

# Model Based Inference

Justin Grimmer

Professor  
Department of Political Science  
Stanford University

May 16th, 2019

# Principal Component Analysis $\rightsquigarrow$ low-dimensional embedding

# A Simple Two-Dimensional Example

Suppose we have the following observations:

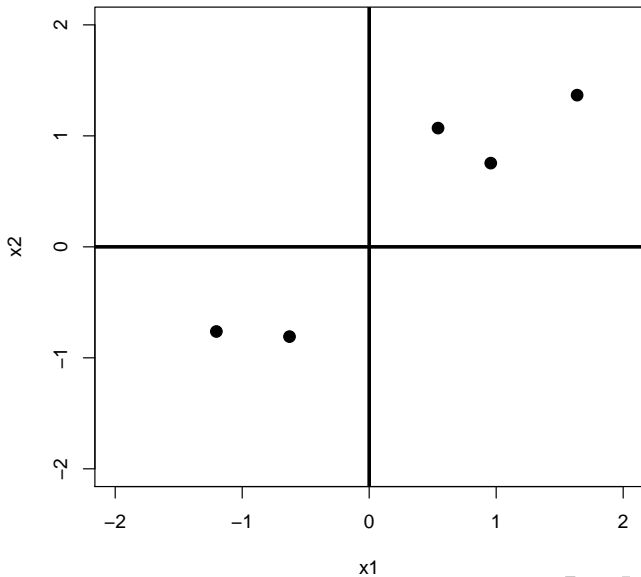
$$x_1 = (0.54, 1.07)$$

$$x_2 = (-1.20, -0.76)$$

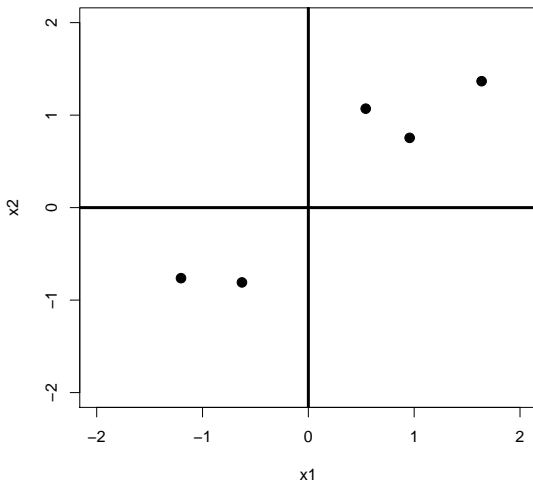
$$x_3 = (-0.63, -0.81)$$

$$x_4 = (0.96, 0.75)$$

$$x_5 = (1.64, 1.37)$$

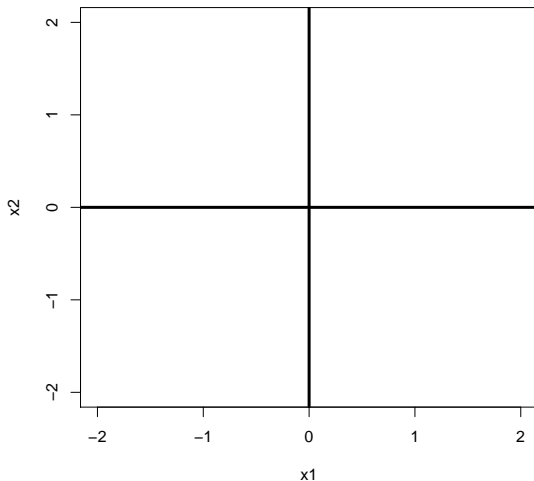


Goal: find line that summarizes bivariate information



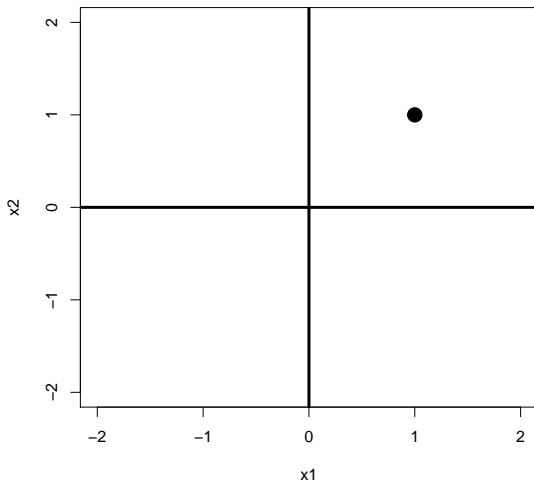
# Vectors to Draw a Line

Suppose  $\mathbf{w}_1 = (1, 1)$



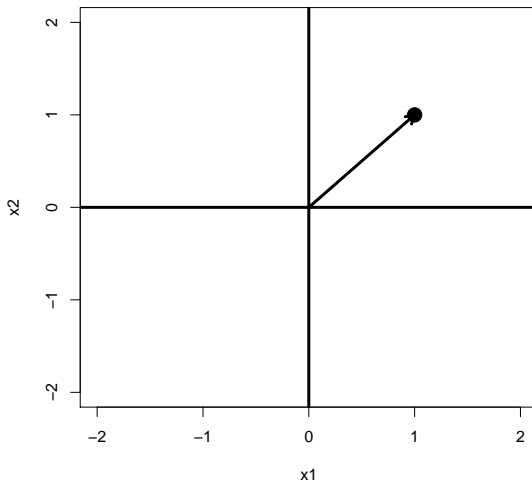
# Vectors to Draw a Line

Suppose  $\mathbf{w}_1 = (1, 1)$



# Vectors to Draw a Line

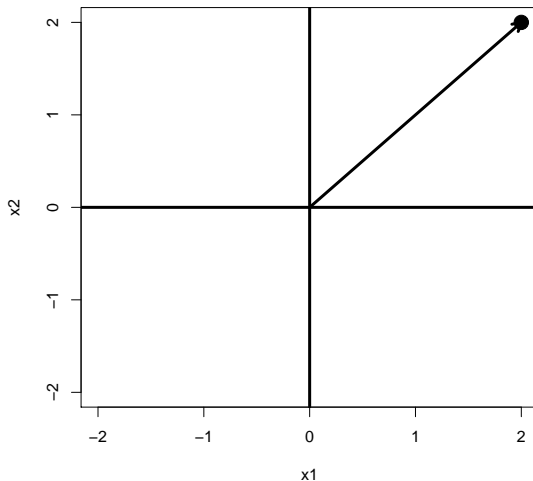
Suppose  $\mathbf{w}_1 = (1, 1)$





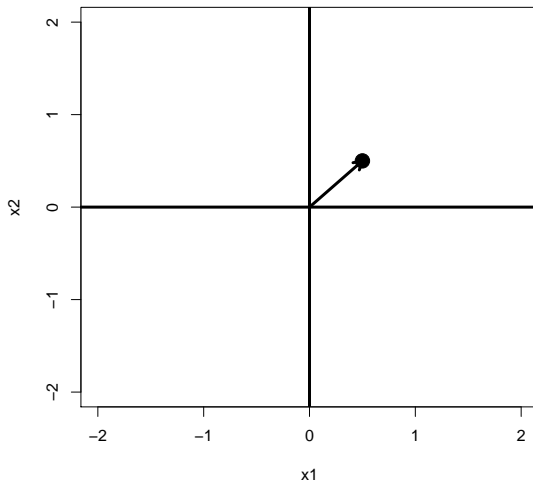
# Vectors to Draw a Line

Suppose  $\mathbf{w}_1 = (1, 1)$   $2\mathbf{w}_1 = (2, 2)$



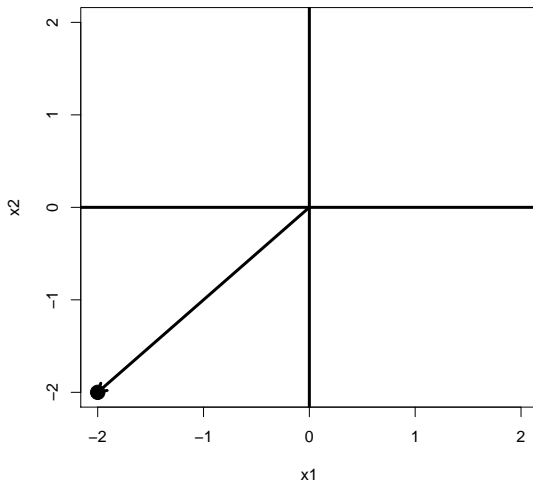
# Vectors to Draw a Line

Suppose  $\mathbf{w}_1 = (1, 1)$   $\frac{1}{2}\mathbf{w}_1 = (1/2, 1/2)$



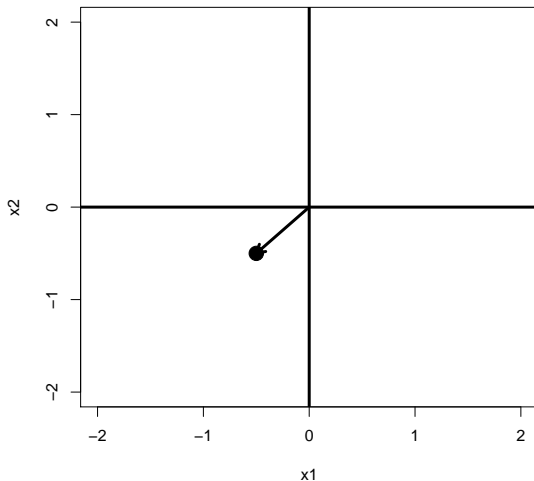
# Vectors to Draw a Line

Suppose  $\mathbf{w}_1 = (1, 1)$   $-2\mathbf{w}_1 = (-2, -2)$



# Vectors to Draw a Line

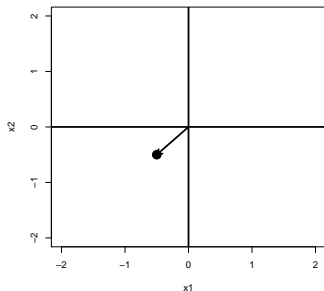
Suppose  $\mathbf{w}_1 = (1, 1)$      $-\frac{1}{2}\mathbf{w}_1 = (-1/2, -1/2)$



