Lecture 8: Nonlinear dimensionality reduction: Diffusion Maps; Graph Partitioning

Foundations of Data Science: Algorithms and Mathematical Foundations

Mihai Cucuringu
mihai.cucuringu@stats.ox.ac.uk

CDT in Mathematics of Random System
University of Oxford

22 September, 2023

Diffusion maps

- introduced in S. Lafon's Ph.D. Thesis in 2004 as a nonlinear dimensionality reduction tool
- connected data analysis and clustering techniques based on eigenvectors of similarity matrices with the geometric structure of non-linear manifolds
- diffusion maps have gained a lot of popularity over the years years
- often called *Laplacian eigenmaps*, these manifold learning techniques
  - identify significant variables that live in a lower dimensional space
  - while preserving the local proximity between data points

Diffusion maps

- consider a set of N points $V = \{x_1, x_2, \ldots, x_N\}$ in an D-dimensional space $\mathbb{R}^D$
- each point (typically) characterizes an image (or an audio stream, text string, etc.)
- if two images x_i and x_j are similar, then $||x_i - x_j||$ is small
- a popular measure of similarity between points in $\mathbb{R}^D$ is defined using the Gaussian kernel

$$w_{ij} = e^{-||x_i - x_j||^2/\epsilon}$$

  so that the closer x_i is from x_j, the larger w_{ij}
- the matrix $W = (w_{ij})_{1 \leq i,j \leq N}$ is symmetric and has positive coefficients
- to normalize W, we define the diagonal matrix D, with $D_{ii} = \sum_{j=1}^{N} w_{ij}$ and define L by

$$L = D^{-1} W,$$

  such that every row of L sums to 1.

Diffusion maps

- define the symmetric matrix $S = D^{-1/2} W D^{-1/2}$
- note S is similar to L, since one can write

$$S = D^{1/2} D^{-1} \; W D^{-1/2} = D^{1/2} L D^{-1/2} \tag{1}$$

- as a symmetric matrix, S has an orthogonal basis of eigenvectors $v_0, v_1, \ldots, v_{N-1}$, and N real eig-values $1 = \lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{N-1}$
- If we eigen-decompose S as

$$S = V \Lambda V^T$$

with

$$V V^T = V^T V = I$$

$$\Lambda = Diag(\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$$

then L becomes

$$L = \Psi \Lambda \Phi^T \tag{2}$$

where $\Psi = D^{-1/2} V$ and $\Phi = D^{1/2} V$, since following (1)

$$L = D^{-1/2} S D^{1/2} = D^{-1/2} V \; \Lambda \; V^T D^{1/2} = \Psi \Lambda \Phi^T \tag{3}$$

Diffusion maps

- $L = D^{-1}W$ is a row-stochastic matrix, $\lambda_0 = 1$ and $\psi_0 = (1, 1, \ldots, 1)^T$, and we disregard this trivial pair

- interpret L as a random walk matrix on a weighted graph $G = (V, E, W)$, where
  - the set of nodes consists of the points $\{x_1, \ldots, x_N\}$,
  - and there is an edge between nodes i and j if and ony if $w_{ij} > 0$

- recall that $L_{ij} = w_{ij}/\deg(i)$
- since $L_{ij} \geq 0$ and $L\mathbf{1} = \mathbf{1}$, then L is a transition probability matrix

- L_{ij} denotes the transition probability from point x_i to x_j in one step time $\Delta t = \epsilon$

$$Pr\{x(t+\epsilon) = x_j | x(t) = x_i\} = L_{ij}.$$

The choice of ϵ

$$w_{ij} = e^{-||x_i - x_j||^2/\epsilon} \tag{4}$$

The parameter ϵ has a two-fold interpretation:

- ϵ is the squared radius of the neighborhood used to infer local geometric and density information
  - w_{ij} is $O(1)$ when x_i and x_j are in a ball of radius $\sqrt{\epsilon}$
  - but w_{ij} is exponentially small for points that are more than $\sqrt{\epsilon}$ apart
- ϵ represents the discrete time step at which the random walk jumps from one point to another

