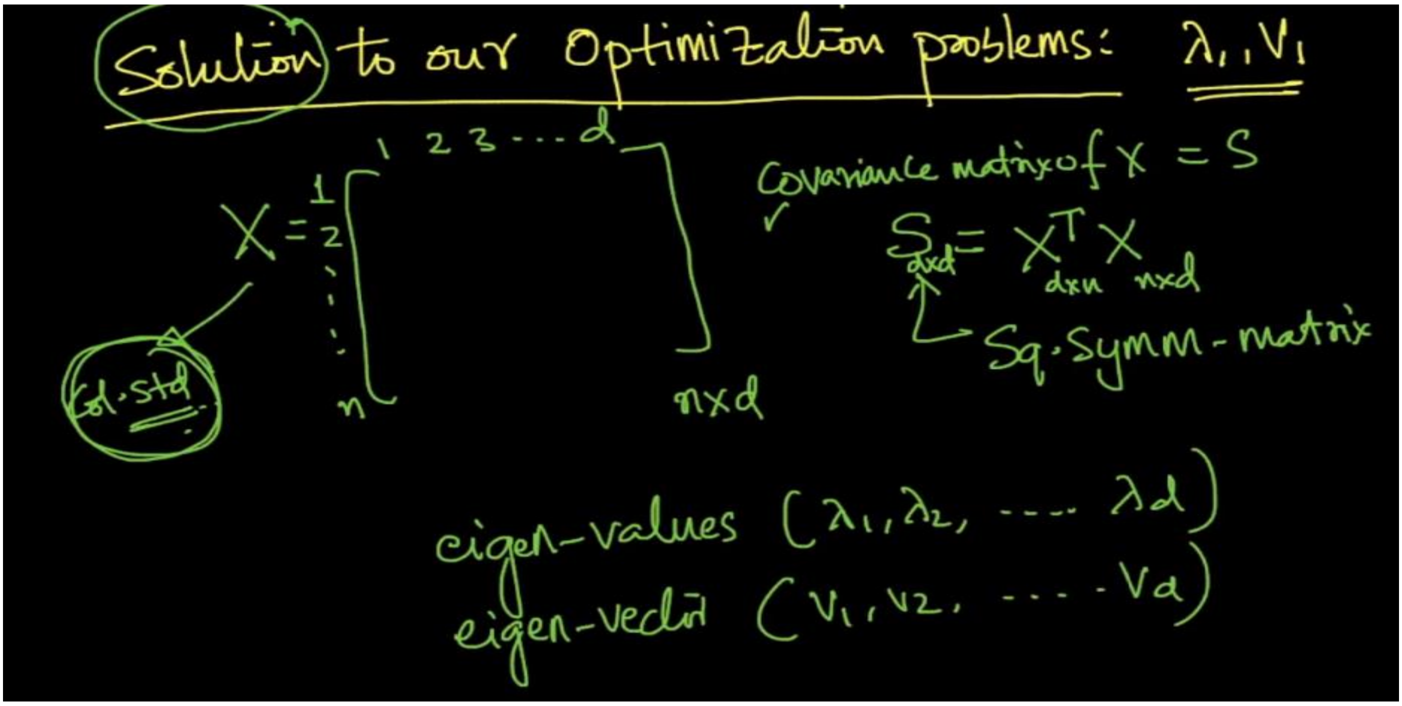
Here we will find the value of max u1, using **Eigen values and Eigen vectors**.

Wkt, co-variance matrix of X = S, where S is symmetric matrix.

And for **d** no of features there will be **d eigen-values** and **d eigen-vector.**

Eigen values are represented by l**ambda**, and Eigen vectors are represented by **v**.

