# Using network theory to detect concept drift

Ezra Nwobodo

University of Bristol

March 22, 2024

# Overview

# What is concept drift and why is it important?

- The underlying distribution of data changes over time.

# What is concept drift and why is it important?

- The underlying distribution of data changes over time.
- Can help us better understand data dynamics.

# What is concept drift and why is it important?

- The underlying distribution of data changes over time.

- Can help us better understand data dynamics.
- Can degrade the performance of a predictive model.

# Illustrative example: Clustering in Customer Purchase Data

- $\mathbf{C}_{t_1}$ and $\mathbf{C}_{t_2}$ represent the clustering configurations at times 1 and 2.
- We say that concept drift has occurred if

$$\mathbf{C}_{t_1} \neq \mathbf{C}_{t_2}.$$

# Illustrative example: Clustering in Customer Purchase Data

- $\mathbf{C}_{t_1}$ and $\mathbf{C}_{t_2}$ represent the clustering configurations at times 1 and 2.
- We say that concept drift has occurred if

$$\mathbf{C}_{t_1} \neq \mathbf{C}_{t_2}.$$

- Detection and adaptation to concept drift ensures that strategies are up-to-date.

# Unsupervised Concept Drift

- Between times $t_0$ and $t_1$ we observe concept drift if, for $n$-dimensional data $\mathbf{X}$,

$$\mathbb{P}_{t_0}(\mathbf{X}) \neq \mathbb{P}_{t_1}(\mathbf{X}).$$

- Between times $t_0$ and $t_1$ we observe concept drift if, for $n$-dimensional data $\mathbf{X}$,
$$\mathbb{P}_{t_0}(\mathbf{X}) \neq \mathbb{P}_{t_1}(\mathbf{X}).$$
- No response variable $Y$ - lack of a clear signal.

# Unsupervised Concept Drift

- Between times $t_0$ and $t_1$ we observe concept drift if, for $n$-dimensional data $\mathbf{X}$,
$$\mathbb{P}_{t_0}(\mathbf{X}) \neq \mathbb{P}_{t_1}(\mathbf{X}).$$
- No response variable $Y$ - lack of a clear signal.
- Difficult for the model to adapt.

# Complications from High-Dimensional Data

- The curse of dimensionality

# Complications from High-Dimensional Data

- The curse of dimensionality
- Interpretability

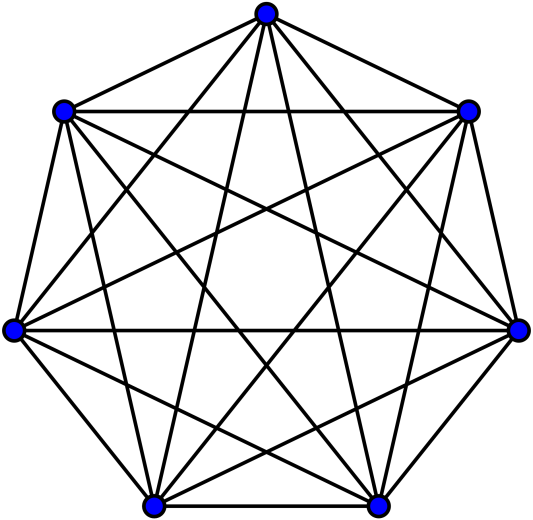
# Existing methods

In the context of unsupervised learning with high-dimensional data:

- Lack of interpretable methods for visualising concept drift, especially for visualising the drift between all covariates at once.

# Framing the problem

# Framing the problem



# Framing the problem

- A network that discretely changes over time with $n$ nodes (covariates).

# Framing the problem

- A network that discretely changes over time with $n$ nodes (covariates).
- Adjacency matrix $\mathbf{A}^{(t)} \in \mathbb{R}^{n\times n}$,
- $\mathbf{A}_{ij}^{(t)} = Cov(\mathbf{c}_i, \mathbf{c}_j)$ at time $t$ with $\mathbf{c}_1, \ldots, \mathbf{c}_n \in \mathbb{R}^m$, representing $m$ observations.

