ITCS 6156/8156 Fall 2023
Machine Learning

Principal Components Analysis & Autoencoders

Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Some content in the slides is based on Dr. Raquel Urtasun’s lecture

Example

- What are the intrinsic latent dimensions in these two datasets?

- How can we find these dimensions from the data?

PCA Toy Example

Consider the following 3D points

1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

If each component is stored in a byte,
we need 18 = 3 x 6 bytes

The example is from Dr. Octavia Camps' slides

PCA Toy Example

Looking closer, we can see that all the points are related geometrically: they are all the same point, scaled by a factor:

$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} = 2 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = 3 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

PCA Toy Example

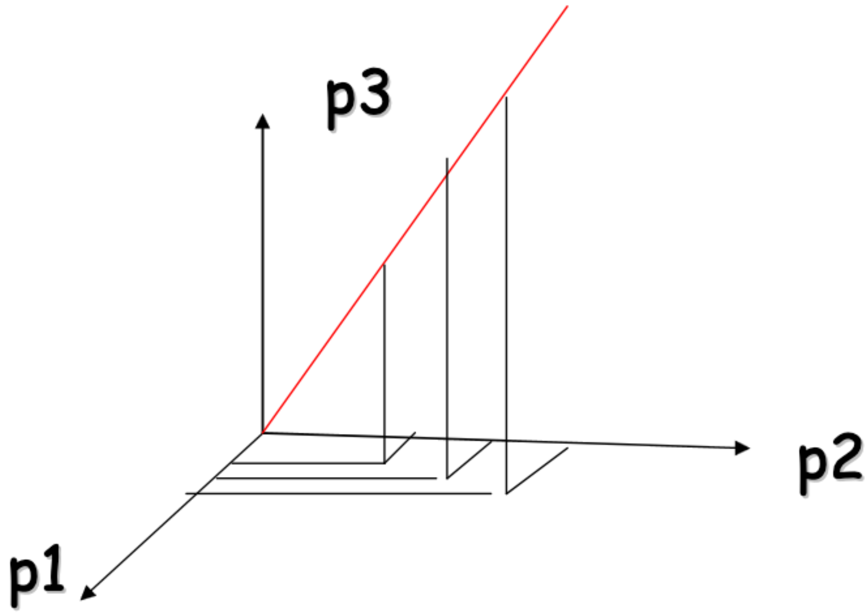
$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} = 2 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = 3 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

They can be stored using only 9 bytes (50% savings!):
Store one point (3 bytes) + the multiplying constants (6 bytes)

Geometrical Interpretation

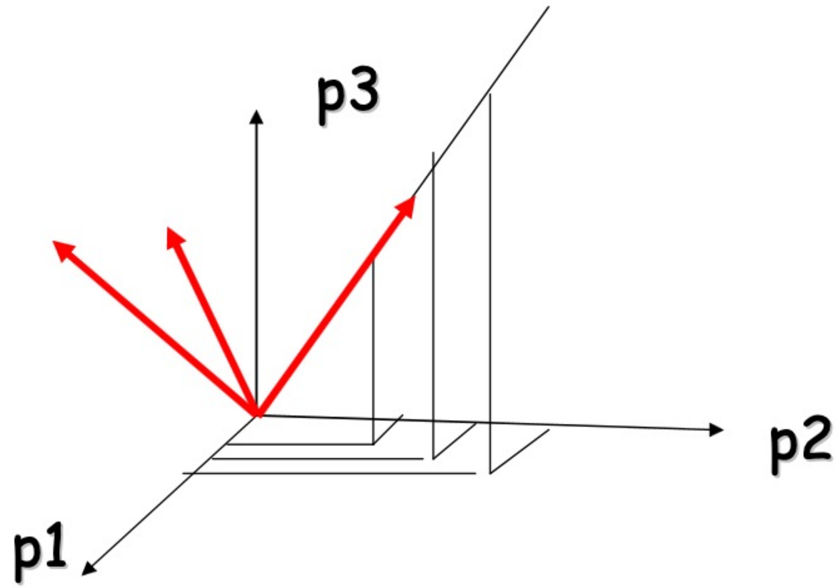
View each point in 3D space.



But in this example, all the points happen to belong to a line: a 1D subspace of the original 3D space.

