Pattern Classification and Recognition:

Feature Extraction/Generation

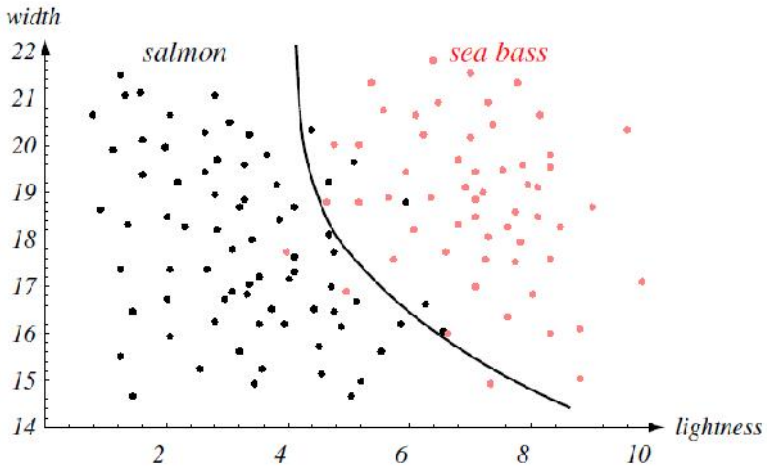
ECE 681

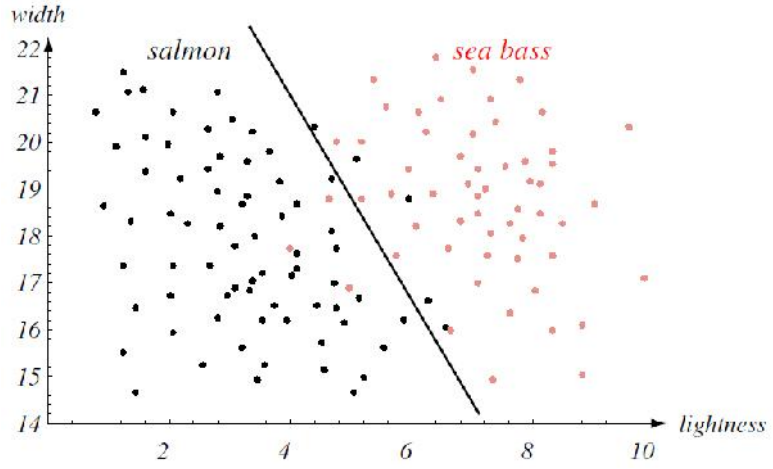
Spring 2016

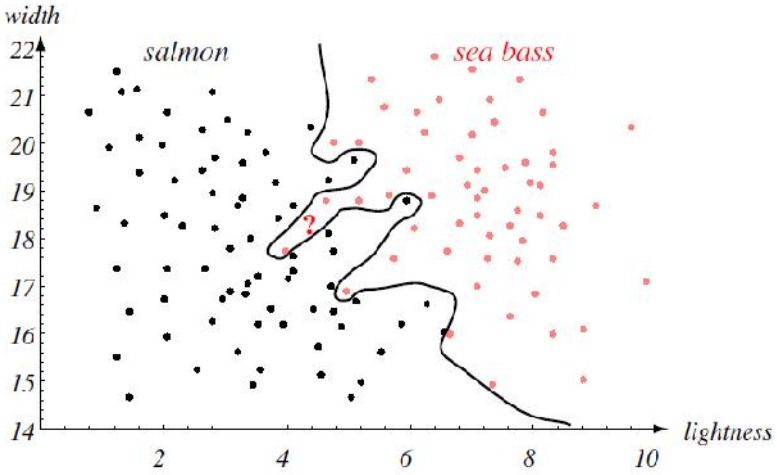
Stacy Tantum, Ph.D.

