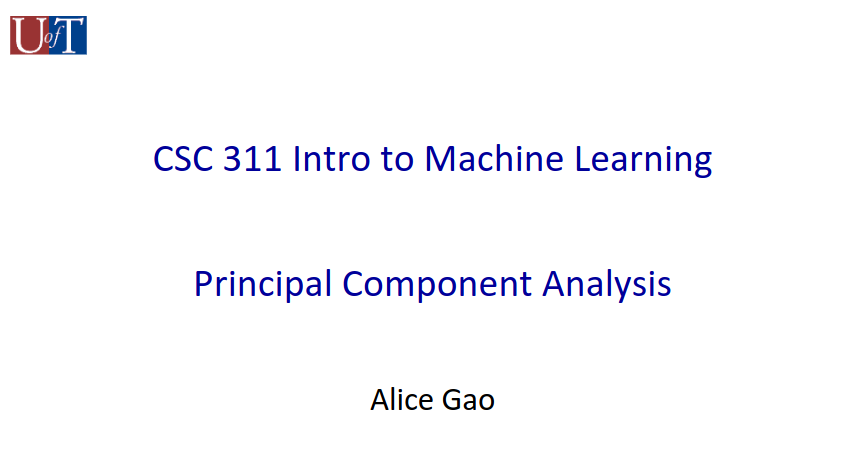
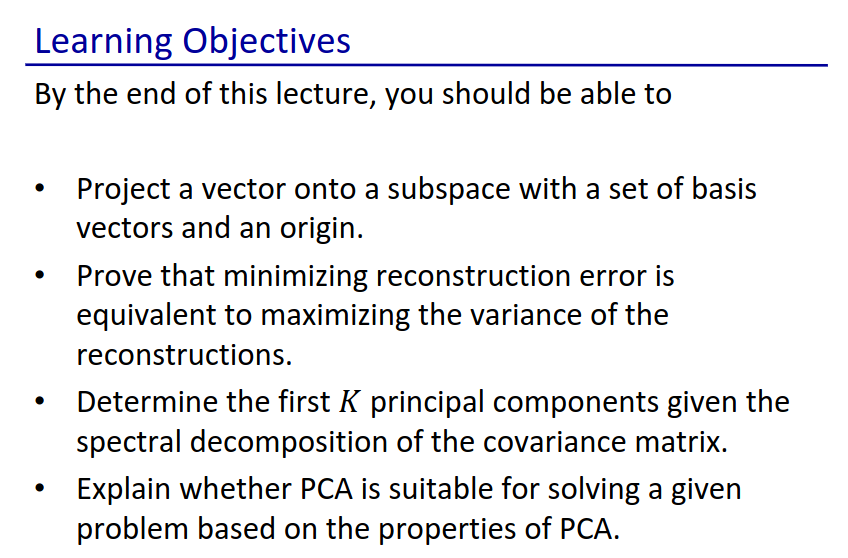
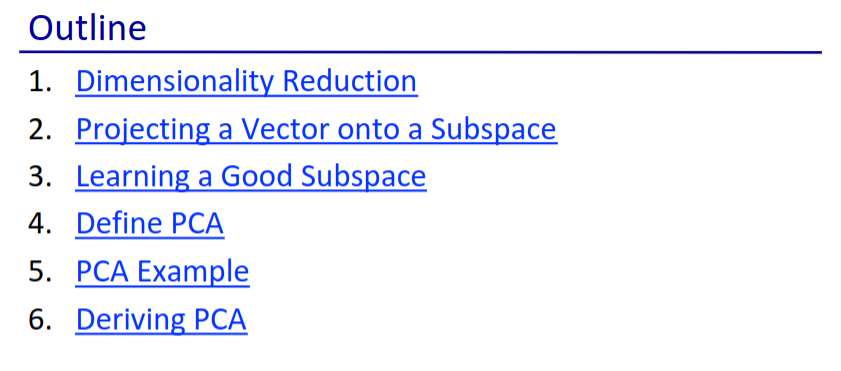
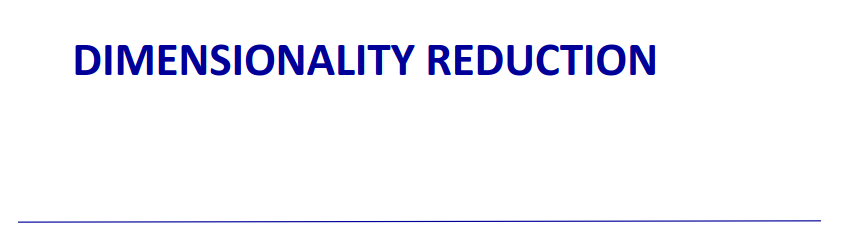
|  |
| --- |
| **Admin stuff**   * Project - New version posted * Test 2 should be graded by tomorrow * Assignment 2 remark forms released now   **Dimensionality reduction**   * Process of reducing the number of dimensions (features) of data by mapping it onto a new subspace * **Some reasons for dimensionality reduction**   + Visualising the data - 2D and 3D data is easier to visualise   + Extract meaningful features - figure out which features are most important   + Compress the data - reduce the space the data takes while preserving as much important info as possible * **Dimensionality reduction techniques**   + Principal component analysis (PCA) - this lecture   + Matrix factorisation   + Autoencoders   **Projecting a vector onto a subspace**   * **Projecting a vector onto a line with an origin of 0**   + - is the magnitude of the projected vector     - is the basis vector for the line * **Projecting a vector onto a general subspace S of K dimensions**  1. Choose an orthonormal basis {u1, … , uk} for S  * Orthonormal basis means that all basis vectors are of length 1 and perpendicular to each other * On the test, the basis will be given * Set of bases put together is   + has dimensions (D x K)     - D = dimensions of the original data  1. Project onto each vector individually to get the **representation/code z**    * is a scalar  * Vectorised:   + has dimensions (K x 1) * is the origin of the subspace   + has dimensions (D x 1)  1. Sum together the projections to get the **reconstruction of** in x  * Vectorised: * has dimensions (D x 1)   **Picking the best subspace**   * We want to pick the best subspace (and thus basis U) to project onto that minimises the amount of information lost when reducing dimensions * **We have 2 criteria to choose the best subspace**   + **Minimise reconstruction error**      - Try to minimise the mean square error of reconstructed vectors   + **Maximise variance of reconstruction**     - Since we want to preserve as much data, we don't want reconstructed vectors to be the same   + These 2 criteria are equivalent, achieving one will get the other     - Proof on slide 21     - **Proof will be on test**   **Principal component analysis (PCA)**   * Unsupervised learning method to choose the best subspace to map the data to based on the above criteria   + Maximising the projected variance   + Minimising the reconstructed error * **Finding the best subspace**   + We use the empirical covariance matrix     - Using spectral decomposition:       * Columns of Q are eigenvectors       * Diagonals of are eigenvalues   + The optimal k bases are the top k eigenvectors of Q, which correspond to the largest eigenvectors of     - Example on slide 33   + **Will not be expected to do spectral decomposition or find empirical covariance matrix on test, but will need to be able to identify which are the best eigenvectors** * **PCA is sensitive to scale**   + We need to normalise the data to avoid bias towards a particular feature   + Features with a wider scale have more variance, which will bias PCA |