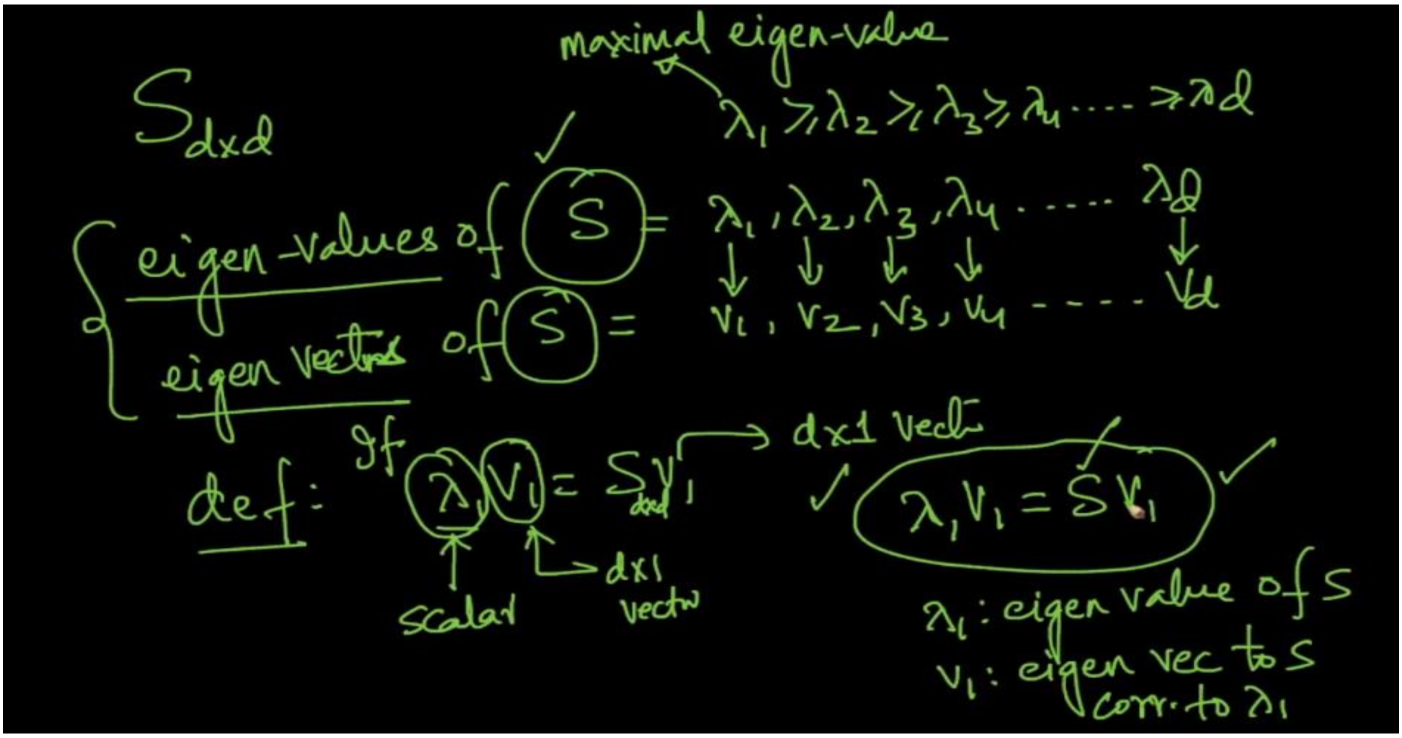


Lambad and V are eigen values and vectors or S respectively, only if:

Lambda$_1$ * V_1 = S_{d*d} V_1

And lambda$_1$ >= lambda$_2$ >= lambda$_3$.............................. >= lambda$_d$

V_1 >= V_2 >= V_3 >=…………………………….. $V_{d,}$ here each eigen vector is linked to each eigen value

