Gov 2018: Unsupervised Learning

Lecture 6: Dimension Reduction

Naijia Liu

Harvard University

March 6th, 2024

Unsupervised Learning

Dimension reduction:

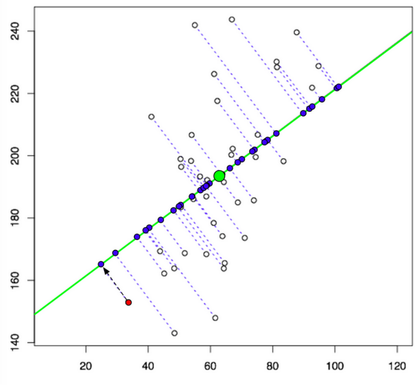
- A particular form of unsupervised learning
- Take high-dimensional features and create a lower-dimensional representation
- Useful for:
  - Visualizing high-dimensional data
  - Preprocessing features for methods that perform poorly in high dimensions (e.g. kNN)
  - Discovering latent concepts underlying the data
  - Combining multiple noisy measurements
- Start with principal component analysis (PCA) and then explore related methods

Finding a Lower Dimensional Representation

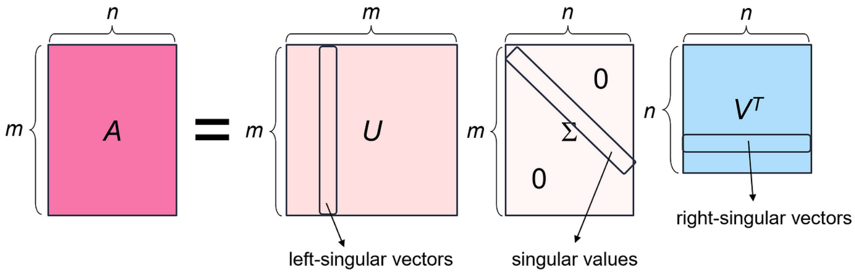
$$\underbrace{\mathbf{X}}_{N\times 2} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{pmatrix} \qquad \underbrace{\mathbf{Z}}_{N\times 1} = \begin{pmatrix} z_{11} \\ z_{21} \\ \vdots \\ z_{N1} \end{pmatrix}$$

Finding a Lower Dimensional Representation

$$\underbrace{\mathbf{X}}_{N\times 2} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{pmatrix} \qquad \underbrace{\mathbf{Z}}_{N\times 1} = \begin{pmatrix} z_{11} \\ z_{21} \\ \vdots \\ z_{N1} \end{pmatrix}$$



Decompose a High Dimensional Matrix



SVD re-expresses a $N \times K$ matrix $\mathbf{X}$ in the following form:

$$\mathbf{X} = \mathbf{UDV}^\top$$

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some constant λ.

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

for some constant λ.

- We want to transform the original matrix by Eigenvector x: $\mathbf{A}\mathbf{x}$.

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some constant λ.

- We want to transform the original matrix by Eigenvector x: $\mathbf{A}\mathbf{x}$.
- This transforming vector is robust after the transformation: $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some constant λ.

- We want to transform the original matrix by Eigenvector x: $\mathbf{A}\mathbf{x}$.
- This transforming vector is robust after the transformation: $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$
- Eignvalue λ tells us the magnitude.

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some constant λ.

- We want to transform the original matrix by Eigenvector x: $\mathbf{A}\mathbf{x}$.
- This transforming vector is robust after the transformation: $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$
- Eignvalue λ tells us the magnitude.
- $\mathbf{x}$ and λ are not unique for most of the matrices.

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some constant λ.

- We want to transform the original matrix by Eigenvector x: $\mathbf{A}\mathbf{x}$.
- This transforming vector is robust after the transformation: $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$
- Eignvalue λ tells us the magnitude.
- $\mathbf{x}$ and λ are not unique for most of the matrices.

Review: Eigenvector Decomposition

For a diagonalizable $N \times N$ matrix, $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is any vector $\mathbf{x}$ that satisfies

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

for some constant λ.

- We want to transform the original matrix by Eigenvector x: $\mathbf{A}\mathbf{x}$.
- This transforming vector is robust after the transformation: $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$
- Eignvalue λ tells us the magnitude.
- $\mathbf{x}$ and λ are not unique for most of the matrices.

It turns out that a $\mathbf{A}$ can be rewritten as $\mathbf{V}\mathbf{D}\mathbf{V}^\top$

Principal Component Analysis: SVD Perspective

- Take the $N \times K$ feature matrix, $\mathbf{X}$

Principal Component Analysis: SVD Perspective

- Take the $N \times K$ feature matrix, $\mathbf{X}$
- Now standardize the columns to create $\widetilde{\mathbf{X}}$, where $\widetilde{\mathbf{X}}_{*,k} = (X_k - \text{mean}(X_k))/\text{s.d.}(X_k)$

Principal Component Analysis: SVD Perspective

- Take the $N \times K$ feature matrix, $\mathbf{X}$
- Now standardize the columns to create $\widetilde{\mathbf{X}}$, where $\widetilde{\mathbf{X}}_{*,k} = (X_k - \text{mean}(X_k))/\text{s.d.}(X_k)$
- We can take the singular value decomposition of $\widetilde{\mathbf{X}} = \mathbf{UDV}^T$

PCA: SVD Perspective

- How should we interpret $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$?

PCA: SVD Perspective

- How should we interpret **U**, **D**, and **V**?
- After SVD of $\widetilde{\mathbf{X}}$, each row of **U** (a $N \times K$ matrix) descibes an observation's *score*, or position in a transformed space

PCA: SVD Perspective

- How should we interpret $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$?
- After SVD of $\widetilde{\mathbf{X}}$, each row of $\mathbf{U}$ (a $N \times K$ matrix) descibes an observation's *score*, or position in a transformed space
- Note that the ordering of the transformed dimensions is arbitrary; we can still recover $\widetilde{\mathbf{X}}$ no matter how they are shuffled

PCA: SVD Perspective

- Variance:

PCA: SVD Perspective

- Variance:
  - The "total" variance is $\sum_k \mathrm{Var}(\widetilde{\mathbf{X}}_{*,k})$

PCA: SVD Perspective

- Variance:
  - The "total" variance is $\sum_k \mathrm{Var}(\widetilde{\mathbf{X}}_{*,k})$
  - The sum of diagonal elements in $\mathbf{D}$ will be $1/N$ the total variance

PCA: SVD Perspective

- Variance:
  - The "total" variance is $\sum_k \text{Var}(\widetilde{\mathbf{X}}_{*,k})$
  - The sum of diagonal elements in $\mathbf{D}$ will be $1/N$ the total variance
  - By convention, the dimensions of the transformed space are ordered according to the variance of $\widetilde{\mathbf{X}}$ that they "explain", corresponding to diagonal elements of $\mathbf{D}$

PCA: SVD Perspective

- Variance:
  - The "total" variance is $\sum_k \text{Var}(\widetilde{\mathbf{X}}_{*,k})$
  - The sum of diagonal elements in $\mathbf{D}$ will be $1/N$ the total variance
  - By convention, the dimensions of the transformed space are ordered according to the variance of $\widetilde{\mathbf{X}}$ that they "explain", corresponding to diagonal elements of $\mathbf{D}$
- Columns of $\mathbf{V}$ are also called the "loadings" of $\widetilde{\mathbf{X}}$ and describe how the transformed space can be mapped back to the feature space

PCA: SVD Perspective

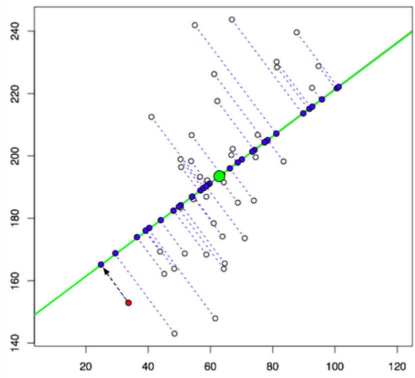
- Variance:
  - The "total" variance is $\sum_k \text{Var}(\widetilde{\mathbf{X}}_{*,k})$
  - The sum of diagonal elements in $\mathbf{D}$ will be $1/N$ the total variance
  - By convention, the dimensions of the transformed space are ordered according to the variance of $\widetilde{\mathbf{X}}$ that they "explain", corresponding to diagonal elements of $\mathbf{D}$
- Columns of $\mathbf{V}$ are also called the "loadings" of $\widetilde{\mathbf{X}}$ and describe how the transformed space can be mapped back to the feature space
- Dimension reduction is achieved by truncating to the first M components (recall how they're ordered)