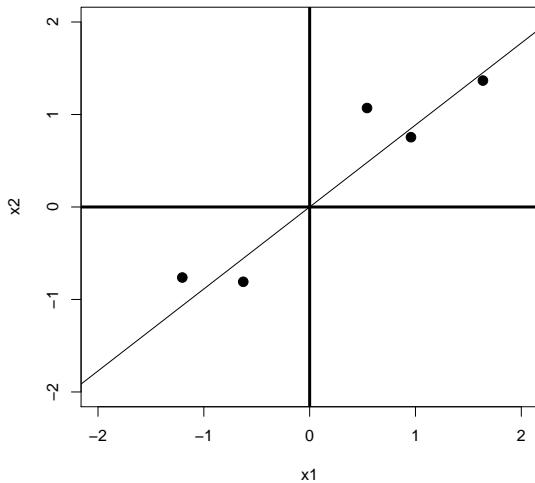
# Vectors to Draw a Line

Suppose  $\mathbf{w}_1 = (1, 1)$

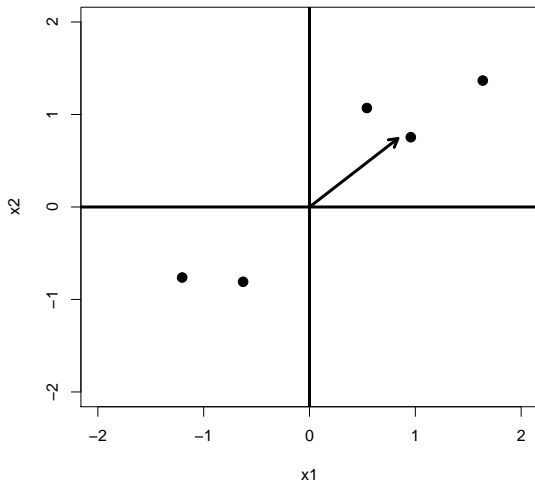
$z_i$  = amount we shrink/flip  $\mathbf{w}_1$  to approximate point  $i$ .



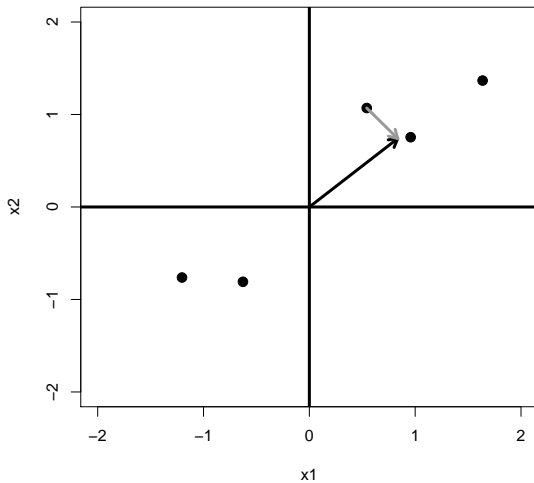
$$\mathbf{w}_1 = (0.75, 0.66)$$



$$\mathbf{w}_1 = (0.75, 0.66) \quad z_1 = 1.12$$

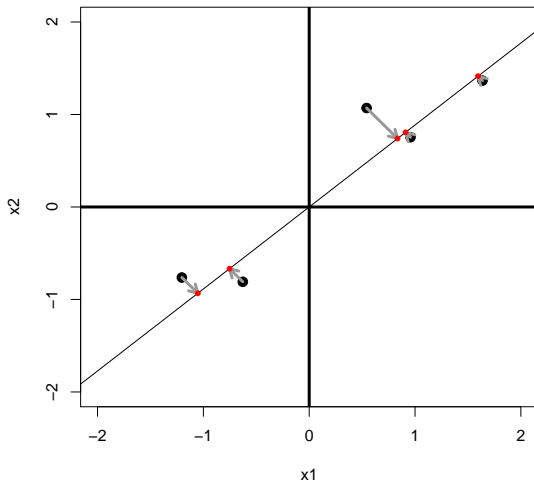


$$\mathbf{w}_1 = (0.75, 0.66) \quad z_1 = 1.12$$





$$\mathbf{w}_1 = (0.75, 0.66) \quad z_1 = 1.12$$



# Algebraic Representation

$$\mathbf{x}_i = z_i \mathbf{w}_1 + \mathbf{e}_i$$

# Algebraic Representation

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2})\end{aligned}$$

# Algebraic Representation

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2})\end{aligned}$$

Find  $\mathbf{w}_1 = (w_{11}, w_{12})$  and  $z_i$  to minimize the error

# Algebraic Representation

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2})\end{aligned}$$

Find  $\mathbf{w}_1 = (w_{11}, w_{12})$  and  $z_i$  to minimize the error

$$\text{error} = \frac{1}{N} \sum_{i=1}^N ((x_{i1}, x_{i2}) - z_i(w_{11}, w_{12}))' ((x_{i1}, x_{i2}) - z_i(w_{11}, w_{12}))$$

# Algebraic Representation

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2})\end{aligned}$$

Find  $\mathbf{w}_1 = (w_{11}, w_{12})$  and  $z_i$  to minimize the error

$$\begin{aligned}\text{error} &= \frac{1}{N} \sum_{i=1}^N ((x_{i1}, x_{i2}) - z_i(w_{11}, w_{12}))' ((x_{i1}, x_{i2}) - z_i(w_{11}, w_{12})) \\ &= \frac{1}{N} \sum_{i=1}^N (x_{i1} - z_i w_{11})^2 + (x_{i2} - z_i w_{12})^2\end{aligned}$$

# Three Dimensional Approximation

$$\mathbf{x}_1 = (0.09, -1.02, -0.10)$$

$$\mathbf{x}_2 = (0.09, 1.41, 0.67)$$

$$\mathbf{x}_3 = (-0.81, -1.46, -0.54)$$

$$\mathbf{x}_4 = (1.43, 0.26, 0.61)$$

$$\mathbf{x}_5 = (1.23, 0.87, 1.33)$$

Find  $\mathbf{w}_1 = (w_{11}, w_{12}, w_{13})$  and  $z_i$  to provide best one dimensional approximation.

## Three-Dimensional Visualization



Three-Dimensional Visualization  
 $\mathbf{w}_1 = (0.48, 0.75, 0.46)$

$$\mathbf{x}_i = z_i \mathbf{w}_1 + \mathbf{e}_i$$

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}, x_{i3}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2}, z_i w_{13} + e_{i3})\end{aligned}$$

$$\mathbf{x}_i = z_i \mathbf{w}_1 + \mathbf{e}_i$$

$$(x_{i1}, x_{i2}, x_{i3}) = (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2}, z_i w_{13} + e_{i3})$$

Find  $\mathbf{w}_1 = (w_{11}, w_{12}, w_{13})$  and  $z_i$  to minimize the error

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}, x_{i3}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2}, z_i w_{13} + e_{i3})\end{aligned}$$

Find  $\mathbf{w}_1 = (w_{11}, w_{12}, w_{13})$  and  $z_i$  to minimize the error

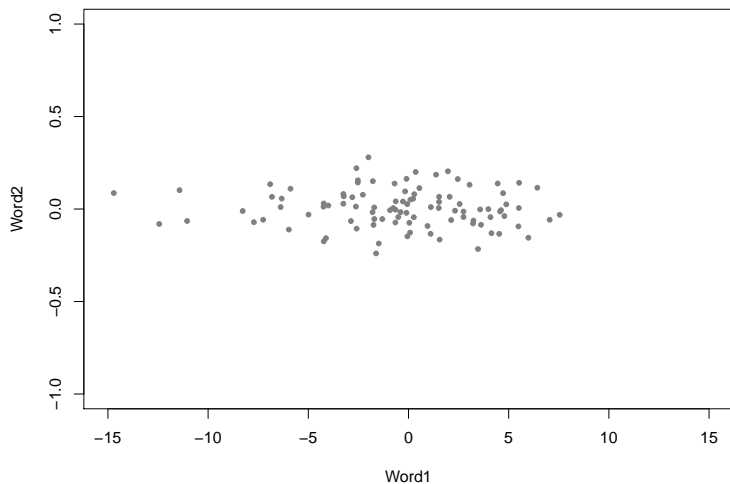
$$\begin{aligned}\text{error} &= \frac{1}{N} \sum_{i=1}^N ((x_{i1}, x_{i2}, x_{i3}) - z_i (w_{11}, w_{12}, w_{13}))' \\ &\quad ((x_{i1}, x_{i2}, x_{i3}) - z_i (w_{11}, w_{12}, w_{13}))\end{aligned}$$

$$\begin{aligned}\mathbf{x}_i &= z_i \mathbf{w}_1 + \mathbf{e}_i \\ (x_{i1}, x_{i2}, x_{i3}) &= (z_i w_{11} + e_{i1}, z_i w_{12} + e_{i2}, z_i w_{13} + e_{i3})\end{aligned}$$