Geometrical Interpretation

Consider a new coordinate system where one of the axes is along the direction of the line:



In this coordinate system, every point has <u>only one</u> non-zero coordinate: we <u>*only*</u> need to store the direction of the line (a 3 bytes image) and the non-zero coordinate for each of the points (6 bytes).

Principal Components Analysis

- Given a set of points, how do we know if they can be compressed like in the previous example?
  - The answer is to look into the correlation between the points
  - The tool for doing this is called PCA

Principal Components Analysis

- An exploratory technique used to reduce the dimensionality of the data set to 2D or 3D
- **Aim**: find a small number of "directions" in input space that explain variation in input data; re-represent data by projecting along those directions
- **Important assumption:** variation contains information
- Can be used to:
  - Reduce number of dimensions in data
  - Find patterns in high-dimensional data
  - Visualize data of high dimensionality
- Example applications:
  - Face recognition
  - Image compression
  - Gene expression analysis

Covariance

- Variance and Covariance are a measure of the “spread” of a set of points around their center of mass (mean)
- Variance – measure of the deviation from the mean for points in one dimension e.g. heights
- Covariance as a measure of how much each of the dimensions vary from the mean with respect to each other.
- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions e.g. number of hours studied & marks obtained.
- The covariance between one dimension and itself is the variance

Covariance

$$\text{covariance}(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

- So, if you had a 3-dimensional data set (x,y,z), then you could measure the covariance between the x and y dimensions, the y and z dimensions, and the x and z dimensions. Measuring the covariance between x and x , or y and y , or z and z would give you the variance of the x , y and z dimensions respectively.

Covariance Matrix

- Representing Covariance between dimensions as a matrix e.g. for 3 dimensions:

$$C = \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix}$$

Variances

- Diagonal is the variances of x, y and z
- cov(x,y) = cov(y,x) hence matrix is symmetrical about the diagonal
- N-dimensional data will result in NxN covariance matrix

Covariance

- What is the interpretation of covariance calculations?

  e.g.: 2 dimensional data set

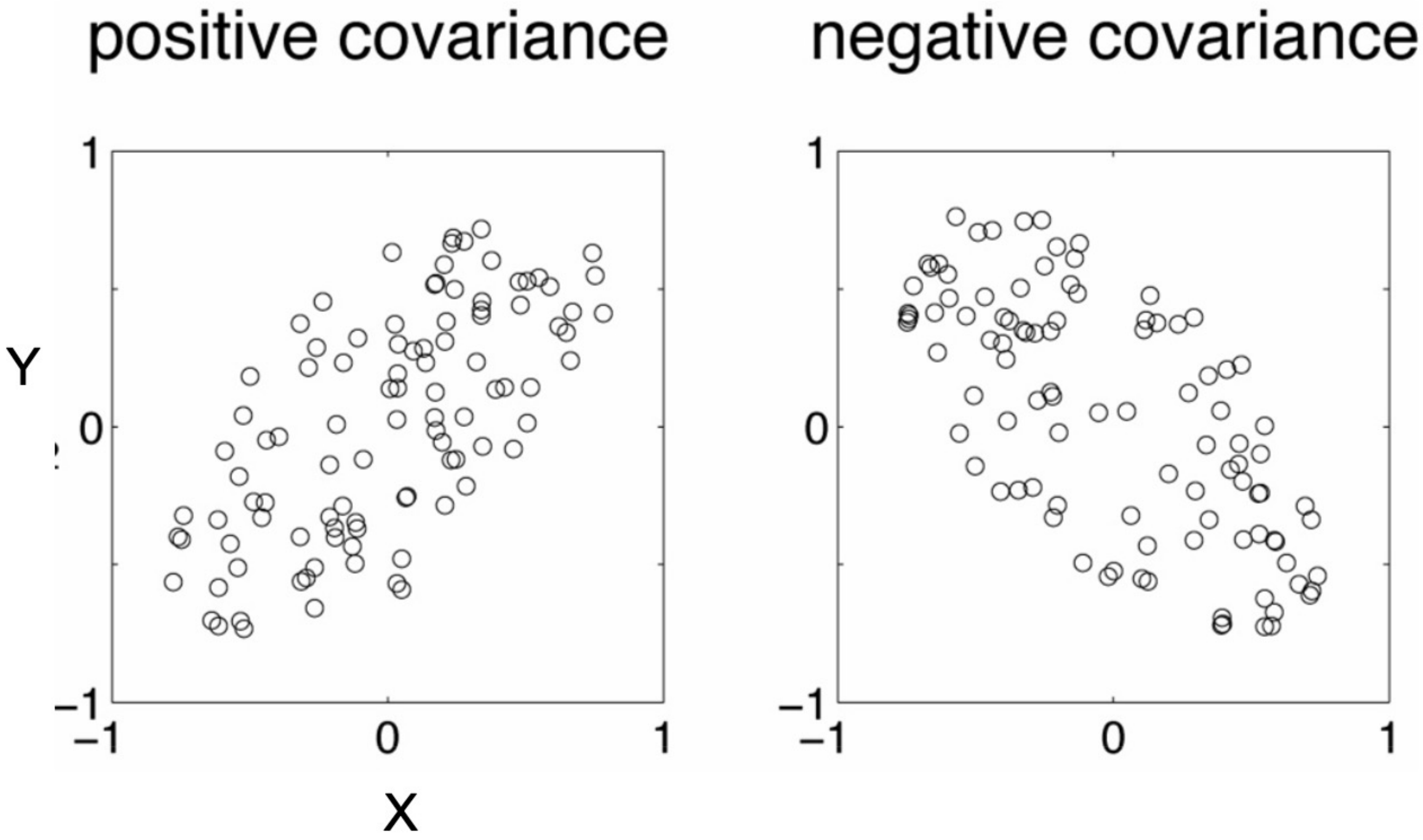
  x: number of hours studied for a subject

