



**ETC3250**

# **Business Analytics**

**Week 5**

**Other dimensionality reduction methods**

23 August 2016

# Dimensionality reduction

- Why dimensionality reduction?
  - Curse of dimensionality
  - Intrinsic dimensionality
  - Visualization
- Dimensionality reduction methods
  - Feature selection vs feature extraction (examples? advantages?)
  - Unsupervised vs supervised
  - Linear vs nonlinear
- Principal components analysis (PCA)
  - PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
  - PCA: linear and unsupervised feature extraction

# Dimensionality reduction

- Why dimensionality reduction?
  - Curse of dimensionality
  - Intrinsic dimensionality
  - Visualization
- Dimensionality reduction methods
  - Feature selection vs feature extraction (examples? advantages?)
  - Unsupervised vs supervised
  - Linear vs nonlinear
- Principal components analysis (PCA)
  - PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
  - PCA: linear and unsupervised feature extraction

# Dimensionality reduction

- Why dimensionality reduction?
  - Curse of dimensionality
  - Intrinsic dimensionality
  - Visualization
- Dimensionality reduction methods
  - Feature selection vs feature extraction (examples? advantages?)
  - Unsupervised vs supervised
  - Linear vs nonlinear
- Principal components analysis (PCA)
  - PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
  - PCA: linear and unsupervised feature extraction

# Dimensionality reduction in regression

- How would you reduce dimensionality in linear regression?
- How would use PCA to reduce dimensionality in linear regression?
- *Principal components regression (PCR): use PCA to construct the first  $M$  principal components,  $Z_1, \dots, Z_M$ , and then fit a linear regression model using these components.*
- *Assumption: the directions in which  $X_1, \dots, X_p$  show the most variation are the directions that are associated with  $Y$ . When is PCR better than traditional least squares?*

# Dimensionality reduction in regression

- How would you reduce dimensionality in linear regression?
- How would use PCA to reduce dimensionality in linear regression?
- *Principal components regression (PCR): use PCA to construct the first  $M$  principal components,  $Z_1, \dots, Z_M$ , and then fit a linear regression model using these components.*
- *Assumption: the directions in which  $X_1, \dots, X_p$  show the most variation are the directions that are associated with  $Y$ . When is PCR better than traditional least squares?*

# Dimensionality reduction in regression

- How would you reduce dimensionality in linear regression?
- How would use PCA to reduce dimensionality in linear regression?
- *Principal components regression (PCR): use PCA to construct the first  $M$  principal components,  $Z_1, \dots, Z_M$ , and then fit a linear regression model using these components.*
- *Assumption: the directions in which  $X_1, \dots, X_p$  show the most variation are the directions that are associated with  $Y$ . When is PCR better than traditional least squares?*

# Partial least squares

- With PCR, the components are identified in an **unsupervised way**. No guarantee that the directions that **best explain the predictors** will also be the **best directions to use for predicting the response  $Y$** . **How would you use the response  $Y$  to reduce dimensionality?**
- Partial least squares (PLS) is a **supervised** alternative to PCR. Same procedure as PCR, but the new features are identified in a supervised way. Roughly speaking, PLS attempts to find directions that help **explain both the response and the predictors**.



# Partial least squares

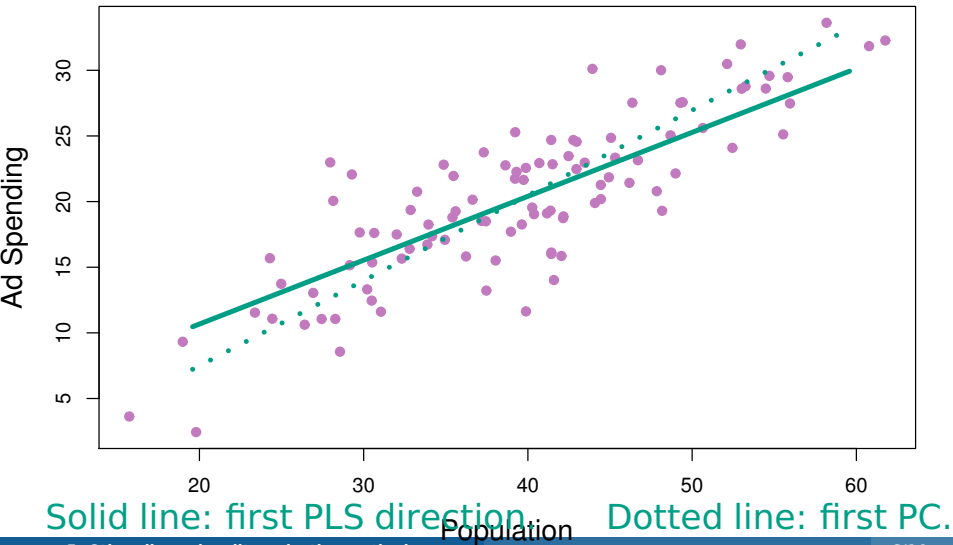
- With PCR, the components are identified in an **unsupervised way**. No guarantee that the directions that **best explain the predictors** will also be the **best directions to use for predicting the response  $Y$** . **How would you use the response  $Y$  to reduce dimensionality?**
- Partial least squares (PLS) is a **supervised** alternative to PCR. Same procedure as PCR, but the new features are identified in a supervised way. Roughly speaking, PLS attempts to find directions that help **explain both the response and the predictors**.