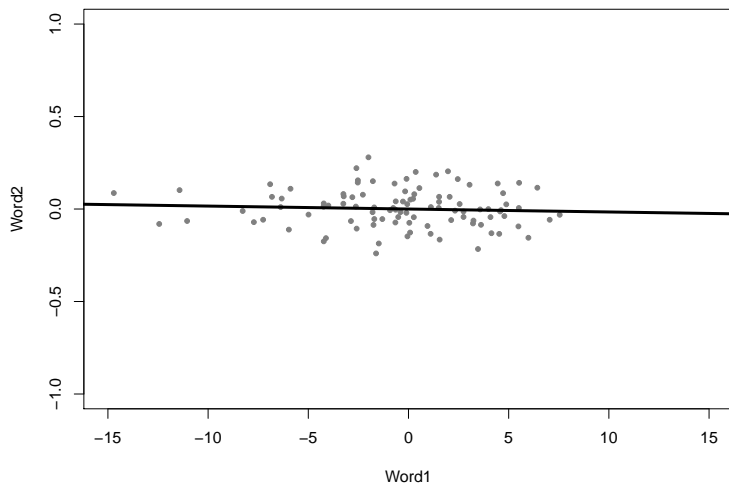
Find  $\mathbf{w}_1 = (w_{11}, w_{12}, w_{13})$  and  $z_i$  to minimize the error

$$\begin{aligned}\text{error} &= \frac{1}{N} \sum_{i=1}^N ((x_{i1}, x_{i2}, x_{i3}) - z_i(w_{11}, w_{12}, w_{13}))' \\ &\quad ((x_{i1}, x_{i2}, x_{i3}) - z_i(w_{11}, w_{12}, w_{13})) \\ &= \frac{1}{N} \sum_{i=1}^N (x_{i1} - z_i w_{11})^2 + (x_{i2} - z_i w_{12})^2 + (x_{i3} - z_i w_{13})^2\end{aligned}$$

# Principal Component Analysis

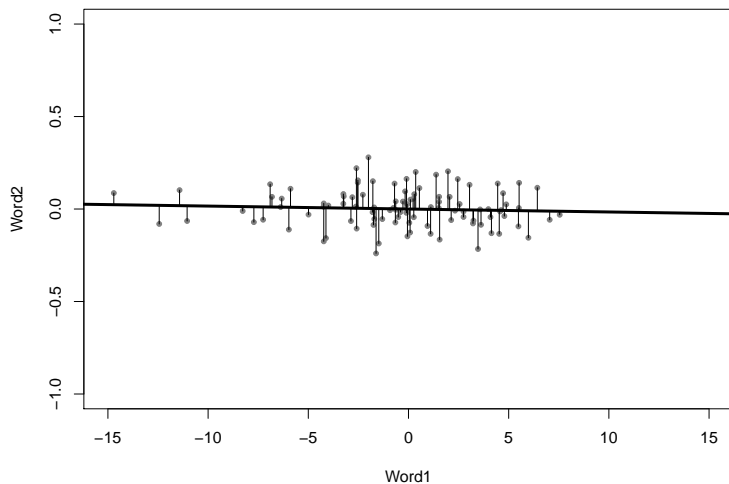


# Principal Component Analysis





# Principal Component Analysis



# PCA Output

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

# PCA Output

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

Principal Component Output:

# PCA Output

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

Principal Component Output:

1)  $K$  Principal Components  $\mathbf{w}_k$

# PCA Output

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

Principal Component Output:

1)  $K$  Principal Components  $\mathbf{w}_k$

$$\mathbf{w}_k = (w_{1k}, w_{2k}, \dots, w_{Jk})$$

# PCA Output

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

Principal Component Output:

1)  $K$  Principal Components  $\mathbf{w}_k$

$$\mathbf{w}_k = (w_{1k}, w_{2k}, \dots, w_{Jk})$$

2)  $K$  component vector describing loadings on principal components for each document

# PCA Output

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

Principal Component Output:

1)  $K$  Principal Components  $\mathbf{w}_k$

$$\mathbf{w}_k = (w_{1k}, w_{2k}, \dots, w_{Jk})$$

2)  $K$  component vector describing loadings on principal components for each document

$$\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{Ki})$$

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.

If

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

Then  $\mathbf{x}$  is an *eigenvector* and  $\lambda$  is the associated *eigenvalue*



# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.

If

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

Then  $\mathbf{x}$  is an *eigenvector* and  $\lambda$  is the associated *eigenvalue*

- $\mathbf{A}$  stretches the eigenvector  $\mathbf{x}$

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.

If

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

Then  $\mathbf{x}$  is an *eigenvector* and  $\lambda$  is the associated *eigenvalue*

- $\mathbf{A}$  stretches the eigenvector  $\mathbf{x}$
- $\mathbf{A}$  stretches  $\mathbf{x}$  by  $\lambda$

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

*Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.*

*If*

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

*Then  $\mathbf{x}$  is an **eigenvector** and  $\lambda$  is the associated **eigenvalue***

- $\mathbf{A}$  stretches the eigenvector  $\mathbf{x}$
- $\mathbf{A}$  stretches  $\mathbf{x}$  by  $\lambda$
- To find eigenvectors/values: (eigen in R)

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.

If

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

Then  $\mathbf{x}$  is an *eigenvector* and  $\lambda$  is the associated *eigenvalue*

- $\mathbf{A}$  stretches the eigenvector  $\mathbf{x}$
- $\mathbf{A}$  stretches  $\mathbf{x}$  by  $\lambda$
- To find eigenvectors/values: (eigen in R)
  - Find  $\lambda$  that solves  $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.

If

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

Then  $\mathbf{x}$  is an **eigenvector** and  $\lambda$  is the associated **eigenvalue**

- $\mathbf{A}$  stretches the eigenvector  $\mathbf{x}$
- $\mathbf{A}$  stretches  $\mathbf{x}$  by  $\lambda$
- To find eigenvectors/values: (eigen in R)
  - Find  $\lambda$  that solves  $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$
  - Find vectors in **null space** of:

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

Suppose  $\mathbf{A}$  is an  $N \times N$  matrix and  $\lambda$  is a scalar.

If

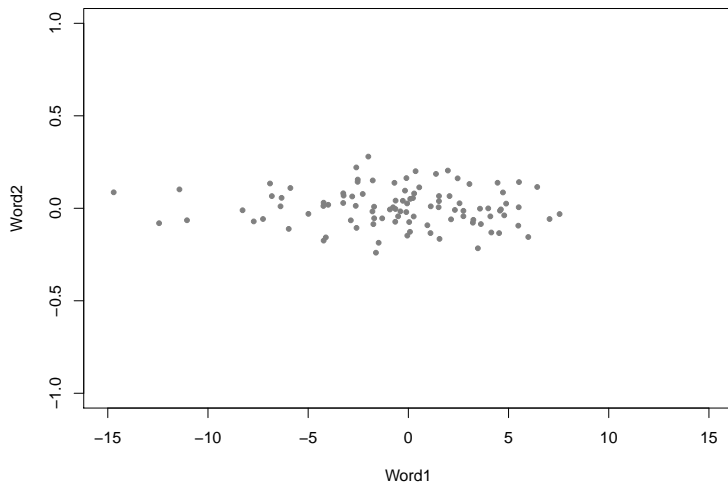
$$\mathbf{Ax} = \lambda \mathbf{x}$$

Then  $\mathbf{x}$  is an **eigenvector** and  $\lambda$  is the associated **eigenvalue**

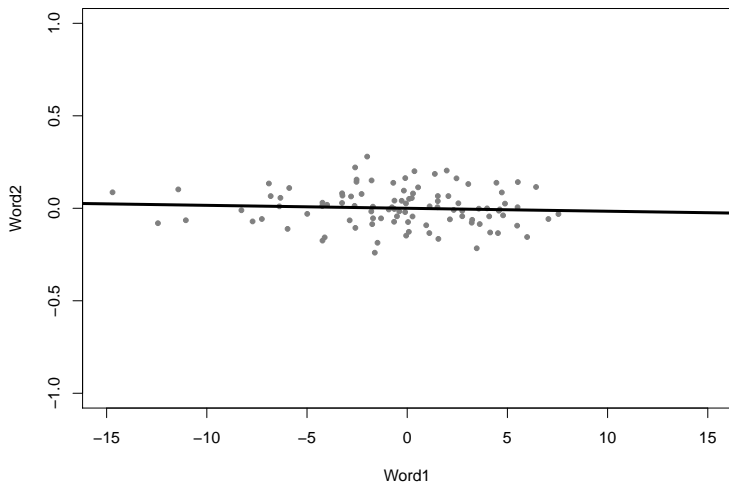
- $\mathbf{A}$  stretches the eigenvector  $\mathbf{x}$
- $\mathbf{A}$  stretches  $\mathbf{x}$  by  $\lambda$
- To find eigenvectors/values: (eigen in R )
  - Find  $\lambda$  that solves  $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$
  - Find vectors in **null space** of:

$$(\mathbf{A} - \lambda \mathbf{I}) = 0$$

# Finding a Lower Dimensional Space (Manifold Learning)

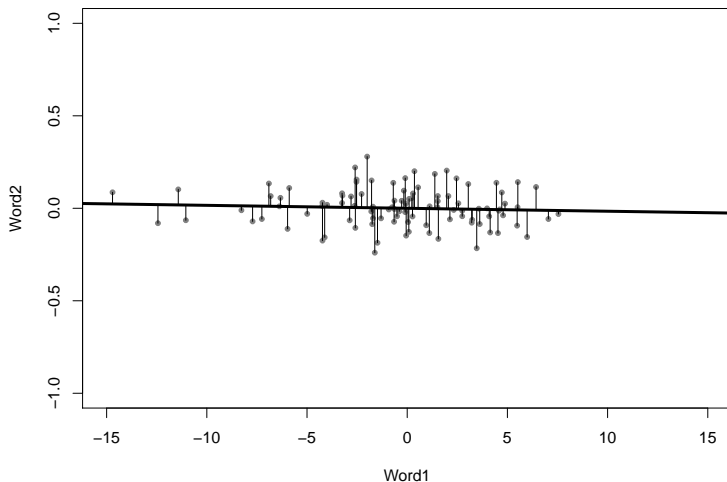


# Finding a Lower Dimensional Space (Manifold Learning)

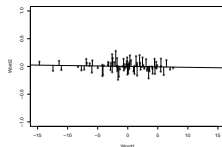




# Finding a Lower Dimensional Space (Manifold Learning)

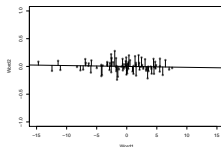


# Finding a Lower Dimensional Space (Manifold Learning)



Original data:

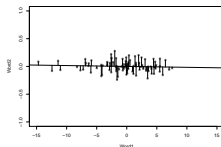
# Finding a Lower Dimensional Space (Manifold Learning)



