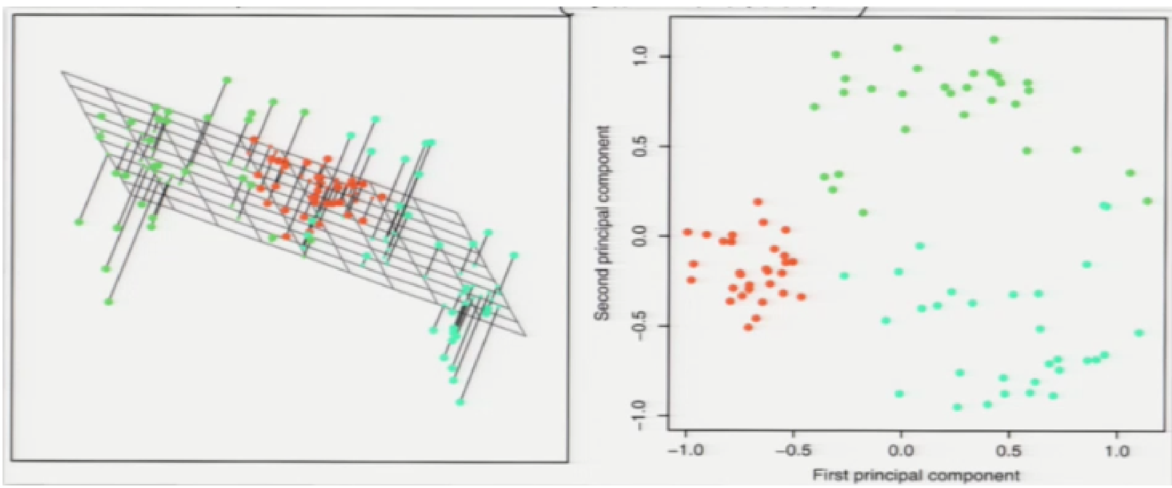
04/11/2016

Unsupervised Learning

- We have sample points, but no labels!
- No classes, no y-values nothing to predict.
- Goal: Discover structure in the data.
- Example:
  - Clustering: partition data into groups of similar/nearby points.
  - Dimensionality reduction: data often lies near a low-dimensional subspace (or manifold) in feature space; matrices have low-rank approximations.
  - Density estimation: fit a continues distribution do discrete data.

Principal Component Analysis (PCA)

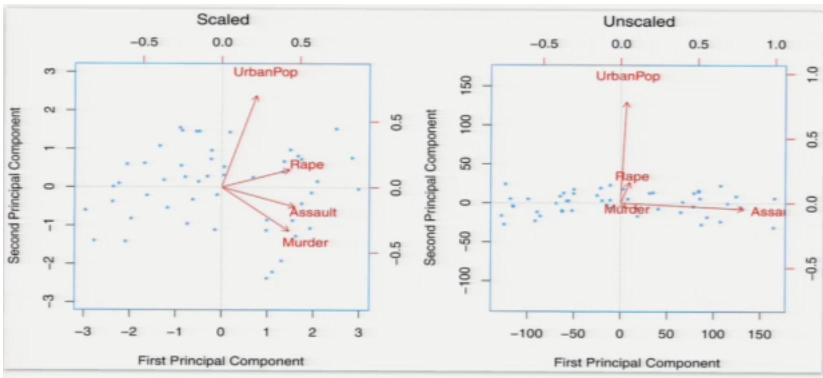
- Goal: Given sample points in $\mathbb{R}^d$, find k directions that capture the variation (dimensionality reduction).



