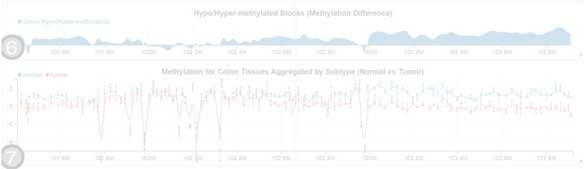


Unsupervised Introduction to Data Science: Dimensionality Reduction

Héctor Corrada Bravo

University of Maryland, College Park, USA

CMSC320: 2020-04-28

Unsupervised Learning

Unsupervised data: characterize patterns in predictor space where observation measurements are represented.

Mathematically, characterize $p(X)$ over p-dimensional predictor space.

Clustering methods assume that this space $p(X)$ can be partitioned into subspaces containing "similar" observations.

Unsupervised Learning: Dimensionality Reduction

Dimensionality reduction: assume observations can be represented in a space with dimension much lower than p.

There are two general strategies for dimensionality reduction:

- data transformations into spaces of smaller dimension that capture global properties of a data set X,
- data embeddings into lower dimensional spaces that retain local properties of a data set X.

We will only see the first.

Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction method.

Goal: *embed data in high dimensional space (e.g., observations with a large number of variables), onto a small number of dimensions*.

Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction method.

Goal: *embed data in high dimensional space (e.g., observations with a large number of variables), onto a small number of dimensions*.

Most frequent use is in Exploratory Data Analysis and visualization

Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction method.

Goal: *embed data in high dimensional space (e.g., observations with a large number of variables), onto a small number of dimensions*.

Most frequent use is in Exploratory Data Analysis and visualization

Also be helpful in regression (linear or logistic) where we can transform input variables into a smaller number of predictors for modeling.

Principal Component Analysis

Mathematically, the PCA problem is:

Given:

- Data set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i$ is the vector of p variable values for the i-th observation.

Return:

- Matrix $[\phi_1, \phi_2, \ldots, \phi_p]$ of *linear transformations* that retain *maximal variance*.

Principal Component Analysis

Think of the first vector ϕ_1 as a linear transformation that embeds observations into 1 dimension:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p$$

where ϕ_1 is selected so that the resulting dataset $\{z_1, \ldots, z_n\}$ has *maximum variance*.

Principal Component Analysis

In order for this to make sense mathematically:

- data has to be centered, i.e., each X_j has mean equal to zero
- transformation vector ϕ_1 has to be normalized, i.e., $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

Principal Component Analysis

Find ϕ_1 by solving optimization problem:

$$\max_{\phi11,\phi_{21},\ldots,\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2$$

$$\mathrm{s.\,t.}\sum_{j=1}^{p}\phi_{j1}^2 = 1$$

Principal Component Analysis

Conceptually: *maximize variance* but *subject to normalization constraint*.

The second transformation ϕ_2 is obtained next solving a similar problem with the added constraint that ϕ_2 **is orthogonal** to ϕ_1.

Principal Component Analysis

Taken together $[\phi_1, \phi_2]$ define a pair of linear transformations of the data into 2 dimensional space.

$$Z_{n\times 2} = X_{n\times p}[\phi_1, \phi_2]_{p\times 2}$$

Principal Component Analysis

Each of the columns of the Z matrix are called *Principal Components*.

The units of the PCs are *meaningless*.

In particular, comparing numbers *across* PCs doesn't make mathematical sense.