Original data:

$$\mathbf{x}_i = (x_{i1}, x_{i2})$$

# Finding a Lower Dimensional Space (Manifold Learning)

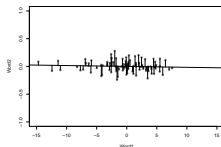


Original data:

$$\mathbf{x}_i = (x_{i1}, x_{i2})$$

Which we approximate with

# Finding a Lower Dimensional Space (Manifold Learning)



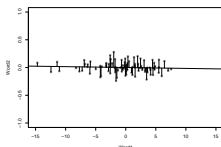
Original data:

$$\mathbf{x}_i = (x_{i1}, x_{i2})$$

Which we approximate with

$$\begin{aligned}\tilde{\mathbf{x}}_i &= z_i \mathbf{w}_1 \\ &= z_i (w_{11}, w_{12})\end{aligned}$$

# Finding a Lower Dimensional Space (Manifold Learning)



Original data  $\mathbf{x}_i \in \mathbb{R}^J$

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$$

Which we approximate with  $L \leq J$  weights  $z_{il}$  and vectors  $\mathbf{w}_l \in \mathbb{R}^J$

$$\tilde{\mathbf{x}}_i = z_{i1}\mathbf{w}_1 + z_{i2}\mathbf{w}_2 + \dots + z_{iL}\mathbf{w}_L$$

Define  $\theta = (\underbrace{\mathbf{Z}}_{N \times L}, \underbrace{\mathbf{W}_L}_{L \times J})$

# Principal Component Analysis $\rightsquigarrow$ Objective function

Consider 1-dimensional case ( $L = 1$ ), centered data, and  $\|\mathbf{w}_1\| = 1$ .

# Principal Component Analysis $\rightsquigarrow$ Objective function

Consider 1-dimensional case ( $L = 1$ ), centered data, and  $\|\mathbf{w}_1\| = 1$ .

$$f(\boldsymbol{\theta}, \mathbf{X}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - z_{i1} \mathbf{w}_1\|^2$$



# Principal Component Analysis $\rightsquigarrow$ Objective function

Consider 1-dimensional case ( $L = 1$ ), centered data, and  $\|\mathbf{w}_1\| = 1$ .

$$\begin{aligned} f(\boldsymbol{\theta}, \mathbf{X}) &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - z_{i1} \mathbf{w}_1\|^2 \\ &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1} \mathbf{w}_1)' (\mathbf{x}_i - z_{i1} \mathbf{w}_1) \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Objective function

Consider 1-dimensional case ( $L = 1$ ), centered data, and  $\|\mathbf{w}_1\| = 1$ .

$$\begin{aligned} f(\boldsymbol{\theta}, \mathbf{X}) &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - z_{i1} \mathbf{w}_1\|^2 \\ &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1} \mathbf{w}_1)' (\mathbf{x}_i - z_{i1} \mathbf{w}_1) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - 2z_{i1} \mathbf{w}_1' \mathbf{x}_i + z_{i1}^2 \right) \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Objective function

Consider 1-dimensional case ( $L = 1$ ), centered data, and  $\|\mathbf{w}_1\| = 1$ .

$$\begin{aligned} f(\boldsymbol{\theta}, \mathbf{X}) &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - z_{i1} \mathbf{w}_1\|^2 \\ &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1} \mathbf{w}_1)' (\mathbf{x}_i - z_{i1} \mathbf{w}_1) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - 2z_{i1} \mathbf{w}_1' \mathbf{x}_i + z_{i1}^2 \right) \end{aligned}$$

$$\mathbf{w}_1' \mathbf{w}_1 = 1$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Optimization:

# Principal Component Analysis $\rightsquigarrow$ Optimization

Optimization:

$$\frac{\partial f(\boldsymbol{\theta}, \mathbf{X})}{\partial z_{i1}} = -\frac{2\mathbf{w}'_1 \mathbf{x}_i + 2z_{i1}}{N}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Optimization:

$$\begin{aligned}\frac{\partial f(\boldsymbol{\theta}, \mathbf{X})}{\partial z_{i1}} &= -\frac{2\mathbf{w}'_1 \mathbf{x}_i + 2z_{i1}}{N} \\ 0 &= -\frac{2\mathbf{w}'_1 \mathbf{x}_i + 2z_{i1}^*}{N}\end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Optimization:

$$\begin{aligned}\frac{\partial f(\boldsymbol{\theta}, \mathbf{X})}{\partial z_{i1}} &= -\frac{2\mathbf{w}'_1 \mathbf{x}_i + 2z_{i1}}{N} \\ 0 &= -\frac{2\mathbf{w}'_1 \mathbf{x}_i + 2z_{i1}^*}{N} \\ z_{i1}^* &= -\mathbf{w}'_1 \mathbf{x}_i\end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Substituting in  $z_{i1}^*$



# Principal Component Analysis $\rightsquigarrow$ Optimization

Substituting in  $z_{i1}^*$

$$= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1)' (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1)$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Substituting in  $z_{i1}^*$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1)' (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \underbrace{\mathbf{x}_i' \mathbf{x}_i}_{\text{Constant}} - 2z_{i1}^* \underbrace{\mathbf{w}_1' \mathbf{x}_i}_{z_{i1}^*} + (z_{i1}^*)^2 \underbrace{\mathbf{w}_1' \mathbf{w}_1}_1 \right) \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Substituting in  $z_{i1}^*$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1)' (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \underbrace{\mathbf{x}_i' \mathbf{x}_i}_{\text{Constant}} - 2z_{i1}^* \underbrace{\mathbf{w}_1' \mathbf{x}_i}_{z_{i1}^*} + (z_{i1}^*)^2 \underbrace{\mathbf{w}_1' \mathbf{w}_1}_1 \right) \\ &= -\frac{1}{N} \sum_{i=1}^N (z_{i1}^*)^2 + c \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Substituting in  $z_{i1}^*$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1)' (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \underbrace{\mathbf{x}_i' \mathbf{x}_i}_{\text{Constant}} - 2z_{i1}^* \underbrace{\mathbf{w}_1' \mathbf{x}_i}_{z_{i1}^*} + (z_{i1}^*)^2 \underbrace{\mathbf{w}_1' \mathbf{w}_1}_1 \right) \\ &= -\frac{1}{N} \sum_{i=1}^N (z_{i1}^*)^2 + c \\ &= -\frac{1}{N} \sum_{i=1}^N \mathbf{w}_1' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_1 \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Substituting in  $z_{i1}^*$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1)' (\mathbf{x}_i - z_{i1}^* \mathbf{w}_1) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \underbrace{\mathbf{x}_i' \mathbf{x}_i}_{\text{Constant}} - 2z_{i1}^* \underbrace{\mathbf{w}_1' \mathbf{x}_i}_{z_{i1}^*} + (z_{i1}^*)^2 \underbrace{\mathbf{w}_1' \mathbf{w}_1}_1 \right) \\ &= -\frac{1}{N} \sum_{i=1}^N (z_{i1}^*)^2 + c \\ &= -\frac{1}{N} \sum_{i=1}^N \mathbf{w}_1' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_1 \\ &= -\mathbf{w}_1' \boldsymbol{\Sigma} \mathbf{w}_1 \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \boldsymbol{\Sigma} \mathbf{w}_1$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \boldsymbol{\Sigma} \mathbf{w}_1$$

where  $\boldsymbol{\Sigma}$  is the :

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$



# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

$$\mathbf{z}_1 = (\mathbf{w}_1 \mathbf{x}_1, \mathbf{w}_1 \mathbf{x}_2, \dots, \mathbf{w}_1 \mathbf{x}_N)$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \boldsymbol{\Sigma} \mathbf{w}_1$$

where  $\boldsymbol{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

$$\begin{aligned} \mathbf{z}_1 &= (\mathbf{w}_1 \mathbf{x}_1, \mathbf{w}_1 \mathbf{x}_2, \dots, \mathbf{w}_1 \mathbf{x}_N) \\ \text{var}(\mathbf{z}_1) &= E[\mathbf{z}_1^2] - E[\mathbf{z}_1]^2 \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

$$\begin{aligned} \mathbf{z}_1 &= (\mathbf{w}_1 \mathbf{x}_1, \mathbf{w}_1 \mathbf{x}_2, \dots, \mathbf{w}_1 \mathbf{x}_N) \\ \text{var}(\mathbf{z}_1) &= E[\mathbf{z}_1^2] - E[\mathbf{z}_1]^2 \\ &= \frac{1}{N} \sum_{i=1}^N z_{i1}^2 - 0 \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

$$\begin{aligned} \mathbf{z}_1 &= (\mathbf{w}_1 \mathbf{x}_1, \mathbf{w}_1 \mathbf{x}_2, \dots, \mathbf{w}_1 \mathbf{x}_N) \\ \text{var}(\mathbf{z}_1) &= E[\mathbf{z}_1^2] - E[\mathbf{z}_1]^2 \\ &= \frac{1}{N} \sum_{i=1}^N z_{i1}^2 - 0 \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{w}_1' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_1 = \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

$$\begin{aligned} \mathbf{z}_1 &= (\mathbf{w}_1 \mathbf{x}_1, \mathbf{w}_1 \mathbf{x}_2, \dots, \mathbf{w}_1 \mathbf{x}_N) \\ \text{var}(\mathbf{z}_1) &= E[\mathbf{z}_1^2] - E[\mathbf{z}_1]^2 \\ &= \frac{1}{N} \sum_{i=1}^N z_{i1}^2 - 0 \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{w}_1' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_1 = \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 \end{aligned}$$

Minimize reconstruction error

# Principal Component Analysis $\rightsquigarrow$ Optimization

$$= -\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1$$

where  $\mathbf{\Sigma}$  is the :

- Empirical covariance matrix  $\rightsquigarrow \frac{1}{N} \mathbf{X}' \mathbf{X}$
- **Variance** of the projected data. Define

$$\begin{aligned} \mathbf{z}_1 &= (\mathbf{w}_1 \mathbf{x}_1, \mathbf{w}_1 \mathbf{x}_2, \dots, \mathbf{w}_1 \mathbf{x}_N) \\ \text{var}(\mathbf{z}_1) &= E[\mathbf{z}_1^2] - E[\mathbf{z}_1]^2 \\ &= \frac{1}{N} \sum_{i=1}^N z_{i1}^2 - 0 \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{w}_1' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_1 = \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 \end{aligned}$$

Minimize reconstruction error  $\rightsquigarrow$  maximize variance of projected data

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints



# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) = \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1)$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned} g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned}g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \\ \mathbf{\Sigma} \mathbf{w}_1^* &= \lambda_1 \mathbf{w}_1^*\end{aligned}$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned}g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \\ \mathbf{\Sigma} \mathbf{w}_1^* &= \lambda_1 \mathbf{w}_1^*\end{aligned}$$

$\mathbf{w}_1^*$  = Eigenvector of  $\mathbf{\Sigma}$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned}g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \\ \mathbf{\Sigma} \mathbf{w}_1^* &= \lambda_1 \mathbf{w}_1^*\end{aligned}$$

$\mathbf{w}_1^*$  = Eigenvector of  $\mathbf{\Sigma}$  (!!!!!!!)

