Dimension Reduction

April 6, 2021

Slides Courtesy of Jiayu Zhou

Some slides from "Principal Component Analysis" by Frank Wood

Announcements

- HW 8: Individual & Due April 9 at midnight
- HW 9: Group & Due April 16 at midnight
- HW 10: Probably due April 21
- Optional HW 11 $\rightarrow$ Dropping one instead
- Exam 3 will be "take home"
- Remaining Lectures: Clustering, Neural Networks, Open Topics (Piazza Post)
- Mimir Code: Code 6 & 7 need to be changed.
  All due April 21? 25?

Today's Lecture

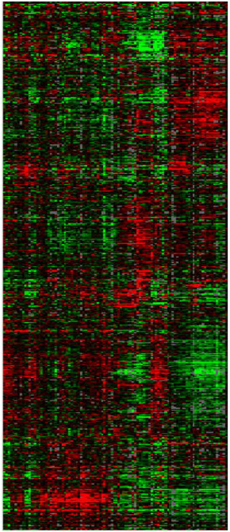
1. Feature Reduction
2. Principal Component Analysis (PCA)
   - Introduction
   - Derivation

Feature Reduction

Feature Reduction

- **Dimension/ality reduction:** reduce the number of random variables we need to consider by obtaining a set of principal values
- In ML, typically refer to dimension reduction as **feature reduction** because dimension corresponds to the number of features
- Think of it as trying to find the most important features for the model's prediction
- PCA
  - Main linear technique for dimension/feature reduction
  - Linear mapping of data to lower-dimension space such that the variance is maximized

High-dimensional Data



Gene expression

Face images

Challenges with High-dimensional Data

- Most machine learning and data mining techniques may not be effective for high-dimensional data
  - **Curse of Dimensionality**
  - Model accuracy and efficiency degrade rapidly as the dimension increases

Challenges with High-dimensional Data

- Most machine learning and data mining techniques may not be effective for high-dimensional data
    - **Curse of Dimensionality**
    - Model accuracy and efficiency degrade rapidly as the dimension increases
- **Intrinsic Dimension**
    - Number of variables needed for minimal representation of data
    - May be small
    - For example, the number of genes responsible for a certain type of disease may be small

Feature Reduction

- Feature Reduction
  - All original features are used
  - The transformed features are linear combinations of the original features
- Feature Selection
  - Only a subset of original features are used

Feature Reduction

- **Feature reduction** refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Criterion for feature reduction can be different based on different problem settings
  - Supervised setting: maximize model's classification ability
  - Unsupervised setting: minimize information loss

Feature Reduction

- **Feature reduction** refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Criterion for feature reduction can be different based on different problem settings
  - Supervised setting: maximize model's classification ability
  - Unsupervised setting: minimize information loss
- Given a set of data points of p variables $\{x_1, x_2, \ldots, x_n\}$, compute the linear transformation (projection)

$$G \in \mathbb{R}^{p \times d} : \mathbf{x} \in \mathbb{R}^p \to \mathbf{y} = G^T \mathbf{x} \in \mathbb{R}^d \quad \text{(typically } d \ll p\text{)}$$

Other Benefits of Feature Reduction

- **Visualization**: projection of high-dimensional data onto 2D or 3D
- **Data compression**: efficient storage and retrieval
- **Noise removal**: positive effect on query accuracy

Applications of Feature Reduction

- Face recognition
- Handwritten digit recognition
- Text mining
- Image retrieval
- Microarray data analysis
- Protein classification

Feature Reduction Techniques

- Unsupervised
  - Latent Semantic Indexing (LSI): truncated SVD
  - Independent Component Analysis (ICA)
  - Principal Component Analysis (PCA)
  - Canonical Correlation Analysis (CCA)
- Supervised
  - Linear Discriminant Analysis (LDA)

Principal Component Analysis (PCA)

Principal Component Analysis

- **Principal Component Analysis (PCA)**
  - Reduces the dimensionality of a dataset by finding **a new set of variables**, smaller than the original set of variables
  - Retains most of the sample's variance
  - Useful for the compression and classification of data

Principal Component Analysis

- **Principal Component Analysis (PCA)**
  - Reduces the dimensionality of a dataset by finding **a new set of variables**, smaller than the original set of variables
  - Retains most of the sample's variance
  - Useful for the compression and classification of data
- **Principal Components (PCs)**: the new variables
  - Are uncorrelated
  - Are ordered by the fraction of total information each retains, where *information* means the variation present in the sample, given by the correlations between the original variables

