Dimensionality Reduction

Lecture 5

Outline

1. Overview
   a) What is Dimensionality Reduction?
   b) Why?

2. Principal Component Analysis (PCA)
   a) Objectives
   b) Explaining variability
   c) SVD

3. Related approaches
   a) ICA
   b) Autoencoders

Example 1: Sportsball

- Consider watching a sportsball game in 4K UHD (3840 × 2160)
  - Assume fixed camera

- How many pixels?

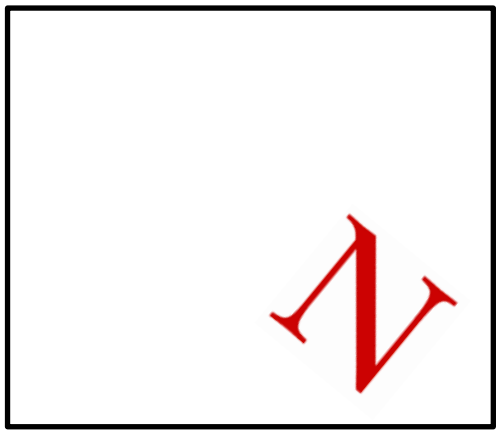
  >8M

- If all we care about is the location of the ball, how many dimensions of interest?

  3: (x, y, z)

Example 2: Image Embedding

- Take a small, known picture (e.g. N), embed it within a large white space (100x100)



- How many pixels?

  10K

- How many degrees of freedom?

  3 (h_transform, v_transform, rotation)

Dimensionality Reduction

- The removal of attributes thought to be irrelevant *for the task at hand*

- Many approaches
  - We focus on unsupervised, linear

Why Reduce Dimensionality?

- Noise reduction
- Imputing (missing) data
  - Estimate values for small d, reconstruct approximately (e.g. recommender systems!)
- Visualize data/results
  - Better for *hu-mahns*!
  - Think back to the effect of high-D in clustering!
- Reduce computational cost of other processes
  - Compression, clustering, classification, …

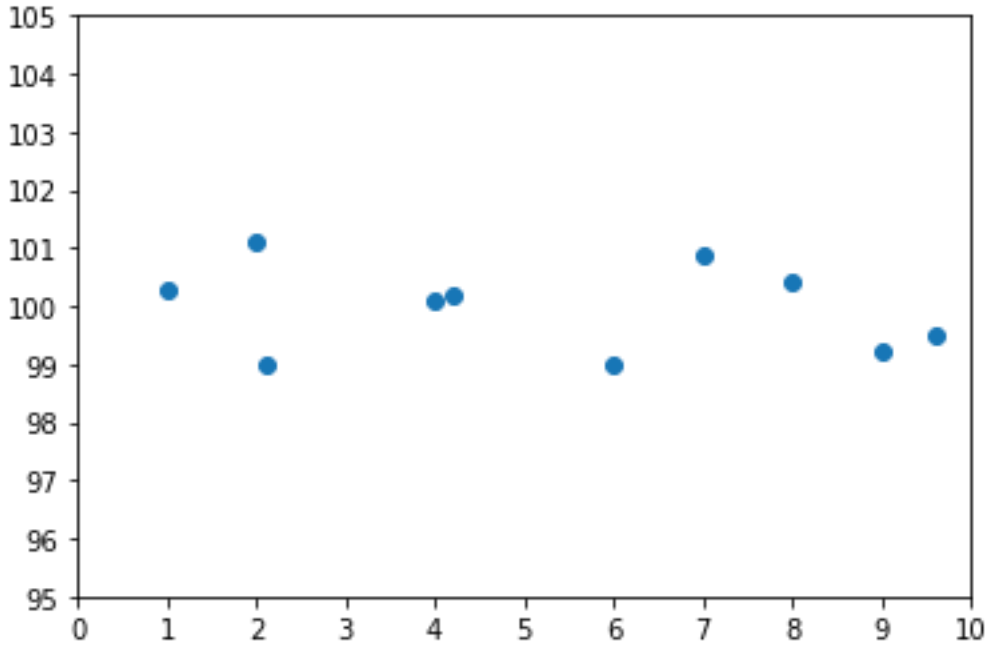
Principal Component Analysis (PCA)

- An approach that learns rotation axes from input data

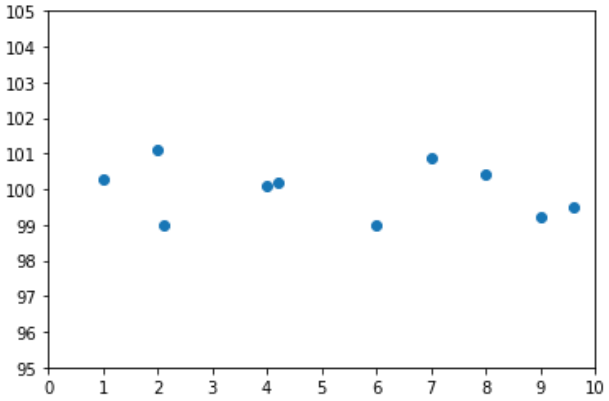
- Axes optimize two equivalent objectives
  - Maximize projected variance
  - Minimize reconstruction error