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned}g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \\ \mathbf{\Sigma} \mathbf{w}_1^* &= \lambda_1 \mathbf{w}_1^*\end{aligned}$$

$\mathbf{w}_1^*$  = Eigenvector of  $\mathbf{\Sigma}$  (!!!!!)

We want  $\mathbf{w}_1$  to maximize variance and

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned}g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \\ \mathbf{\Sigma} \mathbf{w}_1^* &= \lambda_1 \mathbf{w}_1^*\end{aligned}$$

$\mathbf{w}_1^*$  = Eigenvector of  $\mathbf{\Sigma}$  (!!!!!)

We want  $\mathbf{w}_1$  to maximize variance and

$$\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 = \lambda_1$$

# Principal Component Analysis $\rightsquigarrow$ Optimization

Maximize variance, subject to constraints

$$\begin{aligned}g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X}) &= \mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1) \\ \frac{\partial g(\mathbf{z}^*, \mathbf{w}_1, \mathbf{X})}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 \\ \mathbf{\Sigma} \mathbf{w}_1^* &= \lambda_1 \mathbf{w}_1^*\end{aligned}$$

$\mathbf{w}_1^*$  = Eigenvector of  $\mathbf{\Sigma}$  (!!!!!)

We want  $\mathbf{w}_1$  to maximize variance and

$$\mathbf{w}_1' \mathbf{\Sigma} \mathbf{w}_1 = \lambda_1$$

So  $\mathbf{w}_1$  is eigenvector associated with the largest eigenvalue  $\lambda_1$



# An Introduction to Eigenvectors, Values, and Diagonalization

## Theorem

Suppose  $\mathbf{A}$  is an *invertible*  $N \times N$  matrix with  $N$  linearly independent eigenvectors. Then we can write  $\mathbf{A}$  as,

$$\mathbf{A} = \mathbf{W}' \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N \end{pmatrix} \mathbf{W}$$

where  $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N)$  is an  $N \times N$  matrix with the  $N$  eigenvectors as column vectors.

# An Introduction to Eigenvectors, Values, and Diagonalization

## Definition

*Suppose  $A$  is a covariance matrix. Then, we can write  $A$  as*

$$\mathbf{A} = \mathbf{W}' \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N \end{pmatrix} \mathbf{W}$$

*Where  $\lambda_1 > \lambda_2 > \dots > \lambda_N \geq 0$ .*

*We will call  $\mathbf{w}_1$  the first eigenvector,  $\mathbf{w}_2$  the second eigenvector, ...,  $\mathbf{w}_j$  the  $j^{\text{th}}$  eigenvector.*

# Back to Principal Components

## Theorem

*Suppose we want to approximate  $N$  observations  $\mathbf{x}_i \in \mathbb{R}^J$  with  $L < J$  orthogonal-unit length vectors  $\mathbf{w}_l \in \mathbb{R}^J$  with associated scores  $z_{il}$  to minimize reconstruction error:*

# Back to Principal Components

## Theorem

*Suppose we want to approximate  $N$  observations  $\mathbf{x}_i \in \mathbb{R}^J$  with  $L < J$  orthogonal-unit length vectors  $\mathbf{w}_l \in \mathbb{R}^J$  with associated scores  $z_{il}$  to minimize reconstruction error:*

$$f(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

# Back to Principal Components

## Theorem

*Suppose we want to approximate  $N$  observations  $\mathbf{x}_i \in \mathbb{R}^J$  with  $L < J$  orthogonal-unit length vectors  $\mathbf{w}_l \in \mathbb{R}^J$  with associated scores  $z_{il}$  to minimize reconstruction error:*

$$f(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

*The optimal solution sets each  $\mathbf{w}_l$  to be the  $l^{\text{th}}$  eigenvector of the empirical covariance matrix.*

# Back to Principal Components

## Theorem

*Suppose we want to approximate  $N$  observations  $\mathbf{x}_i \in \mathbb{R}^J$  with  $L < J$  orthogonal-unit length vectors  $\mathbf{w}_l \in \mathbb{R}^J$  with associated scores  $z_{il}$  to minimize reconstruction error:*

$$f(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

*The optimal solution sets each  $\mathbf{w}_l$  to be the  $l^{\text{th}}$  eigenvector of the empirical covariance matrix. Further  $z_{il}^* = \mathbf{w}_l' \mathbf{x}_i$  so that the  $L$  dimensional representation is:*

# Back to Principal Components

## Theorem

*Suppose we want to approximate  $N$  observations  $\mathbf{x}_i \in \mathbb{R}^J$  with  $L < J$  orthogonal-unit length vectors  $\mathbf{w}_l \in \mathbb{R}^J$  with associated scores  $z_{il}$  to minimize reconstruction error:*

$$f(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

*The optimal solution sets each  $\mathbf{w}_l$  to be the  $l^{\text{th}}$  eigenvector of the empirical covariance matrix. Further  $z_{il}^* = \mathbf{w}_l' \mathbf{x}_i$  so that the  $L$  dimensional representation is:*

$$\mathbf{x}_i^L = (\mathbf{w}_1' \mathbf{x}_i, \mathbf{w}_2' \mathbf{x}_i, \dots, \mathbf{w}_L' \mathbf{x}_i)$$

# Application of Principal Components in R

Consider press releases from 2005 US Senators



# Application of Principal Components in R

Consider press releases from 2005 US Senators

Define  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$  as the rate senator  $i$  uses  $J$  words.

# Application of Principal Components in R

Consider press releases from 2005 US Senators

Define  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$  as the rate senator  $i$  uses  $J$  words.

$$x_{ij} = \frac{\text{No. Times } i \text{ uses word } j}{\text{No. words } i \text{ uses}}$$

# Application of Principal Components in R

Consider press releases from 2005 US Senators

Define  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$  as the rate senator  $i$  uses  $J$  words.

$$x_{ij} = \frac{\text{No. Times } i \text{ uses word } j}{\text{No. words } i \text{ uses}}$$

dtm:  $100 \times 2796$  matrix containing word rates for senators

# Application of Principal Components in R

Consider press releases from 2005 US Senators

Define  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})$  as the rate senator  $i$  uses  $J$  words.

$$x_{ij} = \frac{\text{No. Times } i \text{ uses word } j}{\text{No. words } i \text{ uses}}$$

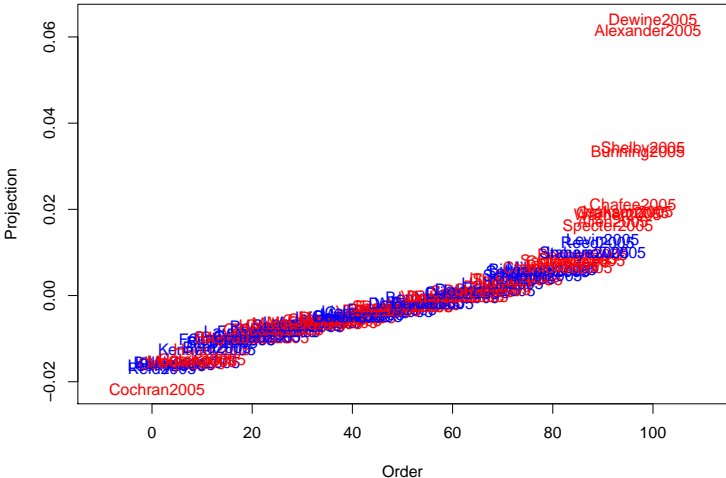
dtm:  $100 \times 2796$  matrix containing word rates for senators

prcomp(dtm) applies principal components

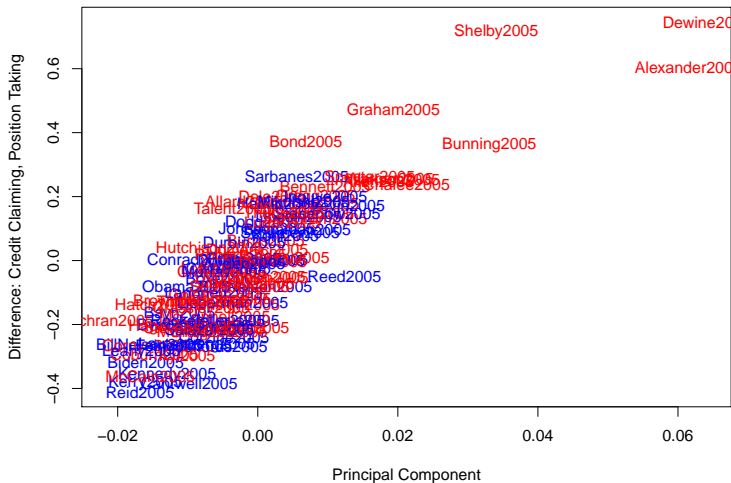
```
load("SenateTDM.RData")
dtm<- t(tdm)
for(z in 1:100){
dtm[z,]<- dtm[z,]/sum(dtm[z,])
}

store<- prcomp(dtm, scale = F)
scores<- store$x[,1]
```

# Application of Principal Components in R



# Application of Principal Components in R



# Probabilistic Principal Components (Tipping and Bishop 1999)

$$\mathbf{x}|\mathbf{w} \sim \text{Multivariate Normal}(\mathbf{Z}\mathbf{w} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

$$\mathbf{w} \sim \text{Multivariate Normal}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{x} \sim \text{Multivariate Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}' + \sigma^2\mathbf{I}$$

- 1) Log-likelihood  $\rightsquigarrow$  straightforward
- 2) Optimization via **EM**-Algorithm
- 3) Corresponds to traditional PCA is  $\lim_{\sigma^2 \rightarrow 0}$
- 4) Closely related to Factor analysis.