PCA: Geometric Perspective

$$\underbrace{\mathbf{X}}_{N\times 2} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{pmatrix} \qquad \underbrace{\mathbf{Z}}_{N\times 1} = \begin{pmatrix} z_{11} \\ z_{21} \\ \vdots \\ z_{N1} \end{pmatrix}$$

PCA: Geometric Perspective

$$\underbrace{\mathbf{X}}_{N\times 2} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{pmatrix} \qquad \underbrace{\mathbf{Z}}_{N\times 1} = \begin{pmatrix} z_{11} \\ z_{21} \\ \vdots \\ z_{N1} \end{pmatrix}$$



PCA: Geometric Perspective

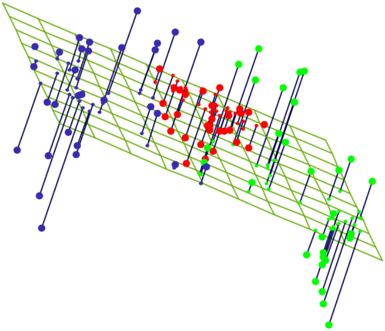
The best 2-dimensional representation is the plane that is closest to the original N observations

$$\underbrace{\mathbf{X}}_{N\times 3} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \\ x_{N1} & x_{N2} & x_{N3} \end{pmatrix} \qquad \underbrace{\mathbf{Z}}_{N\times 2} = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \\ z_{N1} & z_{N2} \end{pmatrix}$$

PCA: Geometric Perspective

The best 2-dimensional representation is the plane that is closest to the original N observations

$$\underbrace{\mathbf{X}}_{N\times 3} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \\ x_{N1} & x_{N2} & x_{N3} \end{pmatrix} \qquad \underbrace{\mathbf{Z}}_{N\times 2} = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \\ z_{N1} & z_{N2} \end{pmatrix}$$



Linear Dimension Reduction

- Methods for deriving a simplified representation of features from a large set of variables

Linear Dimension Reduction

- Methods for deriving a simplified representation of features from a large set of variables
- Goal is for these simplified features to somehow still capture most of the variation in the raw data

Linear Dimension Reduction

- Methods for deriving a simplified representation of features from a large set of variables
- Goal is for these simplified features to somehow still capture most of the variation in the raw data
- $\mathbf{Z}_{*,1}, \ldots, \mathbf{Z}_{*,M}$ represent $M \leq K$ linear combinations of the original K predictors $(\mathbf{X}_{*,1}, \ldots, \mathbf{X}_{*,K})$

$$Z_m = \sum_{k=1}^{K} v_{km} X_k, \quad \text{where} \quad \sum_{k=1}^{K} v_{km}^2 = 1$$

Linear Dimension Reduction

- $\mathbf{Z}_{*,1}, \ldots, \mathbf{Z}_{*,M}$ represent $M \leq K$ linear combinations of the original K predictors $(\mathbf{X}_{*,1}, \ldots, \mathbf{X}_{*,K})$

$$Z_m = \sum_{k=1}^{K} v_{km} X_k, \quad \text{where} \quad \sum_{k=1}^{K} v_{km}^2 = 1$$

  - Loading vector for m-th reduced dimension is $\mathbf{v_m}$ (a unit vector, so that $\mathbf{v}_m^T \mathbf{v}_m = 1$)

Linear Dimension Reduction

- $\mathbf{Z}_{*,1}, \ldots, \mathbf{Z}_{*,M}$ represent $M \leq K$ linear combinations of the original K predictors $(\mathbf{X}_{*,1}, \ldots, \mathbf{X}_{*,K})$

$$Z_m = \sum_{k=1}^{K} v_{km} X_k, \quad \text{where} \quad \sum_{k=1}^{K} v_{km}^2 = 1$$

  - Loading vector for m-th reduced dimension is $\mathbf{v_m}$ (a unit vector, so that $\mathbf{v}_m^T \mathbf{v}_m = 1$)
  - $\sum_{j=1}^{p} v_{jm}^2 = 1$: ensures that Z_m doesn't get arbitrarily large (consistent with $\mathbf{v}_m$ being a direction in space, or a unit vector)

Linear Dimension Reduction

- $\mathbf{Z}_{*,1}, \ldots, \mathbf{Z}_{*,M}$ represent $M \leq K$ linear combinations of the original K predictors $(\mathbf{X}_{*,1}, \ldots, \mathbf{X}_{*,K})$

$$Z_m = \sum_{k=1}^{K} v_{km} X_k, \quad \text{where} \quad \sum_{k=1}^{K} v_{km}^2 = 1$$

  - Loading vector for m-th reduced dimension is $\mathbf{v_m}$ (a unit vector, so that $\mathbf{v}_m^T \mathbf{v}_m = 1$)
  - $\sum_{j=1}^{p} v_{jm}^2 = 1$: ensures that Z_m doesn't get arbitrarily large (consistent with $\mathbf{v}_m$ being a direction in space, or a unit vector)
  - $z_{im} = \mathbf{v_m^T x_i}$: scalar projection of data point $\mathbf{x_i}$ on to the mth principal component direction

Linear Dimension Reduction

- $\mathbf{Z}_{*,1}, \ldots, \mathbf{Z}_{*,M}$ represent $M \leq K$ linear combinations of the original K predictors $(\mathbf{X}_{*,1}, \ldots, \mathbf{X}_{*,K})$

$$Z_m = \sum_{k=1}^{K} v_{km} X_k, \quad \text{where} \quad \sum_{k=1}^{K} v_{km}^2 = 1$$

  - Loading vector for m-th reduced dimension is $\mathbf{v_m}$ (a unit vector, so that $\mathbf{v}_m^T \mathbf{v}_m = 1$)
  - $\sum_{j=1}^{p} v_{jm}^2 = 1$: ensures that Z_m doesn't get arbitrarily large (consistent with $\mathbf{v}_m$ being a direction in space, or a unit vector)
  - $z_{im} = \mathbf{v_m^T x_i}$: scalar projection of data point $\mathbf{x_i}$ on to the mth principal component direction
  - z_{im} tells us how far to go along $\mathbf{v_m}$ to get as close to $\mathbf{x_i}$ as possible

Interpretation of Principal Components

- We showed that finding the closest line (or plane, in higher dimensions) is equivalent to maximizing the variance of the projected data.

Interpretation of Principal Components

- We showed that finding the closest line (or plane, in higher dimensions) is equivalent to maximizing the variance of the projected data.

Interpretation of Principal Components

- We showed that finding the closest line (or plane, in higher dimensions) is equivalent to maximizing the variance of the projected data.
- Proportion of variance explained (PVE)

$$\frac{\text{Variance explained by } m\text{th PC}}{\text{Total variance}} = \frac{\sum_{i=1}^{N} \left(\sum_{k=1}^{K} v_{km} x_{ij} \right)^2}{\sum_{k=1}^{K} \sum_{i=1}^{N} x_{ik}^2}$$

Interpretation of Principal Components

- We showed that finding the closest line (or plane, in higher dimensions) is equivalent to maximizing the variance of the projected data.
- Proportion of variance explained (PVE)

$$\frac{\text{Variance explained by } m\text{th PC}}{\text{Total variance}} = \frac{\sum_{i=1}^{N}\left(\sum_{k=1}^{K} v_{km}x_{ij}\right)^2}{\sum_{k=1}^{K}\sum_{i=1}^{N} x_{ik}^2}$$

- Can also be used to get the cumulative PVE of the first k PC

Interpretation of Principal Components

- We showed that finding the closest line (or plane, in higher dimensions) is equivalent to maximizing the variance of the projected data.
- Proportion of variance explained (PVE)

$$\frac{\text{Variance explained by } m\text{th PC}}{\text{Total variance}} = \frac{\sum_{i=1}^{N} \left(\sum_{k=1}^{K} v_{km} x_{ij} \right)^2}{\sum_{k=1}^{K} \sum_{i=1}^{N} x_{ik}^2}$$

- Can also be used to get the cumulative PVE of the first k PC
- The cumulative PVE of all K PC will be 1

Why do we care?