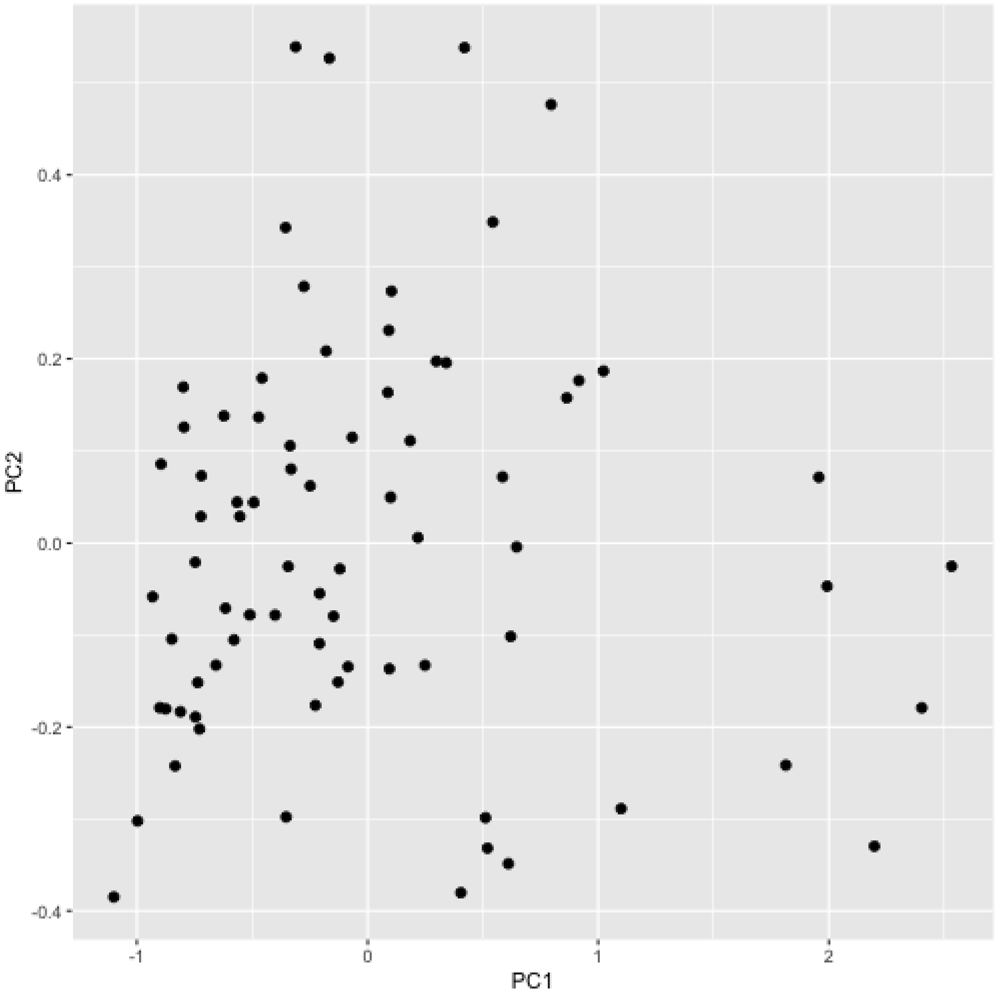
Principal Component Analysis

In practice, may also use a scaling transformation on the variables X_j to have unit variance.

In general, if variables X_j are measured in different units (e.g, miles vs. liters vs. dollars), variables should be scaled to have unit variance.

Conversely, if they are all measured in the same units, they should be scaled.

Principal Component Analysis



Mortgage affordability data embedded into the first two principal components.

Principal Component Analysis

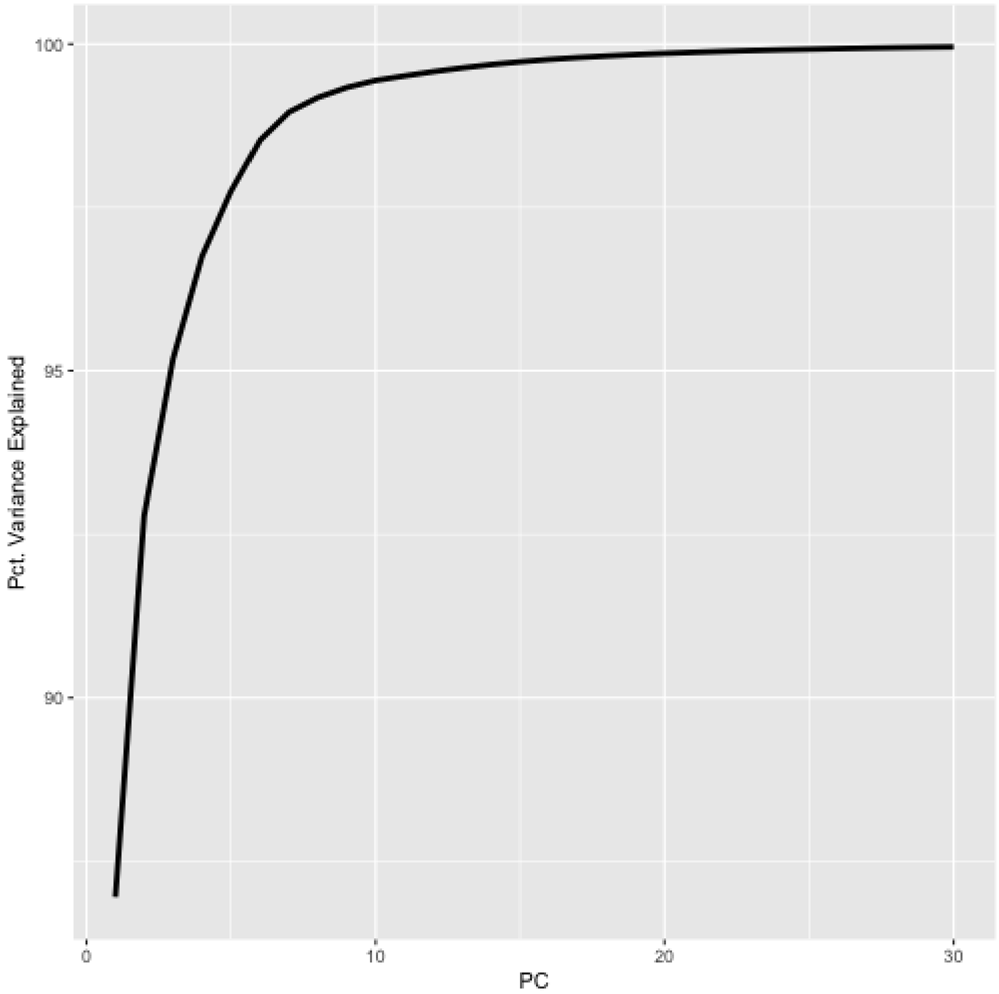
A natural question that arises: How many PCs should we consider in post-hoc analysis?