Data Distribution (Input in Regression/Classification)

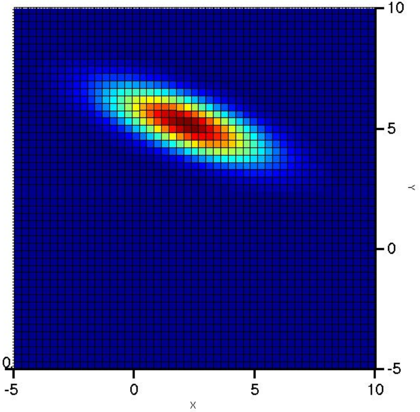


Figure: Gaussian PDF

Uncorrelated Projections of Principal Variation

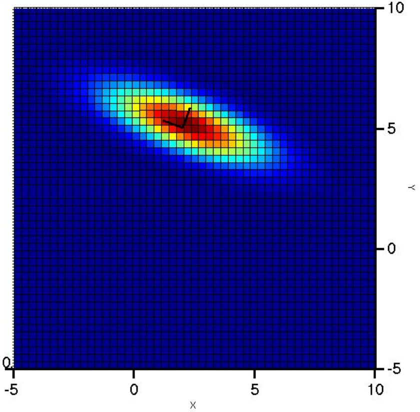
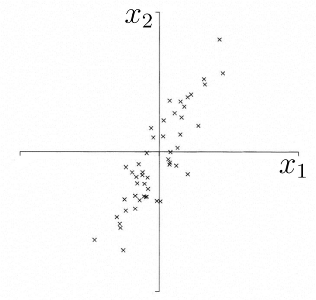
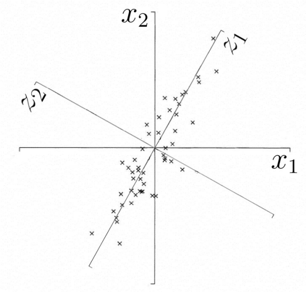


Figure: Gaussian PDF with PC eigenvectors

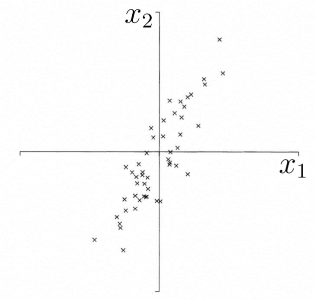
Geometric Picture of Principal Components (PCs)

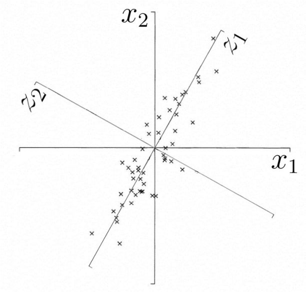




- The 1st PC z_1 is a minimum distance fit to a line in X space
- The 2nd PC z_2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC

Geometric Picture of Principal Components (PCs)





- The 1st PC z_1 is a minimum distance fit to a line in X space
- The 2nd PC z_2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC
- PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous PCs

In Other Words

- In PCA, we identify a first axis that accounts for largest amount of variance in training dataset
- Second axis will be orthogonal to the first and account for largest amount of remaining variance
- For higher dimensions, we can find a 3^{rd} & 4^{th}, and so on, as many axes as the number of dimensions in dataset
- These *axes* $==$ **principal components** of data

Idea

- We want to find the best fit hyperplane (closest to data) and project our data onto it
- Want a projection that: preserves most of the variance $\rightarrow$ which means it loses the least info $\rightarrow$ which means it minimizes the squared distance between the original dataset and its projection on axis

How to Implement PCA

High-level view of the algorithm:

- Compute the covariance matrix of the data
- Compute the eigenvalues and eigenvectors of this covariance matrix
- Use the eigenvalues and eigenvectors to select the most important feature vectors
- Transform your data onto those vectors for reduced dimensionality

PCA Algorithm: Covariance Matrix

- First normalize data to have zero-mean and unit-variance
    - This ensures that each feature will be weighted equally

PCA Algorithm: Covariance Matrix

- First normalize data to have zero-mean and unit-variance
  - This ensures that each feature will be weighted equally
- Covariance of 2 variables measures how correlated they are
  - Positive covariance: when one variable increases/decreases, the other increases/decreases
  - Negative covariance: values of feature variables change in opposite directions
  - Covariance matrix: array where each value specifies covariance between 2 feature variables based on x-y position in the matrix