- PCA will recover the eigenvector (characteristic vector) with the largest eigenvalue (characteristic value)

Why do we care?

- PCA will recover the eigenvector (characteristic vector) with the largest eigenvalue (characteristic value)
- This helps us to identify underlying structures in highly collinear data

Why do we care?

- PCA will recover the eigenvector (characteristic vector) with the largest eigenvalue (characteristic value)
- This helps us to identify underlying structures in highly collinear data
- We can use it to analyze high-dimensional data like voting records

Why do we care?

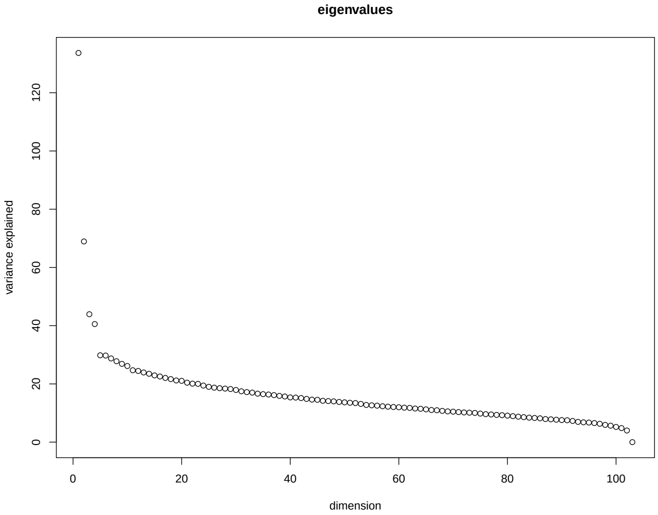
- PCA will recover the eigenvector (characteristic vector) with the largest eigenvalue (characteristic value)
- This helps us to identify underlying structures in highly collinear data
- We can use it to analyze high-dimensional data like voting records
- Or to reduce dimension for visualization

PCA on Senate Rollcall Votes

R Code

```
> rollcall <- read.dta('sen112kh.dta')
> X <- rollcall[,grep('^V', colnames(rollcall))]
> rollcall.pca <- svd(scale(X))
> rollcall <- read.dta('sen112kh.dta')
> rollcall[1:5, 1:12] cong id state dist lstate party eh1 eh2 name V1 V2 V3 1
  112 99911 99 0 USA 100 NA NA OBAMA 9 9 9 2 112 49700 41 0 ALABAMA 200 0 1
  SESSIONS 1 1 6 3 112 94659 41 0 ALABAMA 200 0 1 SHELBY 1 1 1 4 112 40300 81 0
  ALASKA 200 0 1 MURKOWSKI 1 1 1 5 112 40900 81 0 ALASKA 100 0 1 BEGICH 1 1 1
> X <- rollcall[,grep('^V', colnames(rollcall))]
> X <- as.matrix(X)
> rollcall.pca <- svd(scale(X))
> z1 <- rollcall.pca$u[,1]
> z2 <- rollcall.pca$u[,2]
```

PCA on Senate Rollcall Votes



PCA on Senate Rollcall Votes

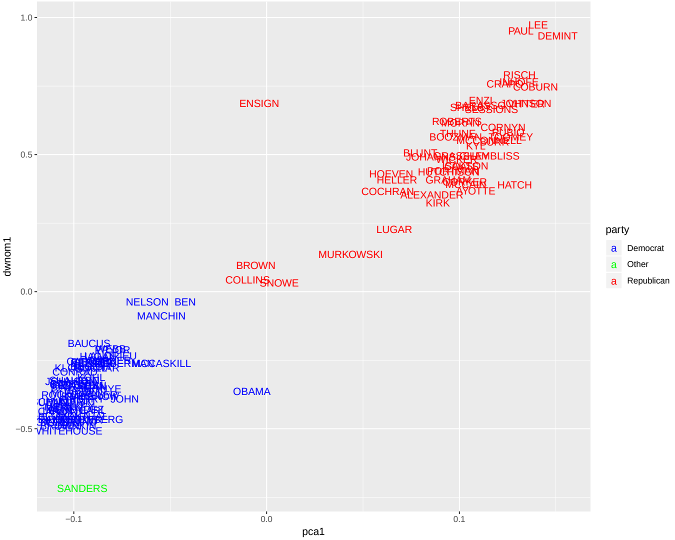
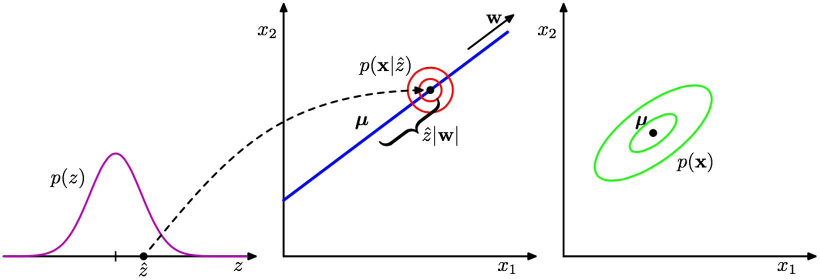


Figure: Note: Y axis is DW-Nominate Score

Probabilistic PCA



- An observation draws its score in the 1D latent space from standard normal (left)
- This score is mapped to the 2D observed space and noise is added (center)
- This implies that overall observed data follows the green distribution (right)

Image Matrix

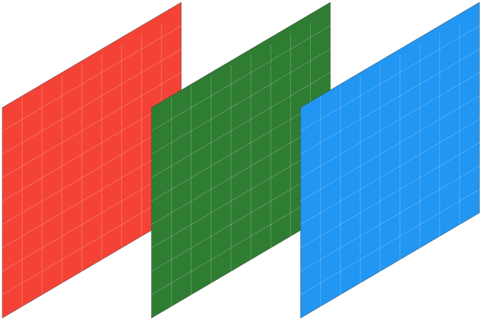
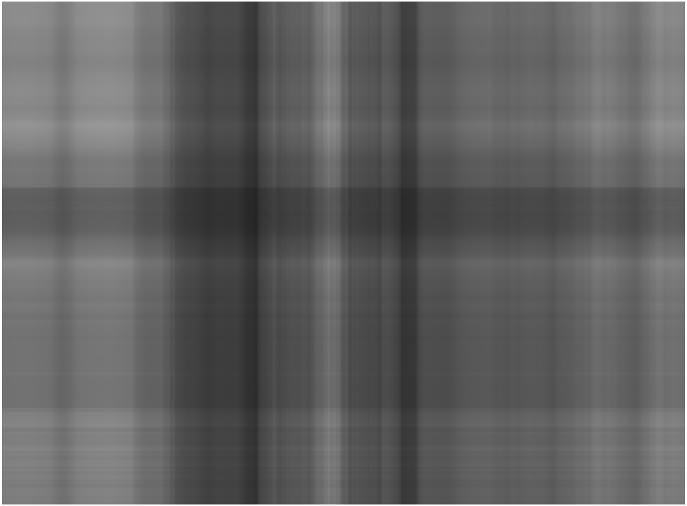


Image Matrix

Image Matrix

Convert image array to matrix of pixel intensities (average RGB channels)

Image Matrix



Reconstructed matrix using only first SVD dimension scores and loadings.