  y: marks obtained in that subject

  covariance value is say: 104.53

  what does this value mean?

Covariance



Covariance

- Exact value is not as important as it's sign.
- A <u>positive value</u> of covariance indicates both dimensions increase or decrease together e.g. as the number of hours studied increases, the marks in that subject increase.
- A <u>negative value</u> indicates while one increases the other decreases, or vice-versa e.g. active social life at UNCC vs performance in CS dept.
- If <u>covariance is zero</u>: the two dimensions are independent of each other e.g. heights of students vs the marks obtained in a subject

Covariance

- Why bother with calculating covariance when we could just plot the 2 values to see their relationship?

  Covariance calculations are used to find relationships between dimensions in high dimensional data sets (usually greater than 3) where visualization is difficult.

Eigenvalues & eigenvectors

- $A\mathbf{x}=\lambda\mathbf{x} \Leftrightarrow (A-\lambda \mathrm{I})\mathbf{x}=0$

- How to calculate **x** and λ:
  - Calculate $det(A-\lambda I)$, yields a polynomial (degree n)
  - Determine roots to $det(A-\lambda I)=0$, roots are eigenvalues λ
  - Solve $(A- \lambda I)\ \mathbf{x}=0$ for each λ to obtain eigenvectors **x**

Eigenvalues & eigenvectors

- Vectors **x** having same direction as *A***x** are called *eigenvectors* of *A* (*A* is an n by n matrix).
- In the equation *A***x**=λ**x**, λ is called an *eigenvalue* of *A*.

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} x \begin{pmatrix} 3 \\ 2 \end{pmatrix} \square \begin{pmatrix} 12 \\ 8 \end{pmatrix} \square 4 x \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

PCA

- By finding the eigenvalues and eigenvectors of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.

- This is the principal component.

Principle of Maximal Variance

- Least loss of information
- Best capture the "spread"
- What is the direction of maximal variance?
- Given any direction ($\hat{\mathbf{u}}$), the projection of $\mathbf{x}$ on $\hat{\mathbf{u}}$ is given by:

$$\mathbf{x}_i^\top \hat{\mathbf{u}}$$

Centered data

- Direction of maximal variance can be obtained by maximizing

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^\top \hat{\mathbf{u}})^2 &= \frac{1}{N}\sum_{i=1}^{N}\hat{\mathbf{u}}^\top \mathbf{x}_i \mathbf{x}_i^\top \hat{\mathbf{u}} \\ &= \hat{\mathbf{u}}^\top \left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \mathbf{x}_i^\top\right)\hat{\mathbf{u}} \end{aligned}$$