Example

Consider the following data…

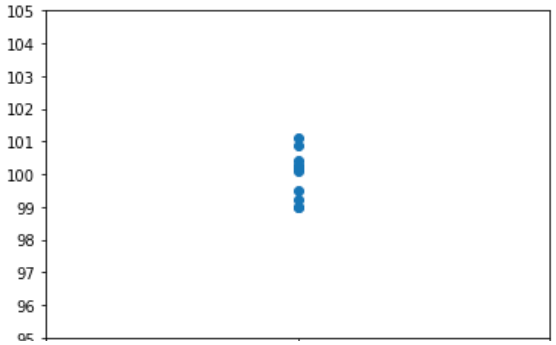


Intuition: Projection Variance

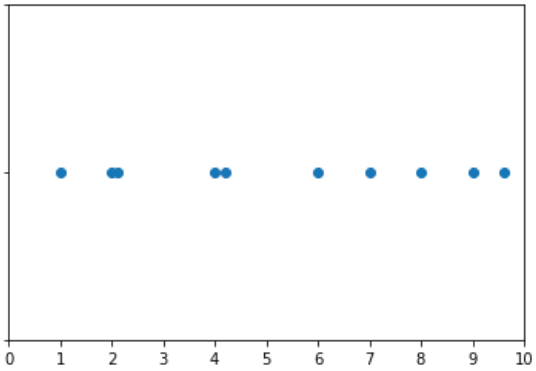


y x

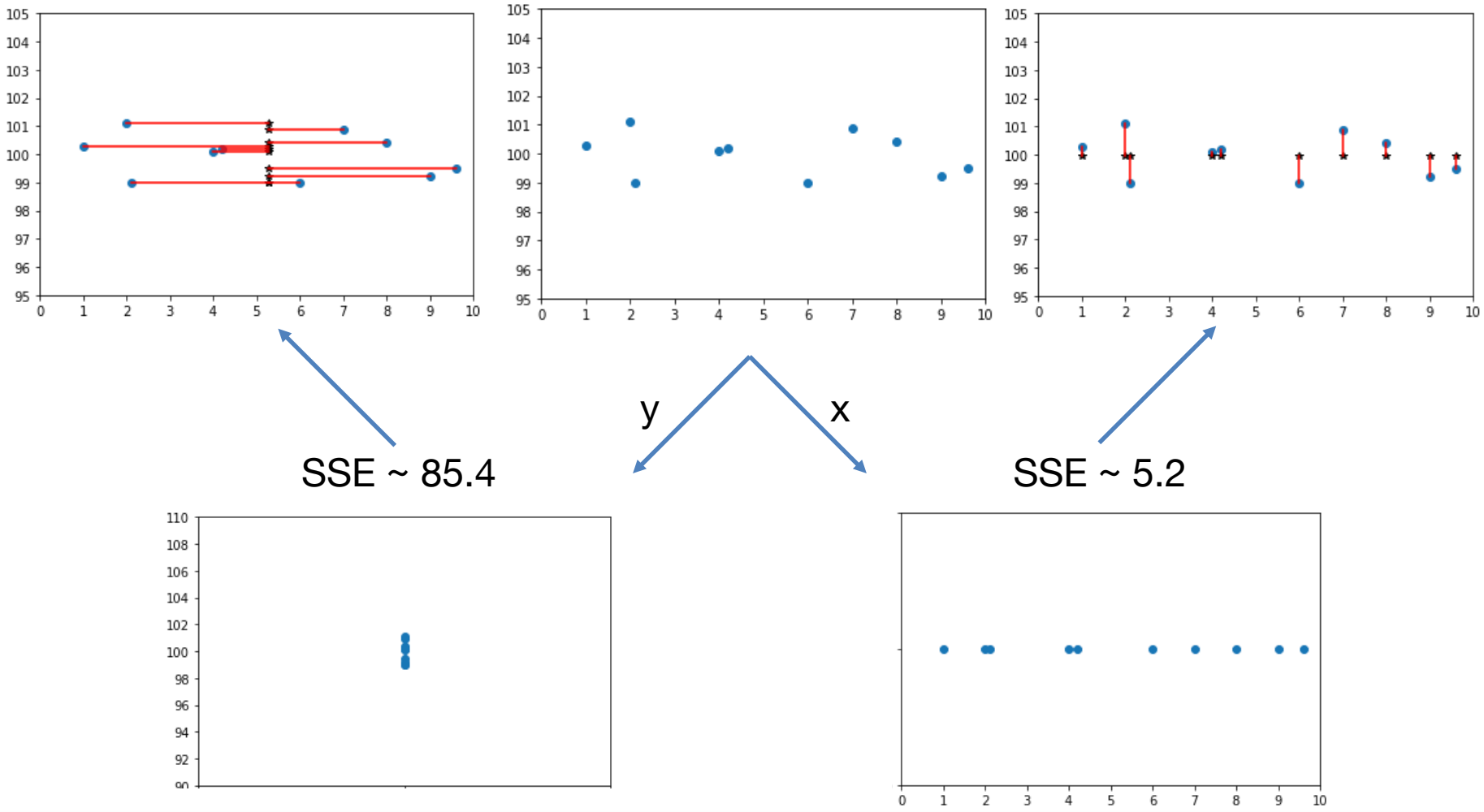
Sample Variance ~ 0.6



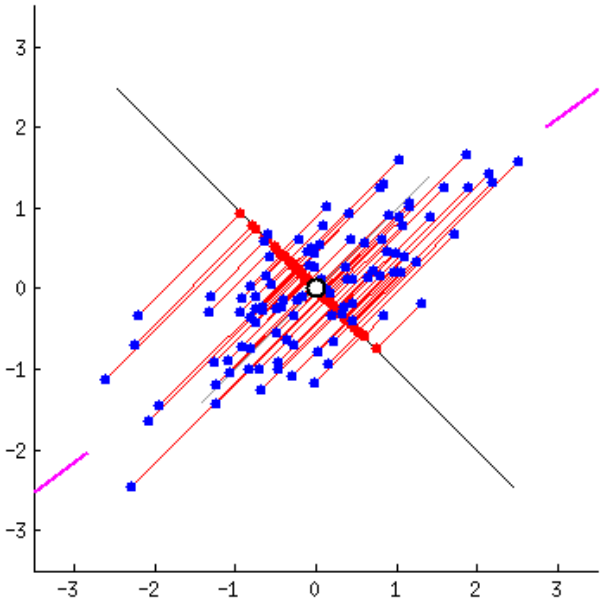
Sample Variance ~ 9.5



Intuition: Reconstruction Error



Minimization in Action



Intuition: Equivalent Objectives

Variance of Data **(Fixed)**

=

Captured Variance (Want Large!)

+

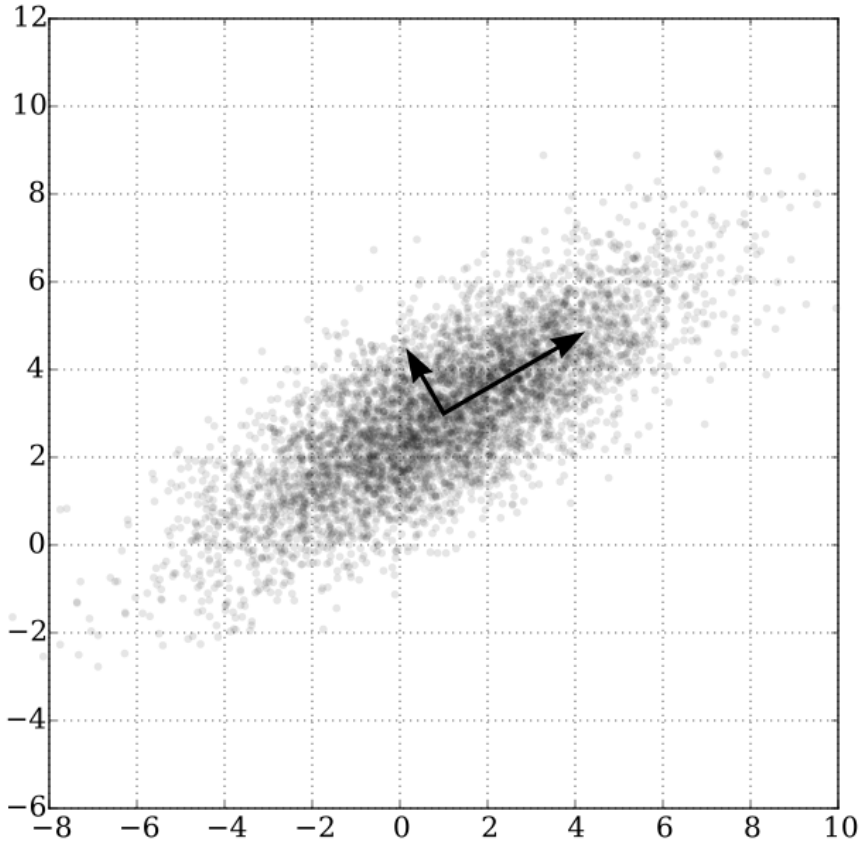
Reconstruction Error (Want Small!)

PCA: Big Picture Idea

- Find an axis (i.e. basis vector) that explains the most variance in the data
- Now find a second axis that is orthogonal to the first, and, given this constraint, explains the most variance
- Now find a third axis…
- Keep those that explain *sufficient* variation

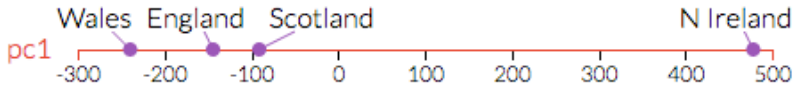
Example 2D

Any Information Lost?



Example 17D (setosa.io)

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175