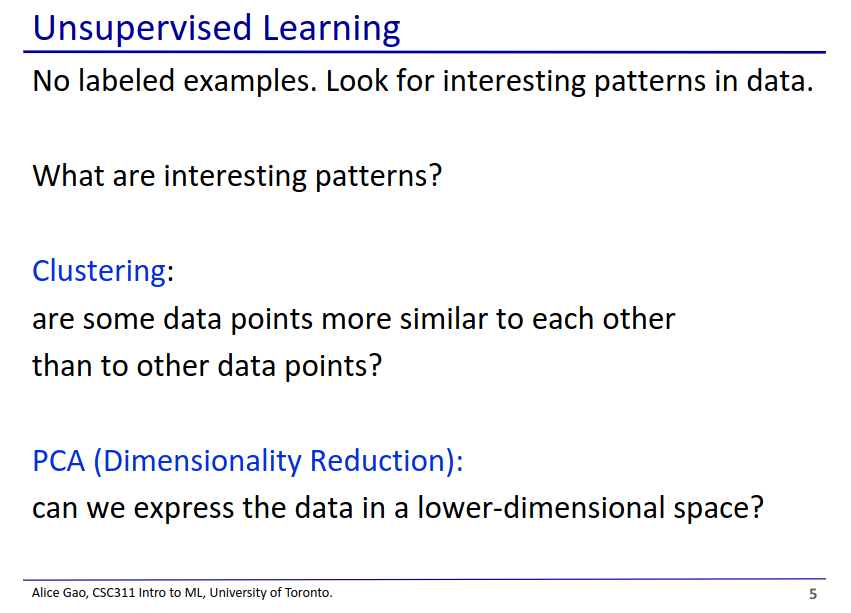




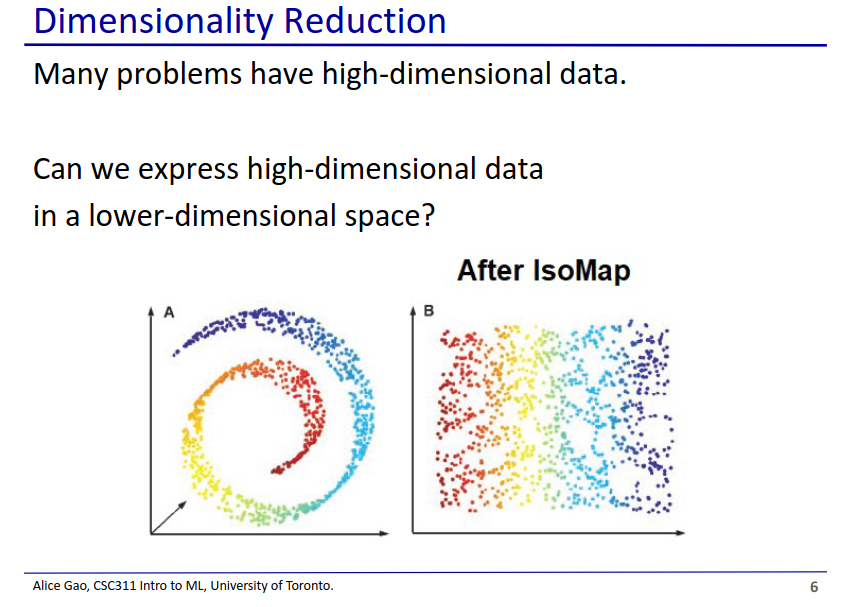




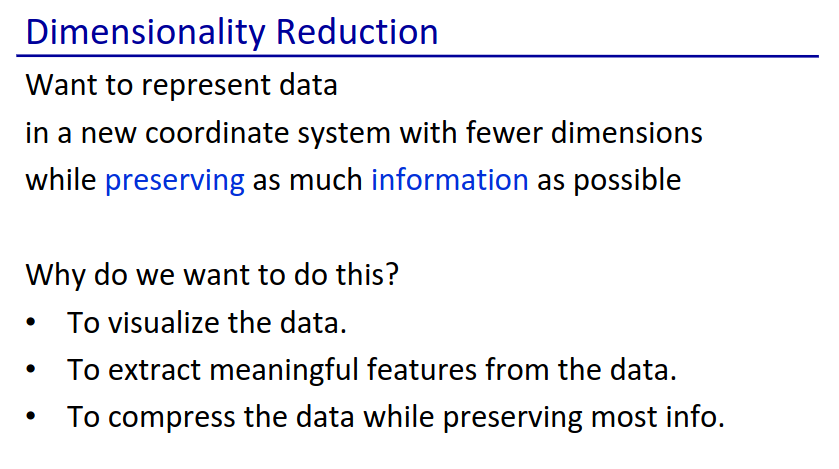
* We are projecting our feature vector onto a lower-dimensional space



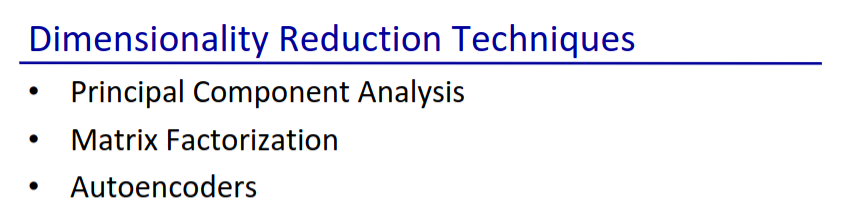
* PCA
  + Suppose we get some data and it has a lot of features
  + But data with a lot of dimensions is hard to work with (recall curse of dimensionality for KNN)
  + So how can we decrease the number of dimensions?

