

Lecture 26: PCA (cont.)

Big Data and Machine Learning for Applied Economics
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

November 11, 2021

Agenda

- 1 Factor Models
- 2 PCA: Theory
 - PC Computation
- 3 Factor Interpretation
- 4 Review & Next Steps
- 5 Further Readings

Factor Models

- ▶ Text is super high dimensional
- ▶ There is often abundant *unlabeled* text
- ▶ Some times unsupervised factor model is a popular and useful strategy with text data
- ▶ You can first fit a factor model to a giant corpus, get a few factors (reduce the dimensionality)
- ▶ Use these factors for supervised learning on a subset of labeled documents.
- ▶ The unsupervised dimension reduction facilitates the supervised learning

Topic Models: Example

- ▶ We have 6,166 reviews, with an average length of 90 words per review, [we8there.com](#).
- ▶ A useful feature of these reviews is that they contain both text and a multidimensional rating on overall experience, atmosphere, food, service, and value.
- ▶ For example, one user submitted a glowing review for Waffle House #1258 in Bossier City, Louisiana:

I normally would not review a Waffle House but this one deserves it. The workers, Amanda, Amy, Cherry, James and J.D. were the most pleasant crew I have seen. While it was only lunch, B.L.T. and chili, it was great. The best thing was the 50's rock and roll music, not too loud not too soft. This is a rare exception to what you all think a Waffle House is. Keep up the good work.

Overall: 5, Atmosphere: 5, Food: 5, Service: 5, Value: 5.

Topic Models: Example

- ▶ We can apply PCA to get a factor representation of the review text.
- ▶ PC1 looks like it will be big and positive for positive reviews,

```
pca <- prcomp(x, scale=TRUE) # can take a long time
```

```
tail(sort(pca$rotation[,1]))
```

```
##      food great      staff veri      excel food high recommend      great food
## 0.007386860 0.007593374 0.007629771 0.007821171 0.008503594
##      food excel
## 0.008736181
```

- ▶ while PC4 will be big and negative

```
tail(sort(pca$rotation[,4]))
```

```
##      order got after minut      never came      ask check readi order drink order
## 0.05918712 0.05958572 0.06099509 0.06184512 0.06776281 0.07980788
```

Factor Models

- ▶ Lets assume that we have a data matrix $X_{n \times p}$
- ▶ A factor model looks like

$$\begin{aligned} X_1 &= \phi_{11}f_1 + \cdots + \phi_{1k}f_k \\ X_2 &= \phi_{21}f_1 + \cdots + \phi_{2k}f_k \\ &\vdots \\ X_p &= \phi_{p1}f_1 + \cdots + \phi_{pk}f_k \end{aligned} \tag{1}$$

- ▶ where
 - ▶ X_j are the inputs of the regressions (independent vars)
 - ▶ The f_k $k = 1, \dots, K$ are the factors, unobserved, and that we want to estimate
 - ▶ ϕ_{jk} are called loadings or rotations—these
 - ▶ When you use a K that is much smaller than p , factor models provide a parsimonious representation for X .

Factor Models: PCA

- ▶ How do you estimate a Factor Model with PCA?
- ▶ We are trying to learn from high-dimensional X some low-dimensional summaries that contain the information necessary to make good decisions.

Factor Models: PCA

- ▶ How do you estimate a Factor Model with PCA?
- ▶ We are trying to learn from high-dimensional X some low-dimensional summaries that contain the information necessary to make good decisions.
- ▶ Suppose that there is only one underlying factor f_1

$$\begin{aligned}x_1 &= \phi_{11}f_1 \\x_2 &= \phi_{21}f_1 \\&\vdots \\x_p &= \phi_{p1}f_1\end{aligned}\tag{2}$$