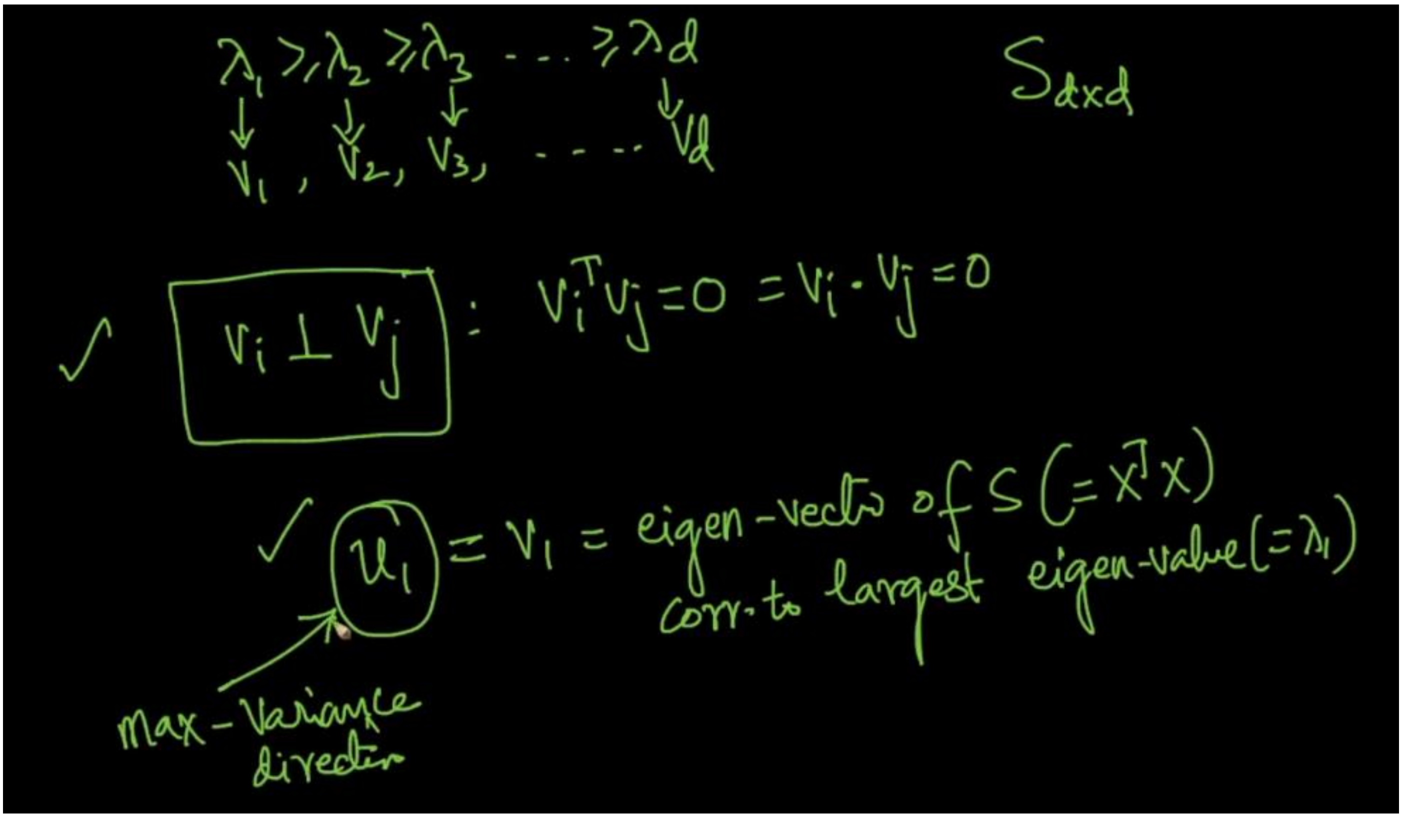


Each Eigen vector is perpendicular to another, that means there dot product is 0, since cos 90 is 0.

