Artificial Intelligence II (CS4442 & CS9542)

Unsupervised Learning: Dimensionality Reduction

Boyu Wang
Department of Computer Science
University of Western Ontario

Unsupervised Learning

Unsupervised learning

- In supervised learning, data is in the form of pairs (x, y), and the goal is to learn a model h such that $h(x)$ can approximate y well.
- In unsupervised learning, the data just contains x!
- The goal of unsupervised learning is to summarize or discover patterns or structure in the data

Unsupervised learning

- In supervised learning, data is in the form of pairs (x, y), and the goal is to learn a model h such that $h(x)$ can approximate y well.
- In unsupervised learning, the data just contains x!
- The goal of unsupervised learning is to summarize or discover patterns or structure in the data
- A variety of problems and uses:
  1. Dimensionality reduction: compression, visualization, feature extraction
  2. Clustering: discover the group structure of data, divide the data into different regions
  3. Density estimation: outlier detection, change point detection

Unsupervised learning

- In supervised learning, data is in the form of pairs (x, y), and the goal is to learn a model h such that $h(x)$ can approximate y well.
- In unsupervised learning, the data just contains x!
- The goal of unsupervised learning is to summarize or discover patterns or structure in the data
- A variety of problems and uses:
  1. Dimensionality reduction: compression, visualization, feature extraction
  2. Clustering: discover the group structure of data, divide the data into different regions
  3. Density estimation: outlier detection, change point detection
- The definition of “ground truth” is often not clear: we don’t have the label y
- Can also be used as a pre-processing step for supervised learning

Dimensionality Reduction

High-dimensional data

- High-Dimensions = Lot of Features

  Document classification

  Features per document =

  thousands of words/unigrams

  millions of bigrams, contextual information

  Surveys - Netflix

  480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

Figure credit: Maria-Florina Balcan

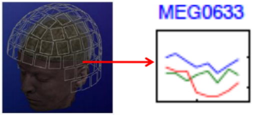
High-dimensional data

- High-Dimensions = Lot of Features

MEG Brain Imaging

120 locations x 500 time points

x 20 objects



Or any high-dimensional image data

Figure credit: Maria-Florina Balcan

What is dimensionality reduction?

Take data in a high dimensional space and map it into a new space whose dimensionality is much smaller (even 2D or 3D for visualization).

What is dimensionality reduction?

Take data in a high dimensional space and map it into a new space whose dimensionality is much smaller (even 2D or 3D for visualization).

- Principles to follow

  1. Do not loss too much information
  2. Approximately preserve similarity/distance relationships between instances.

- Motivations

  1. Computational: compress the data as a pre-processing step to speedup subsequent operations on the data
  2. Visualization: visualize the data for exploratory analysis by mapping the input data into 2D or 3D spaces
  3. Feature extraction: to generate a smaller and more effective or useful set of features.

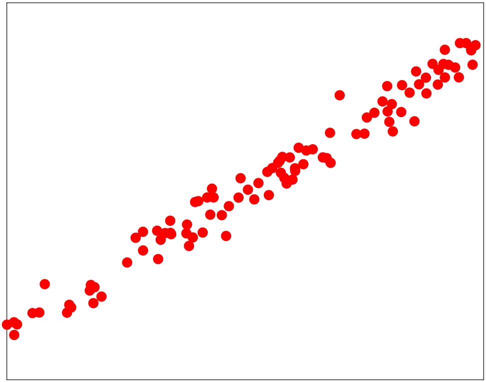
Dimensionality reduction techniques

- Linear methods
  - Principal component analysis (PCA)
  - Independent component analysis (ICA)
  - Canonical correlation analysis (CCA)
  - ...
- Nonlinear methods
  - Kernel PCA
  - Isomap
  - Locally linear embedding (LLE)
  - Autoencoders
  - ...

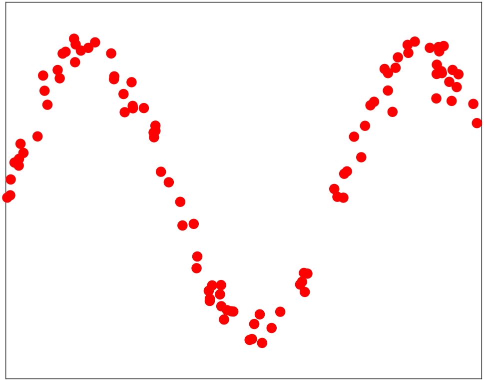
Dimensionality reduction techniques

- Linear methods
  - Principal component analysis (PCA)
  - Independent component analysis (ICA)
  - Canonical correlation analysis (CCA)
  - ...
- Nonlinear methods
  - Kernel PCA
  - Isomap
  - Locally linear embedding (LLE)
  - Autoencoders
  - ...