Image Matrix

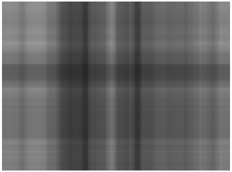


Image Matrix

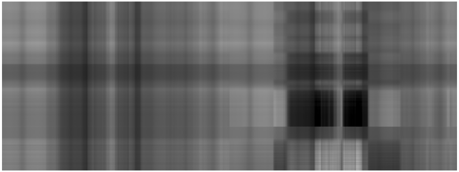


Image Matrix

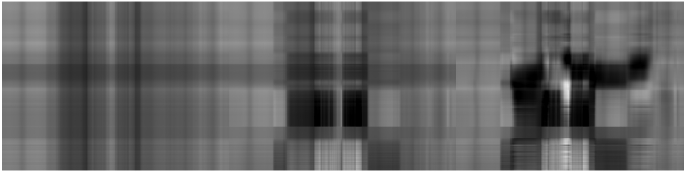


Image Matrix

Image Matrix

Image Matrix

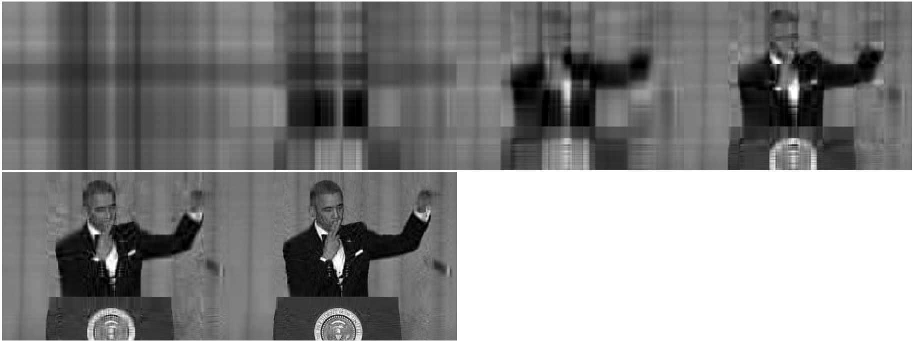


Image Matrix

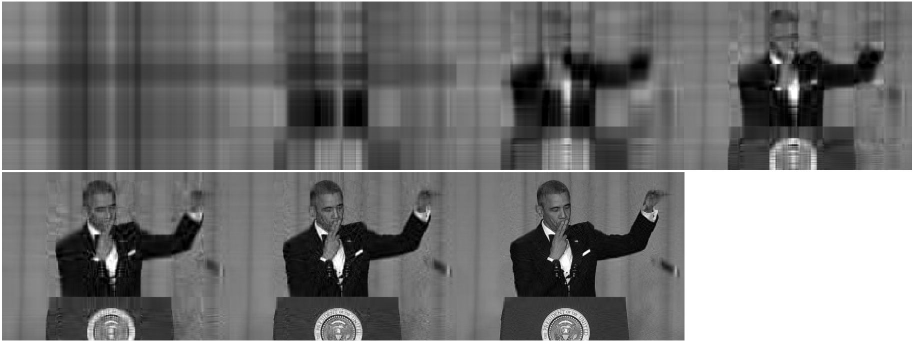


Image Matrix

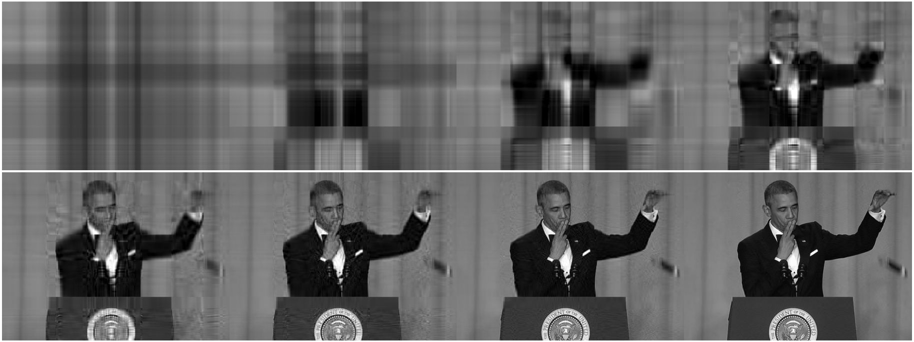
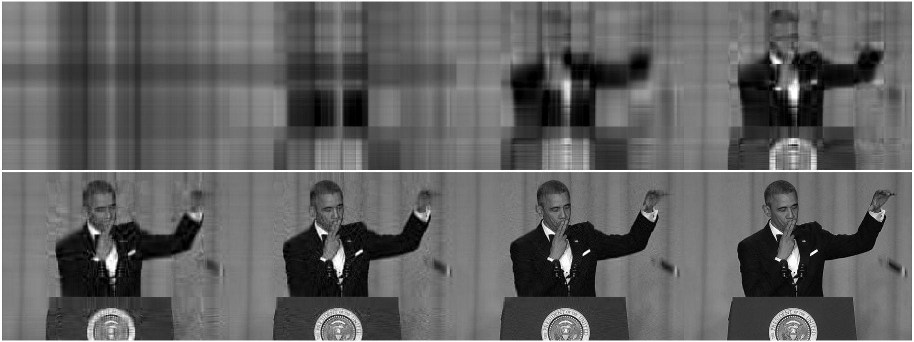


Image Matrix



Reconstruction with successively larger number of dimensions (1, 2, 4, 8, 16, 32, 64, 128)

Image Data

- Alternatively, grayscale images can be treated as a simple vector of pixel intensities (discards all spatial relationships between adjacent pixels).

Image Data

- Alternatively, grayscale images can be treated as a simple vector of pixel intensities (discards all spatial relationships between adjacent pixels).
- Then these can be treated as a standard data matrix (each row becomes an observation)

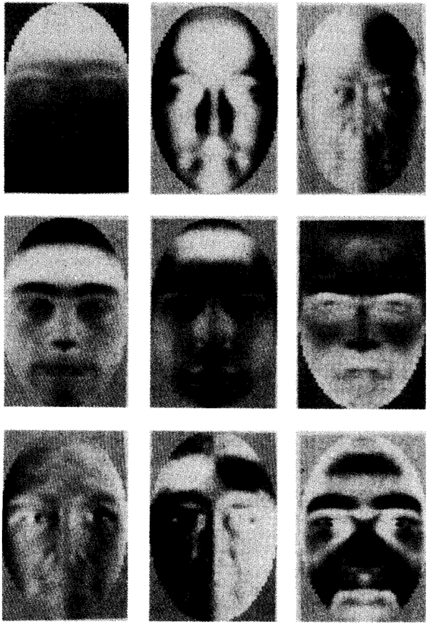
Image Data

- Alternatively, grayscale images can be treated as a simple vector of pixel intensities (discards all spatial relationships between adjacent pixels).
- Then these can be treated as a standard data matrix (each row becomes an observation)
- Essentially discards spatial information

Image Data

- Alternatively, grayscale images can be treated as a simple vector of pixel intensities (discards all spatial relationships between adjacent pixels).
- Then these can be treated as a standard data matrix (each row becomes an observation)
- Essentially discards spatial information
- Classifiers are not invariant to shifting, rotation, resizing of object

Eigenfaces: First 9 Dimensions



Does Exposure to the Refugee Crisis Make Natives More Hostile?

- Causal evidence regarding the impact of the refugee crisis on natives' attitudes, policy preferences, and political engagement. (Hangartner et al, 2018, APSR)
- Leveraging a targeted survey of 2,070 island residents.
- Results show mere exposure suffices in generating lasting increases in hostility.

Use PCA to Reduce Outcome Dimensionality

- Use a set of questions to measure opinion towards Native opinion.
- Built a summary scale that combines the different measures by extracting the first component of a PCA.