One result of PCA is a measure of the variance corresponding to each PC relative to the total variance of the dataset.

From that calculate the *percentage of variance explained* for the m-th PC:

$$PVE_m = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

Principal Component Analysis



We can use this measure to choose number of PCs in an ad-hoc manner. In our case, using more than 10 or so PCs does not add information.

Principal Component Analysis

A useful *rule of thumb*:

- If no apparent patterns in first couple of PCs, stop!
- Otherwise, look at other PCs using PVE as guide.

Principal Component Analysis

A useful *rule of thumb*:

- If no apparent patterns in first couple of PCs, stop!
- Otherwise, look at other PCs using PVE as guide.

There are bootstrap based methods to perform a statistically guided selection of the number of PCs.

Principal Component Analysis

A useful *rule of thumb*:

- If no apparent patterns in first couple of PCs, stop!
- Otherwise, look at other PCs using PVE as guide.

There are bootstrap based methods to perform a statistically guided selection of the number of PCs.

However, there is no commonly agreed upon method for choosing number of PCs used in practice, and methods are somewhat ad-hoc.

Solving the PCA

The Principle Component solutions ϕ are obtained from the *singular value decomposition* of observation matrix $X_{n \times p} = UDV^T$

Solving the PCA

The Principle Component solutions ϕ are obtained from the *singular value decomposition* of observation matrix $X_{n \times p} = UDV^T$

Matrices U and V are orthogonal matrices, $U^TU = I$ and $V^TV = I$

Called the left and right *singular vectors* respectively.

Solving the PCA

The Principle Component solutions ϕ are obtained from the *singular value decomposition* of observation matrix $X_{n \times p} = UDV^T$

Matrices U and V are orthogonal matrices, $U^T U = I$ and $V^T V = I$

Called the left and right *singular vectors* respectively.

D is a diagonal matrix with $d_1 \geq d_2 \geq \ldots d_p \geq 0$. These are referred to as the *singular values*.

Solving the PCA

Using our previous notation V is the transformation matrix $V = [\phi_1, \phi_2, \cdots, \phi_p]$.

Principal components Z are given by the columns of UD. Since U is orthogonal, d_j^2 equals the variance of the jth PC.

Solving the PCA

From this observation we also see that we can write original observations x_i in terms of PCs z and transformations ϕ.