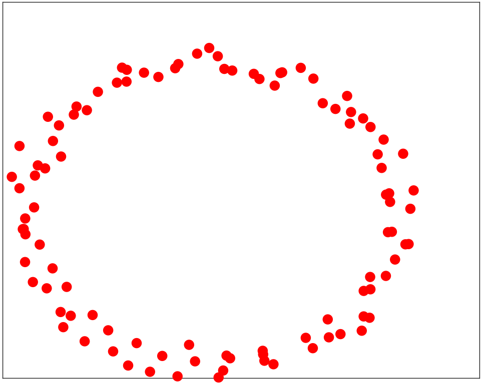
What is the true dimensionality of this data?



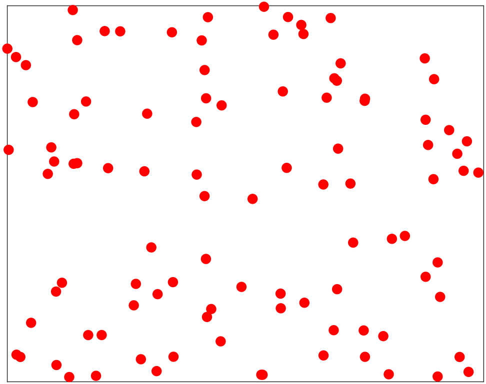
What is the true dimensionality of this data?



What is the true dimensionality of this data?



What is the true dimensionality of this data?



Remarks

- All dimensionality reduction techniques are based on an implicit assumption that the data lies along some low-dimensional manifold
- This is the case for the first three examples, which lie along a 1D manifold despite being plotted in 2D
- In the last example, the data has been generated randomly in 2D, so no dimensionality reduction is possible without losing information
- The first example is easier than the second and third examples

Principal Component Analysis

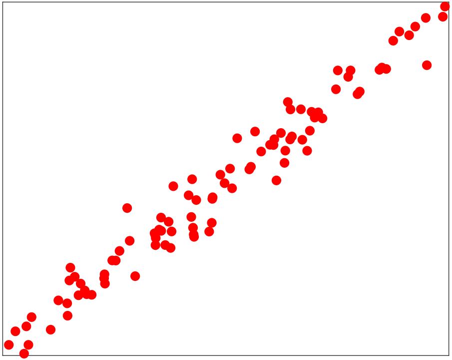
PCA motivations

- Given: m data points $\{x_i\}_{i=1}^m$, $x_i \in \mathbb{R}^n$. For convenience assume $\sum_{i=1}^m x_i = 0$.
- Suppose we want a 1-dimensional representation of that data, instead of n-dimensional

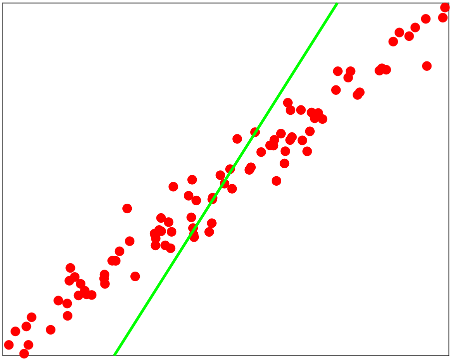
PCA motivations

- Given: m data points $\{x_i\}_{i=1}^m, x_i \in \mathbb{R}^n$. For convenience assume $\sum_{i=1}^m x_i = 0$.
- Suppose we want a 1-dimensional representation of that data, instead of n-dimensional
- Specifically, we will:
  1. Choose a line in $\mathbb{R}^n$ that best represents the data.
  2. Project the data points to along the line.