PCA Algorithm: Covariance Matrix

- First normalize data to have zero-mean and unit-variance
  - This ensures that each feature will be weighted equally
- Covariance of 2 variables measures how correlated they are
  - Positive covariance: when one variable increases/decreases, the other increases/decreases
  - Negative covariance: values of feature variables change in opposite directions
  - Covariance matrix: array where each value specifies covariance between 2 feature variables based on x-y position in the matrix

$$\Sigma = \frac{1}{n-1}((X - \bar{x})^T(X - \bar{x}))$$

PCA Algorithm: Eigenvectors & Eigenvalues

- Eigenvectors == Principal Components
  - From the covariance matrix
  - Represent vector directions of new feature space
- Eigenvalues
  - Represent magnitude of vectors
  - Quantify the contributing variance of each vector

PCA Algorithm: Eigenvectors & Eigenvalues

- Eigenvectors == Principal Components
  - From the covariance matrix
  - Represent vector directions of new feature space
- Eigenvalues
  - Represent magnitude of vectors
  - Quantify the contributing variance of each vector
- Magnitude of eigenvector's corresponding eigenvalue
  - High magnitude (length) $\rightarrow$ data has high variance along that vector in feature space $\rightarrow$ vector holds a lot of information about dataset
  - Small eigenvalue $\rightarrow$ low variance $\rightarrow$ data doesn't vary greatly along that vector
    - Changing the value of this feature vector doesn't affect data, so we can say the feature isn't that important and remove it

PCA Algorithm: Selection

- Have list of eigenvectors sorted in order of importance to dataset (sort based on eigenvalues)
- Need to select most important feature vectors and discard the rest

PCA Algorithm: Selection

Explained Variance Percentage

- Quantifies how much information (variance) can be associated to each of the PCs
- For example: a dataset with 10 feature vectors has the following eigenvalues: $[12, 10, 8, 7, 5, 1, 0.1, 0.03, 0.005, 0.0009]$
  Σ of array $=$ 43.1359
  Σ of [12, 10, 8, 7, 5, 1] $=$ 43
  So 43/43.14 $=$ 99.68% of total variance

PCA Algorithm: Selection

Explained Variance Percentage

- Quantifies how much information (variance) can be associated to each of the PCs
- For example: a dataset with 10 feature vectors has the following eigenvalues: $[12, 10, 8, 7, 5, 1, 0.1, 0.03, 0.005, 0.0009]$
  Σ of array = 43.1359
  Σ of [12, 10, 8, 7, 5, 1] = 43
  So 43/43.14 = 99.68% of total variance
- Define a threshold (usually 95% or higher) and keep or discard feature vectors (or use *top k*)

PCA Algorithm: Projection Matrix

- Final step is to build a projection matrix to project our data onto the new vectors
- Projection matrix = concatenate most important eigenvectors
- Calculate the dot product of original data and our projection matrix

Extra Slides

Algebraic Definition of PCs

Given a sample of n observations on a vector of p variables

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p, \ \mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{pj})$$

define the first principal component of the sample by the linear transformation

$$z_1 = \mathbf{a}_1^T \mathbf{x}_j = \sum_{i=1}^{p} a_{i1} x_{ij}, j = 1, \ldots, n$$

where the vector

$$\mathbf{a}_1 = (a_{11}, a_{21}, \ldots, a_{p1})$$

is chosen such that $var[z_1]$ is maximum.

Algebraic Definition of PCs

To find $\mathbf{a}_1$ first note that

$$\mathsf{Var}[z_1] = \mathbb{E}((z_1 - \bar{z}_1)^2) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{a}_1^T\mathbf{x}_i - \mathbf{a}_1^T\bar{\mathbf{x}})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbf{a}_1^T(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{a}_1 = \mathbf{a}_1^T S\mathbf{a}_1$$

where

- $S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$: the covariance matrix
- $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$: the mean.

Algebraic Definition of PCs

To find $\mathbf{a}_1$ first note that

$$\mathsf{Var}[z_1] = \mathbb{E}((z_1 - \bar{z}_1)^2) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{a}_1^T\mathbf{x}_i - \mathbf{a}_1^T\bar{\mathbf{x}})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbf{a}_1^T(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{a}_1 = \mathbf{a}_1^T S\mathbf{a}_1$$

where

- $S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$: the covariance matrix
- $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$: the mean.

In the following, we assume the data is centered, i.e., $\bar{\mathbf{x}} = 0$, e.g., each feature has zero mean.