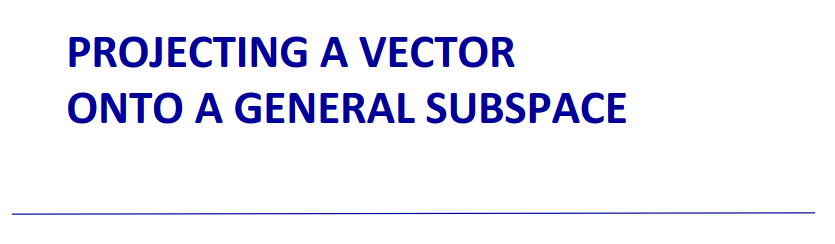


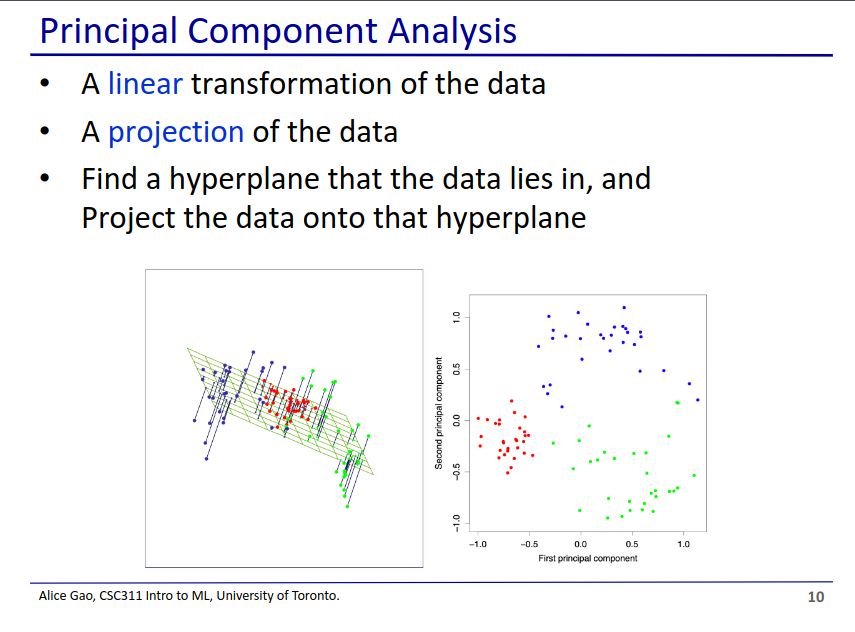
* On the left we have 3D data going around a spiral
  + Note how the Z dimension changes as we move along the spiral
* We can then project it onto a 2D space on the right
* However PCA cannot do this
  + PCA is only does linear dimensionality reduction
  + To do the above we would need a non-linear dimensionality reduction



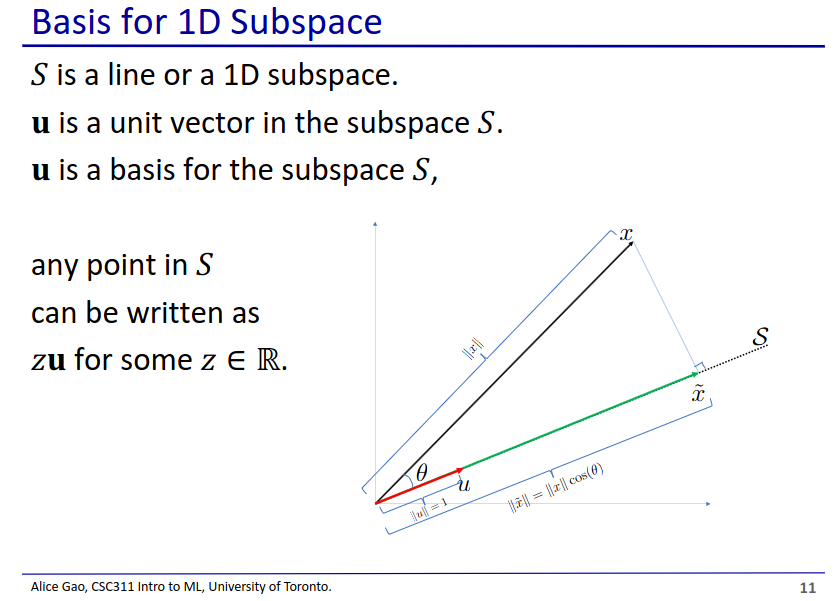
* We want to make our data in a lower dimension while keeping as much information as possible
* Why do we want to do this?
  + To visualise the data
    - We are 3D beings, visualising things in 4+ dimensions is hard
  + To extract meaningful features from the data
    - The reduced features are more meaningful
  + To compress the data
    - Saves space