# Unfolded Adjacent Spectral Embedding

- Dimensionality reduction technique.
- $d$-dimensional representation of data.
- Embed adjacency matrices at different time points while maintaining *stability*.

# Unfolded Adjacent Spectral Embedding

- Dimensionality reduction technique.
- $d$-dimensional representation of data.
- Embed adjacency matrices at different time points while maintaining *stability*.
- Extension of singular value decomposition (SVD).

# Unfolded Adjacent Spectral Embedding

- Define the *unfolding* $\mathbf{A} = [\mathbf{A}^{(1)} | \cdots | \mathbf{A}^{(T)}] \in \mathbb{R}^{n \times nT}$,

# Unfolded Adjacent Spectral Embedding

- Define the *unfolding* $\mathbf{A} = [\mathbf{A}^{(1)}| \cdots |\mathbf{A}^{(T)}] \in \mathbb{R}^{n \times nT}$,
- Compute the (truncated) singular value decomposition:

$$\mathbf{A} \approx \mathbf{U_A \Sigma_A V_A^\top},$$

# Unfolded Adjacent Spectral Embedding

- Define the *unfolding* $\mathbf{A} = [\mathbf{A}^{(1)} | \cdots | \mathbf{A}^{(T)}] \in \mathbb{R}^{n \times nT}$,
- Compute the (truncated) singular value decomposition:

$$\mathbf{A} \approx \mathbf{U_A \Sigma_A V_A^\top},$$

- $\mathbf{\Sigma_A} \in \mathbb{R}^{d \times d}$ diagonal and contains the $d$ largest singular values of $\mathbf{A}$,

# Unfolded Adjacent Spectral Embedding

- Define the *unfolding* $\mathbf{A} = [\mathbf{A}^{(1)}| \cdots |\mathbf{A}^{(T)}] \in \mathbb{R}^{n \times nT}$,
- Compute the (truncated) singular value decomposition:

$$\mathbf{A} \approx \mathbf{U}_{\mathbf{A}} \mathbf{\Sigma}_{\mathbf{A}} \mathbf{V}_{\mathbf{A}}^{\top},$$

- $\mathbf{\Sigma}_{\mathbf{A}} \in \mathbb{R}^{d \times d}$ diagonal and contains the $d$ largest singular values of $\mathbf{A}$,
- $\mathbf{U}_{\mathbf{A}} \in \mathbb{O}^{n \times d}$ and $\mathbf{V}_{\mathbf{A}} \in \mathbb{O}^{nT \times d}$ contain the left and right singular vectors respectively.

# Unfolded spectral embedding

$$\mathbf{A} \approx \mathbf{U_A \Sigma_A V_A^\top} =$$

# Unfolded spectral embedding

$$\mathbf{A} \approx \mathbf{U_A \Sigma_A V_A^\top} = \mathbf{U_A \Sigma_A^{1/2} \Sigma_A^{1/2} V_A^\top}$$

# Unfolded spectral embedding

$$\mathbf{A} \approx \mathbf{U_A}\mathbf{\Sigma_A}\mathbf{V_A}^\top = \mathbf{U_A}\mathbf{\Sigma_A}^{1/2}\mathbf{\Sigma_A}^{1/2}\mathbf{V_A}^\top = \hat{\mathbf{X}}\hat{\mathbf{Y}}^\top$$

# Unfolded spectral embedding

$$\mathbf{A} \approx \mathbf{U_A \Sigma_A V_A^\top} = \mathbf{U_A \Sigma_A^{1/2} \Sigma_A^{1/2} V_A^\top} = \hat{\mathbf{X}}\hat{\mathbf{Y}}^\top$$

- Embedding: $\hat{\mathbf{Y}} = \mathbf{V_A \Sigma_A^{1/2}} \in \mathbb{R}^{nT \times d}$.

