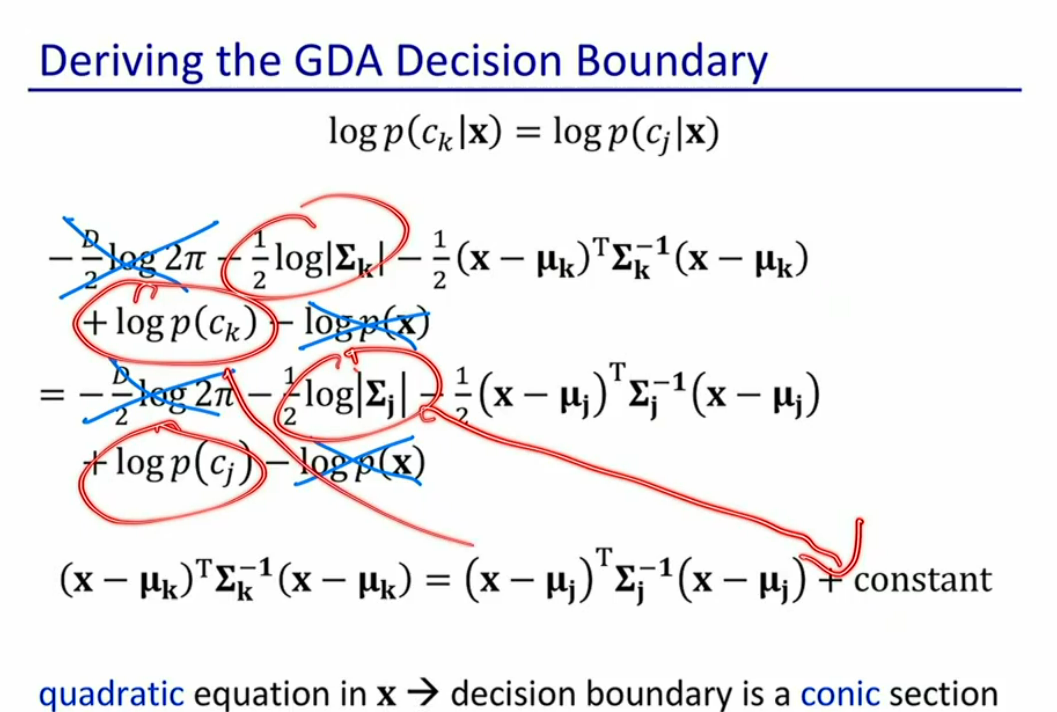
* Example of hyperbolic decision boundary



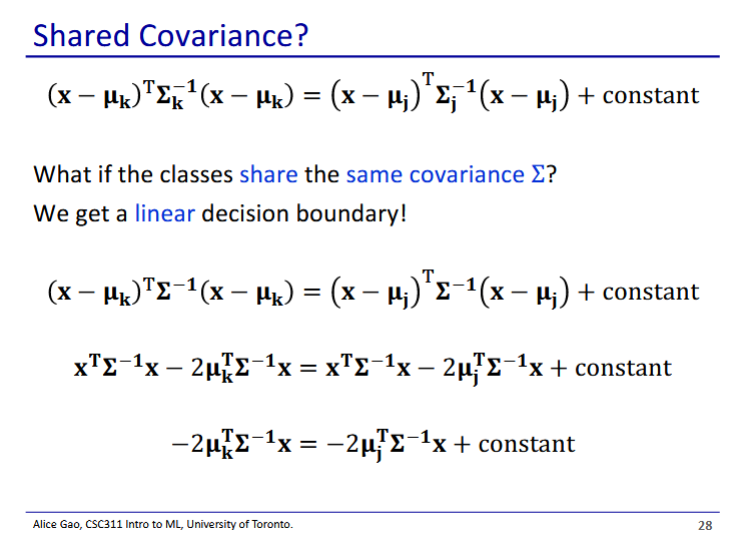
* To derive the decision boundary we should first look at how we make the prediction (bayes rule)
* We take bayes rule and then take the log of bayes rule (line 2)
* We then substitute with the multivariate GDA probability density function (slide 22)
  + This is how we get line 3



* The decision boundary is where the probabilities of both classes are equal
  + Thus we equate the 2 equations for k and j and solve
  + Equation is line 3 of previous slide, one for k and one for j



* The x’ed out terms cancel out
* The circled terms have nothing to do with x and thus become part of the constant (don’t care about)
* Thus we get a conic section
  + The left side is a quadratic equation in x, has 2 terms with x dotted together
  + If we write it out non-vectorised, we end up with an x^2 somewhere
* This is the most general case



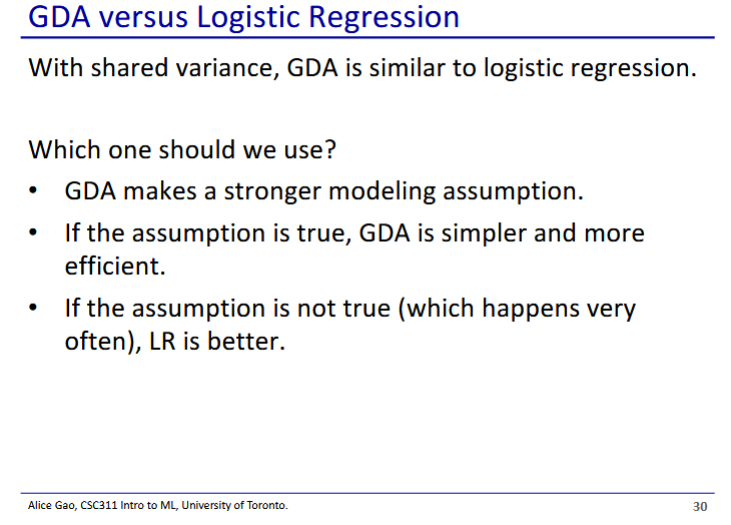
* Class have the same covariance = linear decision boundary

* (terms without x put into constant)

* + Since is a scalar,
  + is a symmetrical matrix thus

* Probably won’t need to derive this during a test, but do know that if the covariance is the same we get a linear decision boundary





* GDA makes a fairly strong assumption about the data
  + Assumes data has to be generated from a Gaussian distribution with a mean and covariance
* When the assumption is satisfied, GDA works very well
* However in most cases this assumption is not satisfied, which means we may want to use a different model