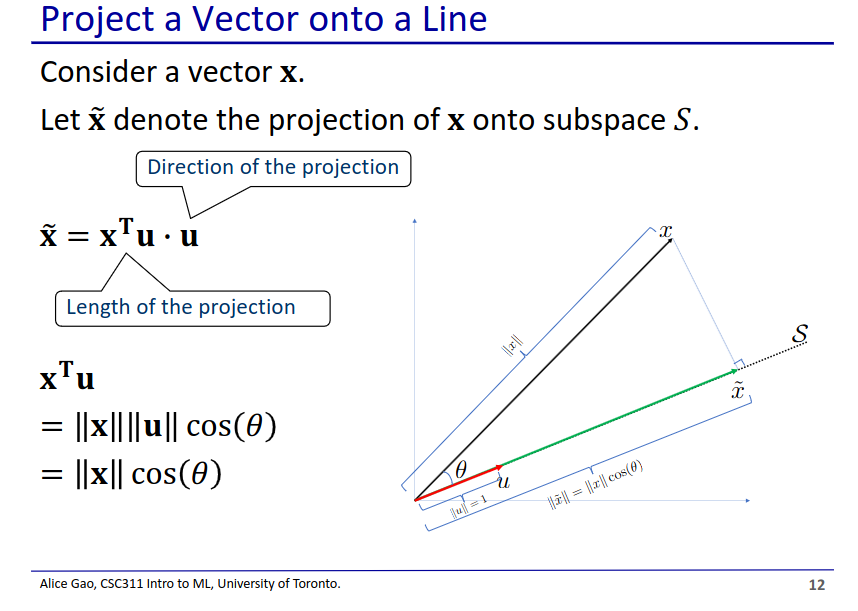




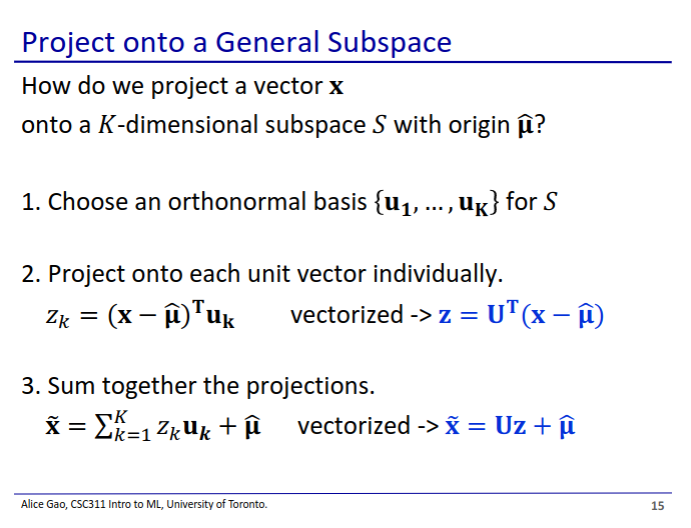
* Linear transformation - we take the data and project it onto a hyperplane
* In the below example we projected 3D data onto a 2D hyperplane



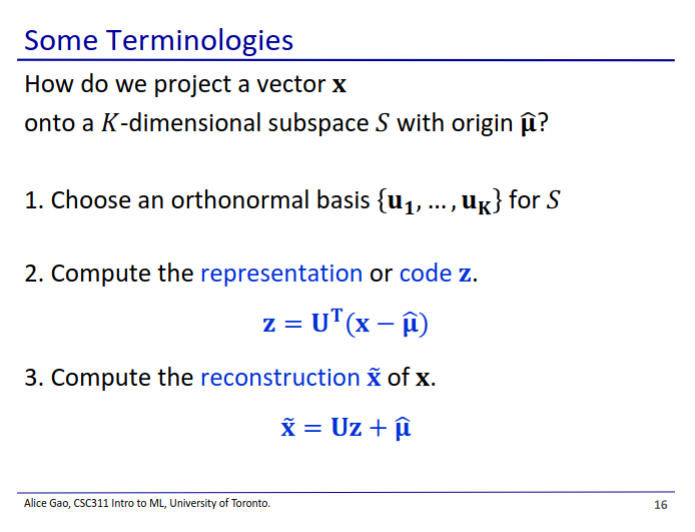
* S represents our subspace (1 dimensional)
  + **u** is the unit vector and the basis for S
* We can write any point in S as z**u**
  + z is a scalar that gives us the length
  + **u** is a vector that gives us the direction

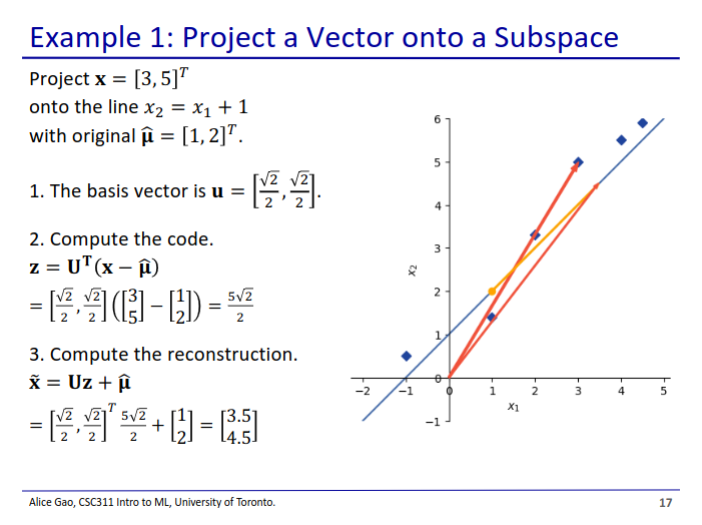


* This slide assumes that the origin of the subspace is the origin, but this doesn’t need to be the case

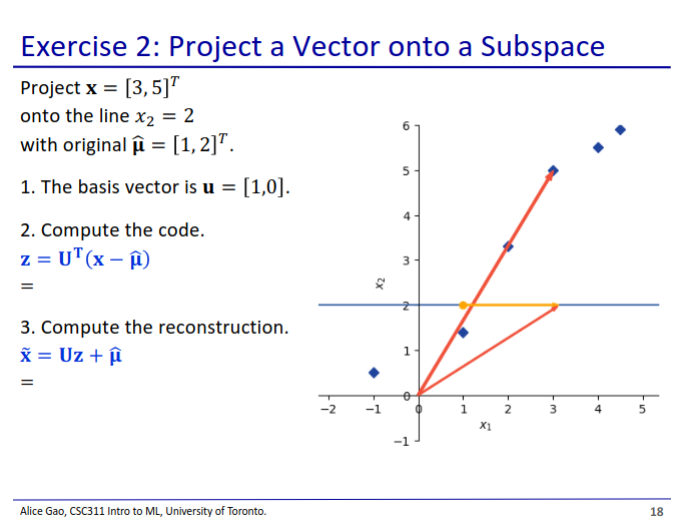


* Subspace S has K-dimensions
  + Thus it has K basis vectors
  + Not technically a subspace since it doesn’t include the origin, but that’s ok
* Orthonormal basis - each basis has a length of 1 and are orthogonal to each other

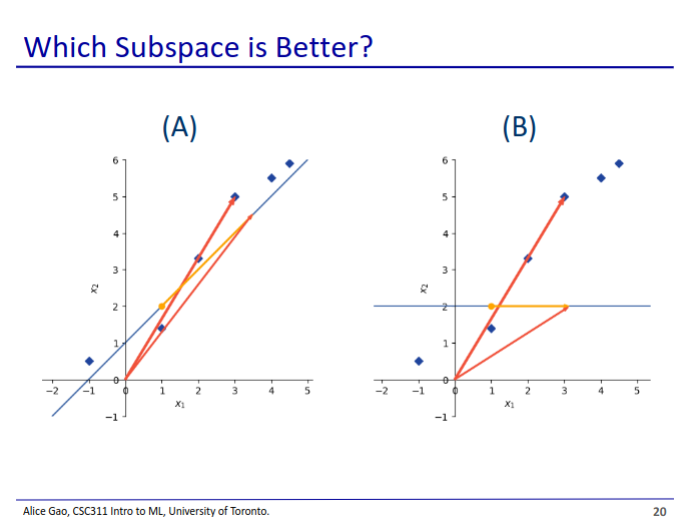




* This basis vector is very simple (is a diagonal line)
* The code (z) on this graph is the length of the projection (yellow line)
* Then using the code and the original we can reconstruct the projection in the original dimension