Factor Models: PCA

- ▶ or in a more compact form

$$X = \phi_1 f_1 \quad (3)$$

- ▶ and

$$f_1 = \phi_1^{-1} X \quad (4)$$

- ▶ so we don't have to deal with inverses (things do not change), let's call $\delta_1 = \phi_1^{-1}$

$$f_1 = \delta_1 X \quad (5)$$

Detour: Algebra Review

- ▶ Let $A_{m \times m}$. It exists
 - ▶ a scalar λ such that $Ap = \lambda p$ for a vector $p_{m \times 1}$,
 - ▶ if $p \neq 0$, then λ is an eigenvalue of A .
 - ▶ and p is an eigenvector of A corresponding to the eigenvalue λ .
- ▶ $A_{m \times m}$ with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$, then:

$$\text{tr}(A) = \sum_{i=1}^m \lambda_i \quad (6)$$

$$\det(A) = \prod_{i=1}^m \lambda_i \quad (7)$$

- ▶ If $A_{m \times m}$ has m different eigenvalues, then the associated eigenvectors are all linearly independent.

Detour: Algebra Review

- Spectral decomposition:

$$A = P\Lambda P \quad (8)$$

- where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$ and P is the matrix whose columns are the corresponding eigenvectors.

$$A = \begin{pmatrix} p_1 & p_2 & \dots & \dots & p_m \end{pmatrix} \begin{pmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & \lambda_m \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ \vdots \\ p_m \end{pmatrix} \quad (9)$$

$$A = \sum_{i=1}^m \lambda_i p_i p_i' \quad (10)$$

Principal Component Analysis

$$f_1 = \delta_1 X \quad (11)$$

- ▶ The task is then finding the best linear combination of the original variables.
- ▶ What is best?

Principal Component Analysis

$$f_1 = \delta_1 X \quad (11)$$

- ▶ The task is then finding the best linear combination of the original variables.
- ▶ What is best?
- ▶ PCA response: the one that preserves the most information
- ▶ In other words, we are going to try to generate an index that reproduces (the best it can) the information (variability) of the original variables
- ▶ How we do that?

Principal Component Analysis

$$f_1 = \delta_1 X \quad (11)$$

- ▶ The task is then finding the best linear combination of the original variables.
- ▶ What is best?
- ▶ PCA response: the one that preserves the most information
- ▶ In other words, we are going to try to generate an index that reproduces (the best it can) the information (variability) of the original variables
- ▶ How we do that?
- ▶ Maximize the variance

Principal Component Analysis

- ▶ The problem then looks like

$$\max V(f_1) = \max V(\delta_1 X) \quad (12)$$

- ▶ where

- ▶ $X = (x_1, \dots, x_K)_{N \times K}$,
- ▶ $S = V(X)$
- ▶ $\delta_1 \in K$

- ▶ Let's set up the problem as

$$\max_{\delta} \delta_1 X \delta_1' \quad (13)$$

- ▶ What is the solution to this problem?

Principal Component Analysis

- ▶ The problem then looks like

$$\max V(f_1) = \max V(\delta_1 X) \quad (12)$$

- ▶ where

- ▶ $X = (x_1, \dots, x_K)_{N \times K}$,
- ▶ $S = V(X)$
- ▶ $\delta_1 \in K$

- ▶ Let's set up the problem as

$$\max_{\delta} \delta_1 X \delta_1' \quad (13)$$

- ▶ What is the solution to this problem?
- ▶ Bring δ to infinity.

Principal Component Analysis

- ▶ Let's "fix" the problem by normalizing δ

$$\begin{aligned} \max_{\delta} \quad & \delta_1 S \delta_1' \\ \text{subject to} \quad & \delta_1 \delta_1' = 1 \end{aligned} \tag{14}$$

- ▶ Let us call the solution to this problem δ_1^* .
- ▶ $f_1^* = \delta_1^* X$ is the 'best' linear combination of X .
- ▶ Intuition: X has K columns and $f_1^* = \delta_1^* X$ has only one. The factor built with the first principal component is the best way to represent the K variables of X using a single single variable.