Algebraic Derivation of PCs

To find $\mathbf{a}_1$: $\max_{\mathbf{a}_1} \text{Var}[z_1]$, s.t. $\mathbf{a}_1^T \mathbf{a}_1 = 1$
Let λ be a Lagrange multiplier

$$L = \mathbf{a}_1^T S \mathbf{a}_1 - \lambda(\mathbf{a}_1^T \mathbf{a}_1 - 1)$$
$$\frac{\partial L}{\partial \mathbf{a}_1} = S\mathbf{a}_1 - \lambda \mathbf{a}_1 = 0$$

Algebraic Derivation of PCs

To find $\mathbf{a}_1$: $\max_{\mathbf{a}_1}$ Var$[z_1]$, s.t. $\mathbf{a}_1^T\mathbf{a}_1 = 1$
Let λ be a Lagrange multiplier

$$L = \mathbf{a}_1^T S\mathbf{a}_1 - \lambda(\mathbf{a}_1^T\mathbf{a}_1 - 1)$$
$$\frac{\partial L}{\partial \mathbf{a}_1} = S\mathbf{a}_1 - \lambda\mathbf{a}_1 = 0$$

- Therefore $\mathbf{a}_1$ is an eigenvector of S corresponding to the largest eigenvalue $\lambda = \lambda_1$

Algebraic Derivation of PCs

- We want an uncorrelated direction for our next direction z_2, i.e., $\text{Cov}[z_2, z_1] = 0$.

Algebraic Derivation of PCs

- We want an uncorrelated direction for our next direction z_2, i.e., $\text{Cov}[z_2, z_1] = 0$.
- To find the next coefficient vector $\mathbf{a}_2$:

$$\max_{\mathbf{a}_2} \text{Var}[z_2] \quad \mathbf{a}_2^T \mathbf{a}_2 = 1, \text{Cov}[z_2, z_1] = 0$$

- We note that

$$\text{Cov}[z_2, z_1] = \mathbf{a}_1^T S \mathbf{a}_2 = \lambda_1 \mathbf{a}_1^T \mathbf{a}_2$$

Therefore

$$\max_{\mathbf{a}_2} \text{Var}[z_2] \quad \mathbf{a}_2^T \mathbf{a}_2 = 1, \mathbf{a}_1^T \mathbf{a}_2 = 0$$

Algebraic Derivation of PCs

$$
\begin{aligned}
L &= \mathbf{a}_2^T S \mathbf{a}_2 - \lambda(\mathbf{a}_2^T \mathbf{a}_2 - 1) - \gamma \mathbf{a}_1^T \mathbf{a}_2 \\
\Rightarrow \frac{\partial L}{\partial \mathbf{a}_2} &= S\mathbf{a}_2 - \lambda \mathbf{a}_2 - \gamma \mathbf{a}_1 = 0 \Rightarrow \gamma = 0 \\
&\Rightarrow S\mathbf{a}_2 = \lambda \mathbf{a}_2
\end{aligned}
$$

- We find that $\mathbf{a}_2$ is also an eigenvector of S
- whose eigenvalue $\lambda = \lambda_2$ is the second largest.

Algebraic Derivation of PCs

- In general

$$\text{Var}[z_k] = \mathbf{a}_k^T S \mathbf{a}_k = \lambda_k$$

  - The kth largest eigenvalue of S is the variance of the kth PC.
  - The kth PC retains the kth greatest fraction of the variation in the sample.

Algebraic Derivation of PCs

- Main steps for computing PCs
  - Standarize the dataset $X \in \mathbb{R}^{n \times p}$
  - Form the covariance matrix $S = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = \frac{1}{n} X^T X$
  - Compute its eigenvectors $\{\mathbf{a}_i\}_{i=1}^{p}$
  - Use the first d eigenvectors $\{\mathbf{a}_i\}_{i=1}^{d}$ to form the d PCs.
  - The transformation $G = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d]$
- Apply PCA: $\mathbf{x} \in \mathbb{R}^p \rightarrow G^T \mathbf{x} \in \mathbb{R}^d$

Optimality Property of PCA

- **Main theoretical result:**
  The matrix G consisting of the first d eigenvectors of the covariance matrix S solves the following min problem:

$$\min_{G \in \mathbb{R}^{p \times d}} \|X^T - G(G^T X^T)\|_F^2 \quad \text{subject to:} \quad G^T G = I_d$$

- $G(G^T X^T)$ is the "reconstructed data matrix", and the objective minimizes reconstruction error.
- PCA projection minimizes the reconstruction error among all linear projections of size d.

Application on Image Compression