# Unfolded spectral embedding

$$\mathbf{A} \approx \mathbf{U}_{\mathbf{A}}\mathbf{\Sigma}_{\mathbf{A}}\mathbf{V}_{\mathbf{A}}^{\top} = \mathbf{U}_{\mathbf{A}}\mathbf{\Sigma}_{\mathbf{A}}^{1/2}\mathbf{\Sigma}_{\mathbf{A}}^{1/2}\mathbf{V}_{\mathbf{A}}^{\top} = \hat{\mathbf{X}}\hat{\mathbf{Y}}^{\top}$$

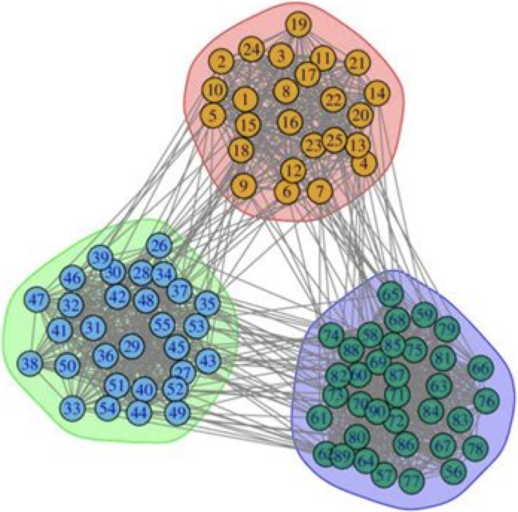
- Embedding: $\hat{\mathbf{Y}} = \mathbf{V}_{\mathbf{A}}\mathbf{\Sigma}_{\mathbf{A}}^{1/2} \in \mathbb{R}^{nT \times d}$.
- Divide $\hat{\mathbf{Y}}$ into $T$ blocks $\hat{\mathbf{Y}}^{(t)}$,

# Unfolded spectral embedding

$$\mathbf{A} \approx \mathbf{U_A}\mathbf{\Sigma_A}\mathbf{V_A}^\top = \mathbf{U_A}\mathbf{\Sigma_A}^{1/2}\mathbf{\Sigma_A}^{1/2}\mathbf{V_A}^\top = \hat{\mathbf{X}}\hat{\mathbf{Y}}^\top$$

- Embedding: $\hat{\mathbf{Y}} = \mathbf{V_A}\mathbf{\Sigma_A}^{1/2} \in \mathbb{R}^{nT\times d}$.
- Divide $\hat{\mathbf{Y}}$ into $T$ blocks $\hat{\mathbf{Y}}^{(t)}$,
- Rows $\hat{\mathbf{Y}}_i$ of each block are the $d$-dimensional points to be plotted to represent the drift at time $t$.

# Dynamic Stochastic Block Model



# Toy Model

Connection probabilities:

$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.24 & 0.06 & 0.54 & 0.06 \\ 0.06 & 0.60 & 0.12 & 0.06 \\ 0.54 & 0.12 & 0.06 & 0.10 \\ 0.06 & 0.06 & 0.10 & 0.18 \end{pmatrix}, \ \mathbf{B}^{(2)} = \begin{pmatrix} 0.48 & 0.48 & 0.12 & 0.06 \\ 0.48 & 0.48 & 0.12 & 0.06 \\ 0.12 & 0.12 & 0.27 & 0.10 \\ 0.06 & 0.06 & 0.10 & 0.18 \end{pmatrix}.$$

# Toy Model

Connection probabilities:

$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.24 & 0.06 & 0.54 & 0.06 \\ 0.06 & 0.60 & 0.12 & 0.06 \\ 0.54 & 0.12 & 0.06 & 0.10 \\ 0.06 & 0.06 & 0.10 & 0.18 \end{pmatrix}, \ \mathbf{B}^{(2)} = \begin{pmatrix} 0.48 & 0.48 & 0.12 & 0.06 \\ 0.48 & 0.48 & 0.12 & 0.06 \\ 0.12 & 0.12 & 0.27 & 0.10 \\ 0.06 & 0.06 & 0.10 & 0.18 \end{pmatrix}.$$

- At $t = 2$ communities 1 and 2 have merged.