Specifically

$$x_i = z_{i1}\phi_1 + z_{i2}\phi_2 + \cdots + z_{ip}\phi_p$$

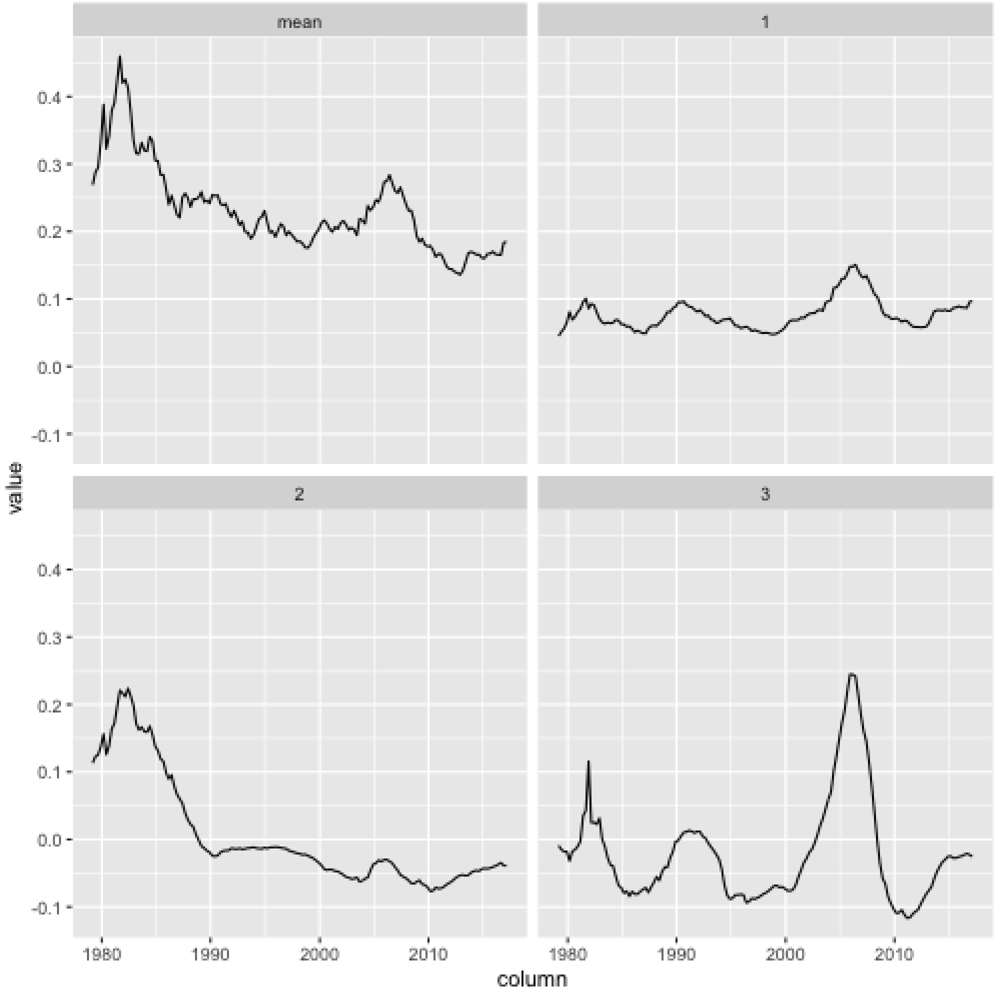
.

Solving the PCA

We can think of the ϕ_j vectors as a basis over which we can represent original observations i.

For this reason, another useful post-hoc analysis is to plot the transformation vectors $\phi_1, \phi_2, \ldots$.

Here we plot the mean time series (since we center observations X before performing the embedding) along with the first three ϕ_j vectors.



Multidimensional Scaling

Multidimensional scaling is a similar approach to PCA but looks at the task in a little different manner.

Given observations $x_1, \ldots, x_N$ in p dimensions, let d_{ij} be the distance between observations i and j. We may also use this algorithm given distances initially instead of p dimensional observations.

Multidimensional Scaling

Multidimensional scaling is a similar approach to PCA but looks at the task in a little different manner.

Given observations $x_1, \ldots, x_N$ in p dimensions, let d_{ij} be the distance between observations i and j. We may also use this algorithm given distances initially instead of p dimensional observations.

Multidimensional Scaling (MDS) seeks to find embeddings $z_1, \ldots, z_N$ of k dimensions for which Euclidean distance (in k dimensional space) is close to the input distances d_{ij}.

Multidimensional Scaling

In *least squares* MDS, we can do this by minimizing

$$S_M(z_1, \ldots, z_N) = \sum_{i \neq j} (d_{ij} - \|z_i - z_j\|)^2$$

A gradient descent algorithm is used to minimize this function.

Multidimensional Scaling

A related method that tends to better capture small distances is given by the *Sammon* mapping:

$$S_{S_m}(z_1, \ldots, z_N) = \sum_{i \neq j} \frac{(d_{ij} - \|z_i - z_j\|)^2}{d_{ij}}$$

Summary

Principal Component Analysis is a conceptually simple but powerful EDA tool. It is very useful at many stages of analyses.

PCA interpretation can be very ad-hoc, however. It is part of large set of unsupervised methods based on *matrix decompositions*, including Kernel PCA, Non-negative Matrix Factorization and others.

Embedding methods seek to capture local properties of observations. Popular recent methods are t-SNE and UMAP.