Occam's Razor

Classifiers should be no more complicated than necessary

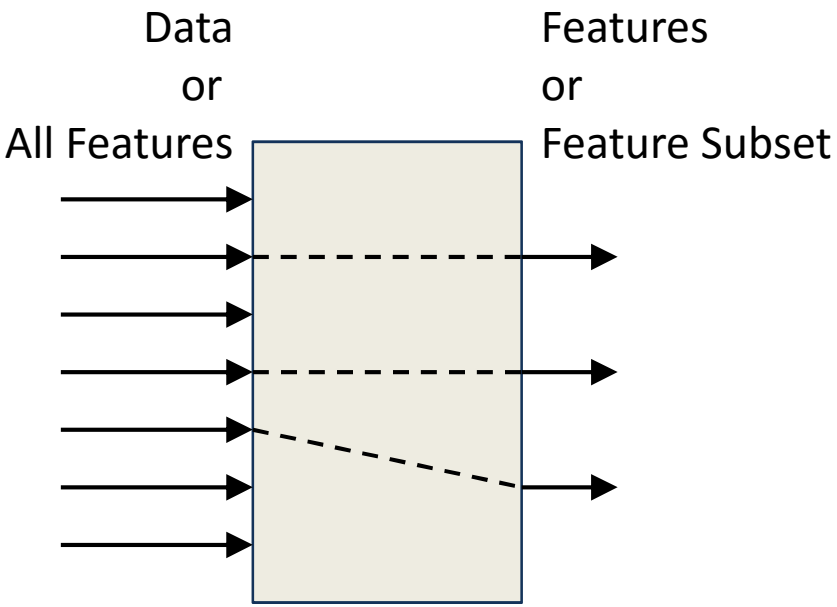




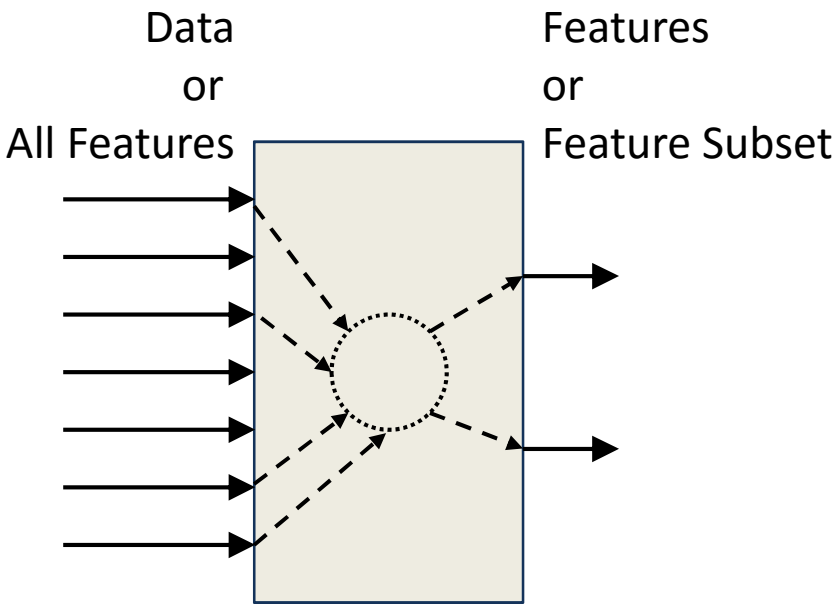


Reducing Complexity

Select Data/Features

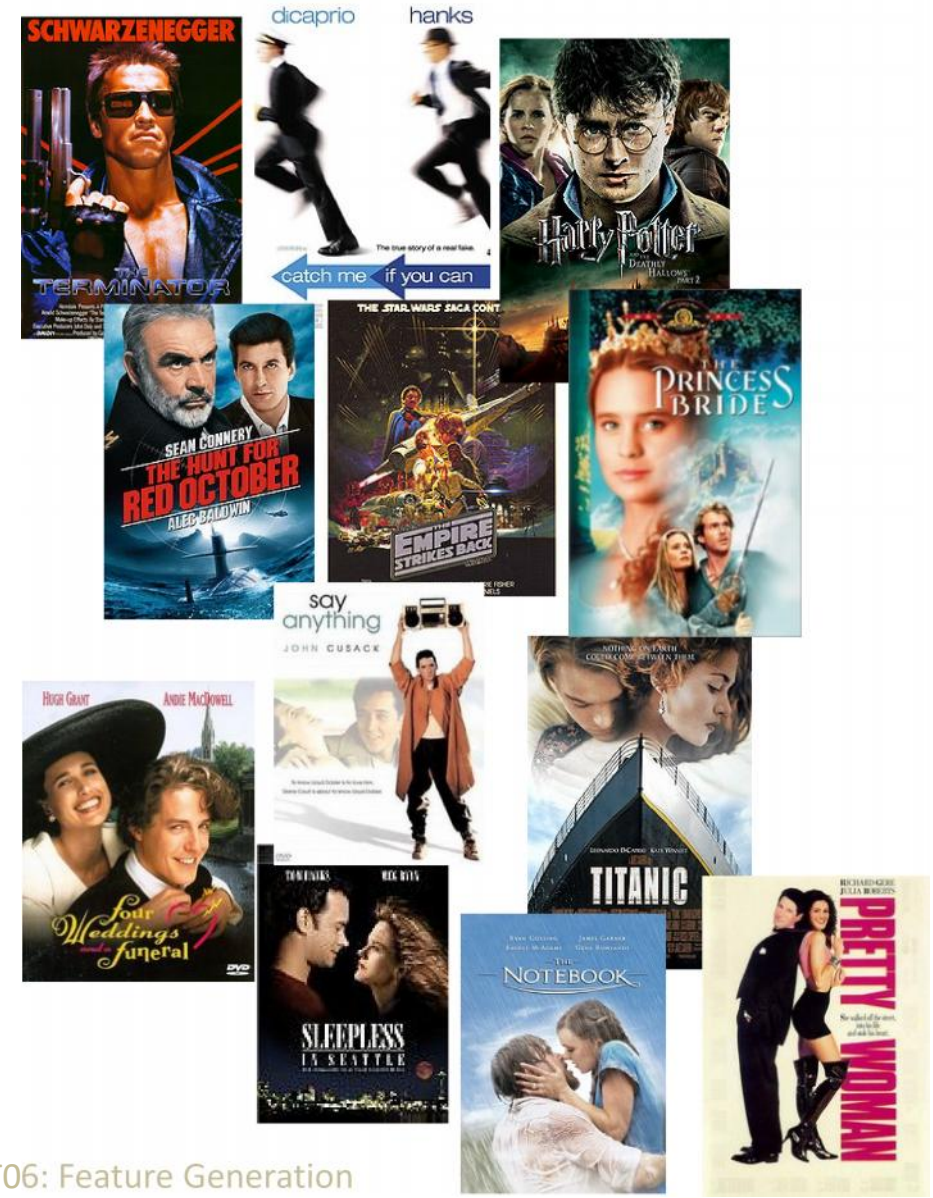


Transform Data/Features