How do we select the number of dimensions  $L? \rightsquigarrow$  **Model**

We want to minimize reconstruction error

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ Model

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)$$

# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ **Model**

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)$$

Four types of terms:

# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ Model

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)$$

Four types of terms:

1)  $\mathbf{x}_i' \mathbf{x}_i$

# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ Model

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)$$

Four types of terms:

- 1)  $\mathbf{x}_i' \mathbf{x}_i$
- 2)  $z_{ij} z_{ik} \mathbf{w}_j' \mathbf{w}_k = z_{ij} z_{ik} 0 = 0$



# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ Model

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)$$

Four types of terms:

- 1)  $\mathbf{x}_i' \mathbf{x}_i$
- 2)  $z_{ij} z_{ik} \mathbf{w}_j' \mathbf{w}_k = z_{ij} z_{ik} 0 = 0$
- 3)  $z_{ij} z_{ij} \mathbf{w}_j' \mathbf{w}_j = z_{ij}^2$

# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ Model

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)$$

Four types of terms:

- 1)  $\mathbf{x}_i' \mathbf{x}_i$
- 2)  $z_{ij} z_{ik} \mathbf{w}_j' \mathbf{w}_k = z_{ij} z_{ik} 0 = 0$
- 3)  $z_{ij} z_{ij} \mathbf{w}_j' \mathbf{w}_j = z_{ij}^2$
- 4)  $\mathbf{x}_i' \sum_{l=1}^L z_{il} \mathbf{w}_l = \sum_{l=1}^L z_{il}^2$

# How do we select the number of dimensions $L$ ? $\rightsquigarrow$ Model

We want to minimize reconstruction error  $\rightsquigarrow$  how well did we do?

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right\|^2$$

Simplifying:

$$\begin{aligned} \text{error}(L) &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right)' \left( \mathbf{x}_i - \sum_{l=1}^L z_{il} \mathbf{w}_l \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right) \end{aligned}$$

Four types of terms:

- 1)  $\mathbf{x}_i' \mathbf{x}_i$
- 2)  $z_{ij} z_{ik} \mathbf{w}_j' \mathbf{w}_k = z_{ij} z_{ik} 0 = 0$
- 3)  $z_{ij} z_{ij} \mathbf{w}_j' \mathbf{w}_j = z_{ij}^2$
- 4)  $\mathbf{x}_i' \sum_{l=1}^L z_{il} \mathbf{w}_l = \sum_{l=1}^L z_{il}^2$

How do we select the number of dimensions  $L? \rightsquigarrow$  **Model**

$$\text{error}(L) = \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right)$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\begin{aligned}\text{error}(L) &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L \mathbf{w}_l \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l \right)\end{aligned}$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\begin{aligned}\text{error}(L) &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L \mathbf{w}_l \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l \right) \\ &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \frac{1}{N} \sum_{l=1}^L \sum_{i=1}^N \mathbf{w}_l' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l\end{aligned}$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\begin{aligned}\text{error}(L) &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L \mathbf{w}_l \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l' \right) \\ &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \frac{1}{N} \sum_{l=1}^L \sum_{i=1}^N \mathbf{w}_l' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l \\ &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \sum_{l=1}^L \mathbf{w}_l' \boldsymbol{\Sigma} \mathbf{w}_l\end{aligned}$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\begin{aligned}\text{error}(L) &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i - \sum_{l=1}^L \mathbf{w}_l \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i \right) - \frac{1}{N} \sum_{l=1}^L \sum_{i=1}^N \mathbf{w}_l' \mathbf{x}_i \mathbf{x}_i' \mathbf{w}_l \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i \right) - \sum_{l=1}^L \mathbf{w}_l' \boldsymbol{\Sigma} \mathbf{w}_l \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}_i' \mathbf{x}_i \right) - \sum_{l=1}^L \lambda_l \mathbf{w}_l' \mathbf{w}_l\end{aligned}$$



How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\begin{aligned}\text{error}(L) &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}'_i \mathbf{x}_i - \sum_{l=1}^L z_{il}^2 \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}'_i \mathbf{x}_i - \sum_{l=1}^L \mathbf{w}_l \mathbf{x}_i \mathbf{x}'_i \mathbf{w}_l \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}'_i \mathbf{x}_i \right) - \frac{1}{N} \sum_{l=1}^L \sum_{i=1}^N \mathbf{w}'_l \mathbf{x}_i \mathbf{x}'_i \mathbf{w}_l \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}'_i \mathbf{x}_i \right) - \sum_{l=1}^L \mathbf{w}'_l \boldsymbol{\Sigma} \mathbf{w}_l \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}'_i \mathbf{x}_i \right) - \sum_{l=1}^L \lambda_l \mathbf{w}'_l \mathbf{w}_l \\ &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{x}'_i \mathbf{x}_i \right) - \sum_{l=1}^L \lambda_l\end{aligned}$$

How do we select the number of dimensions  $L? \rightsquigarrow$  **Model**

If  $L = J$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

If  $L = J$

$$\text{error}(J) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}'_i \mathbf{x}_i) - \sum_{l=1}^J \lambda_l = 0$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

If  $L = J$

$$\text{error}(J) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}'_i \mathbf{x}_i) - \sum_{l=1}^J \lambda_l = 0$$

So for  $L < J$ ,

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

If  $L = J$

$$\text{error}(J) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \sum_{l=1}^J \lambda_l = 0$$

So for  $L < J$ ,

$$0 = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \left( \sum_{l=1}^L \lambda_l + \sum_{j=L+1}^J \lambda_l \right)$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

If  $L = J$

$$\text{error}(J) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \sum_{l=1}^J \lambda_l = 0$$

So for  $L < J$ ,

$$0 = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \left( \sum_{l=1}^L \lambda_l + \sum_{j=L+1}^J \lambda_j \right)$$

$$\sum_{j=L+1}^J \lambda_j = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i' \mathbf{x}_i) - \sum_{l=1}^L \lambda_l$$

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

If  $L = J$

$$\text{error}(J) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}'_i \mathbf{x}_i) - \sum_{l=1}^J \lambda_l = 0$$

So for  $L < J$ ,

$$0 = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}'_i \mathbf{x}_i) - \left( \sum_{l=1}^L \lambda_l + \sum_{j=L+1}^J \lambda_j \right)$$

$$\sum_{j=L+1}^J \lambda_j = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}'_i \mathbf{x}_i) - \sum_{l=1}^L \lambda_l$$

$$\sum_{j=L+1}^J \lambda_j = \text{error}(L)$$

How do we select the number of dimensions  $L? \rightsquigarrow$  **Model**

$$\sum_{j=L+1}^J \lambda_j = \text{error}(L)$$



How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\sum_{j=L+1}^J \lambda_j = \text{error}(L)$$

- Error = Sum of “remaining” eigenvalues

How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\sum_{j=L+1}^J \lambda_j = \text{error}(L)$$

- Error = Sum of “remaining” eigenvalues
- Total variance explained = (sum of included eigenvalues)/(sum of all eigenvalues)

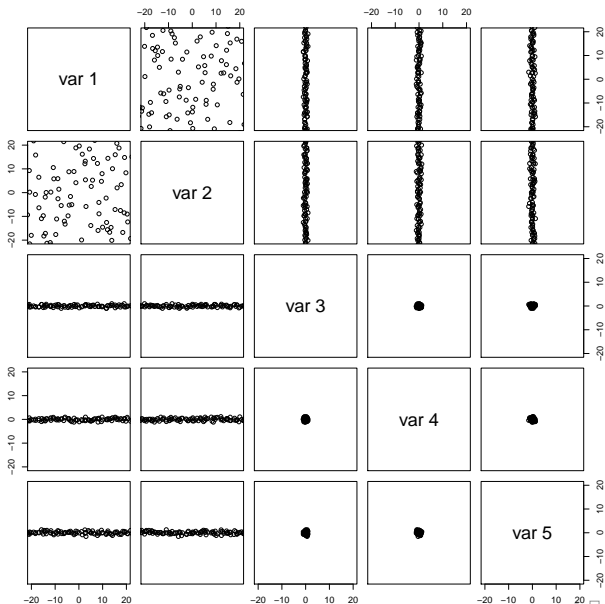
How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**

$$\sum_{j=L+1}^J \lambda_j = \text{error}(L)$$

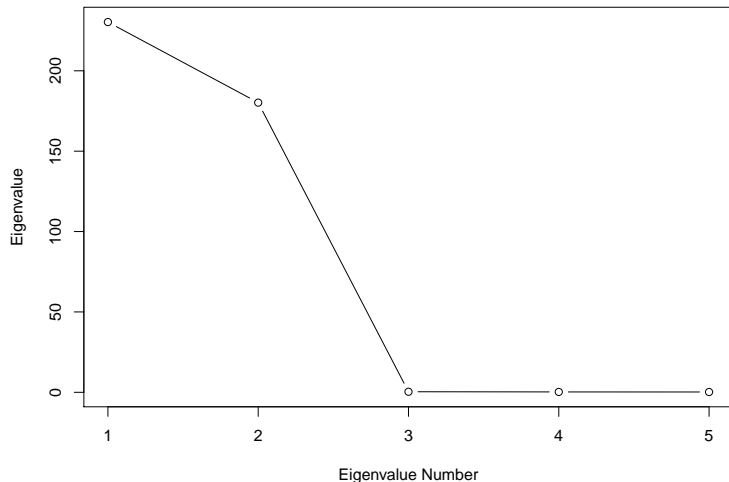
- Error = Sum of “remaining” eigenvalues
- Total variance explained = (sum of included eigenvalues)/(sum of all eigenvalues)

