1 Social Progress Analysis

Tool: Principal Component Analysis

The Analytics Edge: The social progress of nations can be seen as the amalgamation of many features spanning a variety of domains. Providing a measure for social progress can be challenging. We look at a data set of various indicators of social progress and find principal features along which such indicators vary the most with the use of principal component analysis. This aids us in getting a glimpse at how such a social progress index can be computed. We also get to learn one of the useful tools in what is often known as *unsupervised learning.*

1.1 Overview

A recent *New York Times* article discusses the position of the US globally in terms of social progress in light of the 2020 Social Progress Index `https://www.socialprogress.org/`.

The Social Progress Index is a well-established measure, published since 2013, that is meant to catalyze improvement and drive action by presenting social outcome data in a useful and reliable way. Composed of multiple dimensions, the Social Progress Index can be used to benchmark success and provide a holistic, transparent, outcome-based measure of a country's well-being that is independent of economic indicators. Policymakers, businesses, and countries' citizens alike can use it to compare their country against others on different facets of social progress, allowing the identification of specific areas of strength or weakness. (taken from the Methodology summary)

1.2 Available data

The Social Progress Initiative provides a data set of 192 countries over 10 years (2011-2020) with data collected across 54 different indicators form a variety of sources.

Social Progress Index: These indicators are broken into three broad dimensions:

1. Basic human needs
2. Foundations of wellbeing
3. Opportunity

Each dimension is subdivided into four components further, for example, Basic human needs has components Nutrition and Basic Medical care, Water and Sanitation, Shelter, and Personal Safety. Each component comprises of a weighted sum of a disjoint set of the indicators, the weights are determined using PCA. Now the three dimensions are a simple average of each of their four components; and the average of the three components make the *Social Progress Index.*

The dataset we have also contains all components and dimensions along with the various individual indicators.

1.3 Key questions:

1. What can we understand about social progress from the raw data?
2. Can we identify the major differences in social progress occurs?
3. Can we propose some kind of social progress index from our analysis.

1.4 Summary

Data: The data contains variables which are linear combinations of others, moreover there are values which are 'not available'. So we need to prepare the data set for analyses.

Model: Principal component analysis helps us in finding key combination of features which explain maximum variability.

Decision: The model provides weights of different indicators without breaking them into components and dimensions. We still get insight into what are the important areas where the there is a lot of gap in terms of progress. We can use the key principal component to create a ranking as well.

Value: The value of the model is many-fold. We are able to find the key components of variability allowing a nation to address their social progress issues according to these indicators. Morevoer, in a temporal study we may be able to see how different indicators actually become less or more important over time.

2 Principal Component Analysis

2.1 Motivation:

In a large class of data science applications, the user has access to a set of features (independent variables), $\underline{X} = (X_1, \dots, X_p)$ and would like to infer about an output/label Y related to the features. So the problem is to learn/find a function f such that $Y = f(\underline{X})$. Typically one has access to a dataset (training set) to learn about the function f so that given a new input data point $\underline{x}^*$, one can predict the output $y^* = f(\underline{x}^*)$. Algorithms to find/estimate such functions are often termed *supervised learning*. Since the answer is known (in the training set), the learning stops when an acceptable level of precision in performance is achieved.

In contrast, sometimes we have access to some input data/ features $\underline{X} = (X_1, \dots, X_p)$ with no corresponding output. The purpose in such a case is to learn about the structure, distribution, or characteristic of the data. Methods or algorithms to learn in this fashion is often termed *exploratory data analysis* or *unsupervised learning*. The goal often is to discover homogeneous subgroups in the data or important low-dimensional feature sets.

Principal component analysis turns out to be one such unsupervised learning technique.

2.2 Computing Principal Components

2.2.1 Problem set up:

1. n = number of observations;
2. p = number of features;
3. $\underline{X} = (X_1, \dots, X_p)^\top$ = the feature variables;
4. We observe the $n \times p$ data matrix:

$$\boldsymbol{x} = \begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix} =: \begin{bmatrix} \underline{x}_1^\top \\ \vdots \\ \underline{x}_n^\top \end{bmatrix}.$$

   Without loss of generality assume that the sample has mean zero, i.e.,

$$n\bar{x}_j := \sum_{i=1}^{n} x_{ij} = 0, \forall j = 1, \dots, p.$$

5. Denote by S the sample covariance matrix of $\boldsymbol{x}$ given by

$$\mathsf{S} = \frac{1}{n} \sum_{i=1}^{n} \underline{x}_i \underline{x}_i^\top = \frac{1}{n} \boldsymbol{x}^\top \boldsymbol{x}.$$

   Point to ponder: Should we use the divisor n or $n - 1$?

