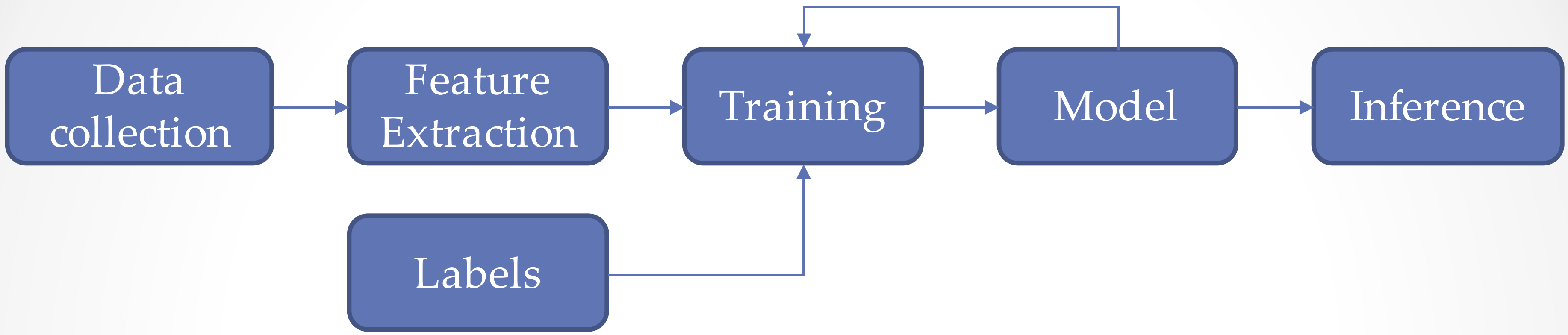
CAS 765 Fall'15
Mobile Computing and Wireless Networking

Rong Zheng

Feature Extraction

Machine Learning Pipeline



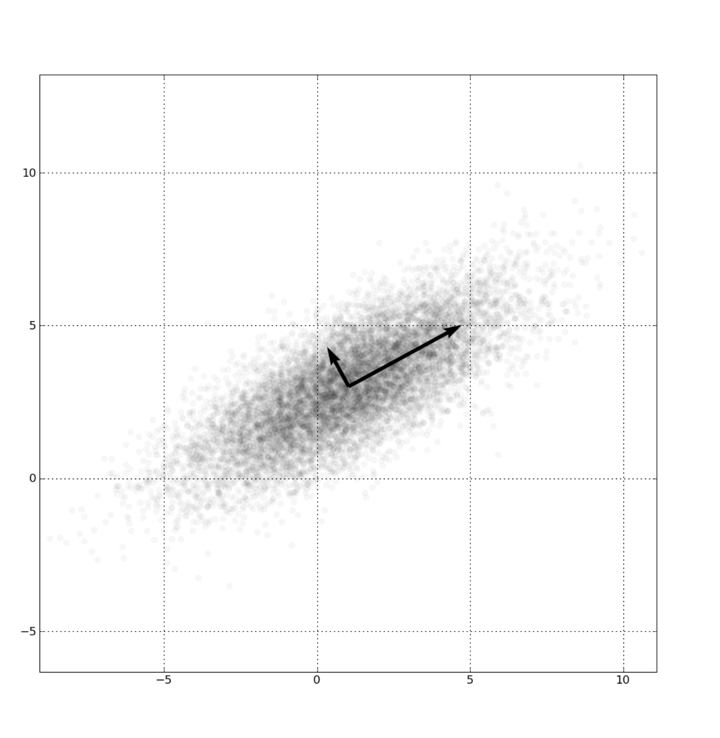
- Feature extraction builds derived values from measured data
  - In some cases, feature selection is further conducted
- Why not use measured data directly?
- Features typically are domain specific
  - E.g., term frequency-inverse document frequency for NLP, zero-crossing for EMG, edge, shape, SIFT in images
  - Requires understanding of the signal characteristics
- Not always possible to know which features are most useful

Learning Objectives

- Features for Electromyogram (EMG), Electroencephalography (EEG)
  - Both are time domain signals
- Dimension reduction: Principle component analysis