Incorporate Problem Domain Knowledge

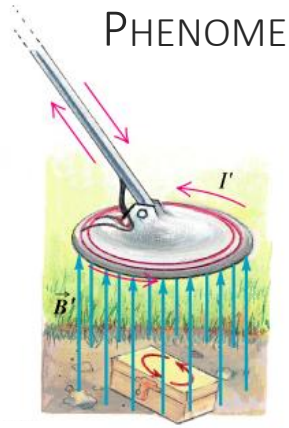
From Knowledge comes Creative Features



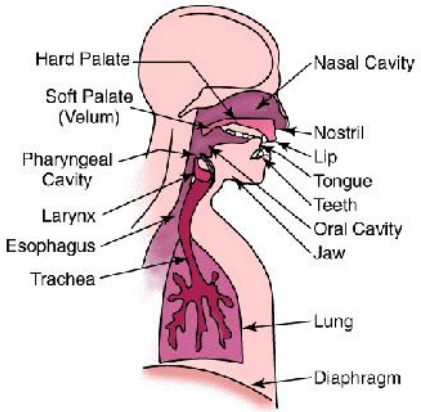
POSSIBLE FEATURES

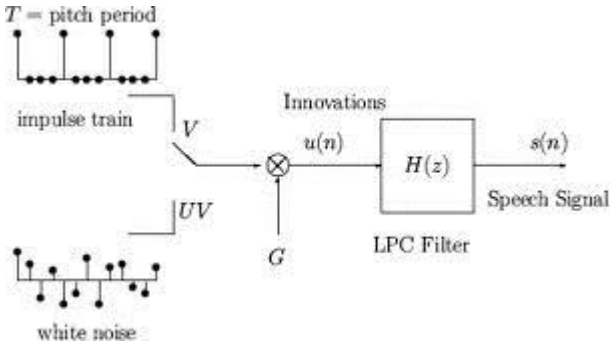
From Knowledge comes Model-Based Features