# Partial least squares

- 1 Standardize the  $p$  predictors
- 2 Compute the first direction  $Z_1$  by setting each  $\phi_{j1}$  in  $Z_1 = \sum_{j=1}^p \phi_{j1} X_j$  equal to the coefficient from the simple linear regression of  $Y$  onto  $X_j$  (see Figure)
- 3 For the second direction  $Z_2$ , we first adjust each of the variables for  $Z_1$ , by regressing each variable on  $Z_1$ , and taking residuals, say  $X'_j$ . These residuals represent the remaining information that has not been explained by the the first PLS direction  $Z_1$ . We then compute  $Z_2$  using  $X'_j$  as  $Z_1$  was computed from  $X_j$
- 4 We can compute  $Z_1, \dots, Z_M$  iteratively. Then we use least squares to fit a linear model using  $Z_1, \dots, Z_M$  as for PCR.

How to choose  $M$ ?

# PLS Example



# Partial least squares

- There are two variants of PLS: PLS1 (one response variable) and PLS2 (at least two response variables).
- In practice, PLS1 is not better than ridge regression or PCR (PLS reduces bias but can potentially increase variance).
- PLS2 is a useful tool for multiresponse regression.
- Other supervised dimensionality reduction methods: CCA, LDA, etc.

# Multidimensional scaling

Suppose that instead of measuring a  $n \times p$  dataset  $\mathbf{X} = [x_{ij}]$ , we were only given a pairwise distance matrix  $\Delta$  where

$$\Delta_{ij} = \|x_i - x_j\|, \quad i, j = 1, \dots, N$$

i.e. we do not know the points themselves. Can we find a lower-dimension representation  $\mathbf{Z}$  for  $\mathbf{X}$  from  $\Delta$ ? (If we have only a distance matrix, we cannot perform PCA.)

*Multidimensional scaling* (MDS) attempts to find a lower dimensional space so that distances between points are preserved as well as possible.

# Multidimensional scaling

Suppose that instead of measuring a  $n \times p$  dataset  $\mathbf{X} = [x_{ij}]$ , we were only given a pairwise distance matrix  $\Delta$  where

$$\Delta_{ij} = \|x_i - x_j\|, \quad i, j = 1, \dots, N$$

i.e. we do not know the points themselves. Can we find a lower-dimension representation  $\mathbf{Z}$  for  $\mathbf{X}$  from  $\Delta$ ? (If we have only a distance matrix, we cannot perform PCA.)

*Multidimensional scaling* (MDS) attempts to find a lower dimensional space so that distances between points are preserved as well as possible.

# Multidimensional scaling

MDS seeks values  $z_1, z_2, \dots, z_n \in \mathbb{R}^k$  that minimize the so-called stress function:

$$S_M(z_1, z_2, \dots, z_n) = \sum_{i \neq j} (\|x_i - x_j\| - \|z_i - z_j\|)^2.$$

(least squares or *Kruskal-Shephard* scaling).

A variation on least squares scaling minimizes

$$S_{Sm}(z_1, z_2, \dots, z_n) = \sum_{i \neq j} \frac{(\|x_i - x_j\| - \|z_i - z_j\|)^2}{\|x_i - x_j\|}$$

(*Sammon mapping*) Difference with least squares scaling?