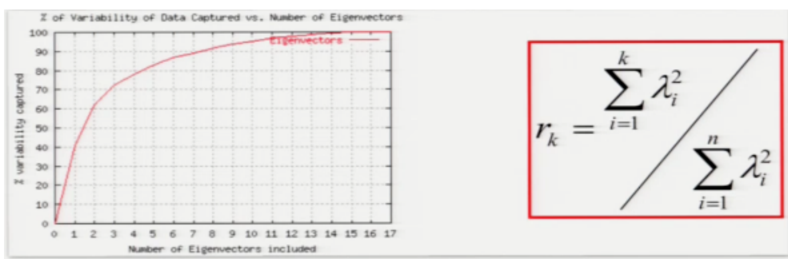
- Why?
  - Find a small basis for representing variations in complex things.
  - Reducing number of dimensions make some computational cheaper, e.g. regression.
  - Remove irrelevant dimensions to reduce overfitting in learning algorithms. Like subset selection, but we can choose features that aren't axis-aligned.
  -

- Let X be and nxd design matrix.
- From now on assume X is centered: mean X_i is zero.
- Let w be a unit vector.
- The orthogonal projection of point x onto vector w is $\tilde{X} = (X \cdot w)w$.
- If w not unit, $\tilde{x} = \frac{x \cdot w}{||w||^2} w$.
- Given orthonormal directions $v_1, \ldots, v_k$ $\tilde{x} = \sum_{i=1}^{k} (x \cdot v_i) v_i$. $x \cdot v_i$ are the coordinates in principal components space.

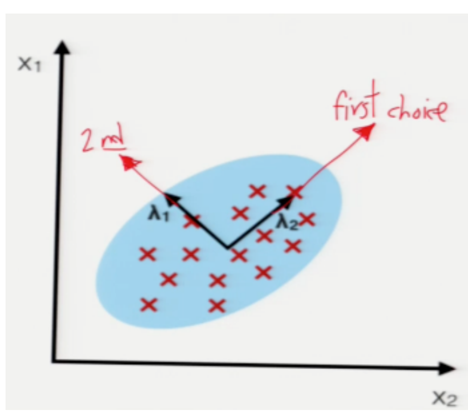
- $X^T X$ is square, symmetric, positive semidefinite, dxd matrix.
- Let $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_d$ be its eigenvalues.
- Let $v_1, v_2, \ldots, v_d$ be corresponding orthogonal unit eigenvectors.
- PCA Alg:
  - Center X.
  - Optional: Normalize X. Units of measurement different?
    - Yes: Normalize.
    - No: Usually don't.



  - Compute unit eigenvectors/values of $X^T X$.
  - Optional: choose k based on the eigenvalue size.



  - For the best k-dimensional subspace, pick directions $v_{d-k+1}, \ldots, v_d$.
  - Compute the coordinates of training/test data in principal components space.
- PCA Derivations:
  1. Fit a Gaussian to data with maximum likelihood estimation. Choose k Gaussian axes of greatest variance.



Recall that MLE estimates a covariance matrix $\hat{\Sigma} = \frac{1}{n} X^T X$

2. Find direction w that maximizes variance of projected data.

$$\begin{aligned}
\text{Var}(\{\tilde{x_1}, \tilde{x_2}, \ldots, \tilde{x_n}\}) &= \frac{1}{n} \sum_{i=1}^{n} \left(X_i \cdot \frac{w}{|w|} \right)^2 \\
&= \frac{1}{n} \frac{|Xw|^2}{|w|^2} \\
&= \frac{1}{n} \frac{w^T X^T X w}{w^T w}
\end{aligned}$$

If w is an eigenvector v_i, Rayleigh quotient $= \lambda_i \to$ of all eigenvector, v_d achieves maximum variance $\frac{\lambda_d}{n}$. One can show v_d beats every other vector too. Then pick v_{d-1}, then $v_{d-2}, \ldots$

3. Find direction w that minimizes "projection error."

$$\begin{aligned}
\sum_{i=1}^{n} |X_i - \tilde{X_i}|^2 = \sum |X_i - \frac{x_i \cdot w}{|w|^2}|^2 = \sum (|X_i|^2 - (X_i \cdot \frac{w}{|w|})^2) \\
= \text{constant} - n(\text{variance from derivation 2}).
\end{aligned}$$

Minimizing projection error $\Leftrightarrow$ maximizing variance.

Eigenfaces

- X contains n images of faces, d pixels each.
- Face recognition: Given a query face, compare it to all training faces; find nearest neighbor in $\mathbb{R}^d$.
- Problem: Each query takes $\Theta(nd)$ time.
- Solution: Run PCA on faces. Reduce to much smaller dimension d'. Now nearest neighbor takes $\mathcal{O}(nd')$ time.