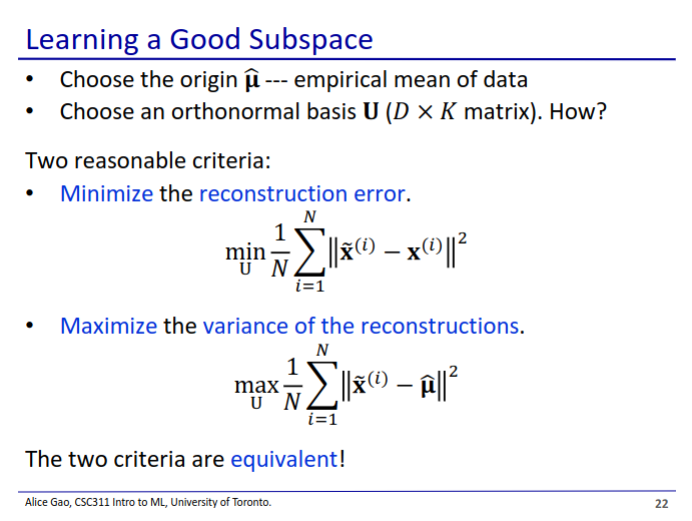


* Compute the code
* Compute the reconstruction

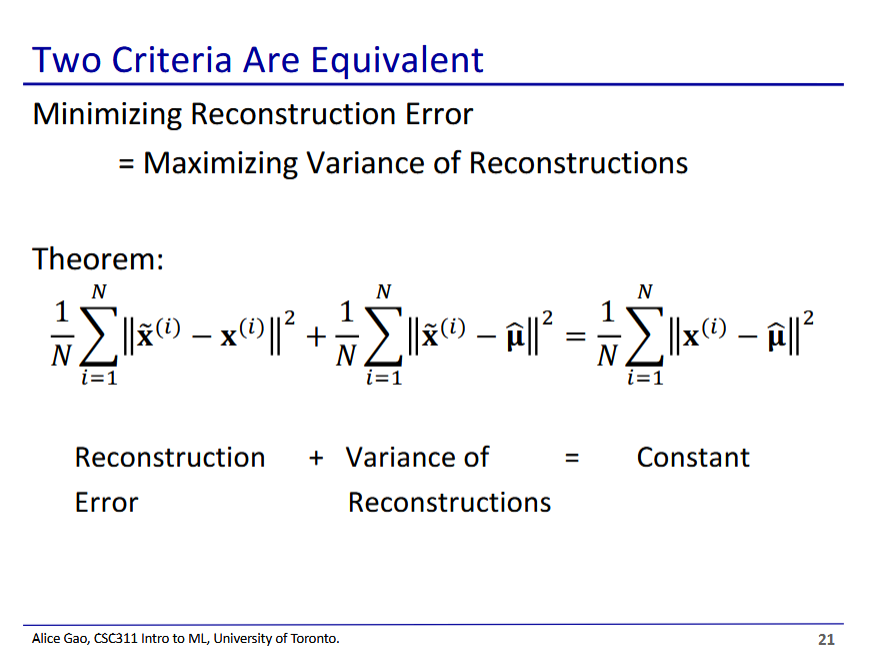


* A is the better subspace
  + The reconstruction is still very similar to the original data point
* In B, the reconstructed points are very far from the original data point
  + We probably lost a lot of information

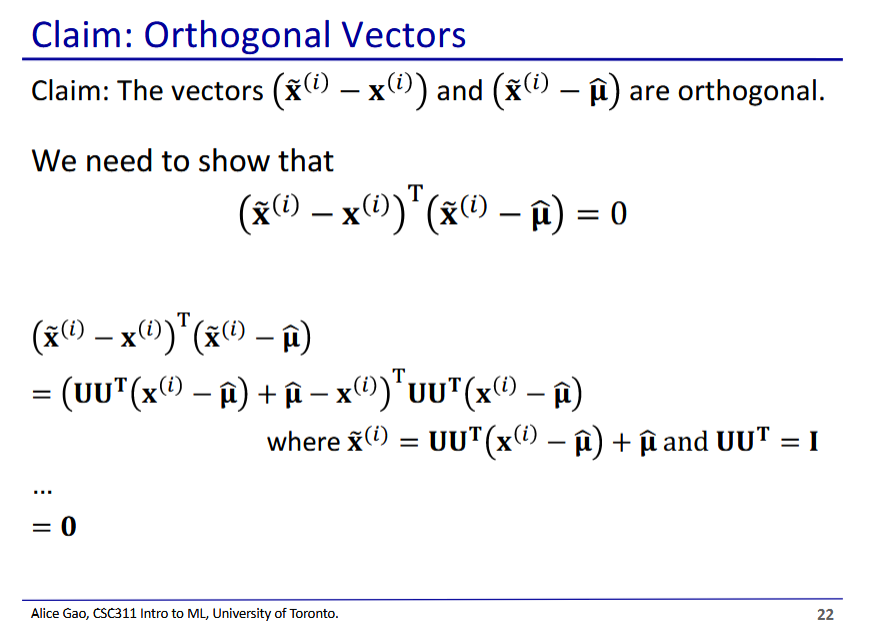




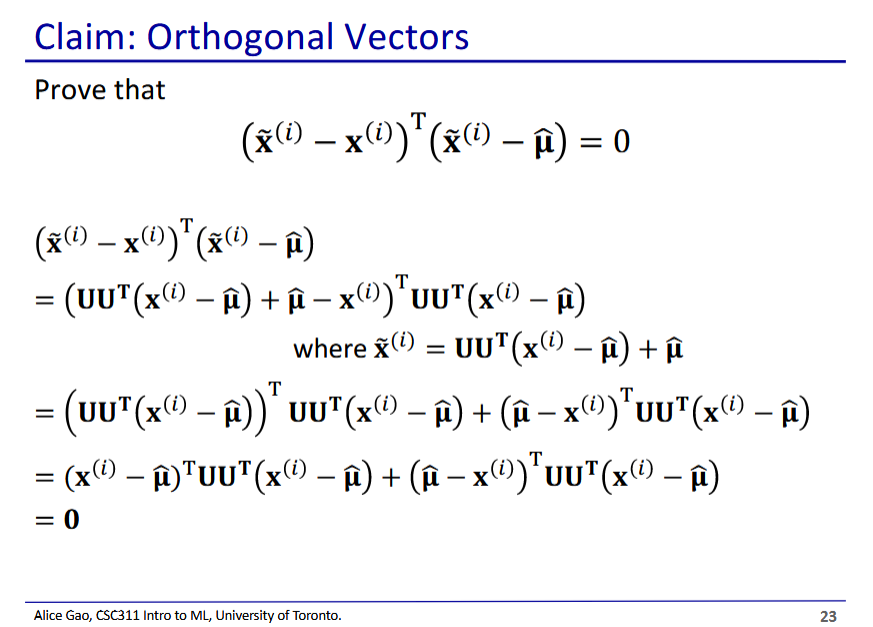
* It makes sense to pick the mean of the data to be the origin of our subspace
* We have 2 criteria to choose the best subspace to project onto
  + Reconstruction error
    - Calculate square err of reconstructed vectors and try to minimise average mean err
  + Variance of reconstruction
    - Maximise the variance of reconstructed vectors (since we want to preserve as much data, we don't want reconstructed vectors to be the same)
* These 2 criteria are actually equivalent