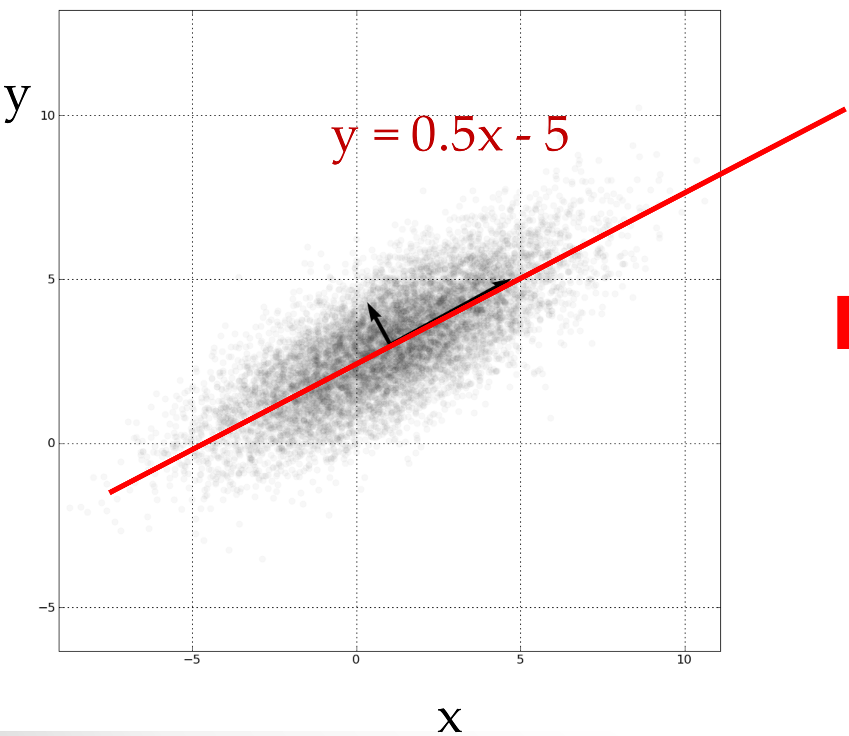
Principle Component Analysis (PCA)

- Convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables
  - Data compression
  - Data visualization
  - Noise reduction
  - Reduction of computation complexity
- Difference between correlation and dependence
  - Independent → uncorrelated
  - Uncorrelated $\nrightarrow$ independence