# Toy Model

Connection probabilities:

$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.24 & 0.06 & 0.54 & 0.06 \\ 0.06 & 0.60 & 0.12 & 0.06 \\ 0.54 & 0.12 & 0.06 & 0.10 \\ 0.06 & 0.06 & 0.10 & 0.18 \end{pmatrix}, \ \mathbf{B}^{(2)} = \begin{pmatrix} 0.48 & 0.48 & 0.12 & 0.06 \\ 0.48 & 0.48 & 0.12 & 0.06 \\ 0.12 & 0.12 & 0.27 & 0.10 \\ 0.06 & 0.06 & 0.10 & 0.18 \end{pmatrix}.$$

- At $t = 2$ communities 1 and 2 have merged.
- Community 4 has same connection probabilities.

# What makes a successful embedding?

- Cross-sectional stability: The embeddings for communities 1 and 2 at time 2 are close.
- Longitudinal stability: The embeddings for community 4 at times 1 and 2 are close.

# Generating our data frames

**Input:** Connection matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$.
**Output:** Data frames $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)} \in \mathbb{R}^{m \times n}$.

# Generating our data frames

**Input:** Connection matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$.
**Output:** Data frames $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)} \in \mathbb{R}^{m \times n}$.

1: Divide covariates into $k = 4$ communities

# Generating our data frames

**Input:** Connection matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$.
**Output:** Data frames $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)} \in \mathbb{R}^{m \times n}$.

1: Divide covariates into $k = 4$ communities
2: Use connection probabilities to simulate adjacency matrix $\mathbf{C}^{(1)}$ where $\mathbf{C}_{ij}^{(1)} = 1$ if and only if $\mathbf{c}_i$ and $\mathbf{c}_j$ are connected

# Generating our data frames

**Input:** Connection matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$.
**Output:** Data frames $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)} \in \mathbb{R}^{m \times n}$.

1: Divide covariates into $k = 4$ communities
2: Use connection probabilities to simulate adjacency matrix $\mathbf{C}^{(1)}$ where $\mathbf{C}_{ij}^{(1)} = 1$ if and only if $\mathbf{c}_i$ and $\mathbf{c}_j$ are connected
3: **for** observation $k \in [m]$ **do**
4: Simulate $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

# Generating our data frames

**Input:** Connection matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$.
**Output:** Data frames $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)} \in \mathbb{R}^{m \times n}$.

1: Divide covariates into $k = 4$ communities
2: Use connection probabilities to simulate adjacency matrix $\mathbf{C}^{(1)}$ where $\mathbf{C}_{ij}^{(1)} = 1$ if and only if $\mathbf{c}_i$ and $\mathbf{c}_j$ are connected
3: **for** observation $k \in [m]$ **do**
4: Simulate $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
5: **for** $i \in [n]$ **do**
6: $\mathbf{D}_{k,i}^{(t)} = \sum_j \mathbf{x}_j$ for j where $\mathbf{c}_i$ and $\mathbf{c}_j$ are connected

# Generating our data frames

**Input:** Connection matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$.
**Output:** Data frames $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)} \in \mathbb{R}^{m \times n}$.

1: Divide covariates into $k = 4$ communities
2: Use connection probabilities to simulate adjacency matrix $\mathbf{C}^{(1)}$ where $\mathbf{C}^{(1)}_{ij} = 1$ if and only if $\mathbf{c}_i$ and $\mathbf{c}_j$ are connected
3: **for** observation $k \in [m]$ **do**
4: Simulate $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x}_1, \ldots, \mathbf{x}_n \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$
5: **for** $i \in [n]$ **do**
6: $\mathbf{D}^{(t)}_{k,i} = \sum_j \mathbf{x}_j$ for j where $\mathbf{c}_i$ and $\mathbf{c}_j$ are connected
7: **end for**
8: **end for**

# Computing the embedding

- Separately compute $Cov(\mathbf{D}^{(1)})$ and $Cov(\mathbf{D}^{(2)})$.
- Let $\mathbf{A}^{(1)} = \text{Hollow}(Cov(\mathbf{D}^{(1)}))$ and $\mathbf{A}^{(2)} = \text{Hollow}(Cov(\mathbf{D}^{(2)}))$ be our adjacency matrices.