Phenomenological Model



$$s(t) = \left(\sum_{n=1}^{N} A_n e^{-\tilde{}_n t} \right) + n(t)$$

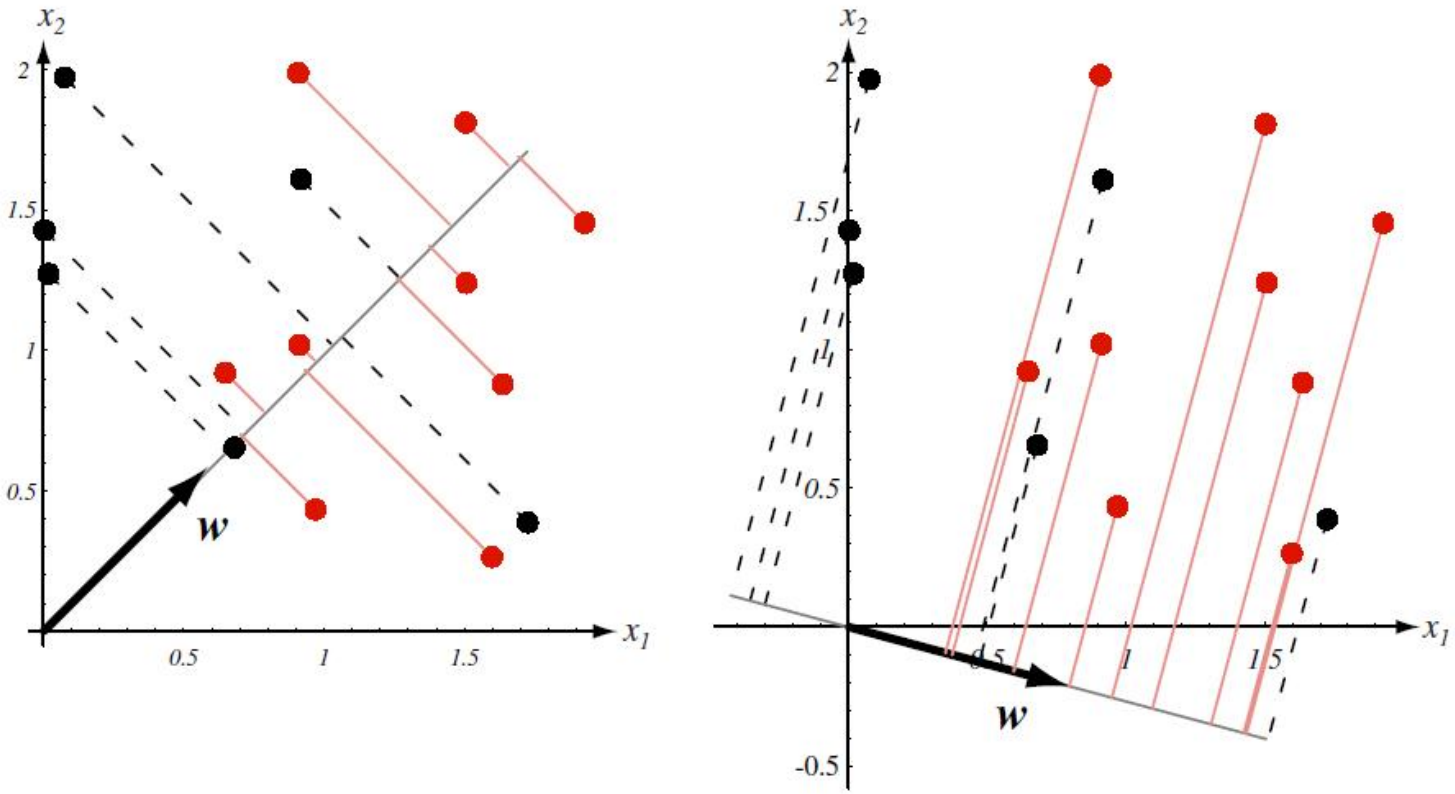




Possible Features

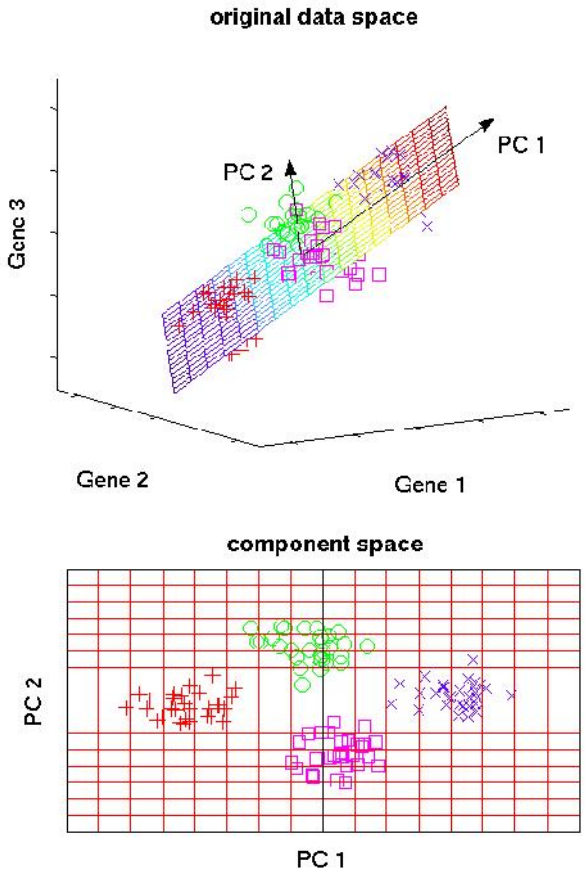
Data-Based Dimensionality Reduction