Deriving PCA via Maximizing Variance

Big picture…

1. Express variance of projected data
2. Maximize, solving for projection vector

P.S. Expect abused notation :)

Some Definitions

- Dataset: $\mathbf{X} = \{ \mathbf{x}_n \}$
  - n = 1 … N
  - Euclidean, dimensionality D

- Goal: find axis of most variance
  - Direction: **u**
    - Euclidean, dimensionality D
    - Assume unit length: $\mathbf{u}^T\mathbf{u} = 1$
  - (More generally $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \ldots]$)

Variance of Projected Data

$$\begin{aligned}
\text{Var}[\,\mathbf{Xu}\,] &= E[\,(\,\mathbf{Xu} - E[\,\mathbf{Xu}\,]\,)^2\,] \\
&= E[\,(\,\mathbf{Xu} - E[\mathbf{X}]\mathbf{u}\,)^2\,] \\
&= E[\,(\,(\,\mathbf{X} - E[\mathbf{X}]\,)\mathbf{u}\,)^2\,] \\
&= \mathbf{u}\,E[\,(\,\mathbf{X} - E[\mathbf{X}]\,)^2\,]\,\mathbf{u} \\
&= \mathbf{u}\,E[\,(\,\mathbf{X} - E[\mathbf{X}]\,)(\,\mathbf{X} - E[\mathbf{X}]\,)^T\,]\,\mathbf{u} \\
&= \mathbf{u}^T\mathbf{S}\,\mathbf{u}
\end{aligned}$$

Can we maximize directly?