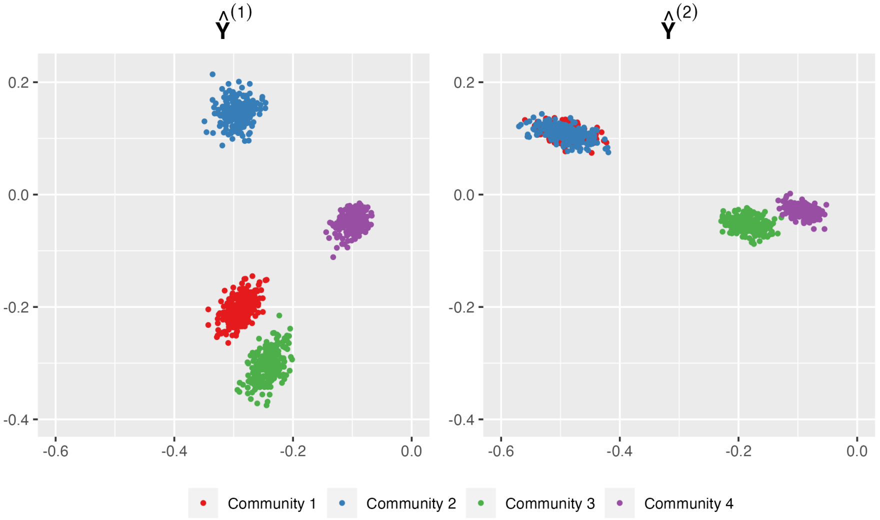
# Computing the embedding

- Separately compute $Cov(\mathbf{D}^{(1)})$ and $Cov(\mathbf{D}^{(2)})$.
- Let $\mathbf{A}^{(1)} = \text{Hollow}(Cov(\mathbf{D}^{(1)}))$ and $\mathbf{A}^{(2)} = \text{Hollow}(Cov(\mathbf{D}^{(2)}))$ be our adjacency matrices.
- Column concatenate: $\mathbf{A} = [\mathbf{A}^{(1)} | \mathbf{A}^{(2)}] \in \mathbb{R}^{n \times 2n}$.
- Compute truncated ($d$-dimensional) SVD: $\mathbf{A} \approx \mathbf{U_A \Sigma_A V_A^\top} = \hat{\mathbf{X}}\hat{\mathbf{Y}}^\top$.