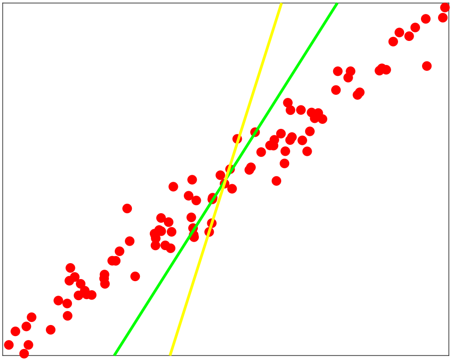
Find the best 1D representation



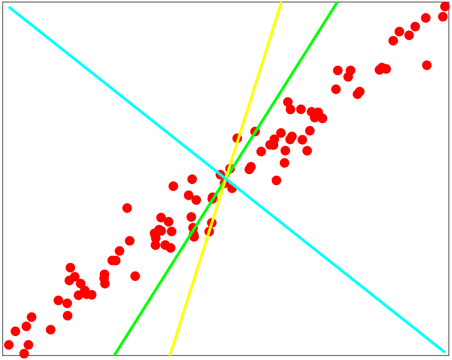
Find the best 1D representation



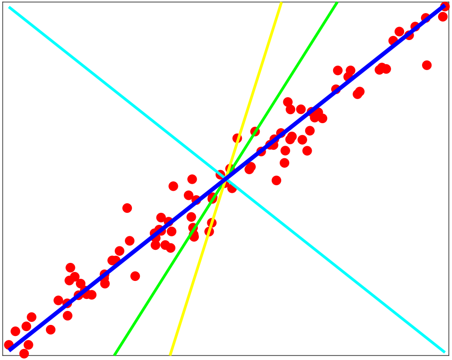
Find the best 1D representation



Find the best 1D representation



Find the best 1D representation



Minimize the reconstruction error

- Every line can be represented as $b + \alpha v$, where $b, v \in \mathbb{R}^n$, and $\alpha \in \mathbb{R}$. For convenience assume $||v||_2^2 = 1$.
- b can be viewed as the position of the line, v determines the direction of the line, and α is the distance between b and a point on the line (i.e., different α's define different points on the line).
- Each instance x_i is associated with a point on the line $\hat{x}_i = b + \alpha_i v$
- We want to choose b, v, and α_i to minimize the total reconstruction error over all data points, measured by Euclidean distance:

$$R = \sum_{i=1}^{m} ||x_i - \hat{x}_i||_2^2$$

where R is the reconstruction error

Constrained optimization problem (I)

$$\min_{b,v,\{\alpha_i\}_{i=1}^m} \sum_{i=1}^{m} ||x_i - (b + \alpha_i v)||_2^2$$
$$\text{s.t. } ||v||_2^2 = 1$$

- Suppose we fix a v satisfying the condition, and find the best b and α_i given v
- Then, we solve:

$$\min R = \min_{b,\{\alpha_i\}_{i=1}^m} \sum_{i=1}^{m} ||x_i - (b + \alpha_i v)||_2^2$$

- Compute the gradient of R with respect to α_i and set it to 0 (note that $||v||_2^2 = 1$):

$$\frac{\partial R}{\partial \alpha_i} = 2\alpha_i - 2v^\top x_i + 2v^\top b = 0 \Rightarrow \alpha_i = v^\top (x_i - b)$$

Constrained optimization problem (II)

$$\min R = \min_{b,\{\alpha_i\}_{i=1}^m} \sum_{i=1}^m ||x_i - (b + \alpha_i v)||_2^2$$

- Compute the gradient of R with respect to b and set it to 0:

$$\nabla_b R = 2mb - 2\sum_{i=1}^m x_i + 2\left(\sum_{i=1}^m \alpha_i\right) v = 0$$

$$\Rightarrow \left(\sum_{i=1}^m \alpha_i\right) v = \sum_{i=1}^m x_i - mb$$

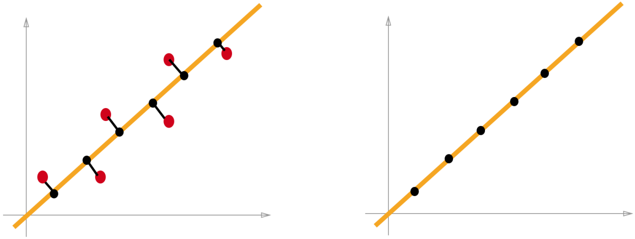
$$\Rightarrow v^\top \left(\sum_{i=1}^m x_i - mb\right) v = \sum_{i=1}^m x_i - mb$$

  This is satisfied when:

$$\sum_{i=1}^m x_i - mb = 0 \Rightarrow b = \frac{1}{m}\sum_{i=1}^m x_i = 0$$

- By substituting α_i, we get $\hat{x}_i = b + v^\top(x_i - b)v = vv^\top x_i$, which means that instances are projected orthogonally on the line to get $\hat{x}_i$.