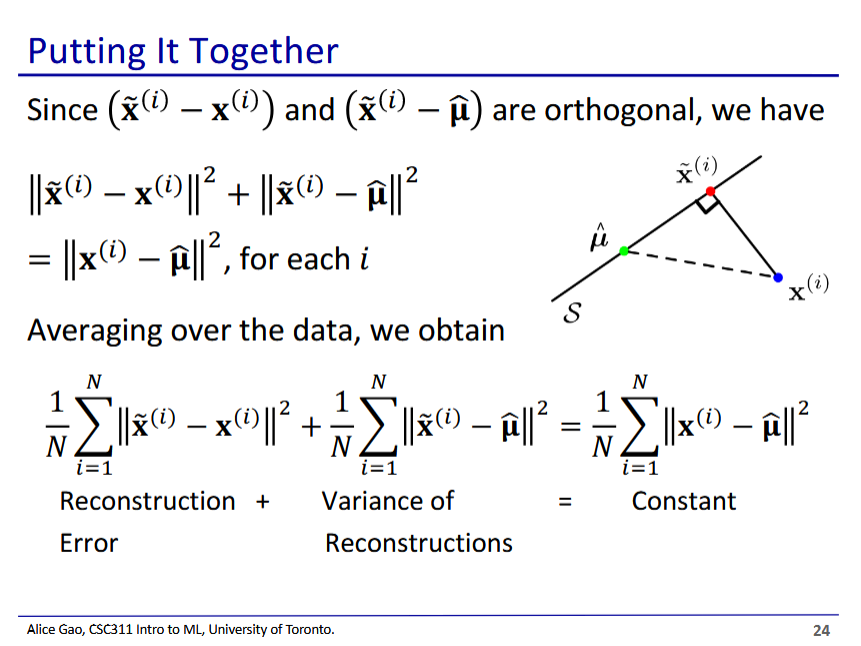
* We want to prove that both criteria add up to a constant
  + In this case it means maximising one is the same as minimising the other
  + Term on the right is a constant because it is the variance of the training vectors (not affected by projections)



* We first need to show that these vectors are orthogonal, which will be useful later in the proof
* Uses a formula relating the reconstructed vector to the original vector

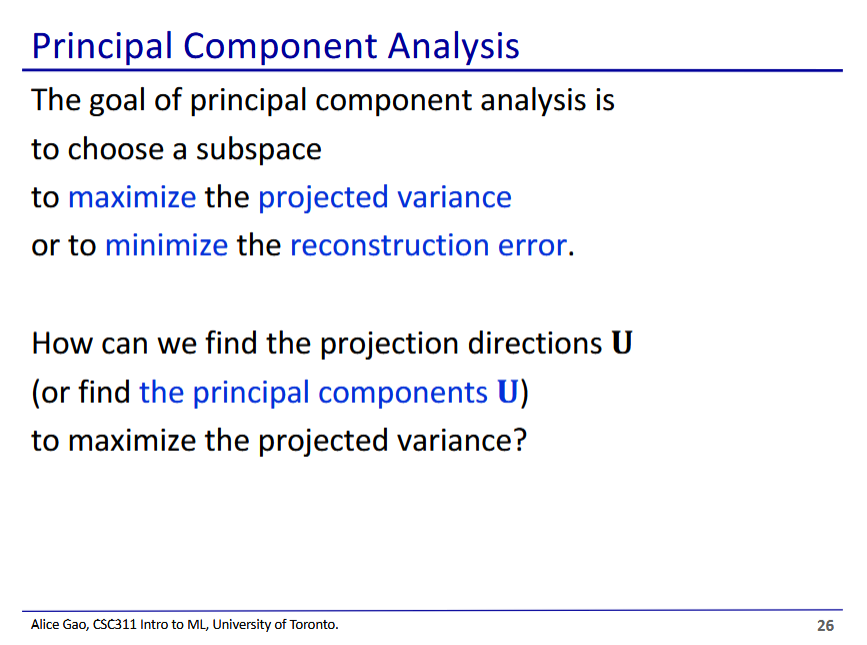


* Hidden slide including extra slides omitted from previous slide

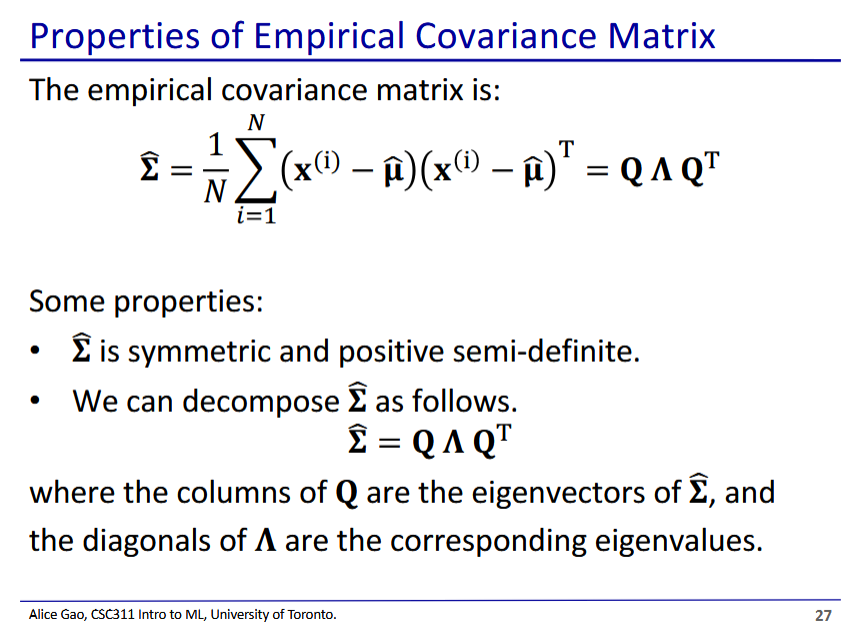


* We use the orthogonal property to figure out what each element of both sums give using the pythagorean theorem
  + Then we add it together to get our constant term
* Term on the right is a constant because it is the variance of the training vectors (not affected by projections)
* We need to know this proof for the final exam