Principal Component Analysis

- ▶ Solution to the problem of the first principal component
- ▶ Let's set the lagrangian

$$\mathcal{L} = \delta_1 S \delta'_1 + \lambda_1 (1 - \delta_1 \delta'_1) \quad (15)$$

- ▶ Rearranging

$$S \delta'_1 = \lambda_1 \delta'_1 \quad (16)$$

- ▶ At the optimum, δ is the eigenvector corresponding to the eigenvalue λ of S .
- ▶ Premultiplying by δ_1 and remembering that $\delta_1 \delta'_1 = 1$:

Principal Component Analysis

$$\delta_1 S \delta_1' = \lambda_1 \quad (17)$$

- ▶ In order to maximize $\delta S \delta$ we must choose λ equal to the maximum eigenvalue of S and δ is the corresponding eigenvector.
- ▶ The problem of finding the best linear combination that reproduces the variability of X is finding the biggest eigenvalue of S and its corresponding eigenvector

Principal Component Analysis

- ▶ The first main component? Are there others?
- ▶ Let's consider the following problem:

$$\max_{\delta_2} \delta_2 S \delta_2' \quad (18)$$

$$\text{st} \quad (19)$$

$$\delta_2 \delta_2' = 1 \quad (20)$$

$$\delta_2 \delta_1' = 0 \quad (21)$$

- ▶ $f_2^* = \delta_2^* X$ is the second principal component : the best linear combination which is orthogonal to the best initial linear combination.
- ▶ Recursively, using this logic you can form q main components.
- ▶ Note that algebraically we could construct $q = K$ factors, actually the number of PC are $\min(n - 1, K)$

q main components

- ▶ Let $\lambda_1, \dots, \lambda_K$ be the eigenvalues of $S = V(X)$, ordered from highest to lowest,
- ▶ p_1, \dots, p_K the corresponding eigenvectors.
- ▶ Call P the matrix of eigenvectors.
- ▶ Then $\delta_j = p_j$, $\forall j$ ('loadings' of the principal components = ordered eigenvectors of S).

Relative importance of factors

- ▶ Now we want to know the relative importance of factors, to have a way of choosing them
- ▶ Let $f_j = X\delta_j$, $j = 1, \dots, K$ be the j -th principal component.

$$V(f_j) = \delta_j S \delta_j' \quad (22)$$

$$= p_j P \Lambda P p_j' \quad (23)$$

$$= \lambda_j \quad (24)$$

(the variance of the j -th principal component is the j -th ordered eigenvalue of S).

- ▶ We this result we can show that the total variance of X is the sum of the variances of x_j , $j = 1, \dots, K$, that is $\text{trace}(S)$

Relative importance of factors

- ▶ We the above result we can show that the total variance of X is the sum of the variances of x_j , $j = 1, \dots, K$, that is $trace(S)$
- ▶ Note the following:

$$trace(S) = trace(P\Lambda P') = trace(PP'\Lambda) = \sum_{j=1}^K \lambda_j = \sum_{j=1}^K V(F_j) \quad (25)$$

- ▶ Then

$$\frac{\lambda_k}{\sum_{j=1}^K \lambda_j} \quad (26)$$

- ▶ measures the relative importance of the j th principal component.