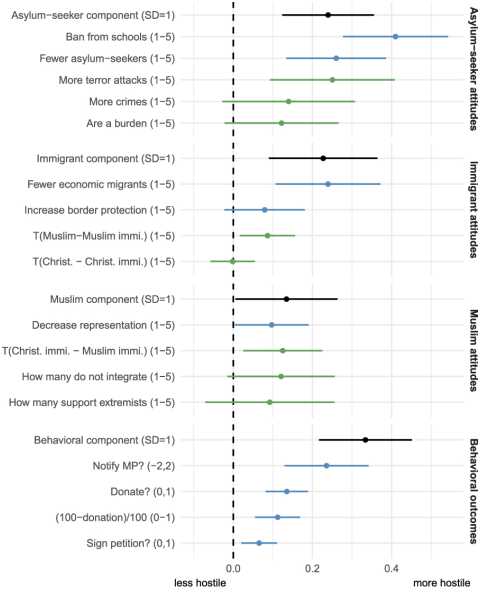
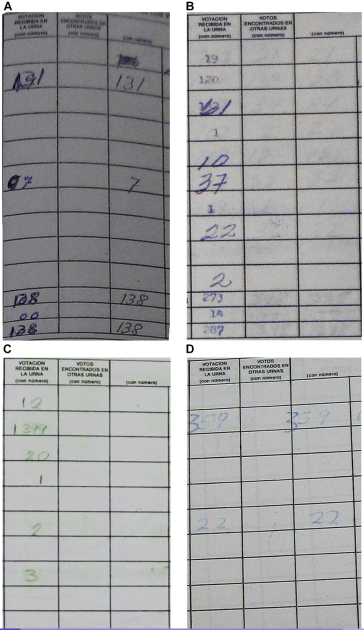


Figure: Black color shows the PCA first component for each set of questions

Fingerprints of Fraud

- How does a non-democratic regime rely on fraud? (Cantu, 2019, APSR)
- Documenting the alteration of vote tallies during the 1988 presidential election in Mexico.
- Authors find evidence of blatant alterations in about a third of the tallies in the country, using image data.

Vote Data


A
B
C
D
VOTACION RECIBIDA EN LA URNA (con número)
VOTOS ENCONTRADOS EN OTRAS URNAS (con número)
(con número)

Feature Extraction from Images

- Transform each picture into a numerical array of size 227 (height) $\times$ 227 (width) $\times$ 3 (RGB color channels)

Feature Extraction from Images

- Transform each picture into a numerical array of size 227 (height) $\times$ 227 (width) $\times$ 3 (RGB color channels)
- Enter first convolutional layer and extract high level visual features.

Feature Extraction from Images

- Transform each picture into a numerical array of size 227 (height) $\times$ 227 (width) $\times$ 3 (RGB color channels)
- Enter first convolutional layer and extract high level visual features.
- Enter second convolutional layer using the output of the previous step.

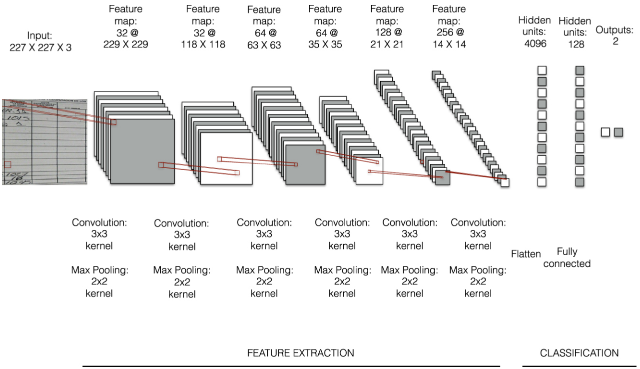
Feature Extraction from Images

- Transform each picture into a numerical array of size 227 (height) $\times$ 227 (width) $\times$ 3 (RGB color channels)
- Enter first convolutional layer and extract high level visual features.
- Enter second convolutional layer using the output of the previous step.
- Enter third convolutional layer ....

Feature Extraction from Images

- Transform each picture into a numerical array of size 227 (height) $\times$ 227 (width) $\times$ 3 (RGB color channels)
- Enter first convolutional layer and extract high level visual features.
- Enter second convolutional layer using the output of the previous step.
- Enter third convolutional layer ....
- Actually we can combine PCA with CNN. (Grag et al, 2019, IEEE).

Model Input



Input:
227 X 227 X 3
Feature map: 32 @ 229 X 229
Feature map: 32 @ 118 X 118
Feature map: 64 @ 63 X 63
Feature map: 64 @ 35 X 35
Feature map: 128 @ 21 X 21
Feature map: 256 @ 14 X 14
Hidden units: 4096
Hidden units: 128
Outputs: 2
Convolution: 3x3 kernel
Max Pooling: 2x2 kernel
Convolution: 3x3 kernel
Max Pooling: 2x2 kernel
Convolution: 3x3 kernel
Max Pooling: 2x2 kernel
Convolution: 3x3 kernel
Max Pooling: 2x2 kernel
Convolution: 3x3 kernel
Max Pooling: 2x2 kernel
Convolution: 3x3 kernel
Max Pooling: 2x2 kernel
Flatten
Fully connected
FEATURE EXTRACTION
CLASSIFICATION

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
    - Links between organizations and political groups
    - Actions taken by two actors jointly
    - Actions by one actor toward another

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another

- Taking dependency seriously

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another

- Taking dependency seriously
  - Most models that we have learned so far assume i.i.d.

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another

- Taking dependency seriously
  - Most models that we have learned so far assume i.i.d.
  - Can we assume that trade flows between countries i and j is i.i.d. compared to trade flows between i and k?

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another
- Taking dependency seriously
  - Most models that we have learned so far assume i.i.d.
  - Can we assume that trade flows between countries i and j is i.i.d. compared to trade flows between i and k?
- Examples:

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another

- Taking dependency seriously
  - Most models that we have learned so far assume i.i.d.
  - Can we assume that trade flows between countries i and j is i.i.d. compared to trade flows between i and k?

- Examples:
  - Social ties

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another

- Taking dependency seriously
  - Most models that we have learned so far assume i.i.d.
  - Can we assume that trade flows between countries i and j is i.i.d. compared to trade flows between i and k?

- Examples:
  - Social ties
  - National alliances

Network Data

- Model relationships among political actors, rather than each unit's behavior in isolation
  - Links between organizations and political groups
  - Actions taken by two actors jointly
  - Actions by one actor toward another

- Taking dependency seriously
  - Most models that we have learned so far assume i.i.d.
  - Can we assume that trade flows between countries i and j is i.i.d. compared to trade flows between i and k?

- Examples:
  - Social ties
  - National alliances
  - Overlapping membership in international institutions

Google PageRank

- Pages with a greater number of incoming edges are more important

Google PageRank

- Pages with a greater number of incoming edges are more important
- Incoming edges analogous to votes of support

Google PageRank

- Pages with a greater number of incoming edges are more important
- Incoming edges analogous to votes of support
- A page with incoming edge from another node with a large number of incoming edges: more important

Google PageRank

- Pages with a greater number of incoming edges are more important
- Incoming edges analogous to votes of support
- A page with incoming edge from another node with a large number of incoming edges: more important
- An "important" senator's Twitter account is followed by another politician whose account has many followers

Google PageRank

- Pages with a greater number of incoming edges are more important
- Incoming edges analogous to votes of support
- A page with incoming edge from another node with a large number of incoming edges: more important
- An "important" senator's Twitter account is followed by another politician whose account has many followers
- An iterative algorithm

$$\text{PageRank}_i = \frac{1-d}{N} + d \times \sum_{j=1}^{N} \frac{A_{ji} \times \text{PageRank}_j}{\text{outdegree}_j}$$

where d is a constant (e.g., 0.85) and N is the number of nodes

Google PageRank

- Pages with a greater number of incoming edges are more important
- Incoming edges analogous to votes of support
- A page with incoming edge from another node with a large number of incoming edges: more important
- An "important" senator's Twitter account is followed by another politician whose account has many followers
- An iterative algorithm

$$\text{PageRank}_i = \frac{1-d}{N} + d \times \sum_{j=1}^{N} \frac{A_{ji} \times \text{PageRank}_j}{\text{outdegree}_j}$$

  where d is a constant (e.g., 0.85) and N is the number of nodes
- Arises from extension of message-passing model in which users start uniformly distributed, but stop browsing at each step with some probability

Twitter Networks among Politicians

R Code

```
## defining colors
> col <- rep("red", n); col[senator$party == "D"] <- "blue"; col[senator$party == "I"] <- "black"
## PageRank
> senator$pagerank <- page.rank(twitter.adj)$vector
> senator[order(senator$pagerank, decreasing=T),][1:5,]
      screen_name          name party state indegree outdegree   pagerank
68   SenPatRoberts   Pat Roberts     R    KS       63        68 0.02100866
7      JohnBoozman  John Boozman     R    AR       55        80 0.01738608
8  SenJohnBarrasso John Barrasso     R    WY       60        87 0.01712930
88        RonWyden     Ron Wyden     D    OR       58         0 0.01679434
60  SenJeffMerkley  Jeff Merkley     D    OR       54        68 0.01611258
```