Most often relies on projecting the data to a lower-dimensional orthogonal space



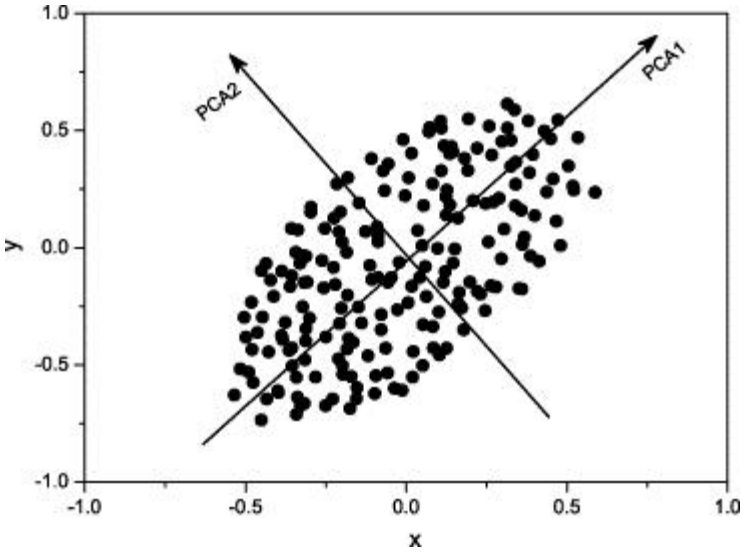
Principal Components Analysis
(Karhunen-Loève Transform)

Linear transformation of a high-dimensional input vector to a lower-dimensional vector with *uncorrelated* components