And **u1 = V_1,** where

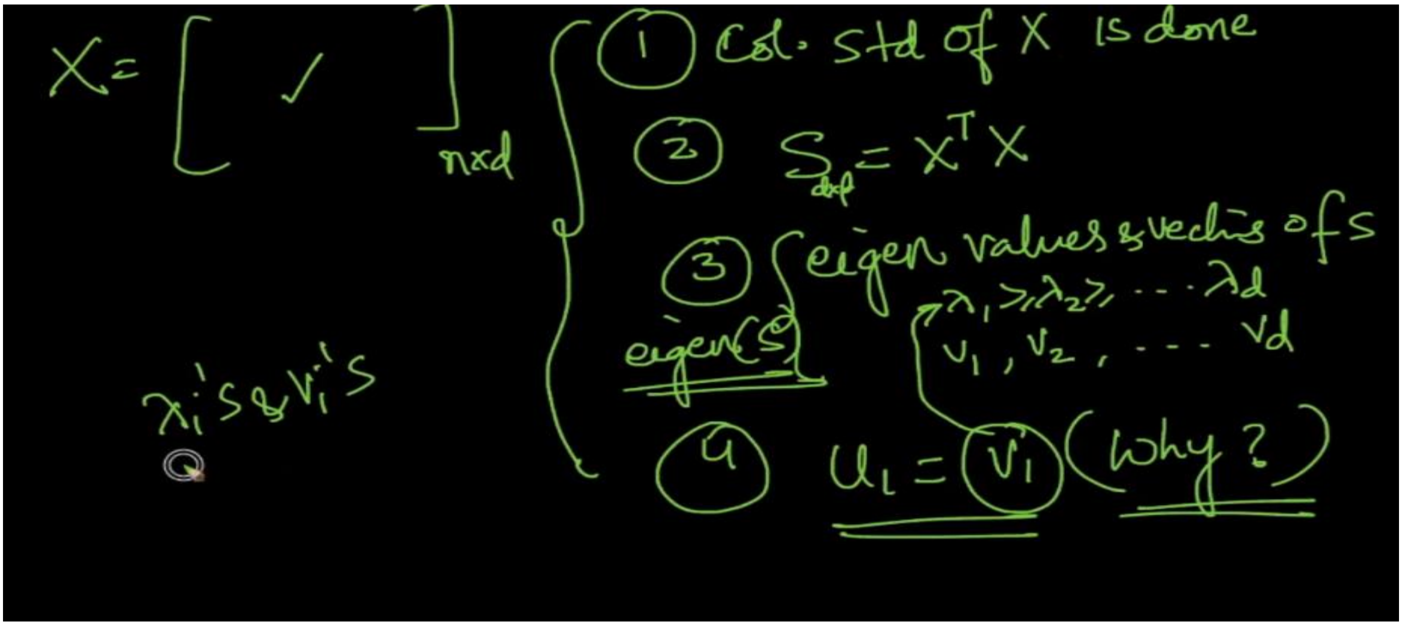
U1 is max-variance direction,

V_1 is eigen vector of S, corresponding to largest eigen value $lamba_1$



Steps to find u1 using eigen vectors an values

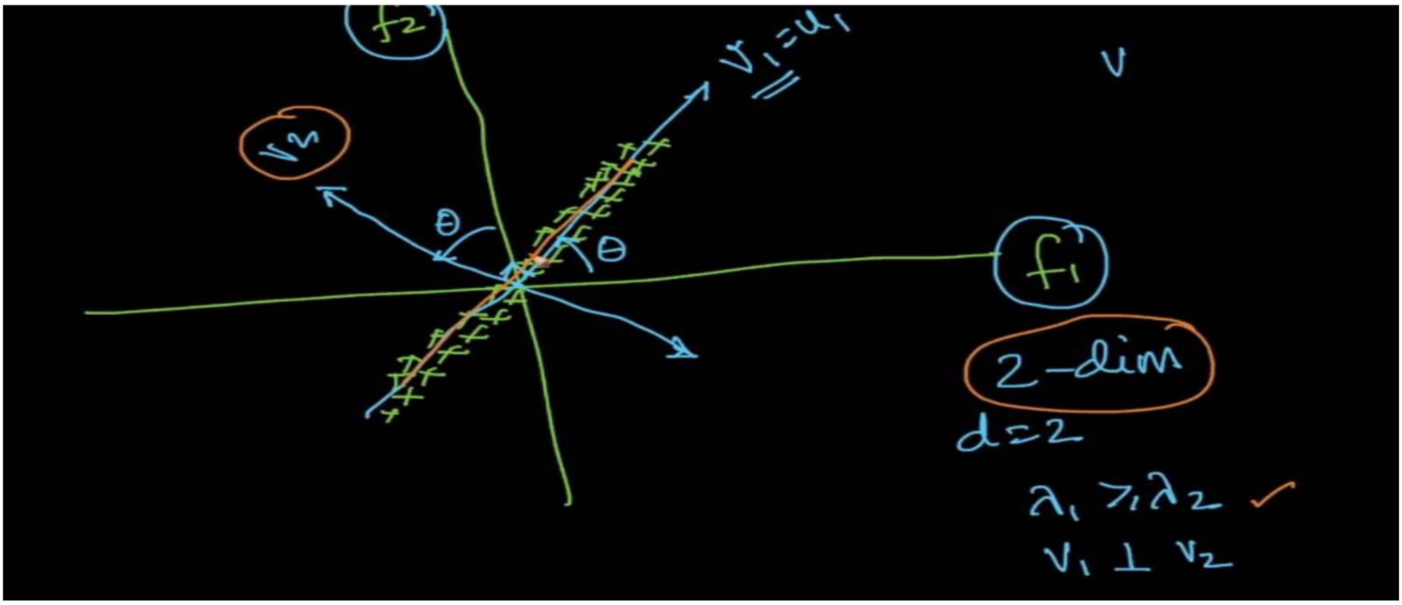
1. Do Column standardization of X.
2. Find co-variance matrix S = X^T.X
3. Find eigen values (lambda$_i$) and eigen vectors (V_i) of S.
4. Then u1 will be equal to V_1.