Twitter Networks among Politicians

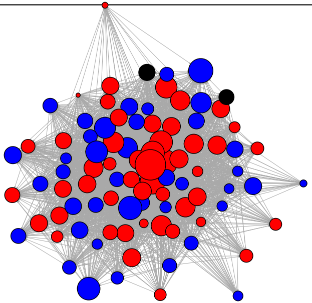
R Code

```
## defining colors
> col <- rep("red", n); col[senator$party == "D"] <- "blue"; col[senator$party == "I"] <- "black"
## PageRank
> senator$pagerank <- page.rank(twitter.adj)$vector
> senator[order(senator$pagerank, decreasing=T),][1:5,]
       screen_name           name party state indegree outdegree   pagerank
68   SenPatRoberts   Pat Roberts     R    KS       63        68 0.02100866
7      JohnBoozman  John Boozman     R    AR       55        80 0.01738608
8  SenJohnBarrasso John Barrasso     R    WY       60        87 0.01712930
88        RonWyden     Ron Wyden     D    OR       58         0 0.01679434
60  SenJeffMerkley  Jeff Merkley     D    OR       54        68 0.01611258

> plot(twitter.adj, vertex.size = senator$pagerank * 1000,
+      vertex.color = col, vertex.label = NA,
+      edge.arrow.size = 0.1, edge.width = 0.5)
```

Twitter Networks among Politicians

R Code

```
## defining colors
> col <- rep("red", n); col[senator$party == "D"] <- "blue"; col[senator$party == "I"] <- "black"
## PageRank
> senator$pagerank <- page.rank(twitter.adj)$vector
> senator[order(senator$pagerank, decreasing=T),][1:5,]
       screen_name          name party state indegree outdegree   pagerank
68   SenPatRoberts   Pat Roberts     R    KS       63        68 0.02100866
7      JohnBoozman  John Boozman     R    AR       55        80 0.01738608
8  SenJohnBarrasso John Barrasso     R    WY       60        87 0.01712930
88        RonWyden     Ron Wyden     D    OR       58         0 0.01679434
60  SenJeffMerkley  Jeff Merkley     D    OR       54        68 0.01611258

> plot(twitter.adj, vertex.size = senator$pagerank * 1000,
+      vertex.color = col, vertex.label = NA,
+      edge.arrow.size = 0.1, edge.width = 0.5)
```



Latent Space Network Models

- Peter D. Hoff, Adrian E. Raftery, & Mark S. Handcock. 2002. "Latent Space Approaches to Social Network Analysis." *JASA* Vol. 97, No. 460.

Latent Space Network Models

- Peter D. Hoff, Adrian E. Raftery, & Mark S. Handcock. 2002. "Latent Space Approaches to Social Network Analysis." *JASA* Vol. 97, No. 460.
- Ties arise stochastically as a function of the distance between two observations. With unweighted ties:

$$p_{ij} = \text{logit}^{-1}(\alpha_i + \beta_j + \mathbf{X}_{ij}^\top \gamma + \delta ||\mathbf{z}_i - \mathbf{z}_j||)$$
$$A_{ij} \sim \text{Bern}(p_{ij})$$

where $\mathbf{X}_{ij}$ is a vector of dyadic characteristics

Latent Space Network Models

- Peter D. Hoff, Adrian E. Raftery, & Mark S. Handcock. 2002. "Latent Space Approaches to Social Network Analysis." *JASA* Vol. 97, No. 460.
- Ties arise stochastically as a function of the distance between two observations. With unweighted ties:

$$p_{ij} = \text{logit}^{-1}(\alpha_i + \beta_j + \mathbf{X}_{ij}^\top \gamma + \delta||\mathbf{z}_i - \mathbf{z}_j||)$$
$$A_{ij} \sim \text{Bern}(p_{ij})$$

  where $\mathbf{X}_{ij}$ is a vector of dyadic characteristics
- Generalizes to all exponential-family distributions

Latent Space Network Models

- Pablo Barberá. 2015. "Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data." *Political Analysis*, 2015, 23 (1), 76-91.

$$
\begin{aligned}
p_{ij} &= \text{logit}^{-1}(\alpha_i + \beta_j + \gamma||\mathbf{z}_i - \mathbf{z}_j||) \\
A_{ij} &\sim \text{Bern}(p_{ij})
\end{aligned}
$$

Latent Space Network Models

- Pablo Barberá. 2015. "Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data." *Political Analysis*, 2015, 23 (1), 76-91.

$$
\begin{aligned}
p_{ij} &= \text{logit}^{-1}(\alpha_i + \beta_j + \gamma||\mathbf{z}_i - \mathbf{z}_j||) \\
A_{ij} &\sim \text{Bern}(p_{ij})
\end{aligned}
$$

(simultaneously scale binary bipartite network of politician and populace Twitter accounts)

Latent Space Network Models

- Pablo Barberá. 2015. "Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data." *Political Analysis*, 2015, 23 (1), 76-91.

$$
\begin{aligned}
p_{ij} &= \text{logit}^{-1}(\alpha_i + \beta_j + \gamma||\mathbf{z}_i - \mathbf{z}_j||) \\
A_{ij} &\sim \text{Bern}(p_{ij})
\end{aligned}
$$

  (simultaneously scale binary bipartite network of politician and populace Twitter accounts)

- In Song Kim and Dmitriy Kunisky. "Mapping Political Communities: A Statistical Analysis of Lobbying Networks in Legislative Politics." Working paper.

$$
\begin{aligned}
\mu_{ij} &= \exp(\alpha_i + \beta_j + \gamma||\mathbf{z}_i - \mathbf{z}_j||) \\
A_{ij} &\sim \text{Poisson}(\mu_{ij})
\end{aligned}
$$

Latent Space Network Models

- Pablo Barberá. 2015. "Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data." *Political Analysis*, 2015, 23 (1), 76-91.

$$p_{ij} = \text{logit}^{-1}(\alpha_i + \beta_j + \gamma||\mathbf{z}_i - \mathbf{z}_j||)$$
$$A_{ij} \sim \text{Bern}(p_{ij})$$

  (simultaneously scale binary bipartite network of politician and populace Twitter accounts)

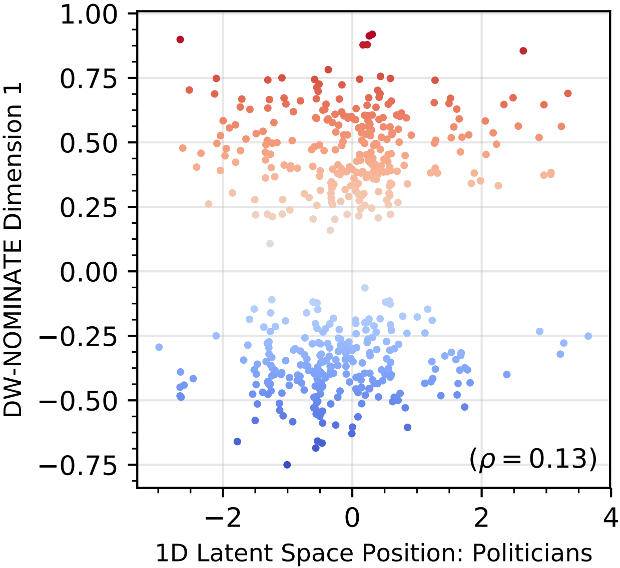
- In Song Kim and Dmitriy Kunisky. "Mapping Political Communities: A Statistical Analysis of Lobbying Networks in Legislative Politics." Working paper.

$$\mu_{ij} = \exp(\alpha_i + \beta_j + \gamma||\mathbf{z}_i - \mathbf{z}_j||)$$
$$A_{ij} \sim \text{Poisson}(\mu_{ij})$$