Look familiar? Covariance!

Maximize, with $\mathbf{u}^\intercal\mathbf{u}=1$

$$\frac{\partial}{\partial u} u^\intercal S u + \lambda(1 - u^\intercal u) = 0$$

$$(S + S^\intercal)u - 2\lambda u = 0$$

$$Su = \lambda u$$

Symmetric!

Familiar?
So u=eigenvector with largest λ

So…

- The projection vector that captures maximum variance is the eigenvector of the covariance matrix that has the largest eigenvalue
  - This is the first *principle component* (PC1)

- Because the covariance matrix is symmetric, all of the eigenvectors are pairwise orthogonal
  - Prove this in HW!

Applying PCA

1. Transform data
   - For each dimension: mean=0, variance=1
   - $x' = (x - \mu) / \sigma$
2. Compute eigendecomposition of the covariance matrix
3. Select *k* principal components
4. Project data (Xu)

Computing the Eigendecomposition

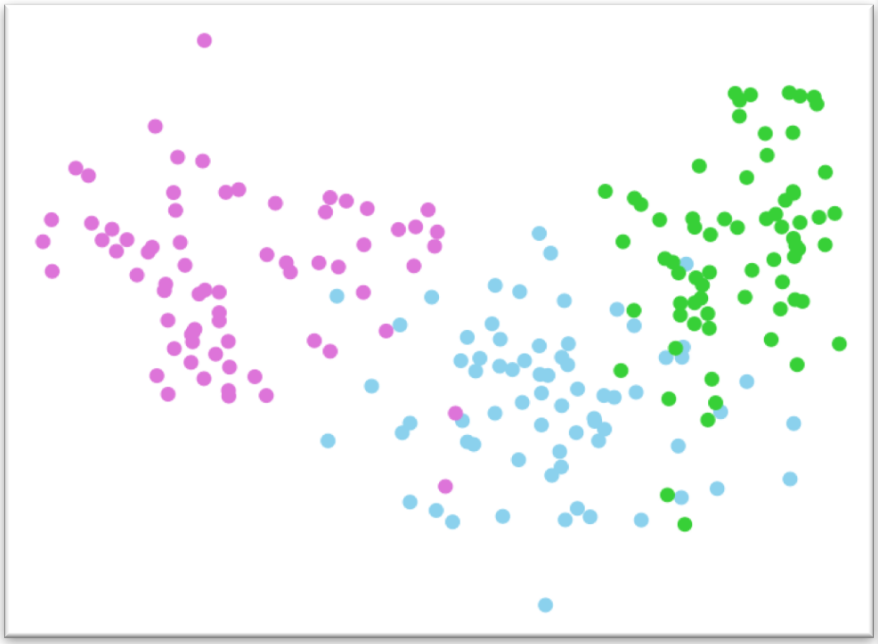
- Values
  - Symbolically via characteristic equation
  - Power iteration
- Vectors
  - System of linear equations w.r.t. characteristic equation or def'n of eigendecomposition
- See LRU for examples
  - Practice in HW!
  - Note issues related to speed vs accuracy

Explaining Variance

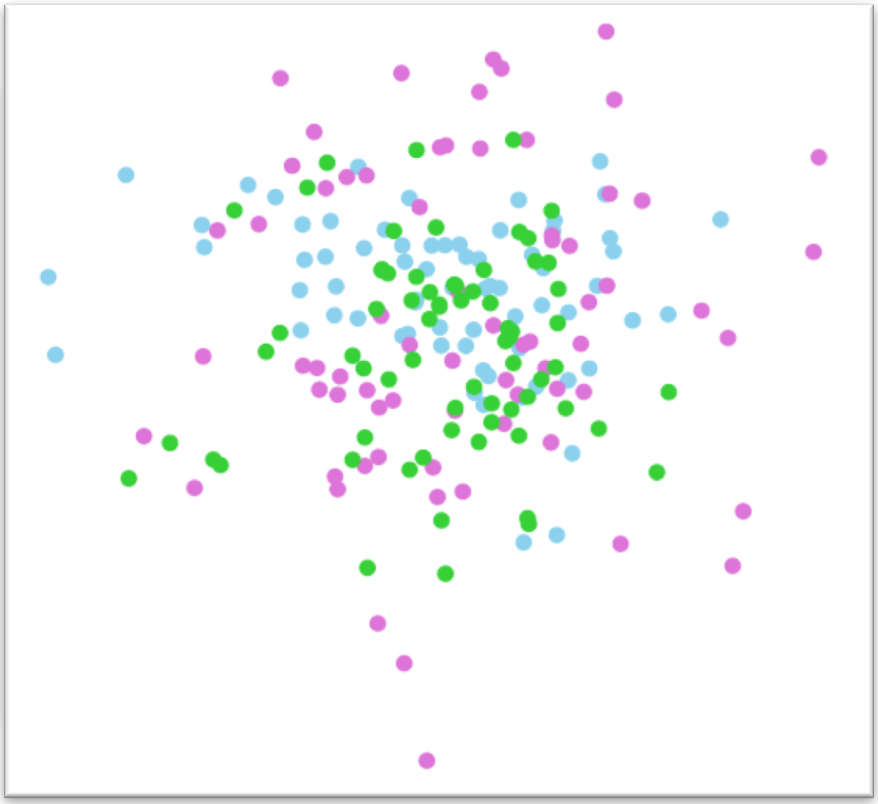
- S is positive semi-definite, so eigenvalues are non-negative
- Eigenvalues represent proportion of variance explained (0 = ignore!)
- SO, sort descending, apply threshold for cumulative proportion of eigenvalues
  - Often 90%
  - Or elbow, constant (e.g. visualization)