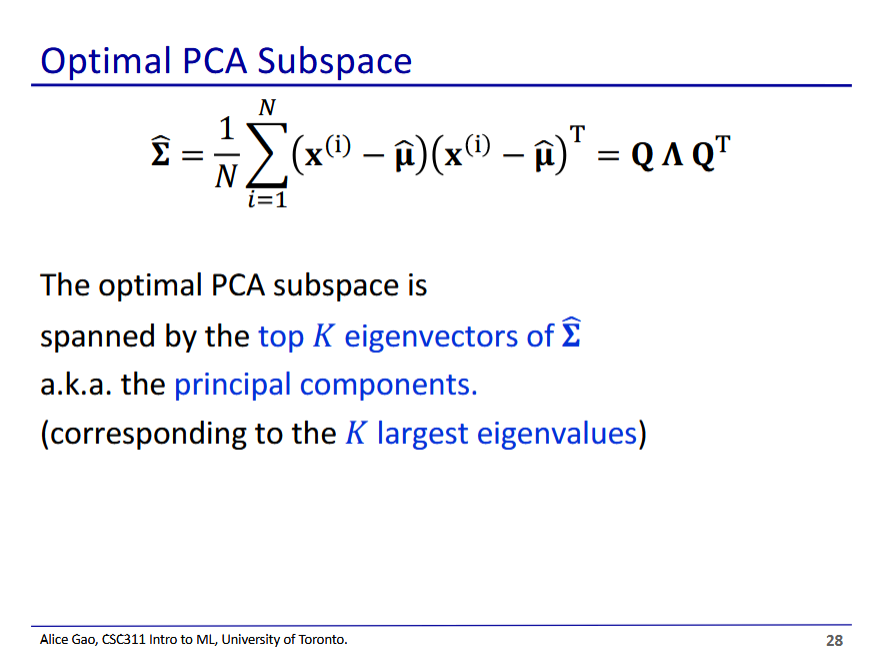




* We choose a subspace that best satisfies either criteria (both criteria are equivalent)

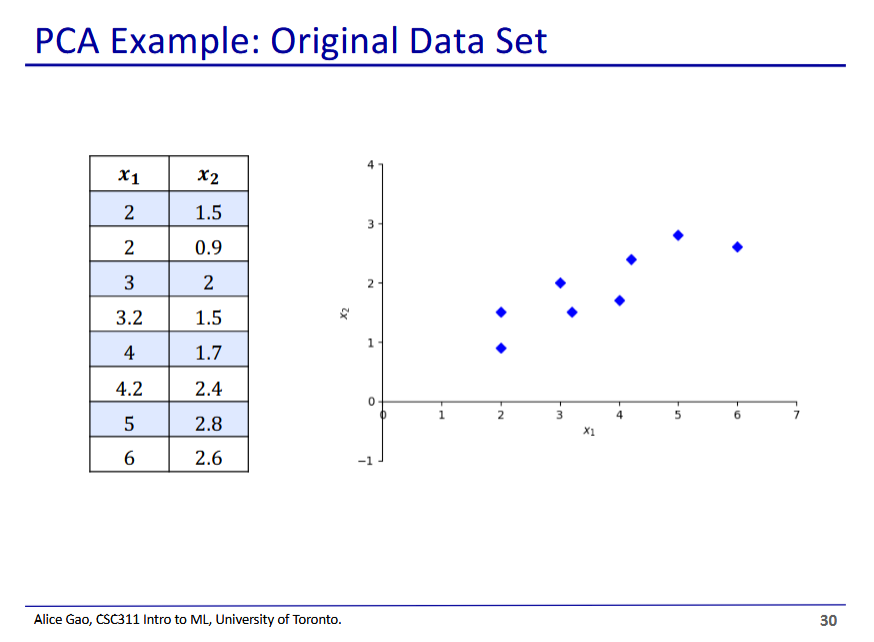


* We use the covariance matrix to help us find the optimal subspace
  + Recall covariance matrix from Gaussian distribution analysis (GDA) 2 weeks ago
* We use special decomposition to get that the empirical covariance matrix

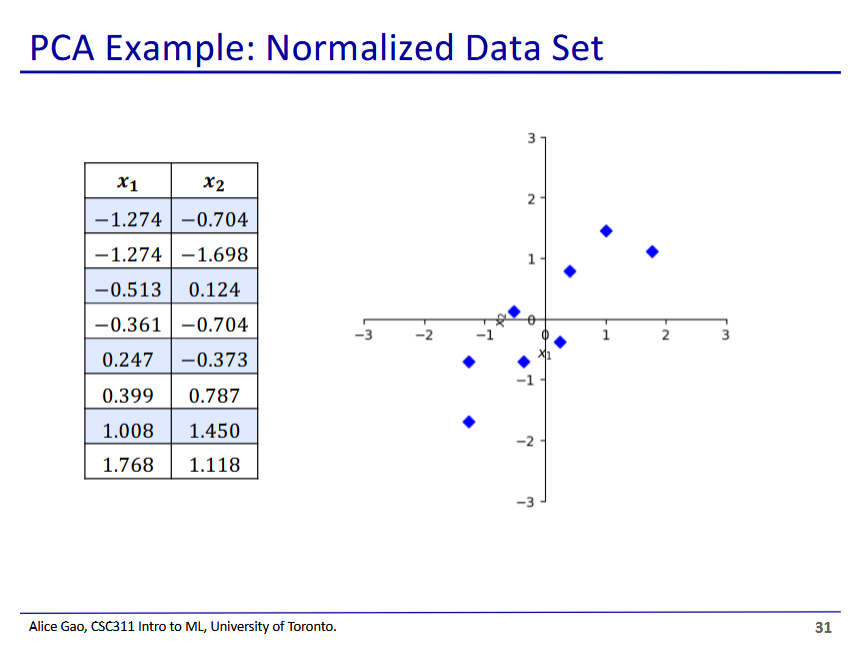


* It is the top K eigenvectors
  + Are the eigenvectors of the K largest eigenvalues
* K is the dimension of the subspace we want