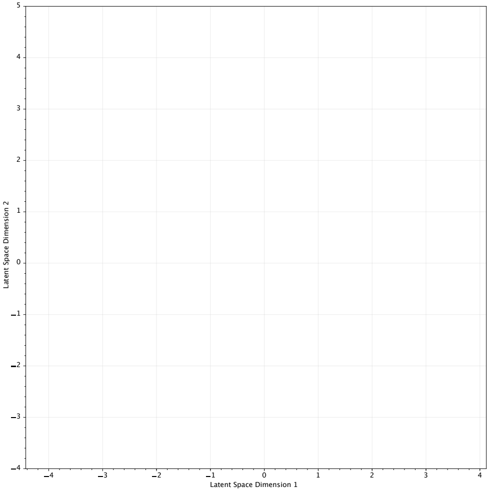
  (simultaneously scale count-valued bipartite network of interest group lobbying on a politician's sponsored bills)

1D Model: Kim & Kunisky, PA, 2021

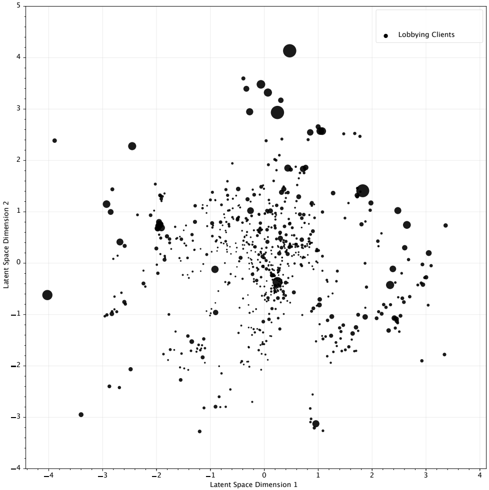
1D Model: Kim & Kunisky, PA, 2021



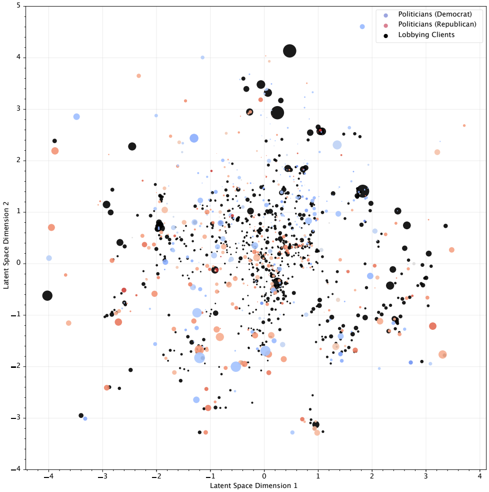
2D Model: Kim & Kunisky



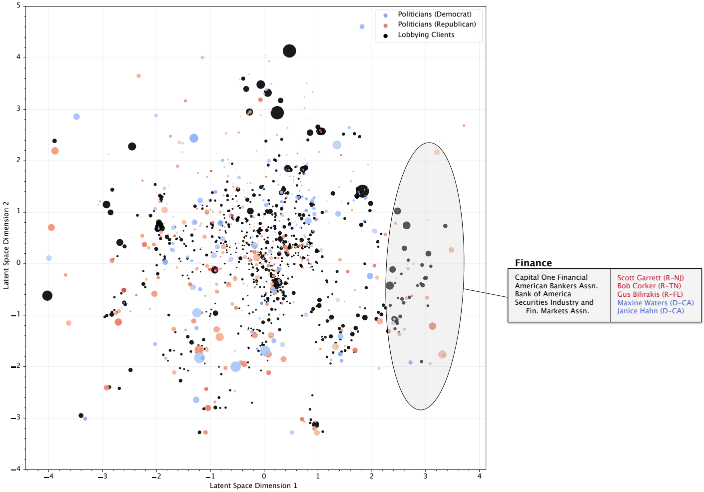
2D Model: Kim & Kunisky



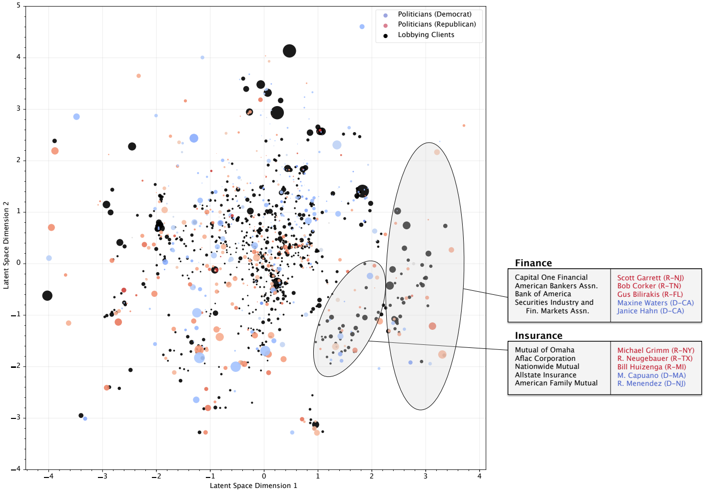
2D Model: Kim & Kunisky



2D Model: Kim & Kunisky



2D Model: Kim & Kunisky



2D Model: Kim & Kunisky

Military & Veterans' Affairs

Iraq & Afghanistan Veterans of America Fleet Reserve Assn. Retired Enlisted Assn. United Spinal Assn. American Legion	Jeff Miller (R–FL) Richard Burr (R–NC) Mike Coffman (R–CO) Dina Titus (D–NV) Rick Larsen (D–WA) Beto O'Rourke (D–TX)

Energy

Berkshire Hathaway Energy Koch Co. Public Sector National Grid USA Renewable Fuels Assn. Alliance of Automobile Mnf. Arch Coal BP America Chevron USA	Pompeo, Mike (R-TX) Flake, Jeff (R-AZ) McKinley, David (R-WV) Barrasso, John (R-WY) Edward Markey (D-MA) Shaheen, Jeanne (D-NH)

Conservation & Land Use

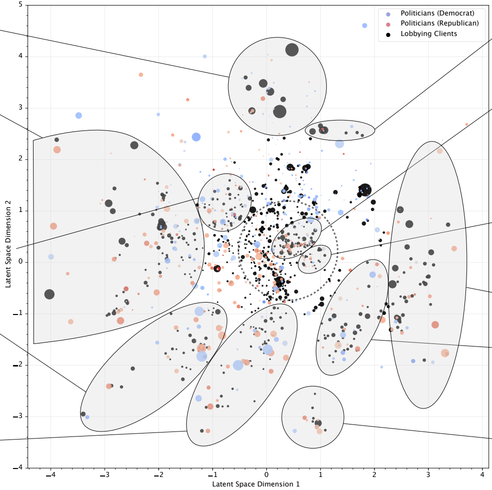
Earthjustice Legal Defense Fund Blue Green Alliance Sierra Club The Wilderness Society Safari Club International American Chemistry Council	Rand Paul (R-KY) David Vitter (R-LA) Paul Gossar (R-AZ) Barbara Boxer (D-CA) Tom Udall (D-NM) Robert Casey (D-PA)

Telecom

Ntl. Assn. of Broadcasters Cincinnatti Bell AT&T Corporation Time Warner Cable Verizon Gov. Relations Twenty-First Century Fox	Bob Goodlatte (R-VA) Mike Rogers (R-MI) Michael McCaul (R-TX) John Rockefeller (D-WV) Ron Wyden (D-OR) Doris Matsui (D-CA)

Technology

Google Yahoo! Amazon Mastercard Dell Texas Instruments Oracle Hewlett Packard	Darrell Issa (R-CA) Jason Chaffetz (R-UT) John Cornyn (R-TX) Orrin Hatch (R-UT) Patrick Leahy (D-VT) Zoe Lofgren (D-CA) Charles Schumer (D-NY)



Gun Rights

Gun Owners of America Everytown for Gun Safety Action Fund Ntl. Rifle Assn. of America	Steve Stockman (R–TX) Jim Jordan (R–OH) Robin Kelly (D–IA) Danny Davis (D–IL)