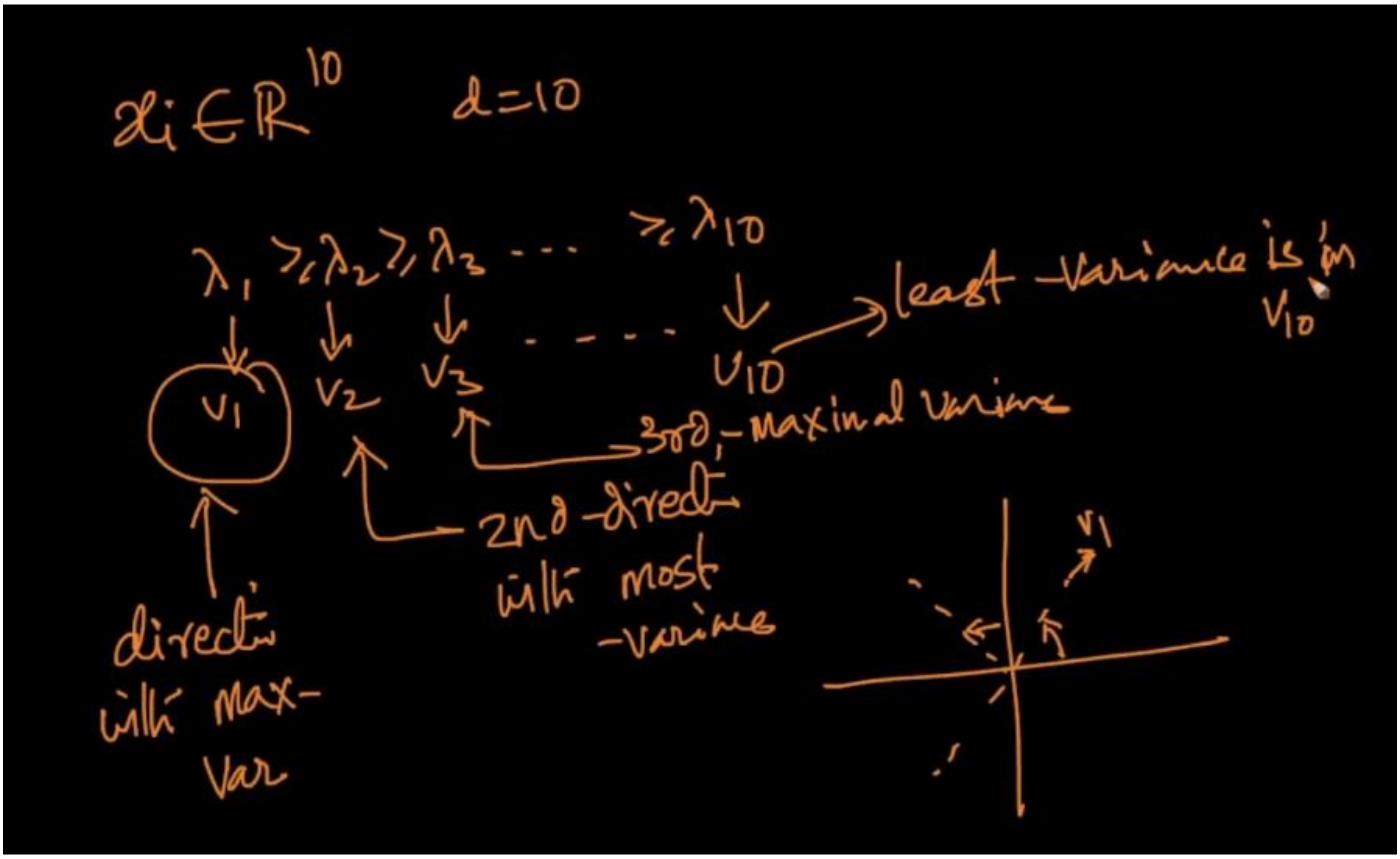
Why u1 = V_1?

Below figure shows V1 fo 2-dimensions, and it's perfectly the containing all the x's.

Here v2 is perpendicular to v1.



Now above image shows only for 2-dimensions, but for more dimensions let's say 10, we will have 10 eigen values and 10 eigen vectors, where V_1 will be the direction with max-variance ie (U1), V_2 will be 2[nd] maximal variance direction, V_3 will be 3[rd] maximal variance direction and so on.