Finding Direction of Maximal Variance

Find:

$$\underset{\hat{\mathbf{u}}:\hat{\mathbf{u}}^\top\hat{\mathbf{u}}=1}{max}\ \hat{\mathbf{u}}^\top\mathbf{S}\hat{\mathbf{u}}$$

where:

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^\top$$

- $\mathbf{S}$ is the sample (empirical) covariance matrix of the mean-centered data

Solution

The solution to the above constrained optimization problem may be obtained using the Lagrange multipliers method. We maximize the following w.r.t. $\hat{\mathbf{u}}$:

$$(\hat{\mathbf{u}}^\top \mathbf{S}\hat{\mathbf{u}}) - \lambda(\hat{\mathbf{u}}^\top \hat{\mathbf{u}} - 1)$$

to get:

$$\begin{aligned} \frac{d}{d\hat{\mathbf{u}}}(\hat{\mathbf{u}}^\top \mathbf{S}\hat{\mathbf{u}}) - \lambda(\hat{\mathbf{u}}^\top \hat{\mathbf{u}} - 1) &= 0 \\ \mathbf{S}\hat{\mathbf{u}} - \lambda\hat{\mathbf{u}} &= 0 \\ \mathbf{S}\hat{\mathbf{u}} &= \lambda\hat{\mathbf{u}} \end{aligned}$$

Obviously, the solution to the above equation is an eigen vector of the matrix $\mathbf{S}$. But which $\mathbf{S}$? Note that for the optimal solution:

$$\hat{\mathbf{u}}^\top \mathbf{S}\hat{\mathbf{u}} = (\hat{\mathbf{u}}^\top \lambda\hat{\mathbf{u}}) = \lambda$$

Thus we should choose the largest possible λ which means that the first solution is the eigen vector of $\mathbf{S}$ with largest eigen value.

PCA Algorithm

1. *Center* $\mathbf{X}$

$$\mathbf{X} = \mathbf{X} - \hat{\boldsymbol{\mu}}$$

2. Compute sample covariance matrix:

$$\mathbf{S} = \frac{1}{N-1}\mathbf{X}^\top\mathbf{X}$$

3. Find eigen vectors and eigen values for $\mathbf{S}$

4. $\mathbf{W}$ consists of first L eigen vectors as columns

    - Ordered by decreasing eigen-values
    - $\mathbf{W}$ is $D \times L$

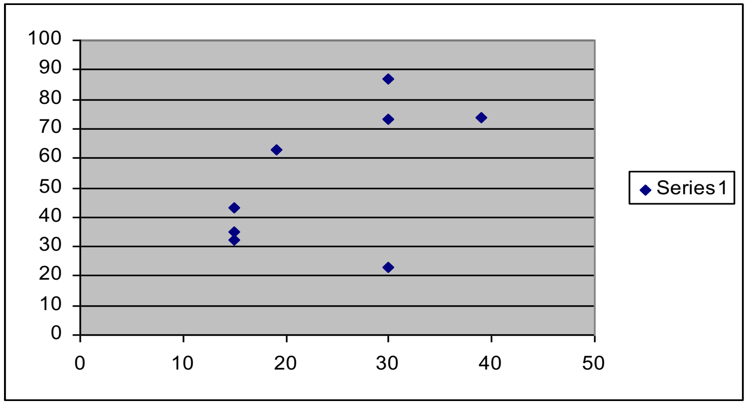
5. Let $\mathbf{Z} = \mathbf{X}\mathbf{W}$

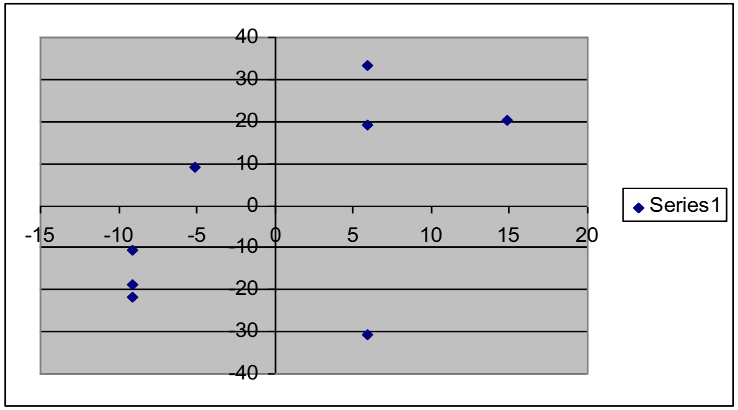
6. Each row in $\mathbf{Z}$ (or $\mathbf{z}_i^\top$) is the lower dimensional embedding of $\mathbf{x}_i$