Wheat Data

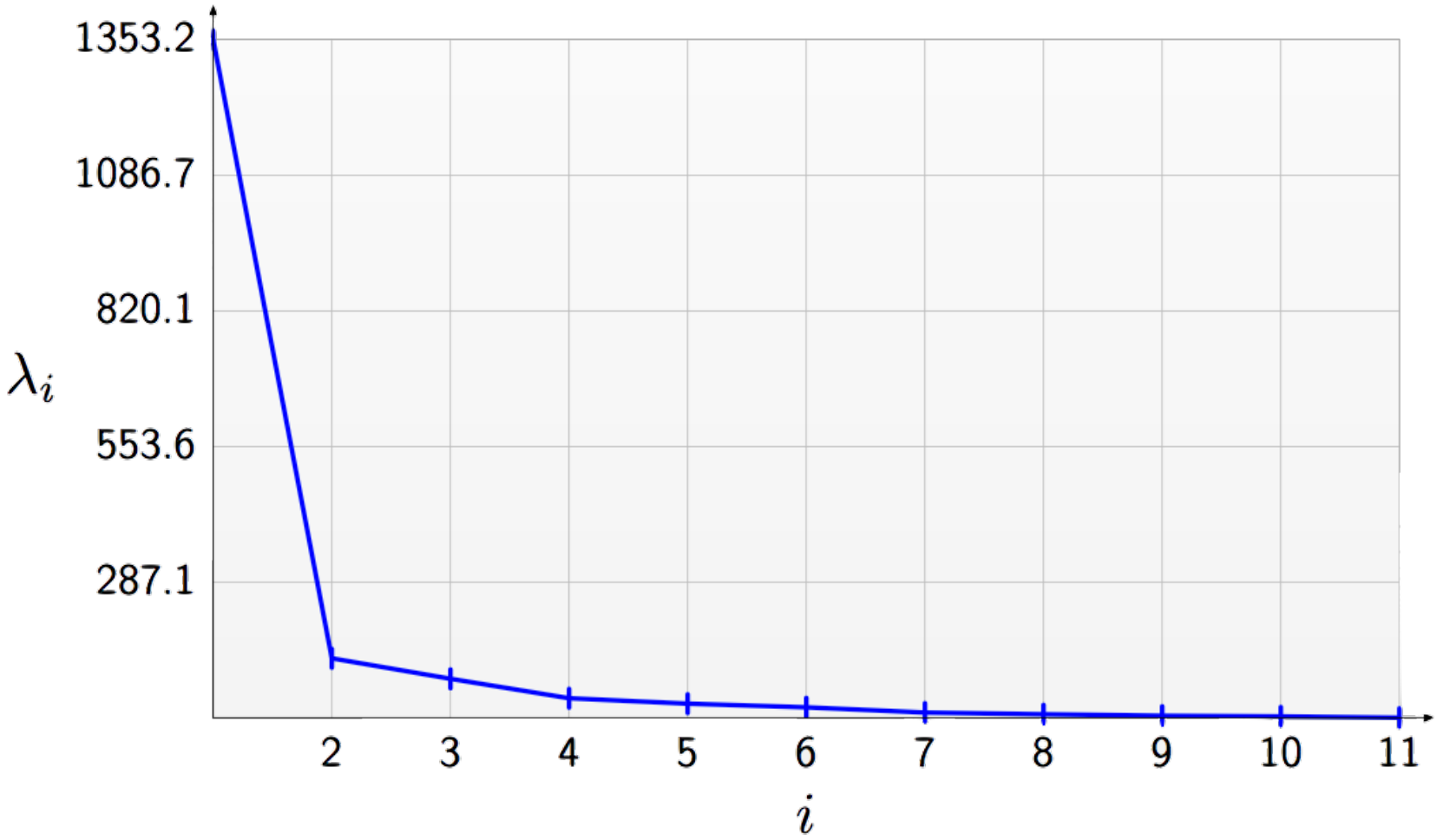
Top 2 Components



Bottom 2 Components



Face Image Dataset



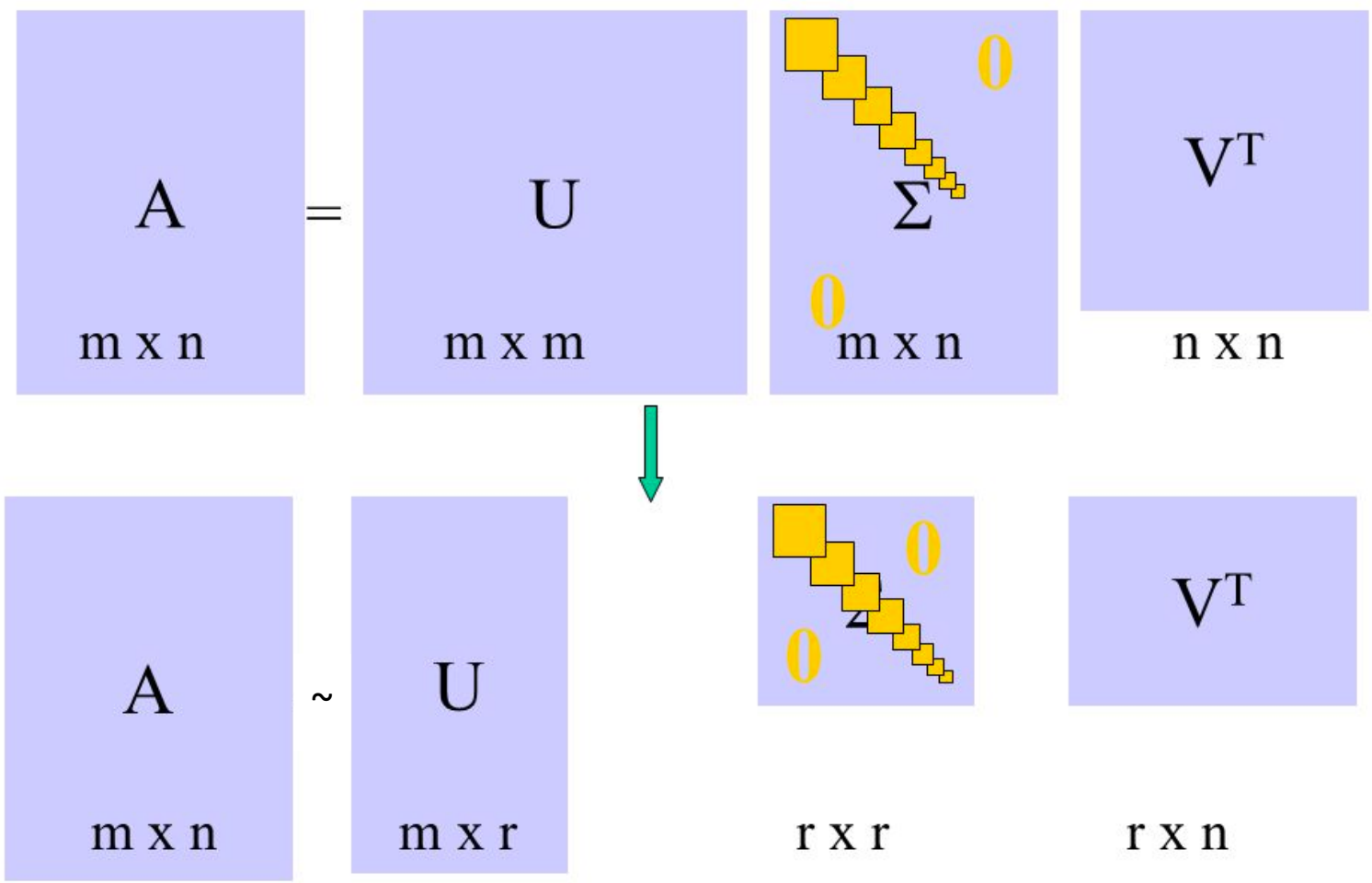
Singular Value Decomposition (SVD)

- $X = USV^T$
  - $X^TX = VS^TU^TUSVT = VSISV^T = VSSV^T = VDV^T$

- A more general decomposition
  - Basis vectors for rows and columns of data
    - vs PCA: only rows
  - Many other uses (e.g. Latent Semantic Analysis)

- SVD provides the same decomposition when mean of the data is 0
  - (S)ingular values = sqrt(eigenvalues)
  - Project via XV^k
  - Practice in HW!

SVD