Selection of factors

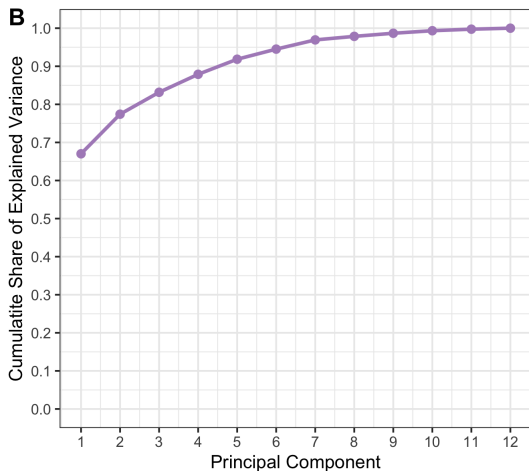
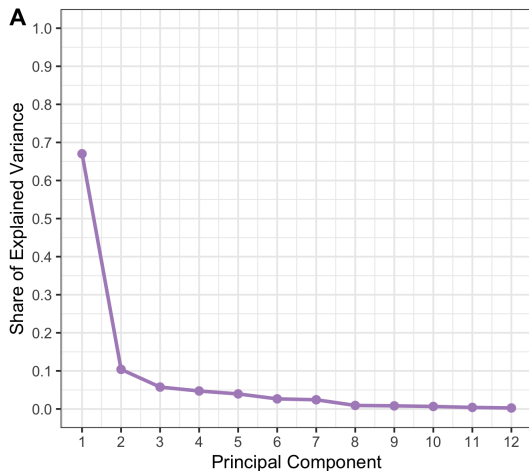
- ▶ Although a matrix X of dimension n times K generally has $\min(n - 1, K)$ different principal components.
- ▶ In practice, we are generally not interested in all the components, but rather stay with the first ones that allow us to visualize or interpret data.
- ▶ Indeed, we would like to keep the minimum number that allows us a good understanding of the data.
- ▶ The natural question that arises here is whether there is an established way to determine the number of principal components to use.
- ▶ Unfortunately, there is no accepted objective way in the literature to answer it.

Selection of factors

- ▶ However, there are three simple approaches that can guide you in deciding the number of relevant major components.
 - ▶ Visual examination of screeplot
 - ▶ Kaiser criterion.
 - ▶ Proportion of variance explained.

Selection of factors

Screplot



Selection of factors

Kaiser criterion

- ▶ Let the columns of X be standardized, so that each variable has unit variance.
- ▶ In this case:

$$\text{trace}(S) = \sum_{j=1}^K V(F_j) = K \quad (27)$$

- ▶ and recall $\sum_{j=1}^K \lambda_j = \sum_{j=1}^K V(F_j)$ then

$$\sum_{j=1}^K \lambda_j = K \quad (28)$$

- ▶ On average, each factor contributes one unit. When $\lambda_j > 1$, that factor it explains the total variance more than the average. → Retain the factors with $\lambda_j > 1$

Selection of factors

Proportion of variance explained

- ▶ Another approach often used in practice is to impose a threshold a priori and choose the main components based on it.
 - ▶ For example, we could define a threshold of 90%, which in the previous example plot would result in 5 main components.
 - ▶ Whereas if it were 70% we would have 2 main components.
- ▶ The threshold to be defined will depend on the application, the context, and the data set. Thresholds between 70% and 90% are typically used.

PC Computation

- ▶ Before I mentioned that data was standardized, that is, re-centered to have zero mean and scaled to have variance one.
- ▶ From a strictly mathematical point of view, there is nothing inherently wrong with making linear combinations of variables with different units of measurement.
- ▶ However, when we use PCA we seek to maximize variance and the variance is affected by the units of measurement.
- ▶ This implies that the principal components based on the covariance matrix S will change if the units of measure of one or more variables change.

PC Computation

- ▶ To prevent this from happening, it is common practice to standardize the variables. That is, each X value is re-centered and divided by the standard deviation:

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \quad (29)$$

- ▶ where \bar{x}_j is the mean and s_j is the standard deviation of column j .
- ▶ Then the initial data matrix X is replaced by the standardized data matrix Z .
- ▶ Note also that when standardizing the data matrix, the covariance matrix S is simply the original data correlation matrix. This is sometimes referred to in the literature as the PCA correlation matrix.

PC Computation

Uniqueness of the main components

- ▶ It is necessary to warn that the "loadings" of the main components δ are unique except for a sign change.
- ▶ This implies that depending on the implementation we can obtain different results in two libraries.
- ▶ The "loadings" will be the same but the signs may differ.
- ▶ The signs may differ because each weight specifies a direction in k -dimensional space and the change of sign has no effect on the direction.

PC Computation

- ▶ As a practical aside, note that `prcomp` converts X here from sparse to dense matrix storage.
- ▶ For really big text DTMs, which will be very sparse, this will cause you to run out of memory.
- ▶ A big data strategy for PCA is to first calculate the covariance matrix for X and then obtain PC rotations as the eigenvalues of this covariance matrix.