Health

American College of Emergency Physicians AARP Healthcare Leadership Council Assn. of American Medical Colleges American Assn. of Nurse Practitioners	Michael Burgess (R–TX) Diane Black (R–TN) Thomas Coburn (R–OK) Larry Bucshon (R–IN) Thomas Harkin (D–IA) Joseph Crowley (D–NY) Donna Edwards (D–MD)

Universities

Northwestern University Vanderbilt University University of Virginia Massachusetts Institute of Technology McGraw Hill Education	Virginia Foxx (R–NC) Luke Messer (R–IN) Chris Collins (R–NY) Spencer Bachus (R–AL) Joe Manchin (D–WV) Henry Johnson (D–GA)

Finance

Capital One Financial American Bankers Assn. Bank of America Securities Industry and Fin. Markets Assn.	Scott Garrett (R–NJ) Bob Corker (R–TN) Gus Bilirakis (R–FL) Maxine Waters (D–CA) Janice Hahn (D–CA)

Insurance

Mutual of Omaha Aflac Corporation Nationwide Mutual Allstate Insurance American Family Mutual	Michael Grimm (R–NY) R. Neugebauer (R–TX) Bill Huizenga (R–MI) M. Capuano (D–MA) R. Menendez (D–NJ)

Travel

US Travel Assn. Marriott International Global Business Travel Assn. Morphotrust USA	Joseph Heck (R–NV) Candice Miller (R–MI) J.A. Emerson (R–MO) Trey Gowdy (R–SC) Mike Quigley (D–IL)

Latent Space Text Models

- Very similar to latent space network models

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

where

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

where

- y_{itj} is the count of word j in party i's manifesto in election (or year) t

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

where

- y_{itj} is the count of word j in party i's manifesto in election (or year) t
- α_{it} is a party-year fixed effect

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

where

- y_{itj} is the count of word j in party i's manifesto in election (or year) t
- α_{it} is a party-year fixed effect
- ψ_j is a word fixed effect

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. “A Scaling Model for Estimating Time-Series Party Positions from Texts.” *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

where

- y_{itj} is the count of word j in party i's manifesto in election (or year) t
- α_{it} is a party-year fixed effect
- ψ_j is a word fixed effect
- β_j is a word-specific coefficient capturing the importance of word j in discriminating between party positions

Latent Space Text Models

- Very similar to latent space network models
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS* Vol. 52, No. 3.

$$\mu_{itj} = \exp(\alpha_{it} + \psi_j + \beta_j \omega_{it})$$
$$y_{itj} \sim \text{Poisson}(\mu_{itj})$$

where

- y_{itj} is the count of word j in party i's manifesto in election (or year) t
- α_{it} is a party-year fixed effect
- ψ_j is a word fixed effect
- β_j is a word-specific coefficient capturing the importance of word j in discriminating between party positions
- ω_{it} is the estimate of party i's position in election t

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ $(N \times K)$...

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ $(N \times K)$...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ ($N \times K$)...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?
- Consider the following procedure (with standardized $\mathbf{X}$):

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ ($N \times K$)...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?
- Consider the following procedure (with standardized $\mathbf{X}$):
  - Examine each feature $\mathbf{X}_j$ and compute $\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ ($N \times K$)...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?
- Consider the following procedure (with standardized $\mathbf{X}$):
  - Examine each feature $\mathbf{X}_j$ and compute $\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Construct i's score on the first latent dimension as weighted sum of its covariates, $Z_{i1} = \sum_j X_{ij} \hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ $(N \times K)$...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?
- Consider the following procedure (with standardized $\mathbf{X}$):
  - Examine each feature $\mathbf{X}_j$ and compute $\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Construct i's score on the first latent dimension as weighted sum of its covariates, $Z_{i1} = \sum_j X_{ij} \hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Then, residualize $\mathbf{X}$ (take out portion that can be explained by first dimension: $\tilde{\mathbf{X}}_j^1 = \mathbf{X}_j - Z_{i1} \cdot \frac{\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Z}_1)}{\hat{\text{Var}}(\mathbf{Z}_1)}$

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ ($N \times K$)...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?
- Consider the following procedure (with standardized $\mathbf{X}$):
  - Examine each feature $\mathbf{X}_j$ and compute $\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Construct i's score on the first latent dimension as weighted sum of its covariates, $Z_{i1} = \sum_j X_{ij} \hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Then, residualize $\mathbf{X}$ (take out portion that can be explained by first dimension: $\tilde{\mathbf{X}}_j^1 = \mathbf{X}_j - Z_{i1} \cdot \frac{\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Z}_1)}{\hat{\text{Var}}(\mathbf{Z}_1)}$
  - Construct i's score on the second dimension as weighted sum of its residualized covariates, $Z_{i2} = \sum_j X_{ij} \hat{\text{Cov}}(\tilde{\mathbf{X}}_j^1, \mathbf{Y})$

Partial Least Squares

- What if we didn't want to recover latent dimensions that optimally summarized $\mathbf{X}$ ($N \times K$)...
- ... but rather dimensions summarizing $\mathbf{X}$ in a way that explains $\mathbf{Y}$?
- Consider the following procedure (with standardized $\mathbf{X}$):
  - Examine each feature $\mathbf{X}_j$ and compute $\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Construct i's score on the first latent dimension as weighted sum of its covariates, $Z_{i1} = \sum_j X_{ij} \hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Y})$
  - Then, residualize $\mathbf{X}$ (take out portion that can be explained by first dimension: $\tilde{\mathbf{X}}_j^1 = \mathbf{X}_j - Z_{i1} \cdot \frac{\hat{\text{Cov}}(\mathbf{X}_j, \mathbf{Z}_1)}{\hat{\text{Var}}(\mathbf{Z}_1)}$
  - Construct i's score on the second dimension as weighted sum of its residualized covariates, $Z_{i2} = \sum_j X_{ij} \hat{\text{Cov}}(\tilde{\mathbf{X}}_j^1, \mathbf{Y})$
  - Repeat for dimensions 3, ..., M

Partial Least Squares

- Then compute partial least square coefficients by regressing $\mathbf{Y}$ on $\mathbf{Z} = [\mathbf{Z}_1, \ldots, \mathbf{Z}_M]$
- If $M = K$, then you can reconstruct each OLS coefficient from weighted combination of $(\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{Y}$
- Rather than than maximizing projected variance, this maximizes covariance between projection and outcome

Ridge Revisited

- In general, ridge regression tends to penalize the low variance principal components (i.e., the component with lower variance)

Ridge Revisited

- In general, ridge regression tends to penalize the low variance principal components (i.e., the component with lower variance)
- Recall that after centering, $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and $\mathbf{X}^\top\mathbf{X} = \hat{\text{Cov}}\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^\top$

Ridge Revisited

- In general, ridge regression tends to penalize the low variance principal components (i.e., the component with lower variance)
- Recall that after centering, $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and $\mathbf{X}^\top\mathbf{X} = \hat{\text{Cov}}\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^\top$

Ridge Revisited

- In general, ridge regression tends to penalize the low variance principal components (i.e., the component with lower variance)
- Recall that after centering, $\mathbf{X} = \mathbf{UDV}^T$ and $\mathbf{X}^\top\mathbf{X} = \hat{\text{Cov}}\mathbf{X} = \mathbf{VD}^2\mathbf{V}^\top$

$$
\begin{aligned}
\mathbf{X}\hat{\beta}^{\text{ridge}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \\
&= \mathbf{UDV}^T(\mathbf{VD}^2\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{VDU}^T\mathbf{y} \\
&= \mathbf{UDV}^T(\mathbf{VD}^2\mathbf{V}^T + \lambda\mathbf{VV}^T)^{-1}\mathbf{VDU}^T\mathbf{y} \\
&= \mathbf{UDV}^T\{\mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})\mathbf{V}^T\}^{-1}\mathbf{VDU}^T\mathbf{y} \\
&= \mathbf{UD}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{DU}^T\mathbf{y} \\
&= \sum_{k=1}^{K} \mathbf{u}_k \underbrace{\frac{d_k^2}{d_k^2 + \lambda}}_{\leq 1} \mathbf{u}_k^T\mathbf{y}
\end{aligned}
$$

- Dimensions explaining less variance in data get more shrinkage