# Link between MDS and PCA

Computation of PCs in PCA:

- Singular value decomposition

- $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}'$  with  $\mathbf{U}'\mathbf{U} = \mathbf{I}$  and  $\mathbf{V}'\mathbf{V} = \mathbf{I}$

- $\Phi = \mathbf{V} \rightarrow \mathbf{Z} = \mathbf{X}\Phi$

- Eigenvalue decomposition

- $\mathbf{C} = \mathbf{X}'\mathbf{X}$  where the columns of  $\mathbf{X}$  are scaled

- $\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}'$  with  $\mathbf{V}'\mathbf{V} = \mathbf{I}$

- $\Phi = \mathbf{V} \rightarrow \mathbf{Z} = \mathbf{X}\Phi$



# Link between MDS and PCA

If  $\Delta_{ij}$  are Euclidean distances between the rows of  $\mathbf{X}$ , then MDS is equivalent to PCA.

Given distance matrix  $\Delta \in \mathbb{R}^{n \times n}$ ,

**1** Recover the inner product  $\mathbf{X}\mathbf{X}'$  from  $\Delta$

- Compute  $A_{ij} = -\frac{1}{2}\Delta_{ij}^2$
- Double center  $A$  to recover  $B$ :  
 $B = (I - M)A(I - M)$  where  $M = \frac{1}{n}\mathbf{1}\mathbf{1}' \in \mathbb{R}^{n \times n}$

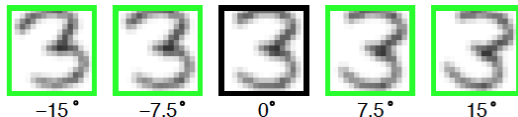
**2** Factorize  $B$  to get the first  $k$  principal component scores

This is called *classical multidimensional scaling*.

$$S_C(z_1, z_2, \dots, z_n) = \sum_{i \neq j} (\langle x_i - \bar{x}, x_j - \bar{x} \rangle - \langle z_i - \bar{z}, z_j - \bar{z} \rangle)^2$$

# Multidimensional scaling

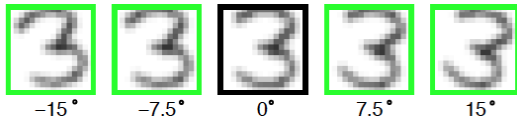
- Multidimensional scaling can be applied to any  $\Delta_{ij}$ , not just Euclidean distances
- In this case, we don't compute principal component scores, and the lower-dimensional representation can be a nonlinear function of the data
- When do we need to use non-Euclidean distances?



We wish to remove the effect of rotation in measuring distances between two digits

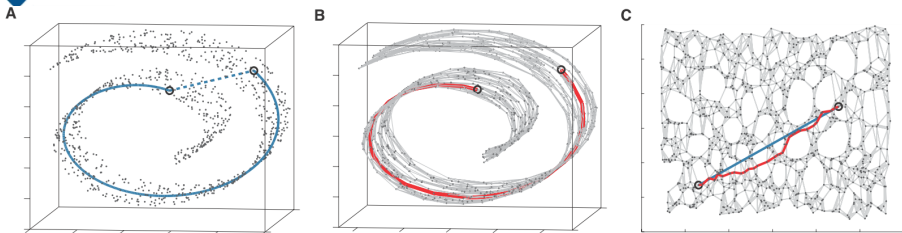
# Multidimensional scaling

- Multidimensional scaling can be applied to any  $\Delta_{ij}$ , not just Euclidean distances
- In this case, we don't compute principal component scores, and the lower-dimensional representation can be a nonlinear function of the data
- When do we need to use non-Euclidean distances?



We wish to remove the effect of rotation in measuring distances between two digits

# Isometric feature mapping



(From Tenenbaum et al. (2000), “A global geometric framework for nonlinear dimensionality reduction”)

- Construct a graph  $G = (V, E)$  based on the structure between  $x_1, \dots, x_n$ .
- Then, define a graph distance  $\Delta_{ij}^{\text{Isomap}}$  between  $i$  and  $j$ , and use MDS for the low-dimensional representation

# Dimensionality reduction methods

- Feature selection vs feature extraction
- Linear and nonlinear
- Unsupervised and supervised
- Low-dimensional representation with maximum variance, that retains local properties of the data, etc.
- **Linear PCA**, Nonlinear PCA, Kernel PCA, Sparse PCA, etc.
- **MDS**, ICA, LDA, etc.
- **PLS**, CCA, FA, etc.
- **Isomap**, diffusion maps, MVU, LLE, t-SNE, autoencoders, etc.