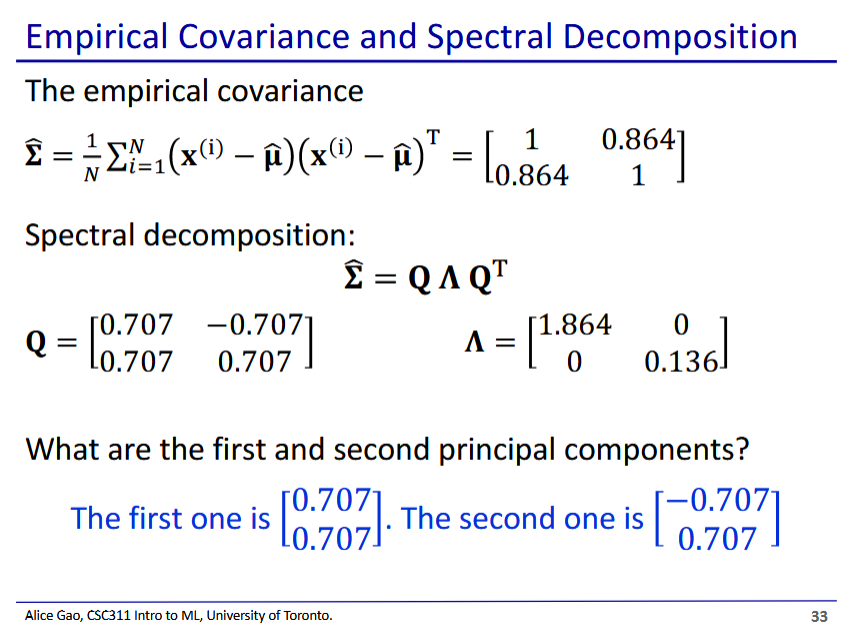




* Even before starting, the first principle component will point to the top right
  + Will capture the most variance in the data

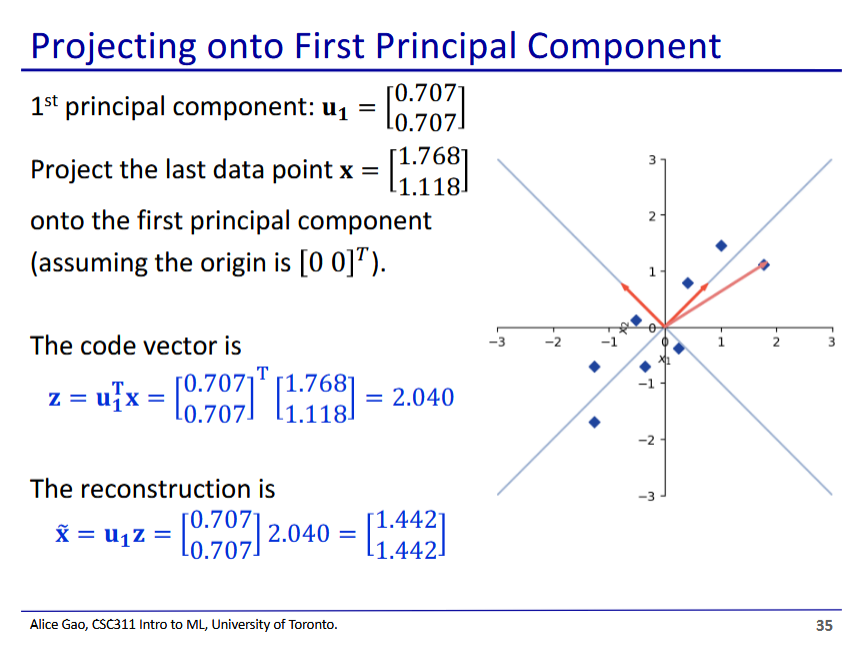


* PCA is sensitive to scale like KNN
  + If a feature has a much larger scale, it will have a larger variance which will bias PCA

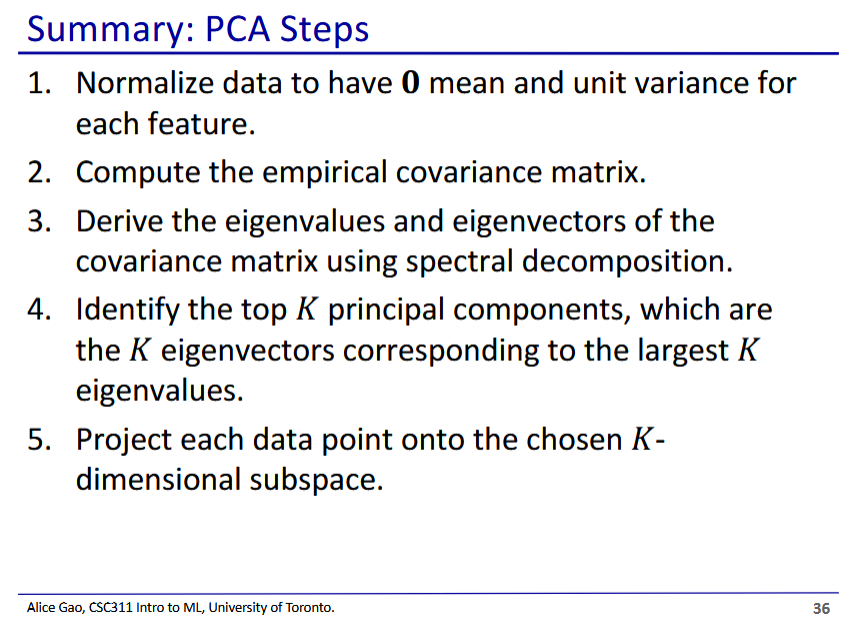


The first principle component pertains to the eigenvector with the largest eigenvalue, second pertains to second and so forth

* We make covariance matrix
  + (2x2) since we have 2 features
* Then we do spectral decomposition
* What are the first and second principal components?
  + We first find the first and second largest eigenvalues (diagonals of matrix on the right), then get their eigenvectors (columns in matrix on left)
  + Q1 and then Q2



* Then we can calculate the reconstruction of the projection in the original dimensions



Expectations for final

* Project the vector onto a subspace (1 dimensional and multidimensional)
  + Also be able to do reconstruction of projected vector
* The proof
* Won't need to calculate covariance or spectral decomposition
  + But we will need to be able to order principal components