PCA vs SVD

- Can calculate SVD without X^TX
  - Saves computation
  - Prevents loss of precision

- Actual performance depends upon N vs D
  - Eigen: $O(nd^2 + d^3)$
  - SVD: $O(\min\{nd^2, n^2d\})$
  - But SVD tends to be the default

Core Assumptions

- Linearity
  - Can be addressed via “kernel” trick
- Focus on orthogonality, variance
  - Other approaches exist, may be more appropriate

Independent Components Analysis (ICA)

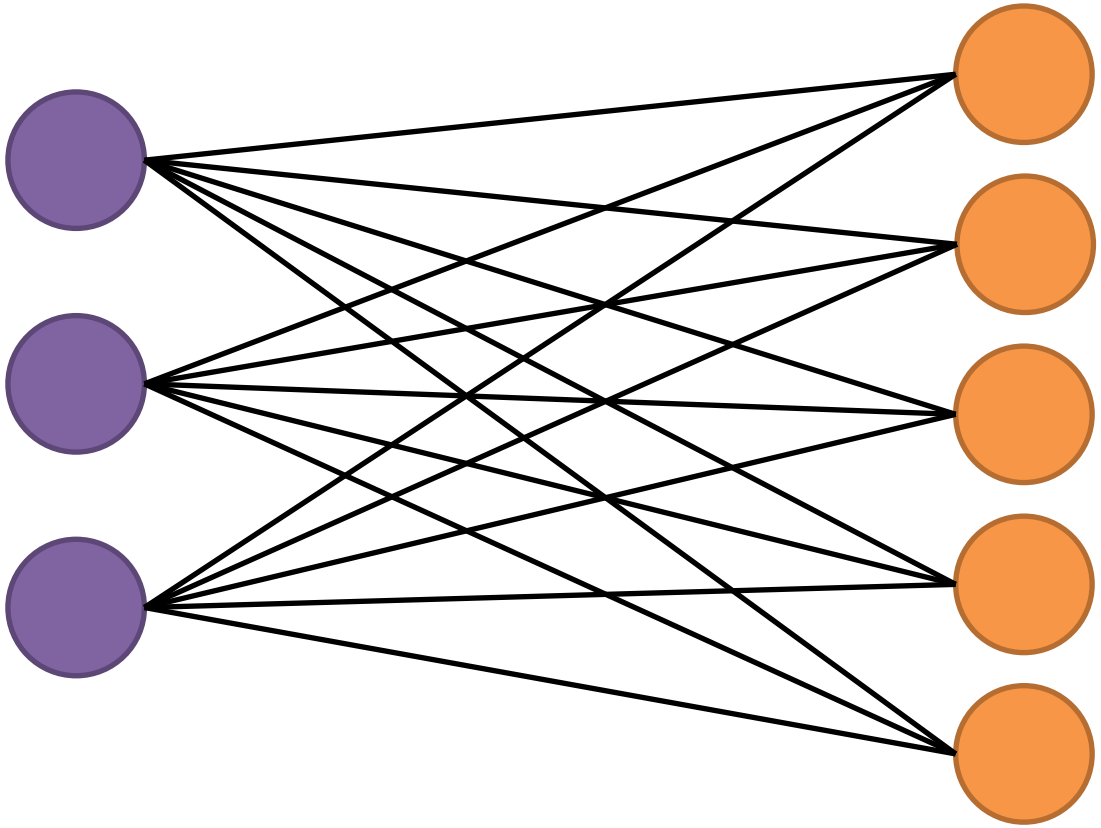
- Whereas PCA seeks correlation via variance, ICA seeks statistically independent components