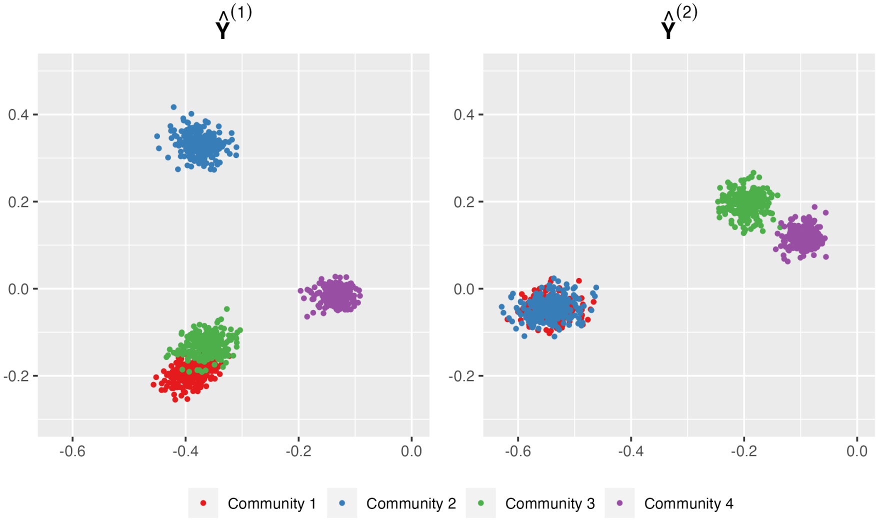
# Computing the embedding

- Separately compute $Cov(\mathbf{D}^{(1)})$ and $Cov(\mathbf{D}^{(2)})$.
- Let $\mathbf{A}^{(1)} = \text{Hollow}(Cov(\mathbf{D}^{(1)}))$ and $\mathbf{A}^{(2)} = \text{Hollow}(Cov(\mathbf{D}^{(2)}))$ be our adjacency matrices.
- Column concatenate: $\mathbf{A} = [\mathbf{A}^{(1)} | \mathbf{A}^{(2)}] \in \mathbb{R}^{n \times 2n}$.
- Compute truncated ($d$-dimensional) SVD: $\mathbf{A} \approx \mathbf{U}_{\mathbf{A}} \mathbf{\Sigma}_{\mathbf{A}} \mathbf{V}_{\mathbf{A}}^{\top} = \hat{\mathbf{X}} \hat{\mathbf{Y}}^{\top}$.
- Plot first 2 entries for each row of $\hat{\mathbf{Y}}^{(1)}$ and $\hat{\mathbf{Y}}^{(2)}$.

# Stability of UASE



# Comparison: Independent Spectral Embedding



# Conclusion

Under some assumptions, you can prove:

# Conclusion

Under some assumptions, you can prove:

- The cross sectional, and longitudinal stability of UASE.

# Conclusion

Under some assumptions, you can prove:

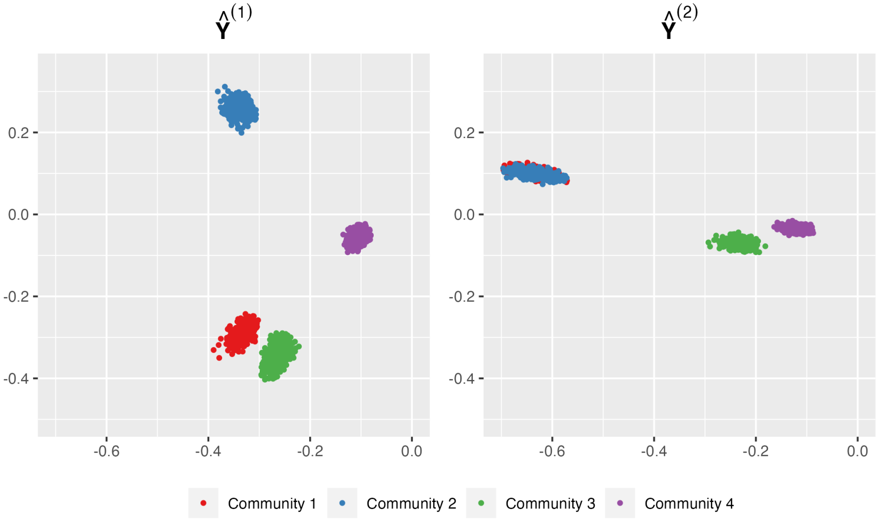
- The cross sectional, and longitudinal stability of UASE.
- That each embedding $\hat{\mathbf{Y}}^{(t)}$ converges to some underlying 'true' or 'noise-free' $\tilde{\mathbf{Y}}^{(t)}$ as the size of n increases, up to rotation.

# Conclusion

Under some assumptions, you can prove:

- The cross sectional, and longitudinal stability of UASE.
- That each embedding $\hat{\mathbf{Y}}^{(t)}$ converges to some underlying 'true' or 'noise-free' $\tilde{\mathbf{Y}}^{(t)}$ as the size of n increases, up to rotation.
- That the points  point clouds converge to a multivariate Gaussian distribution.

# Conclusion



$\hat{Y}^{(1)}$
$\hat{Y}^{(2)}$
0.2
0.0
-0.2
-0.4
-0.6
-0.4
-0.2
0.0
Community 1
Community 2
Community 3
Community 4