- ▶ The first step can be done using sparse matrix algebra.

- ▶ The rotations are then available as

```
## eigen( xvar, symmetric = TRUE)$vec.
```

- ▶ There are also approximate PCA algorithms available for fast factorization on big data. See, for example, the `irlba` package for R.

Factor Interpretation

- ▶ $f_s = X\delta'_s$: 'loadings' often suggest that a factor works as a 'index' of a group of variables.
- ▶ Idea: look at the 'loadings'
- ▶ Caution: factors via principal components are orthogonal recursively.

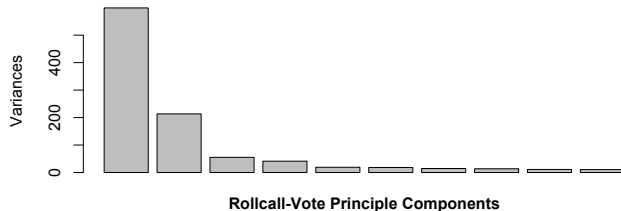
Factor Interpretation: Example

► Congress and Roll Call Voting

- Votes in which names and positions are recorded are called 'roll calls'.
- The site `voteview.com` archives vote records and the R package `pscl` has tools for this data.
- 445 members in the last US House (the 111th)
- 1647 votes: `nea = -1`, `yea=+1`, `missing = 0`.
- This leads to a large matrix of observations that can probably be reduced to simple factors (party).

Factor Interpretation

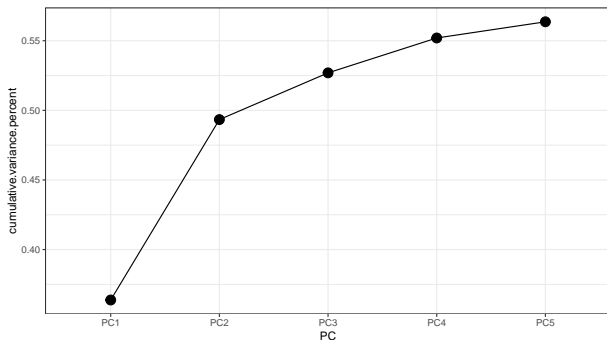
- ▶ Vote components in the 111th house
- ▶ Each PC is $f_s = X\delta'_s$



- ▶ Huge drop in variance from 1st to 2nd and 2nd to 3rd PC.
- ▶ Poli-Sci holds that PC1 is usually enough to explain congress. 2nd component has been important twice: 1860's and 1960's.

Factor Interpretation

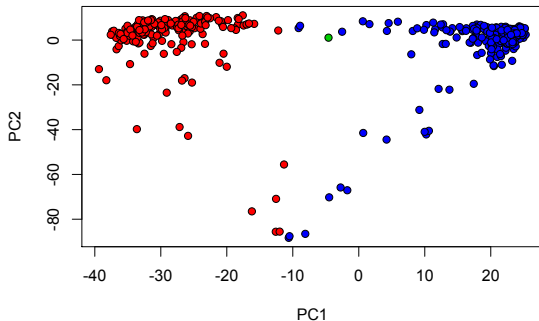
- ▶ Vote components in the 111th house
- ▶ Each PC is $F_s = X\delta_s$



- ▶ Huge drop in variance from 1st to 2nd and 2nd to 3rd PC.
- ▶ Poli-Sci holds that PC1 is usually enough to explain congress. 2nd component has been important twice: 1860's and 1960's.

Factor Interpretation

- ▶ Top two PC directions in the **111th** house



- ▶ Republicans in red and Democrats in blue:
 - ▶ Clear separation on the first principal component.
 - ▶ The second component looks orthogonal to party.