Question: We want a low-dimensional representation of the data. More formally we may ask: can we find

$Z_1, \ldots, Z_q$ with $q << p$ such that

$$\begin{aligned} Z_1 &= \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p = \underline{X}^{\mathsf{T}}\underline{\phi}_1, \\ &\vdots \\ Z_q &= \phi_{1q}X_1 + \phi_{2q}X_2 + \ldots + \phi_{pq}X_p = \underline{X}^{\mathsf{T}}\underline{\phi}_q, \end{aligned}$$

and $\underline{Z} = (Z_1, \ldots, Z_q)^{\mathsf{T}}$ represents $\underline{X}$ 'appropriately'. We also restrict to unit length directions $\underline{\phi}_i$, meaning, $||\underline{\phi}_i|| = \underline{\phi}_i^{\mathsf{T}}\underline{\phi}_i = \sum_{i=1}^{p} \phi_{ji}^2 = 1$.

2.2.2 Linear algebra: eigenvalues, eigenvectors and covariance matrices

First let us note a few things about eigenvalues and eigenvectors of matrices.

1. Any symmetric matrix A has n orthonormal eigenvectors $\underline{v}_1, \ldots, \underline{v}_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$ respectively such that for $1 \leq i, j \leq n$,

   (a) $\mathsf{A}\underline{v}_i = \lambda_i \underline{v}_i$,

   (b) $\underline{v}_i^{\mathsf{T}}\underline{v}_i = 1$,

   (c) $\underline{v}_i^{\mathsf{T}}\underline{v}_j = 0$ for $i \neq j$.

   Moreover, $\underline{v}_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}$.

2. A non-negative definite (symmetric) matrix has all eigenvalues non-negative.

3. For a symmetric matrix A, let $\mathsf{E} = [\underline{v}_1 \ldots \underline{v}_n]$ be the matrix of eigenvectors of A and $\mathsf{D} = \mathsf{diag}(\lambda_1, \ldots, \lambda_\mathsf{n})$ where λ_i is the eigenvalue corresponding to $\underline{v}_i$. Then $\mathsf{A} = \mathsf{EDE}^{\mathsf{T}}$. By convention, the eigenvalues are ordered: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.

4. Moreover in the above scenario, since E is orthonormal $\mathsf{E}^{\mathsf{T}}\mathsf{E} = \mathsf{E}\mathsf{E}^{\mathsf{T}} = \mathsf{I}$ and $\mathsf{E}^{-1} = \mathsf{E}^{\mathsf{T}}$.

2.2.3 Problem approach to find a low-dimensional representation

We consider two approaches here:

1. Project the data points to a hyperplane (of dimension q) which has the smallest average distance from the data cloud.

2. Explain the total variability in the data by breaking up into direction of maximum variance; then choose from among the directions orthogonal to the first one, the one with second maximum variance, so on and so forth.

Both approaches actually lead to the same answer. For computational purposes, it turns out that the second approach is more tractable.

2.2.4 Approach 1: projection onto the closest hyperplane (minimizing residuals)

For the sake of convenience we will carry out the computation when $q = 1$. We want to find the direction $\underline{\phi}_1 = (\phi_{11}, \ldots, \phi_{p1})^T$ such that average error in projecting the data points to this direction is the minimum. Computing mean square errors we have the average error to be:

$$\begin{aligned}\mathsf{MSE}(\underline{\phi}_1) &:= \frac{1}{n}\sum_{i=1}^{n} ||\underline{x}_i - (\underline{x}_i^{\mathsf{T}}\underline{\phi}_1)\underline{\phi}_1||^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{p}(x_{ij} - z_{i1}\phi_{j1})^2\right] \quad \text{where } z_{i1} = \underline{x}_i^{\mathsf{T}}\underline{\phi}_1 \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} z_{i1}^2. \qquad (1)\end{aligned}$$