An Example

Mean1=24.1
Mean2=53.8

X1	X2	X1'	X2'
19	63	-5.1	9.25
39	74	14.9	20.25
30	87	5.9	33.25
30	23	5.9	-30.75
15	35	-9.1	-18.75
15	43	-9.1	-10.75
15	32	-9.1	-21.75
30	73	5.9	19.25





An Example

- C=

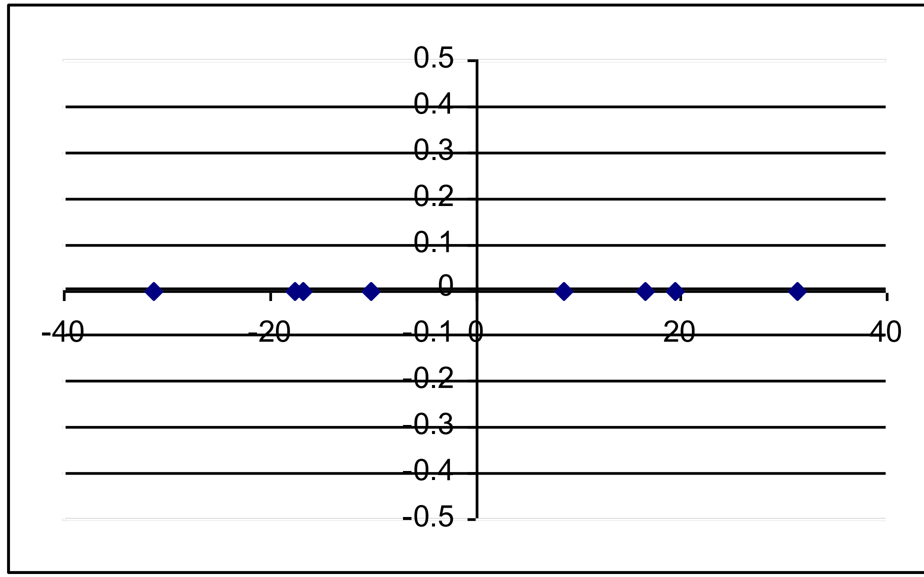
75	106
106	482

- Using MATLAB, we find out:
  - Eigenvectors:
  - e1=(-0.98,-0.21), λ1=51.8
  - e2=(0.21,-0.98), λ2=560.2
  - Thus the second eigenvector is more important!