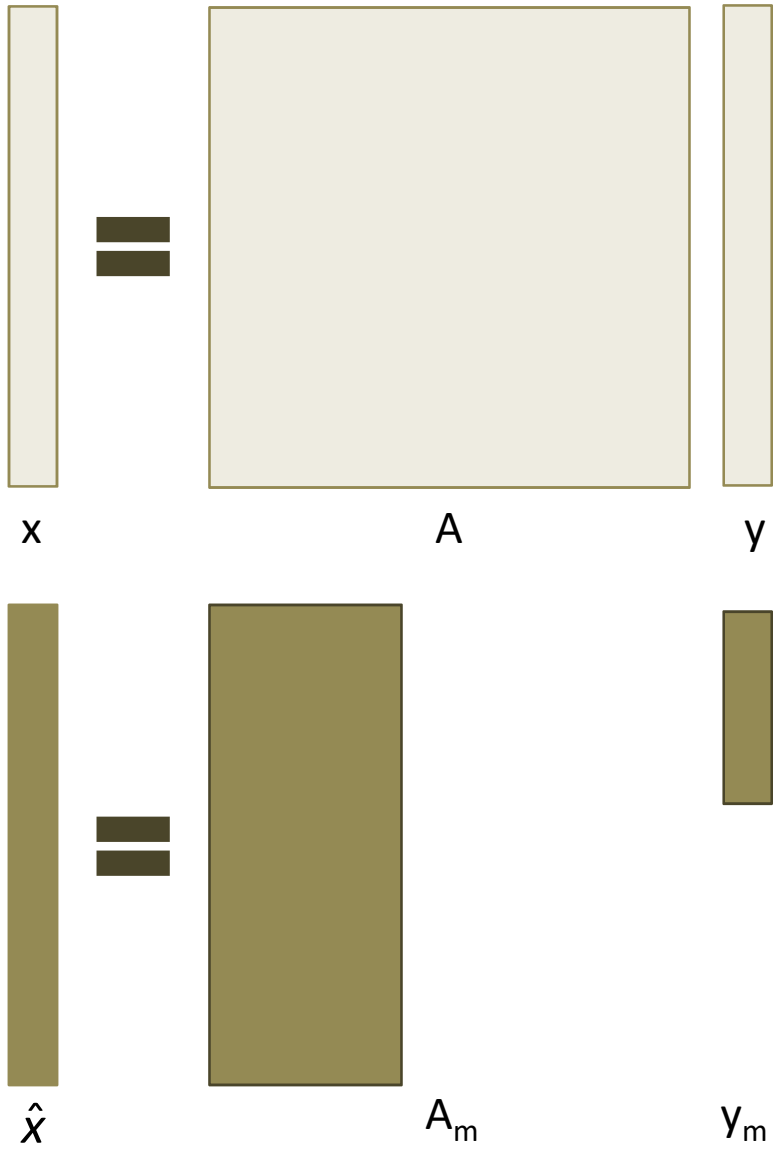


PCA: Maximum Total Variance

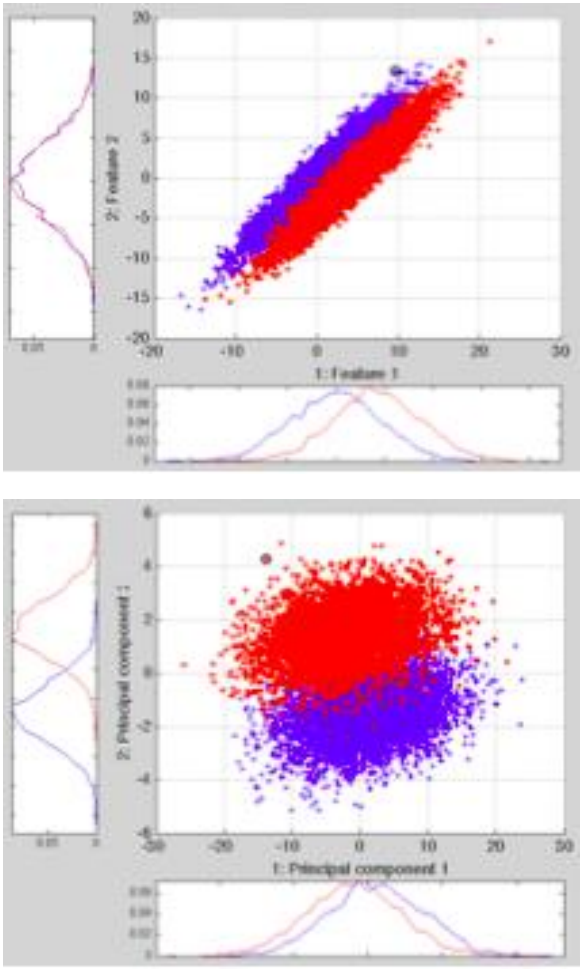
1. Find a projection that explains as much variance in the data as possible
2. Find a projection (orthogonal to those already found) that explains as much of the remaining variance in the data as possible