Since the first term in (1) does not depend on $\underline{\phi}_1$, subject to the constraint that $||\underline{\phi}_1||^2 = 1$,

$$\min_{\underline{\phi}_1} \mathsf{MSE}(\underline{\phi}_1) \Leftrightarrow \max_{\underline{\phi}_1} \frac{1}{n}\sum_{i=1}^{n} z_{i1}^2 = \max_{\underline{\phi}_1} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij}\phi_{j1})^2.$$

Now the sample variance of $Z_1 = \underline{X}^{\mathsf{T}}\underline{\phi}_1$ is

$$\mathsf{V}(\underline{\phi}_1) = \frac{1}{n}\sum_{i=1}^{n}(z_{i1} - \bar{z}_1)^2 = \frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$$

since

$$\bar{z}_1 := \frac{1}{n}\sum_{i=1}^{n} z_{i1} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}\phi_{j1} = \sum_{j=1}^{p}\phi_{j1}\bar{x}_j = 0.$$

Hence finding unit direction $\underline{\phi}_1$ with minimum $\mathsf{MSE}(\underline{\phi}_1)$ is equivalent to finding *weight/loading/rotation* $\underline{\phi}_1$ which maximizes sample variance of $Z_1 = \underline{X}^{\mathsf{T}}\underline{\phi}_1$ given by $\mathsf{V}(\underline{\phi}_1)$ subject to $||\underline{\phi}_1||^2 = 1$.

2.2.5 Approach 2: direction of maximum variance

To begin with, we look at the case when $q = 1$. The idea was already addressed in *Approach 1*. First note that the covariance matrix of the data set $\boldsymbol{x}$ is given by the $p \times p$ matrix $\mathsf{S} = \frac{1}{n}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}$. Now observe that

$$\mathsf{V}(\underline{\phi}_1) = \frac{1}{n}\sum_{i=1}^{n} z_{i1}^2 = \frac{1}{n}\underline{\phi}_1^{\mathsf{T}}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}\underline{\phi}_1 = \underline{\phi}_1^{\mathsf{T}}\mathsf{S}\underline{\phi}_1$$

which we intend to maximize w.r.t. $\underline{\phi}_1$ such that $\underline{\phi}_1^{\mathsf{T}}\underline{\phi}_1 = 1$. Using Lagrange multiplier λ, we need to optimize

$$\mathcal{L}(\underline{\phi}_1, \lambda) = \underline{\phi}_1^{\mathsf{T}}\mathsf{S}\underline{\phi}_1 - \lambda(\underline{\phi}_1^{\mathsf{T}}\underline{\phi}_1 - 1)$$

Taking partial derivatives, we have

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\lambda} &= \underline{\phi}_1^{\mathsf{T}}\underline{\phi}_1 - 1, \\ \frac{\partial\mathcal{L}}{\partial\underline{\phi}_1} &= 2\mathsf{S}\underline{\phi}_1 - 2\lambda\underline{\phi}_1.\end{aligned}$$

Now by equating the above to zero we get

$$\underline{\phi}_1^{\mathsf{T}}\underline{\phi}_1 = 1, \tag{2}$$

$$\mathsf{S}\underline{\phi}_1 = \lambda\underline{\phi}_1. \tag{3}$$

Note that, for the sample covariance matrix S, the following hold:

1. S is a symmetric non-negative definite matrix. Hence all eigenvalues are non-negative.
2. We can write $\mathsf{S} = \mathsf{PDP}^{\mathsf{T}}$ where P has the eigenvectors of S as its columns and the associated eigenvalues $\mathsf{D} = \mathsf{diag}(\lambda_1, \ldots, \lambda_\mathsf{p})$ are such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$.