Factor Interpretation

Far right (very conservative)

```
> sort(votepc[,1])  
      BROUN (R GA-10)      FLAKE (R AZ-6)      HENSARLIN (R TX-5)  
      -39.3739409      -38.2506713      -37.5870597
```

Far left (very liberal)

```
> sort(votepc[,1], decreasing=TRUE)  
      EDWARDS (D MD-4)      PRICE (D NC-4)      MATSUI (D CA-5)  
      25.2915083      25.1591151      25.1248117
```

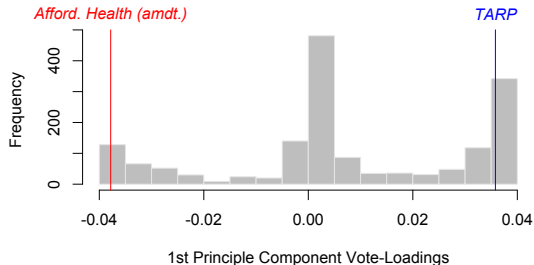
social issues? immigration? no clear pattern

```
> sort(votepc[,2])  
      SOLIS (D CA-32) GILLIBRAND (D NY-20)      PELOSI (D CA-8)  
      -88.31350926      -87.58871687      -86.53585568  
      STUTZMAN (R IN-3)      REED (R NY-29)      GRAVES (R GA-9)  
      -85.59217310      -85.53636319      -76.49658108
```

- PC1 is easy to read, PC2 is ambiguous (is it even meaningful?)

Factor Interpretation

- ▶ **High PC1-loading votes are ideological battles.**
- ▶ These tend to have informative voting across party lines.



- ▶ A vote for Republican amendments to 'Affordable Health Care for America' strongly indicates a negative PC1 (more conservative), while a vote for Troubled Asset Relief Program (TARP) indicates a positive PC1 (more progressive).

Factor Interpretation

- ▶ Look at the largest loadings in δ_2 to discern an interpretation.

```
> loadings[order(abs(loadings[,2]), decreasing=TRUE)[1:5],2]
Vote.1146  Vote.658  Vote.1090  Vote.1104  Vote.1149
0.05605862 0.05461947 0.05300806 0.05168382 0.05155729
```

- ▶ These votes all correspond to near-unanimous symbolic action.

- ▶ For example, 429 legislators voted for resolution 1146:
'Supporting the goals and ideals of a Cold War Veterans Day'
If you didn't vote for this, you weren't in the house.

- ▶ **Mystery Solved:** the second PC is just attendance!

```
> sort(rowSums(votes==0), decreasing=TRUE)
      SOLIS (D CA-32)  GILLIBRAND (D NY-20)      REED (R NY-29)
                1628                1619                1562
STUTZMAN (R IN-3)    PELOSI (D CA-8)    GRAVES (R GA-9)
                1557                1541                1340
```


Principal Component Regression

- ▶ The concept is very simple: instead of regressing onto X , use a lower dimension set of principal components f_s as covariates.
- ▶ This works well for a few reasons:
 - ▶ PCA reduces dimension, which is always good.
 - ▶ Higher variance covariates are good in regression, and we choose the top PCs to have highest variance.
 - ▶ The PCs are independent: no multicollinearity.
- ▶ The 2-stage algorithm is straightforward. For example,

```
mypca = prcomp(X, scale=TRUE)
z = predict(mypca)[,1:K]
reg = glm(y~., data=as.data.frame(z))
```

Review & Next Steps

- ▶ Factor Models
- ▶ PCA Theory
- ▶ PC Computation
- ▶ Factor Interpretation
- ▶ Next class: More on PC regression and LDA
- ▶ Questions? Questions about software?

Further Readings

- ▶ Ahumada, H. A., Gabrielli, M. F., Herrera Gomez, M. H., & Sosa Escudero, W. (2018). Una nueva econometría: Automatización, big data, econometría espacial y estructural.
- ▶ Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for machine learning. Cambridge University Press.
- ▶ James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- ▶ Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
- ▶ Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.
- ▶ Peña, D. (2002). Análisis de datos multivariantes (Vol. 24). Madrid:McGraw-hill.