An Example

- If we only keep one dimension: e2

- We keep the dimension of e2=(0.21,-0.98)
- We can obtain the final data as

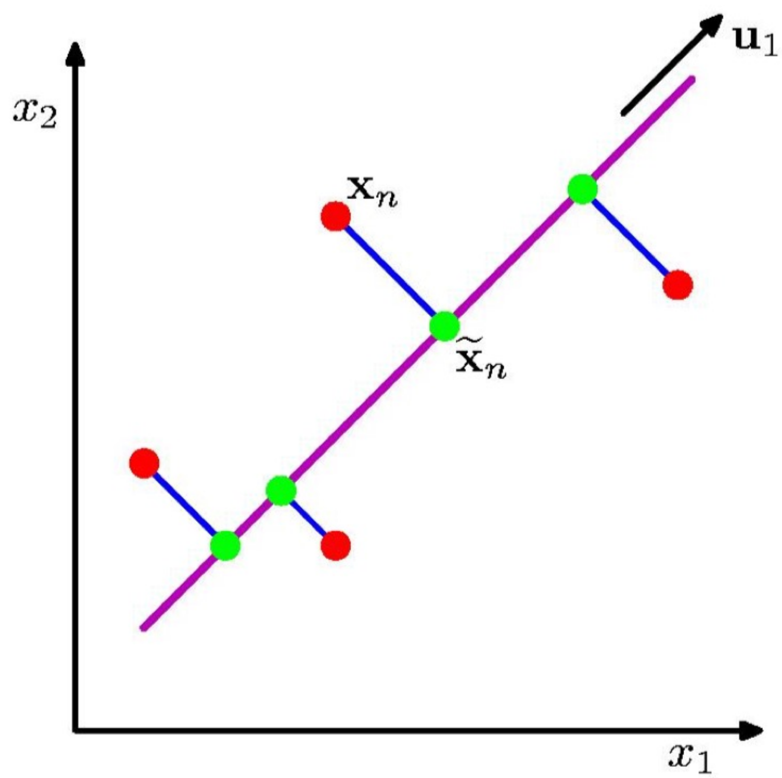


yi
-10.14
-16.72
-31.35
31.374
16.464
8.624
19.404
-17.63

$$y_i = \begin{pmatrix}0.21 & -0.98\end{pmatrix}\begin{pmatrix}x_{i1} \\ x_{i2}\end{pmatrix} = 0.21 * x_{i1} - 0.98 * x_{i2}$$

Two Derivations of PCA

- Two views/derivations:
  - Maximize variance (scatter of green points)
  - Minimize error (red-green distance per datapoint)



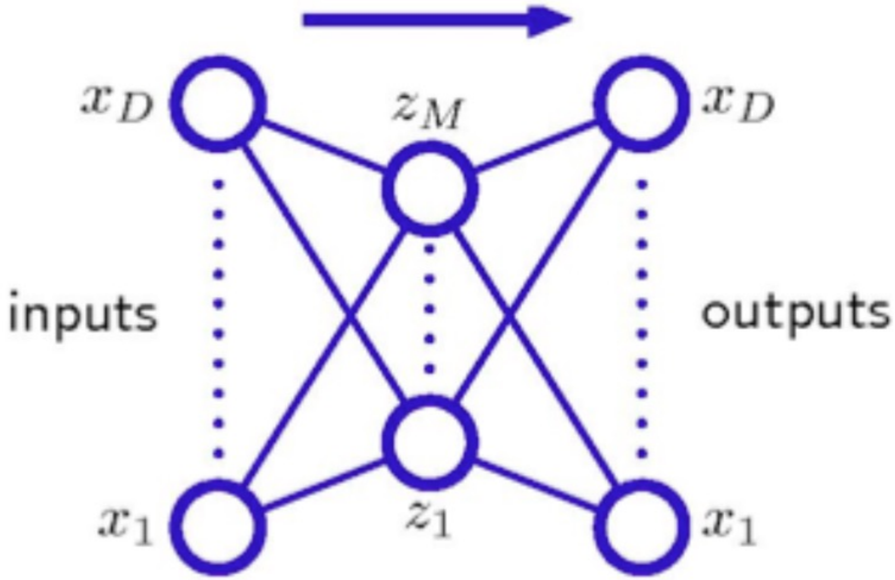
Applying PCA to faces

- Run PCA on 2429 19x19 grayscale images (CBCL data)
- Compresses the data: can get good reconstructions with only 3 components

- PCA for pre-processing: can apply classifier to latent representation
  - PCA with 3 components obtains 79% accuracy on face/non-face discrimination on test data vs. 76.8% for GMM with 84 states
- Can also be good for visualization

Relation to Neural Networks

- PCA is closely related to a particular form of neural network
- An autoencoder is a neural network whose outputs are its own inputs



- The goal is to minimize reconstruction error

Autoencoders

- Define

$$\mathbf{z} = f(W\mathbf{x}); \quad \hat{\mathbf{x}} = g(V\mathbf{z})$$

- Goal:

$$\min_{\mathbf{W},\mathbf{V}} \frac{1}{2N} \sum_{n=1}^{N} ||\mathbf{x}^{(n)} - \hat{\mathbf{x}}^{(n)}||^2$$

- If g and f are linear

$$\min_{\mathbf{W},\mathbf{V}} \frac{1}{2N} \sum_{n=1}^{N} ||\mathbf{x}^{(n)} - VW\mathbf{x}^{(n)}||^2$$

- In other words, the optimal solution is PCA.

Autoencoders: Nonlinear PCA

- What if $g()$ is not linear?
- Then we are basically doing nonlinear PCA
- Some subtleties but in general this is an accurate description

Comparing Reconstructions

Questions?

UNIVERSITY OF NORTH CAROLINA CHARLOTTE