3. Repeat



PCA: Minimum MSE Approximation



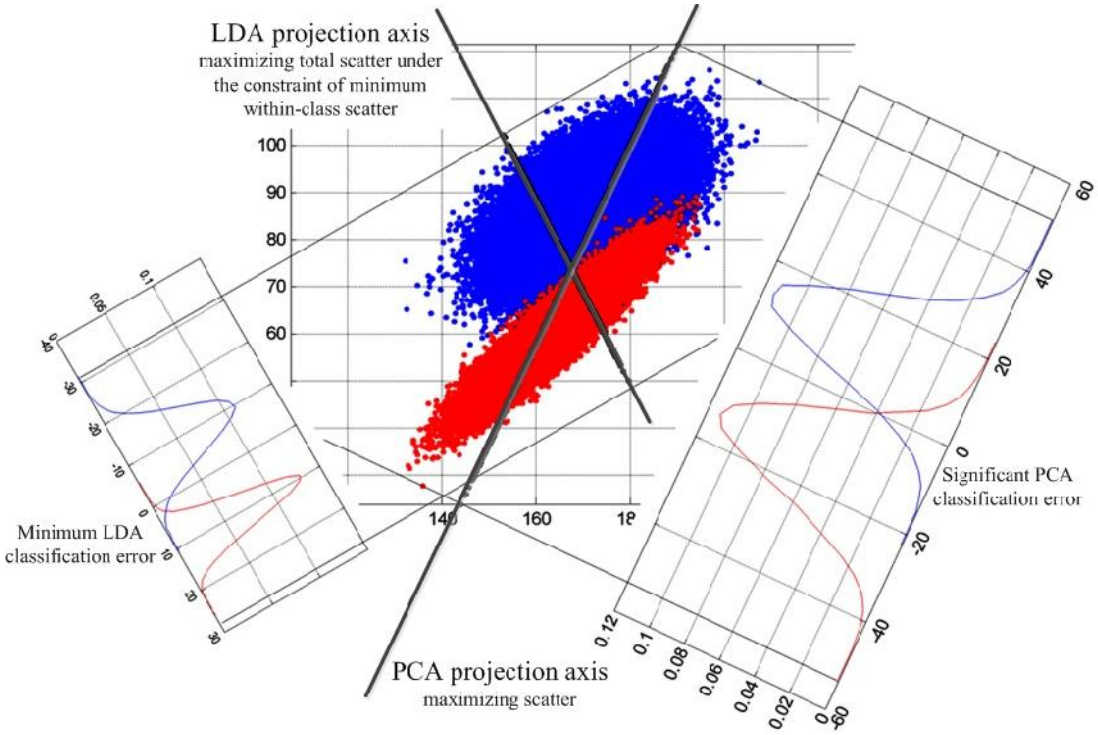
PCA: Dimensionality Reduction



PCA: Limitations

PCA may perform poorly (or fail) if the input isn't Gaussian

PCA seeks only to find projections that maximize the variance of the data, nothing about it is optimized for *classification*