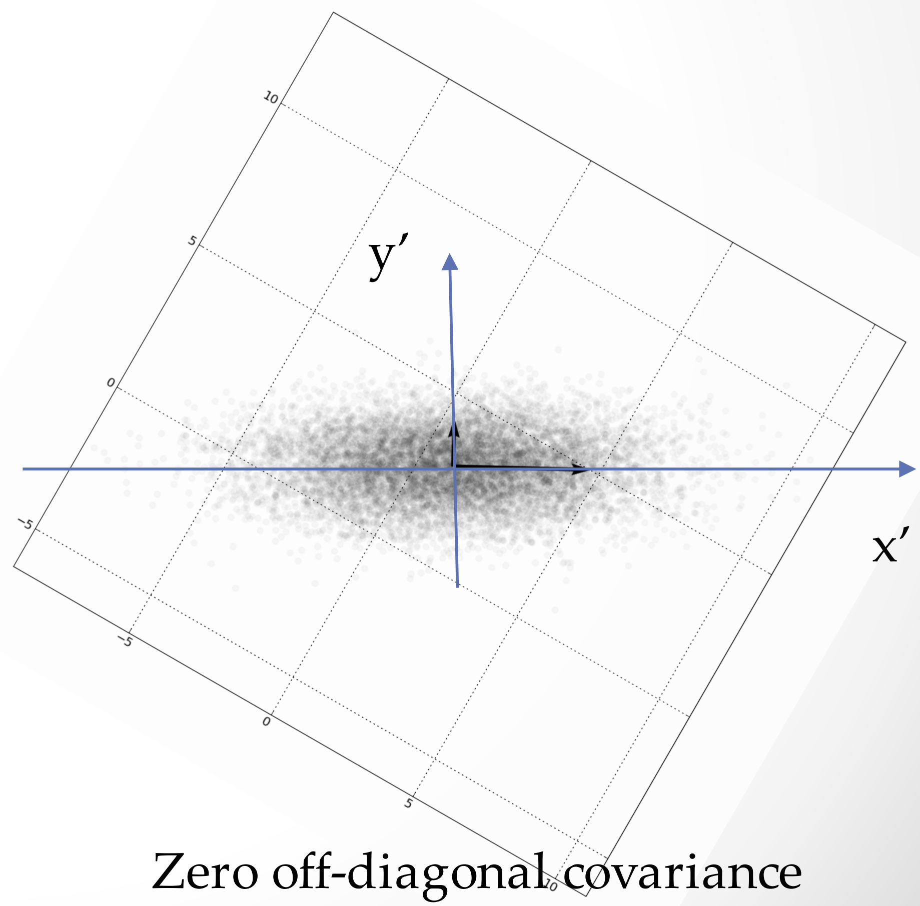


Intuition – De-correlation

- Correlation: observations concentrated around the line y = 0.5x -5



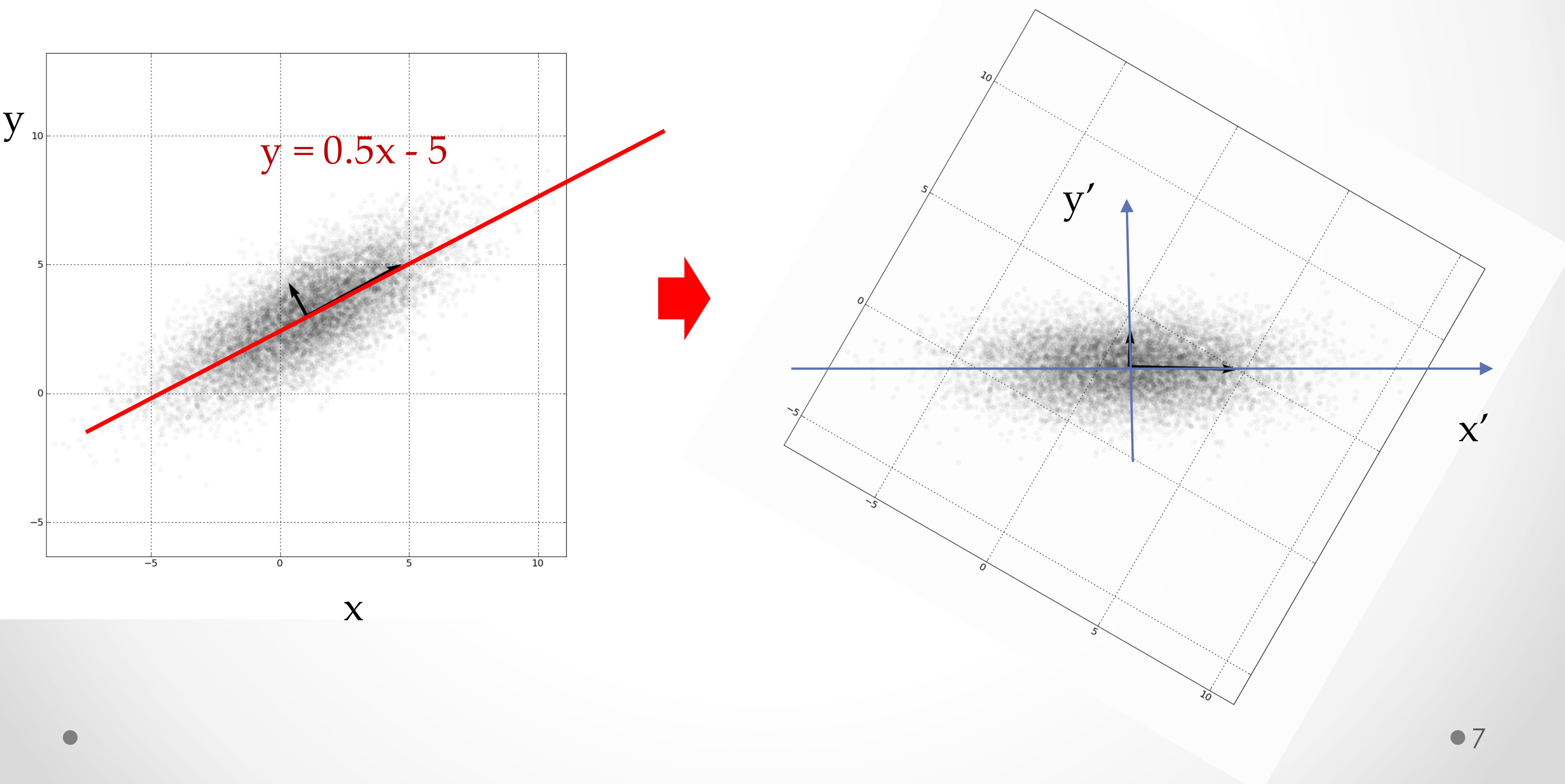
Non-zero off-diagonal covariance matrix



Zero off-diagonal covariance matrix

Intuition – Dimension Reduction

- After projections to another basis, pick the dimensions with higher variance



PCA algorithm

- Reduce data from n-dimensions to k-dimensions (n > k)
- Compute "covariance matrix":

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n}(x^{(i)})(x^{(i)})^T$$

- Compute "eigenvectors" of matrix:

```
[U,S,V] = svd(Σ);
```

$$\Sigma_{\mathrm{nxn}} = \mathbf{U}_{\mathrm{nxn}}\boldsymbol{\Lambda}_{\mathrm{nxn}}\boldsymbol{V}^{\boldsymbol{T}}_{\mathrm{nxn}}$$

where U, V are unitary matrices: $UU^{\mathbf{T}} = I, V^{\mathbf{T}}V = I$

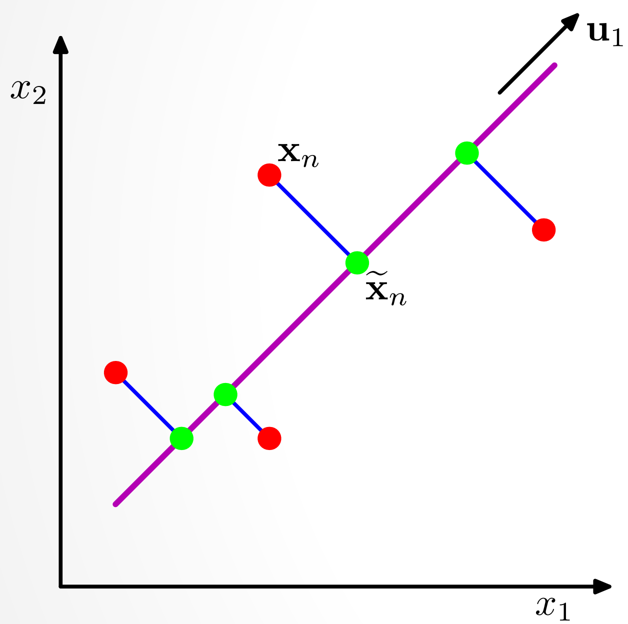
How to Select k?

- Typically, choose k to be smallest value so that