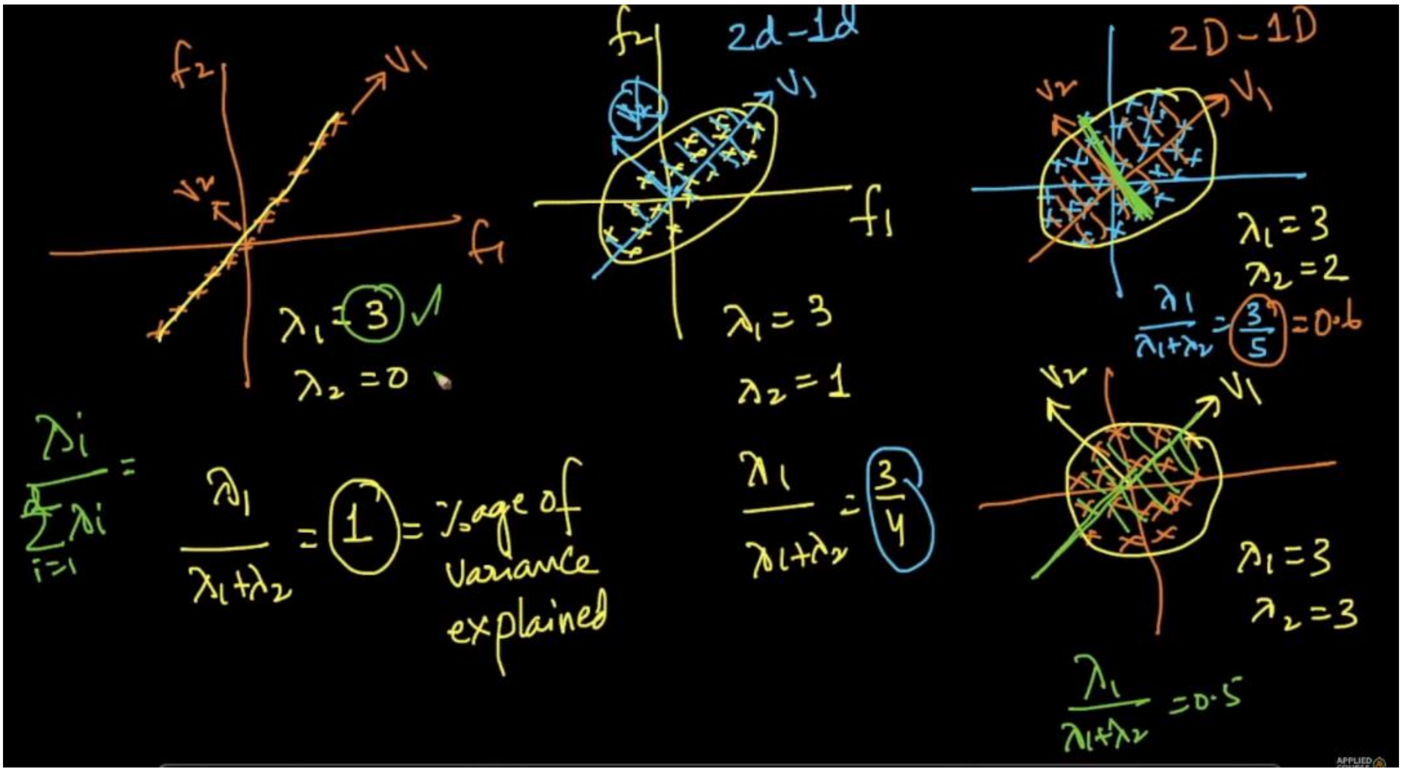


Given eigen values lambda, we can say and find how much percentage of information will be retained by just taking new feature that is our new vector V_1 or percentage of variance explained by

Lambda1 / (lamda1 + lambda2).

in the first case where lambda1 (wrt v1) = 3 and lambda2(wrt v2) =0 i.e, means from 2d => 1d if we simply ignore v2 still we retain 100% of information because lamda1/ (lamda1+lamda2) => 3/(3+0) =>100%

2nd case have ¾ ie 75%, that means V_1 retains 75% of information, and so on.



Comments:

1. Here the optimal unit vector u1 is used to indicate the direction of eigen vector v1 in which the maximum variance is preserved. We get the components of vector from v1 and it's direction is given by u1. Only for indicating the direction of 'v1', we use u1
2. Must read ref: https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues
3. Eigen faces and PCA in face detection: https://sandipanweb.wordpress.com/2018/01/06/eigenfaces-and-a-simple-face-detector-with-pca-svd-in-python/
4. Feature Scaling for PCA **using sklearn:** **https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html**
5. Applying PCA on **iris dataset** **https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60**