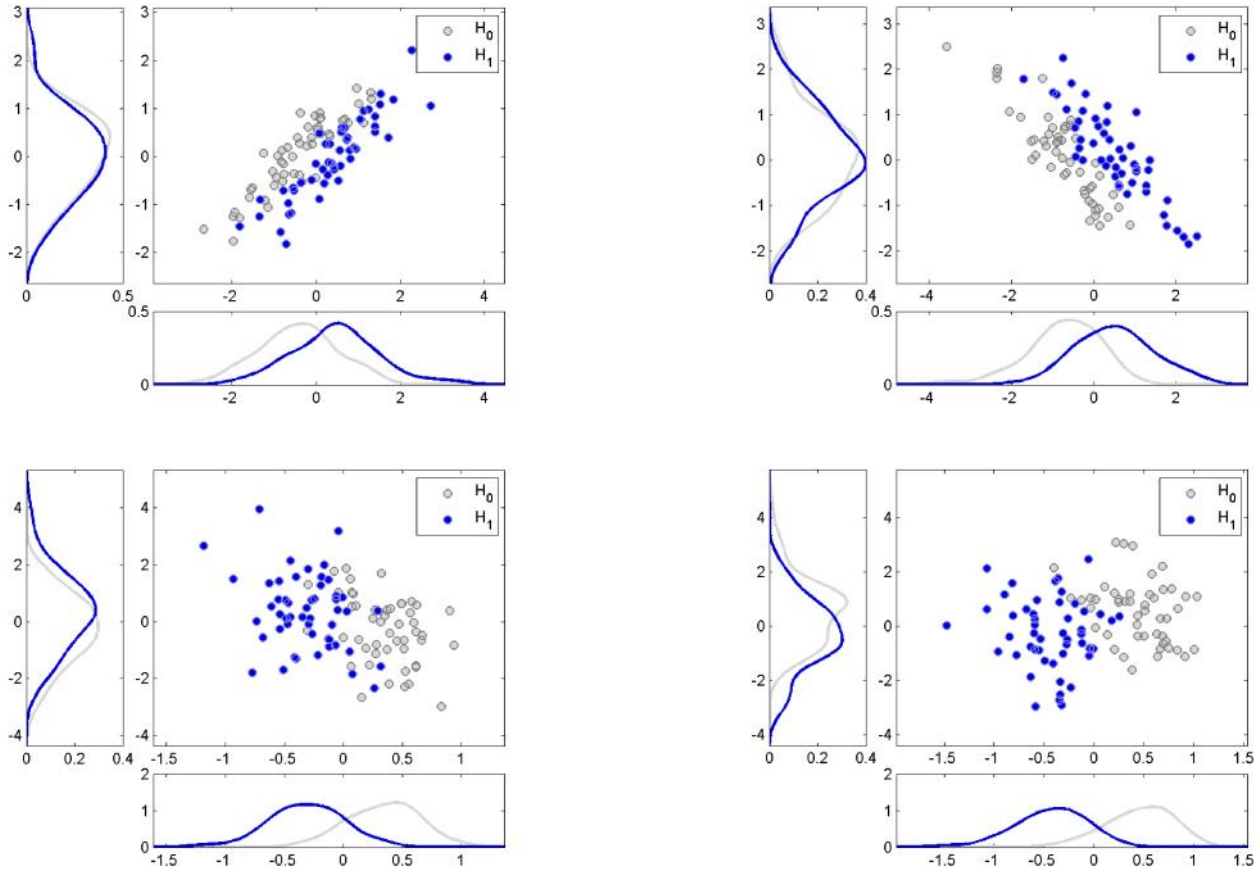


PCA Big Picture

Transformed Variable: $y=A^T x$

$(A = \text{eigenvectors of } R_x)$

Linear projection (transformation) of original data (x) such that components of new data (y) are uncorrelated



PCA Coding Tips

Transformed Variable: $y=A^T x$

$(A = \text{eigenvectors of } R_x)$

See Matlab functions `corr` and `eig`, also `pca`

What do we need for input?

What do we want from output?