Hence (2), (3) implies that λ is an eigenvalue of S and $\underline{\phi}_1$ is its associated eigenvector. Since we have the decomposition $\mathsf{S} = \mathsf{PDP}^{\mathsf{T}}$, choosing the eigenvector corresponding to the maximum eigenvalue λ_1 as $\underline{\phi}_1$ we get $\mathsf{S}\underline{\phi}_1 = \lambda_1\underline{\phi}_1$. Hence

$$\mathsf{V}(\underline{\phi}_1) = \underline{\phi}_1^{\mathsf{T}}\mathsf{S}\underline{\phi}_1 = \lambda\underline{\phi}_1^{\mathsf{T}}\underline{\phi}_1 = \lambda_1.$$

Now the next step is to find a direction $\underline{\phi}_2$ such that the sample variance of $Z_2 = \underline{\phi}_2^{\mathsf{T}}\underline{X}$ is maximized among all directions orthogonal to $\underline{\phi}_1$. That is we need to find a solution to:

$$\max_{\underline{\phi}_2} \mathsf{V}(\underline{\phi}_2) = \max_{\underline{\phi}_2} \underline{\phi}_1^{\mathsf{T}}\mathsf{S}\underline{\phi}_1$$

subject to $\underline{\phi}_2^{\mathsf{T}}\underline{\phi}_2 = 1$ and $\underline{\phi}_2 \perp \underline{\phi}_1$. We can check that the solution is the eigenvector corresponding to the second largest eigenvalue λ_2 so that $S\underline{\phi}_2 = \lambda_2\underline{\phi}_2$ and $\mathsf{V}(\phi_2) = \lambda_2$. We can continue like this.

Thus, the first principal component is the eigenvector corresponding to the largest eigenvalue $\underline{\phi}_1$ of S given by λ_1, the second principal component is the eigenvector $\underline{\phi}_2$ corresponding to the second largest eigenvalue of S given by λ_2, so and and so forth until the q^{th} eigenvector for $q \leq p$. Naturally the variance explained by the first q principal components is given by $\sum_{i=1}^{q}\lambda_i$.

2.2.6 How do we choose the number of principal components?

The total variance in the data set is

$$\begin{aligned}\mathsf{V}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2 &= \mathsf{Tr}(\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}/\mathsf{n}) \\ &= \mathsf{Tr}(\mathsf{S}) = \mathsf{Tr}(\mathsf{PDP}^{\mathsf{T}}) \\ &= \mathsf{Tr}(\mathsf{P}^{\mathsf{T}}\mathsf{PD}) = \mathsf{Tr}(\mathsf{D}) = \sum_{i=1}^{p}\lambda_i.\end{aligned}$$

The variance explained by first q principal components

$$\mathsf{V}(\phi_1) + \ldots \mathsf{V}(\phi_q) = \lambda_1 + \ldots + \lambda_q.$$

We define a quantity R^2 of the projection which explains the proportion of variance explained by the first q components:

$$R^2 = \frac{\sum^q \lambda_i}{\sum^p \lambda_i}$$

Depending on a threshold chosen by the user the number of principal components can be determined. In practice this is done using a scree plot.

In practice we observe that the different attributes have different variances and hence a PCA gets influenced the the variable with largest variance. Hence it is customary to standardize all variables to have unit (or equal variance). In all variances are equal then the total variance $\mathsf{V}(\boldsymbol{x}) \sum^p \lambda_i = p$. In such a case, it is often recommended that the principal components are chosen for eigenvalues with value greater than 1, since they explain more than one variable of the entire set.

2.2.7 Limitations

Albeit being quite useful, we need to keep in mind certain limitations.

- PCA depends on the scaling of the variables and hence one needs to scale the variables, essentially bringing them all down to the same scale.
- We assume that a linear relationships between variables can explain the total variability.
- The eventual principal components loses some interpretability.

3 Principal Component Analysis in R: some notes

1. Two functions are generally used `prcomp` and `princomp`. We use the first one.
2. The output, say `pr.out`$<-$`prcomp(data)` stores a few values
   - `pr.out$center` and `pr.out$scale` are means and standard deviations of the data which are used to standardize the data.
   - `pr.out$sdev` are standard deviations of the principal components (square root of the eigen values of the S matrix in Section 2.2.5)
   - `pr.out$rotation` are the attribute loadings (eigen-vectors) and `pr.out$x` are the individual scores according to the principal components.
3. A Scree plot can be can be useful and `plot(pr.out)` creates such a plot.
4. We tend to plot also the cumulative proportion of variance explained by the principal components as discussed in 2.2.6.
5. We use `library(factoextra)` to do some colorful plots.