$$\frac{\frac{1}{n}\sum_{i=1}^{k}\|S_{ii}\|^2}{\frac{1}{n}\sum_{i=1}^{m}\|S_{ii}\|^2} \geq 1-\varepsilon$$

- Can be thought of as choosing the k dimensions that have the maximum variances

Interpretation– Optimization



N-dim to K-dim projection

$x \rightarrow \tilde{x}$ s.t., J$=\frac{1}{m}\sum_{l=1}^{m}\left\|x^{(l)} - \tilde{x}^{(l)}\right\|$ minimized

$\{u_i\}, i = 1,2,\dots,k$ s.t., $u_i^T u_i = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$

$$\tilde{x}^{(l)} = \underbrace{\sum_{i=1}^{k} {x^{(l)}}^T u_i u_i}_{\text{Projection}} + \underbrace{\sum_{i=k+1}^{n} \bar{x}^T u_i u_i}_{\text{Bias term}}$$

where $\bar{x} = \frac{1}{m}\sum_{l=1}^{m} x^{(l)}$

$$\begin{aligned} \mathrm{J} &= \frac{1}{m}\sum_{l=1}^{m}\left\|x^{(l)} - \tilde{x}^{(l)}\right\| \\ &= \frac{1}{m}\sum_{l=1}^{m}\sum_{i=1}^{k}\left({x^{(l)}}^T u_i - \bar{x}^T u_i\right)^2 \\ &= \sum_{i=k+1}^{n} u_i^T S\, u_i \end{aligned}$$

The general solution to the minimization problem is $\Sigma u_i = \lambda_i u_i$ and J $= \sum_{i=k+1}^{n} \lambda_i$