Recommendation  $\rightsquigarrow$  look for Elbow

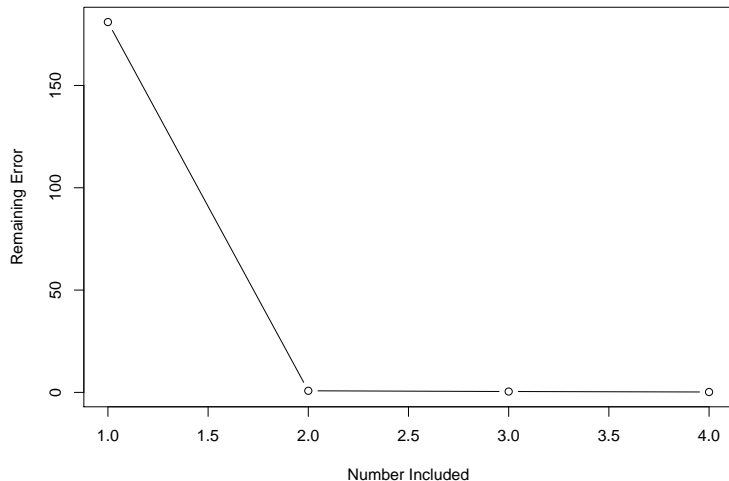
How do we select the number of dimensions  $L? \rightsquigarrow$  **Model**



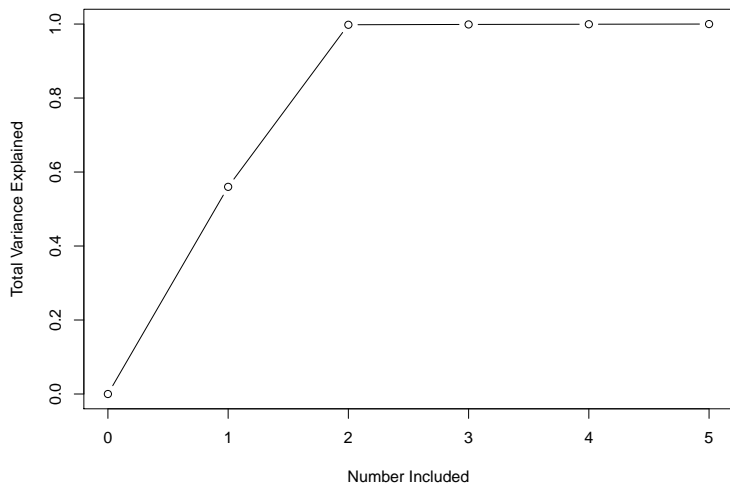
How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  Model



How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**



How do we select the number of dimensions  $L$ ?  $\rightsquigarrow$  **Model**



# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ?



# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ? J

# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ?  $J$ (!!!!)

# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ?  $J$ (!!!!)

- Attempts to assess dimensionality require a **model**  $\rightsquigarrow$  some way to tradeoff accuracy of reconstruction with simplicity

# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ?  $J$ (!!!!)

- Attempts to assess dimensionality require a **model**  $\rightsquigarrow$  some way to tradeoff accuracy of reconstruction with simplicity
- **Any** answer (no matter how creatively obtained) supposes **you have the right function to measure tradeoff**

# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ?  $J$ (!!!!)

- Attempts to assess dimensionality require a **model**  $\rightsquigarrow$  some way to tradeoff accuracy of reconstruction with simplicity
- **Any** answer (no matter how creatively obtained) supposes **you have the right function to measure tradeoff**
- The “right” number of dimensions depends on the **task** you have in mind

# Non-model based evaluations: What's the point?

What is the true underlying dimensionality of  $\mathbf{X}$ ?  $J(!!!!)$

- Attempts to assess dimensionality require a **model**  $\rightsquigarrow$  some way to tradeoff accuracy of reconstruction with simplicity
- **Any** answer (no matter how creatively obtained) supposes **you have the right function to measure tradeoff**
- The “right” number of dimensions depends on the **task** you have in mind

Mathematical model  $\rightsquigarrow$  insufficient to make modeling decision

# Appendix

# Kernel Principal Component Analysis

Define a **Kernel** ( $N \times N$ ) matrix as:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

where  $k(\cdot, \cdot)$  is a function that behaves like a similarity function.



# Kernel Principal Component Analysis

Define a **Kernel** ( $N \times N$ ) matrix as:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

where  $k(\cdot, \cdot)$  is a function that behaves like a similarity function. Where we suppose this matrix emerges from applying  $\phi : \mathcal{R}^J \rightarrow \mathcal{R}^M$  to the data and then taking the inner product:

# Kernel Principal Component Analysis

Define a **Kernel** ( $N \times N$ ) matrix as:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

where  $k(\cdot, \cdot)$  is a function that behaves like a similarity function. Where we suppose this matrix emerges from applying  $\phi : \mathcal{R}^J \rightarrow \mathcal{R}^M$  to the data and then taking the inner product:

$$\mathbf{K} = \mathbf{\Phi} \mathbf{\Phi}' \text{ (The inner product matrix)}$$

# Kernel Principal Component Analysis

Define a **Kernel** ( $N \times N$ ) matrix as:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

where  $k(\cdot, \cdot)$  is a function that behaves like a similarity function. Where we suppose this matrix emerges from applying  $\phi : \mathcal{R}^J \rightarrow \mathcal{R}^M$  to the data and then taking the inner product:

$$\mathbf{K} = \mathbf{\Phi} \mathbf{\Phi}' \text{ (The inner product matrix)}$$

$$= \begin{pmatrix} \phi(\mathbf{x}_1)' \phi(\mathbf{x}_1) & \phi(\mathbf{x}_1)' \phi(\mathbf{x}_2) & \dots & \phi(\mathbf{x}_1)' \phi(\mathbf{x}_N) \\ \phi(\mathbf{x}_2)' \phi(\mathbf{x}_1) & \phi(\mathbf{x}_2)' \phi(\mathbf{x}_2) & \dots & \phi(\mathbf{x}_2)' \phi(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N)' \phi(\mathbf{x}_1) & \phi(\mathbf{x}_N)' \phi(\mathbf{x}_2) & \dots & \phi(\mathbf{x}_N)' \phi(\mathbf{x}_N) \end{pmatrix}$$

# Kernel Principal Component Analysis

Define a **Kernel** ( $N \times N$ ) matrix as:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

where  $k(\cdot, \cdot)$  is a function that behaves like a similarity function. Where we suppose this matrix emerges from applying  $\phi : \mathcal{R}^J \rightarrow \mathcal{R}^M$  to the data and then taking the inner product:

$$\begin{aligned} \mathbf{K} &= \mathbf{\Phi} \mathbf{\Phi}' \text{ (The inner product matrix)} \\ &= \begin{pmatrix} \phi(\mathbf{x}_1)' \phi(\mathbf{x}_1) & \phi(\mathbf{x}_1)' \phi(\mathbf{x}_2) & \dots & \phi(\mathbf{x}_1)' \phi(\mathbf{x}_N) \\ \phi(\mathbf{x}_2)' \phi(\mathbf{x}_1) & \phi(\mathbf{x}_2)' \phi(\mathbf{x}_2) & \dots & \phi(\mathbf{x}_2)' \phi(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N)' \phi(\mathbf{x}_1) & \phi(\mathbf{x}_N)' \phi(\mathbf{x}_2) & \dots & \phi(\mathbf{x}_N)' \phi(\mathbf{x}_N) \end{pmatrix} \end{aligned}$$

Compute PCA of  $\mathbf{\Phi}$  from  $\mathbf{\Phi} \mathbf{\Phi}'$

# Kernel PCA

PCA of  $\mathbf{X}$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ .

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then



# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$(\mathbf{X}\mathbf{X}')\mathbf{u}_1 = \lambda_1\mathbf{u}_1$$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1)\end{aligned}$$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

But  $\mathbf{v}_1$  needs unit length, and

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

But  $\mathbf{v}_1$  needs unit length, and

$$\|\mathbf{v}_1\|^2 = \mathbf{v}_1'\mathbf{v}_1$$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

But  $\mathbf{v}_1$  needs unit length, and

$$\begin{aligned}\|\mathbf{v}_1\|^2 &= \mathbf{v}_1'\mathbf{v}_1 \\&= \mathbf{u}_1'\mathbf{X}\mathbf{X}'\mathbf{u}_1\end{aligned}$$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

But  $\mathbf{v}_1$  needs unit length, and

$$\begin{aligned}\|\mathbf{v}_1\|^2 &= \mathbf{v}_1'\mathbf{v}_1 \\&= \mathbf{u}_1'\mathbf{X}\mathbf{X}'\mathbf{u}_1 \\&= \lambda_1\mathbf{u}_1'\mathbf{u}_1 = \lambda_1\end{aligned}$$

# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

But  $\mathbf{v}_1$  needs unit length, and

$$\begin{aligned}\|\mathbf{v}_1\|^2 &= \mathbf{v}_1'\mathbf{v}_1 \\&= \mathbf{u}_1'\mathbf{X}\mathbf{X}'\mathbf{u}_1 \\&= \lambda_1\mathbf{u}_1'\mathbf{u}_1 = \lambda_1\end{aligned}$$

So first eigenvector of  $\mathbf{X}'\mathbf{X}$  is



# Kernel PCA

PCA of  $\mathbf{X}$  Eigenvectors of  $\mathbf{X}'\mathbf{X}$  ( $\frac{1}{N}$  doesn't affect eigenvectors)

Suppose  $\mathbf{u}_1$  is an eigenvector for  $\mathbf{X}\mathbf{X}'$ , with value  $\lambda_1$ . Then

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_1 &= \lambda_1\mathbf{u}_1 \\(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{u}_1) &= \lambda_1(\mathbf{X}'\mathbf{u}_1) \\&= \lambda_1\mathbf{v}_1\end{aligned}$$

But  $\mathbf{v}_1$  needs unit length, and

$$\begin{aligned}\|\mathbf{v}_1\|^2 &= \mathbf{v}_1'\mathbf{v}_1 \\&= \mathbf{u}_1'\mathbf{X}\mathbf{X}'\mathbf{u}_1 \\&= \lambda_1\mathbf{u}_1'\mathbf{u}_1 = \lambda_1\end{aligned}$$

So first eigenvector of  $\mathbf{X}'\mathbf{X}$  is

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}}\mathbf{X}'\mathbf{u}_1$$

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ .

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

$$\phi(\mathbf{x}_i)' \mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \phi(\mathbf{x}_i)' \mathbf{\Phi}' \mathbf{u}_1$$

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

$$\phi(\mathbf{x}_i)' \mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \phi(\mathbf{x}_i)' \mathbf{\Phi}' \mathbf{u}_1$$

$$\phi(\mathbf{x}_i)' \mathbf{\Phi}' = \left[ \phi(\mathbf{x}_i)' \phi(\mathbf{x}_1), \phi(\mathbf{x}_i)' \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_i)' \phi(\mathbf{x}_N) \right]$$



# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

$$\begin{aligned}\phi(\mathbf{x}_i)' \mathbf{w}_1 &= \frac{1}{\sqrt{\lambda_1}} \phi(\mathbf{x}_i)' \mathbf{\Phi}' \mathbf{u}_1 \\ \phi(\mathbf{x}_i)' \mathbf{\Phi}' &= \left[ \phi(\mathbf{x}_i)' \phi(\mathbf{x}_1), \phi(\mathbf{x}_i)' \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_i)' \phi(\mathbf{x}_N) \right] \\ &= [k(\mathbf{x}_i, \mathbf{x}_1), k(\mathbf{x}_i, \mathbf{x}_2), \dots, k(\mathbf{x}_i, \mathbf{x}_N)]\end{aligned}$$

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

$$\phi(\mathbf{x}_i)' \mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \phi(\mathbf{x}_i)' \mathbf{\Phi}' \mathbf{u}_1$$

$$\begin{aligned} \phi(\mathbf{x}_i)' \mathbf{\Phi}' &= \left[ \phi(\mathbf{x}_i)' \phi(\mathbf{x}_1), \phi(\mathbf{x}_i)' \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_i)' \phi(\mathbf{x}_N) \right] \\ &= [k(\mathbf{x}_i, \mathbf{x}_1), k(\mathbf{x}_i, \mathbf{x}_2), \dots, k(\mathbf{x}_i, \mathbf{x}_N)] = \mathbf{k}(\mathbf{x}_i, *) \end{aligned}$$

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

$$\begin{aligned}\phi(\mathbf{x}_i)' \mathbf{w}_1 &= \frac{1}{\sqrt{\lambda_1}} \phi(\mathbf{x}_i)' \mathbf{\Phi}' \mathbf{u}_1 \\ \phi(\mathbf{x}_i)' \mathbf{\Phi}' &= \left[ \phi(\mathbf{x}_i)' \phi(\mathbf{x}_1), \phi(\mathbf{x}_i)' \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_i)' \phi(\mathbf{x}_N) \right] \\ &= [k(\mathbf{x}_i, \mathbf{x}_1), k(\mathbf{x}_i, \mathbf{x}_2), \dots, k(\mathbf{x}_i, \mathbf{x}_N)] = \mathbf{k}(\mathbf{x}_i, *)\end{aligned}$$

Then, we can obtain projection for observation  $i$  using Kernel with

# Kernel PCA

$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}'$  (assume  $\mathbf{\Phi}$  is mean-centered, for now)

We can obtain  $\mathbf{u}_1$  and  $\lambda_1$  from  $\mathbf{K}$ . We know that

$$\mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \underbrace{\mathbf{\Phi}'}_{\text{Unknown}} \mathbf{u}_1$$

But suppose we want to project a point  $\phi(\mathbf{x}_i)$ , then

$$\begin{aligned}\phi(\mathbf{x}_i)' \mathbf{w}_1 &= \frac{1}{\sqrt{\lambda_1}} \phi(\mathbf{x}_i)' \mathbf{\Phi}' \mathbf{u}_1 \\ \phi(\mathbf{x}_i)' \mathbf{\Phi}' &= [\phi(\mathbf{x}_i)' \phi(\mathbf{x}_1), \phi(\mathbf{x}_i)' \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_i)' \phi(\mathbf{x}_N)] \\ &= [k(\mathbf{x}_i, \mathbf{x}_1), k(\mathbf{x}_i, \mathbf{x}_2), \dots, k(\mathbf{x}_i, \mathbf{x}_N)] = \mathbf{k}(\mathbf{x}_i, *)\end{aligned}$$

Then, we can obtain projection for observation  $i$  using Kernel with

$$\phi(\mathbf{x}_i)' \mathbf{w}_1 = \frac{1}{\sqrt{\lambda_1}} \mathbf{k}(\mathbf{x}_i, *) \mathbf{u}_1$$

# Kernel PCA

Center  $\mathbf{K}$ ?

Use centering matrix  $\mathbf{H}$

$$\begin{aligned}\mathbf{H} &= \mathbf{I}_N - \frac{(\mathbf{1}_N \mathbf{1}_N')}{N} \\ \mathbf{K}_{\text{center}} &= \mathbf{H} \mathbf{K} \mathbf{H}\end{aligned}$$

# Spirling and Indian Treaties

Spirling (2013): model Treaties between US and Native Americans

# Spirling and Indian Treaties

**Spirling (2013):** model Treaties between US and Native Americans  
Why?

# Spirling and Indian Treaties

**Spirling (2013):** model Treaties between US and Native Americans  
Why?

- American political development



# Spirling and Indian Treaties

**Spirling (2013):** model Treaties between US and Native Americans  
Why?

- American political development
- IR Theories of Treaties and Treaty Violations

# Spirling and Indian Treaties

**Spirling (2013):** model Treaties between US and Native Americans  
Why?

- American political development
- IR Theories of Treaties and Treaty Violations
- Comparative studies of indigenous/colonialist interaction

# Spirling and Indian Treaties

**Spirling (2013):** model Treaties between US and Native Americans  
Why?

- American political development
- IR Theories of Treaties and Treaty Violations
- Comparative studies of indigenous/colonialist interaction
- **Political Science question:** how did Native Americans lose land so quickly?

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?  
After stemming, stopping, bag of wording:

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.



# Spirling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spirling uses complicated representation of texts to preserve word order

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order~>  
broad application

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Between Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order↪  
broad application

Peace Between Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Between Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Between Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace **B**etween Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace **Between** Us



# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace **Between** Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace **Be**tween Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Bet**we**en Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Bet**ween** Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order↪  
broad application

Peace Betw**een** Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Between **en** Us

# Spiraling and Indian Treaties

How do we preserve word order and semantic language?

After stemming, stopping, bag of wording:

- Peace Between Us
- No Peace Between Us

are identical.

Spiraling uses complicated representation of texts to preserve word order ~>  
broad application

Peace Between **n** **Us**

# Spiraling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.



# Spiraling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

# Spirling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

Spirling examines 5-character strings,  $s \in \mathcal{A}$

# Spiraling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

Spiraling examines 5-character strings,  $s \in \mathcal{A}$

Define:

# Spirling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

Spirling examines 5-character strings,  $s \in \mathcal{A}$

Define:

$\phi_s : \mathcal{X} \rightarrow \mathbb{R}$  as a function that counts the number of times string  $s$  occurs in document  $\mathbf{x}$ .

# Spiraling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

Spiraling examines 5-character strings,  $s \in \mathcal{A}$

Define:

$\phi_s : \mathcal{X} \rightarrow \mathbb{R}$  as a function that counts the number of times string  $s$  occurs in document  $\mathbf{x}$ .

Define **string kernel** to be,

# Spiraling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

Spiraling examines 5-character strings,  $s \in \mathcal{A}$

Define:

$\phi_s : \mathcal{X} \rightarrow \mathbb{R}$  as a function that counts the number of times string  $s$  occurs in document  $\mathbf{x}$ .

Define **string kernel** to be,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s \in \mathcal{A}} w_s \phi_s(\mathbf{x}_i) \phi_s(\mathbf{x}_j)$$

# Spiraling and Indian Treaties

Consider documents  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , where we have preserved order, punctuation, and all else.

We say  $\mathbf{x}_i \in \mathcal{X}$

Spiraling examines 5-character strings,  $s \in \mathcal{A}$

Define:

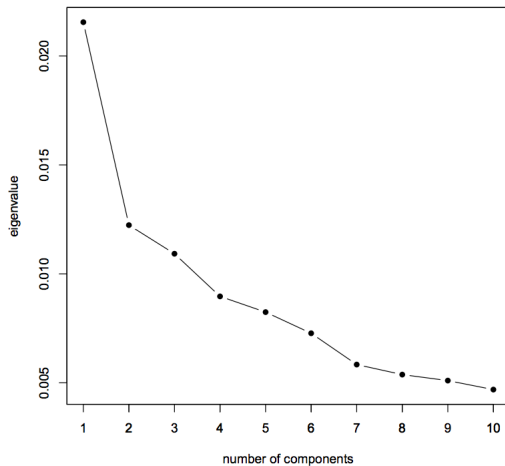
$\phi_s : \mathcal{X} \rightarrow \mathbb{R}$  as a function that counts the number of times string  $s$  occurs in document  $\mathbf{x}$ .

Define **string kernel** to be,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s \in \mathcal{A}} w_s \phi_s(\mathbf{x}_i) \phi_s(\mathbf{x}_j)$$

$\phi(\mathbf{x}_i) \approx \binom{32}{5}$  element long count vector

# Spirling and Indian Treaties





# Spirling and Indian Treaties