Example of PCA projection



Find the direction of the line

$$\min_{b,v,\{\alpha_i\}_{i=1}^m} \sum_{i=1}^m ||x_i - (b + \alpha_i v)||_2^2$$
$$\text{s.t. } ||v||_2^2 = 1$$

- Substituting $\alpha_i = v^\top(x_i - b)$ and $b = 0$ gives

$$\max_v \sum_{i=1}^m v^\top x_i x_i^\top v \tag{1}$$
$$\text{s.t. } ||v||_2^2 = 1$$

- Let $X = [x_1, \ldots x_m]^\top \in \mathbb{R}^{m \times n}$, then (1) is equivalent to

$$\max_v v^\top X^\top X v$$
$$\text{s.t. } ||v||_2^2 = 1$$

- The Lagrangian is: $L(v, \lambda) = v^\top X^\top X v - \lambda ||v||_2^2$
- The solution to the problem, obtained by setting $\nabla_v L = 0$, is: $X^\top X v = \lambda v$

Optimal choice of v

$$X^\top X v = \lambda v$$

- Recall: an eigenvector v of a matrix A satisfies $Av = \lambda v$, where $\lambda \in \mathbb{R}$ is the eigenvalue
- Fact: $X^\top X$ has n non-negative eigenvalues and n orthogonal eigenvectors.
- v should be the eigenvector of $X^\top X$ associated with the largest eigenvalue!
- As $X^\top X$ is the sample correlation/covariance matrix, v essentially finds a direction along which the variance of data points is maximized!

Two equivalent views of PCA

1. Project data onto a low dimension space while keep as much information as possible (minimize the reconstruction error):

$$R = \sum_{i=1}^{m} ||x_i - \hat{x}_i||_2^2$$

   If $\sum_{i=1}^{m} x_i = 0$, we show that it is equivalent to

$$\min_{v} \sum_{i=1}^{m} ||x_i - vv^\top x_i||, \qquad \text{s.t. } ||v||_2^2 = 1 \tag{2}$$

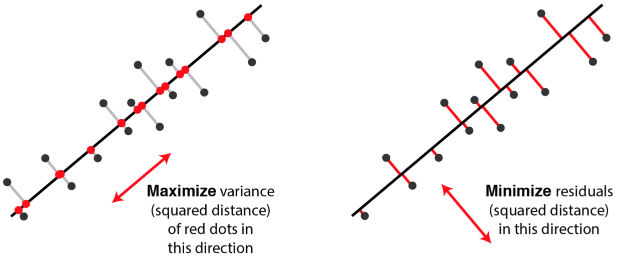
2. Then, we show that (2) is equivalent to

$$\max_{v} v^\top X^\top X v, \qquad \text{s.t. } ||v||_2^2 = 1 \tag{3}$$

   which can be solved by eigenvalue decomposition

3. $X^\top X$ is the sample correlation/covariance matrix $\Rightarrow$ minimize the reconstruction error = maximize the sample covariance of the data!

Two equivalent views of PCA



http://alexhwilliams.info/itsneuronalblog/2016/03/27/pca/

Remarks

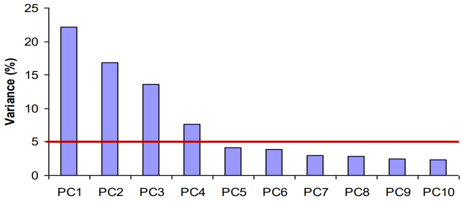
- The first principal component v_1 is the the eigenvector of the sample covariance matrix $X^\top X$ associated with the largest eigenvalue
- The second principal component v_1 is the the eigenvector of the sample covariance matrix $X^\top X$ associated with the second largest eigenvalue
- And so on...
- $v_i^\top v_j = 0$, principal components are orthogonal to each other

How many principal components shall we keep?

- When the eigenvalues are sorted in decreasing order, the proportion of the variance captured by the first *d* components is:

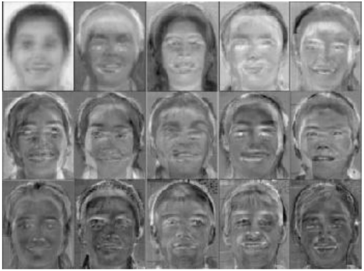
$$\frac{\lambda_1 + \ldots, +\lambda_d}{\lambda_1 + \ldots, +\lambda_d + \lambda_{d+1} + \ldots, \lambda_n}$$

- So if a “big” drop occurs in the eigenvalues at some point, that suggests a good dimension cutoff
- Might lose some info, but if eigenvalues are small, do not lose much



