Principle Component Analysis

Implementing dimensionality reduction

PCA

- Remember that for the matrix X we can perform the Singular Value Decomposition as:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{T}$$

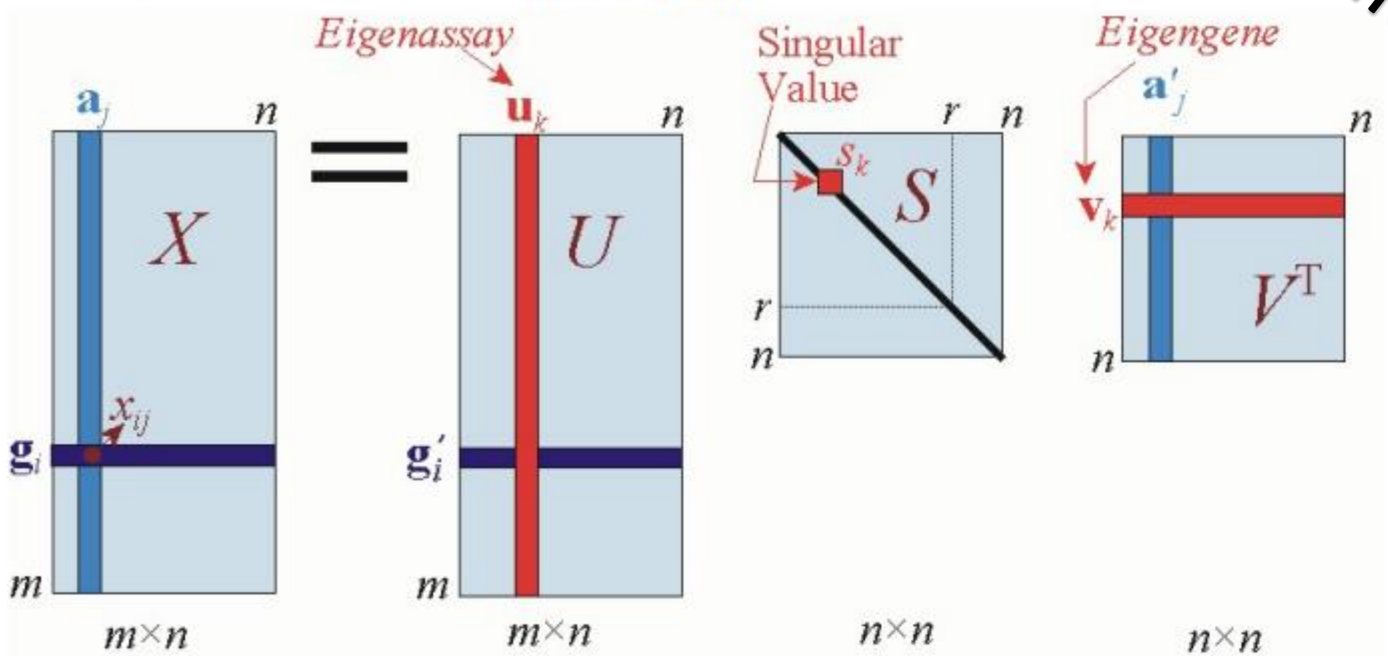
Where **US** gives us the coordinates of a sample in **X** in the space of principle components:

$$\mathbf{X}/\mathbf{V}^{T} = \mathbf{U}\mathbf{S}$$

SVD and PCA

Lecture 5, slide 21

- Singular Value Decomposition
  - For any matrix X: $X = USV^T$



Data X, one row per data point Data is zero-centred	US gives coordinates of rows of X in the space of principle components	S is diagonal, $S_k > S_{k+1}$, S_k is k^{th} largest eigenvalue	Rows of V^T are unit length eigenvectors

Projecting the data into PC space

- Given that *US* is our original observations projected into Principal Component Space, we will need to solve $X = USV^T$ for US.

- To remove the V^T on the right, we can utilise the fact that a square orthogonal matrix multiplied by its inverse (the transpose) is the Identity Matrix and can be removed.

Projecting our data into PC space

$$X = USV^T$$
$$XV = USV^TV$$
$$XV = USI$$
$$XV = US$$

PCA for dimensionality reduction

- Given some observed data *X:*
  - Get the mean of the observed data: μ.
  - Perform a Singular Value Decomposition on the mean centred *X*: obtaining *U,S* and *V*.
  - Mean-centre the points in X.
  - Project points onto the new principle component space.

PCA in practice

- Given an N-by-F matrix of **training** data *Xtrn:*
  - μ = *mean(Xtrn);*
  - *Xtrn_centred = Xtrn –* μ;
  - *[U,S,V'] = svd(Xtrn_centred);*
  - *Xtrn_projected = Xtrn_centred * V;*

  - Given an M-by-F matrix of **testing** data *Xtst*:
  - *Xtst_centred = Xtst –* μ;
  - *Xtst_projected = Xtst_centred * V;*

How to reduce dimensions

- Select f < F principle component features.

  - *f = 10;*
  - *Xtrn_projected = Xtrn_projected(:, 1:f);*
  - *Xtst_projected = Xtst_projected(:, 1:f);*

- Or make the selection during the projection:
  - *Xtrn_projected = Xtrn_centred * V(:, 1:f);*
  - *Xtst_projected = Xtst_centred * V(:, 1:f);*