- Like PCA, achieved via linear transformation

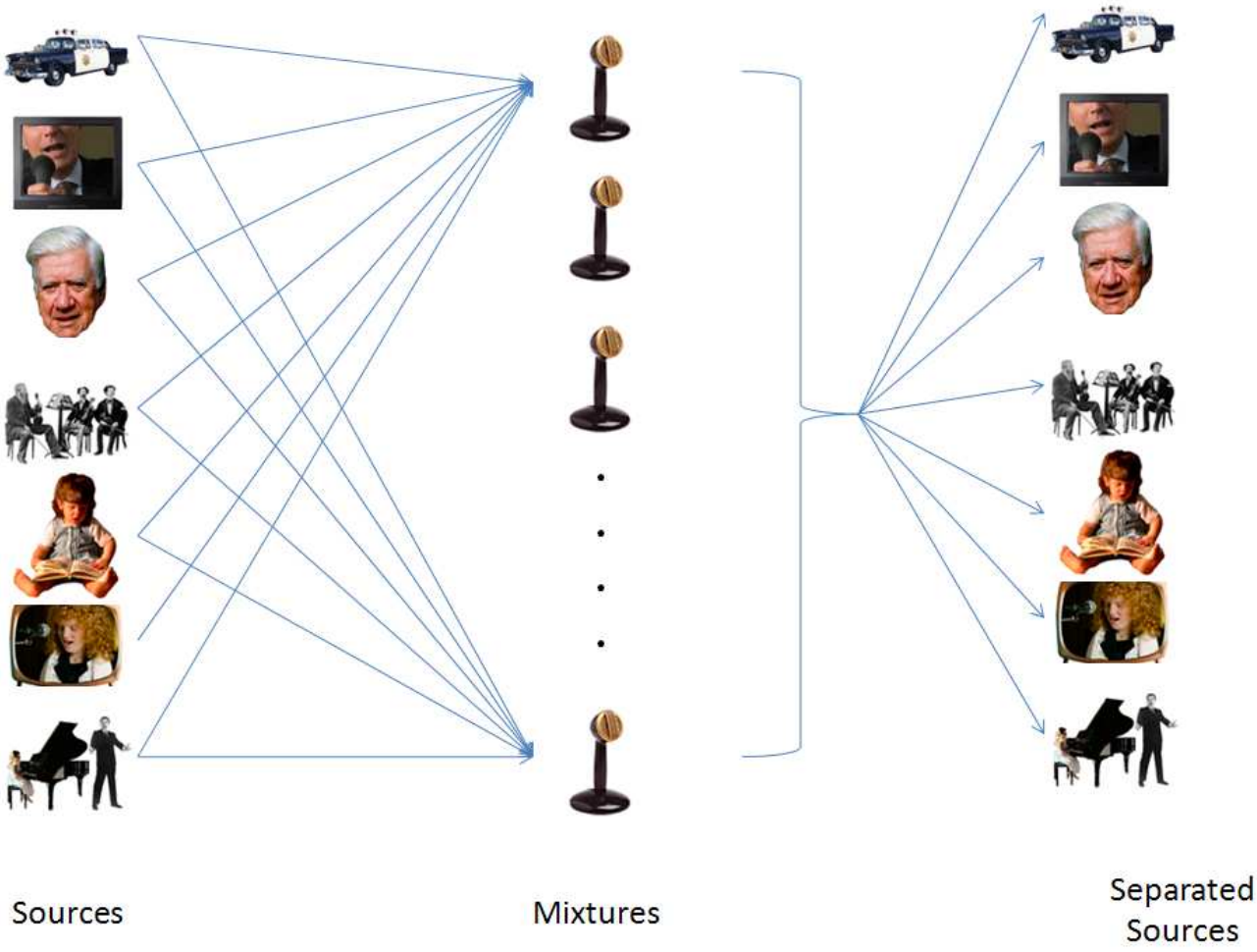
ICA Assumption

Find Hidden via Observable

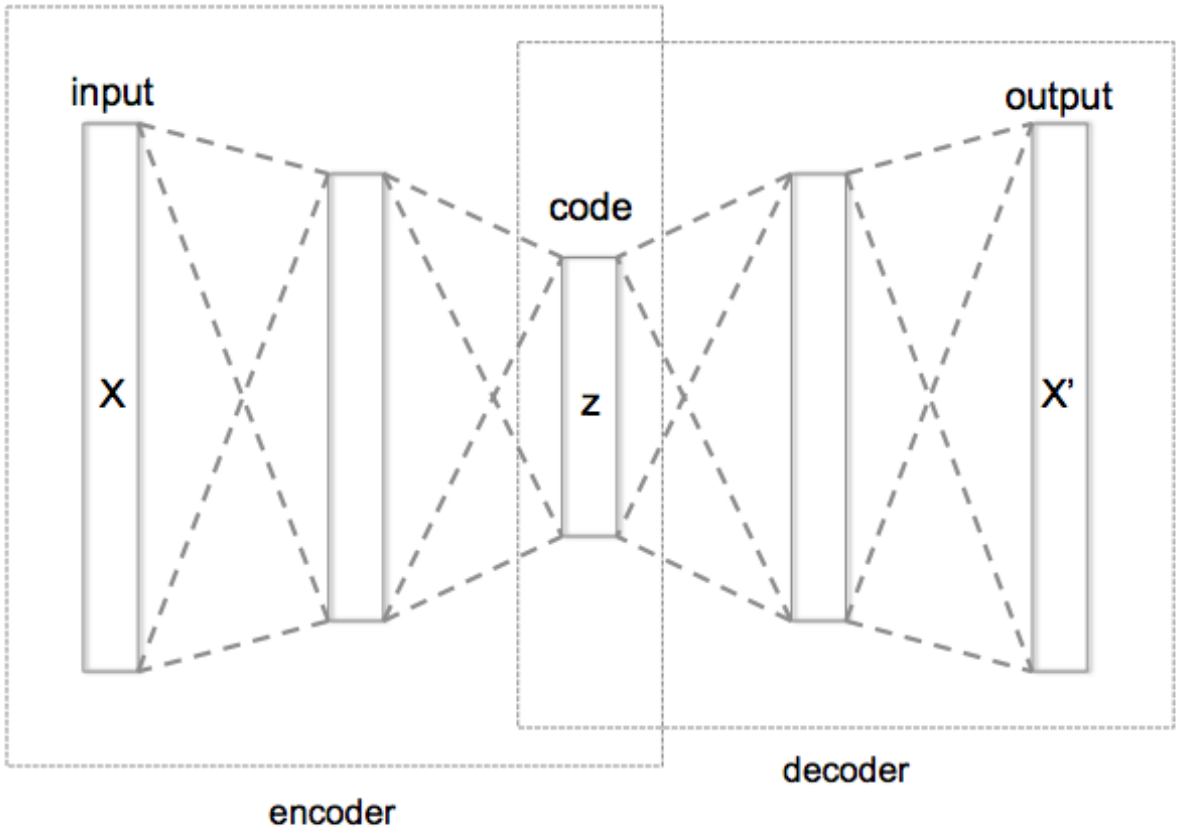
Hidden Ind. Variables **Observable Variables**



ICA Example: Cocktail Party



Autoencoders



Autoencoders

- Also known as an autoassociative neural network
- Architecture
  - Same number of inputs/outputs
  - Fewer "code" nodes than i/o
- Goal: reproduce output
- If only one hidden layer (i.e. code), same as PCA
  - But if more – performs nonlinear PCA :)

More Than You Wanted to Know

Journal of Machine Learning Research 16 (2015) 2859-2900 Submitted 5/14; Revised 3/15; Published 12/15

Linear Dimensionality Reduction:
Survey, Insights, and Generalizations

John P. Cunningham JPC2181@COLUMBIA.EDU
Department of Statistics
Columbia University
New York City, USA

Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
Department of Engineering
University of Cambridge
Cambridge, UK

Editor: Gert Lanckriet

Abstract

Linear dimensionality reduction methods are a cornerstone of analyzing high dimensional data, due to their simple geometric interpretations and typically attractive computational properties. These methods capture many data features of interest, such as covariance, dynamical structure, correlation between data sets, input-output relationships, and margin between data classes. Methods have been developed with a variety of names and motivations in many fields, and perhaps as a result the connections between all these methods have not been highlighted. Here we survey methods from this disparate literature as optimization programs over matrix manifolds. We discuss principal component analysis, factor analysis, linear multidimensional scaling, Fisher's linear discriminant analysis, canonical correlations analysis, maximum autocorrelation factors, slow feature analysis, sufficient dimensionality reduction, undercomplete independent component analysis, linear regression, distance metric learning, and more. This optimization framework gives insight to some rarely discussed shortcomings of well-known methods, such as the suboptimality of certain eigenvector solutions. Modern techniques for optimization over matrix manifolds enable a generic linear dimensionality reduction solver, which accepts as input data and an objective to be optimized, and returns, as output, an optimal low-dimensional projection of the data. This simple optimization framework further allows straightforward generalizations and novel variants of classical methods, which we demonstrate here by creating an orthogonal-projection canonical correlations analysis. More broadly, this survey and generic solver suggest that linear dimensionality reduction can move toward becoming a blackbox, objective-agnostic numerical technology.

Keywords: dimensionality reduction, eigenvector problems, matrix manifolds

1. Introduction

Linear dimensionality reduction methods have been developed throughout statistics, machine learning, and applied fields for over a century, and these methods have become indispensable tools for analyzing high dimensional, noisy data. These methods produce a