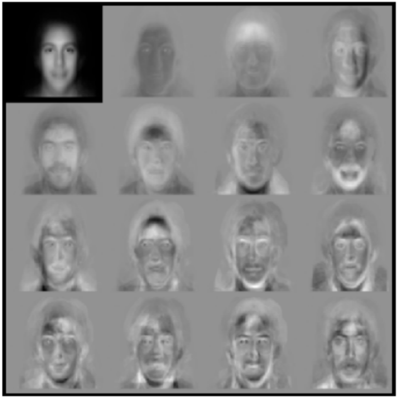
Eigenface example

- A set of faces on the left and the corresponding eigenfaces (principal components) on the right
- The faces have to be centred and scaled ahead of time
- The components are in the same space as the instances (images) and can be used to reconstruct the images



Eigenface example

Top left image is a linear combination of rest



Kernel PCA

Motivations

- PCA cannot distinguish non-linear structure
- We can use a similar idea as in support vector machine: instead of using the points x, we go to some feature mapping: $x \to \phi(x)$
- In the higher dimensional space, we can then do PCA
- The result will be non-linear in the original data space!

Kernel PCA (I)

- Suppose that the mean of the data in feature space is 0: $\sum_{i=1}^{m} \phi(x_i) = 0$
- The sample covariance matrix is:

$$C = \sum_{i=1}^{m} \phi(x_i)\phi(x_i)^\top$$

- The corresponding eigen-decomposition problem is

$$Cv_j = \lambda_j v_j$$

- We want to avoid explicitly going to feature space - instead we want to work with kernels:

$$K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

Kernel PCA (II)

- Re-write the PCA equation:

$$\sum_{i=1}^{m} \phi(x_i)\phi(x_i)^\top v_j = \lambda_j v_j$$

- Assume that the eigenvectors can be written as a linear combination for features:

$$v_j = \sum_{i=1}^{m} a_{ji}\phi(x_i)$$

- By substituting this back into the equation we get:

$$\sum_{i=1}^{m} \phi(x_i)\phi(x_i)^\top \left(\sum_{k=1}^{m} a_{jk}\phi(x_k)\right) = \lambda_j \sum_{k=1}^{m} a_{jk}\phi(x_k)$$

$$\Rightarrow \sum_{i=1}^{m} \phi(x_i) \left(\sum_{k=1}^{m} a_{jk}K(x_i, x_k)\right) = \lambda_j \sum_{k=1}^{m} a_{jk}\phi(x_k)$$

Kernel PCA (IV)

- multiply the equation by $\phi(x_l)^\top$ to the left:

$$\sum_{i=1}^{m} \phi(x_l)^\top \phi(x_i) \left(\sum_{k=1}^{m} a_{jk} K(x_i, x_k) \right) = \lambda_j \sum_{k=1}^{m} a_{jk} \phi(x_l)^\top \phi(x_k)$$

$$\Rightarrow \sum_{i=1}^{m} K(x_l, x_i) \left(\sum_{k=1}^{m} a_{jk} K(x_i, x_k) \right) = \lambda_j \sum_{k=1}^{m} a_{jk} K(x_l, x_k)$$

- By rearranging we get:

$$K^2 a_j = \lambda_j K a_j,$$

where $a_j = [a_{j1}, \ldots, a_{jm}]^\top \in \mathbb{R}^n$

- We can remove a factor of K from both sides of the matrix

$$K a_j = \lambda_j a_j$$

- For a new data point x its projection onto the principal components is:

$$\phi(x)^\top v = \sum_{i=1}^{m} a_{ji} \phi(x)^\top \phi(x_i) = \sum_{i=1}^{m} a_{ji} K(x, x_i)$$

Kernel PCA example

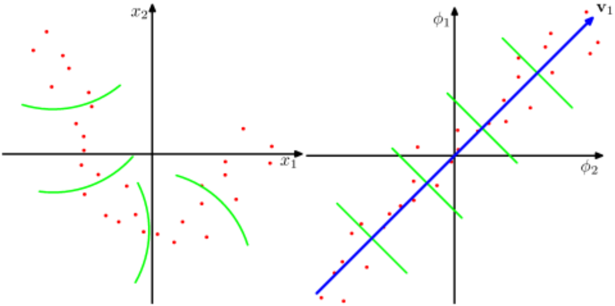


Figure credit: Christopher Bishop

Summary

- Unsupervised learning: discover patterns and structure from data without label information
- Dimensionality reduction: compress and visualize data
- PCA: find a linear projection such that the reconstruction error is minimized $\Leftrightarrow$ the variance of the data points is maximized
- Kernel PCA: a nonlinear extension of PCA using the kernel trick