• Lafon chose ϵ to be the order of the average smallest non-zero value of $||x_i - x_j||^2$

$$\epsilon = \frac{1}{k}\sum_{i=1}^{k} \min_{j:x_j \neq x_i} ||x_i - x_j||^2 \tag{5}$$

• Singer/Coifman et al

- (a) if ϵ is relatively too large compared to $||x_i - x_j||^2$, the entries W will be very close to one
- (b) if ϵ is relatively too small, it leads to almost zero entries for W

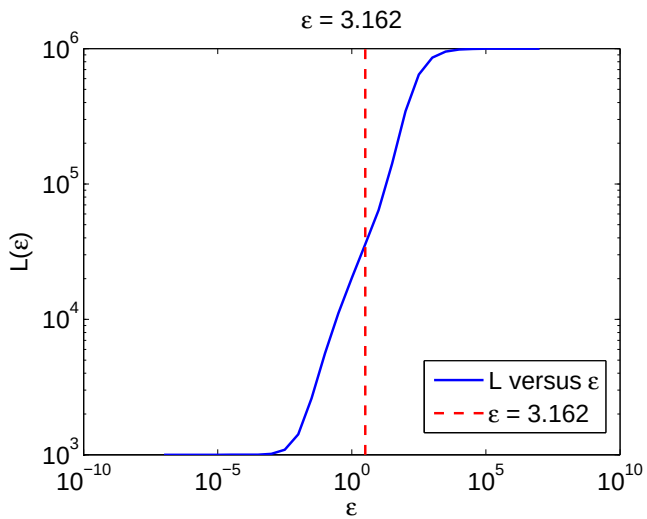
The choice of ϵ (cont)

Neither case too interesting...

- (a): most of the diffusion has taken place
- (b): no diffusion takes places

Pick a value of ϵ that straddles the two boundaries

1. Construct an ϵ-dependent weight matrix $W(\epsilon)$, for several ϵ values
2. Compute
$$T(\epsilon) = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}(\epsilon) \tag{6}$$
3. Plot $T(\epsilon)$ using a logarithmic plot. This plot will have two asymptotic regimes when $\epsilon \to 0$ and $\epsilon \to \infty$
4. Choose ϵ where the logarithmic plot of $T(\epsilon)$ appears linear.



Source: Bubacarr Bah, *Diffusion Maps: Analysis and Applications*

Diffusion maps

- Interpreting the eigenvectors as functions over our data set, the *diffusion map* maps points from the original space to the eigenvectors of L, $\mathcal{L}' : V \mapsto \mathbb{R}^{D-1}$, is defined as

$$\mathcal{L}'_t(x_i) = (\lambda_2^t \psi_2(j), \lambda_3^t \psi_3(j), \ldots, \lambda_D^t \psi_D(j)) \tag{7}$$

  - where we left out the first trivial eigenvalue $\lambda_1 = 0$ & eigenvector $\psi_1 = \mathbf{1}$, as it does not help differentiate points on the graph.
  - still a map to $D-1$ dimensions; but note now that each coordinate has a factor of λ_k^t which (for moderate values of t) will be rather small whenever λ_k is small.

- The *truncated diffusion* map (truncated to k dimensions), which maps $\mathcal{L} : V \mapsto \mathbb{R}^{D-1}$, is given by

$$\mathcal{L}_t(x_j) = (\lambda_2^t \psi_2(j), \lambda_3^t \psi_3(j), \ldots, \lambda_{k+1}^t \psi_{k+1}(j)) \tag{8}$$

- using the left and right eigenvectors of L

$$L = \sum_{r=0}^{N-1} \lambda_r \phi_r \psi_r \quad (9) \qquad L_{ij} = \sum_{r=0}^{N-1} \lambda_r \phi_r(i) \psi_r(j) \quad (10)$$

- note that

$$L^t = \sum_{r=0}^{N-1} \lambda_r^t \phi_r \psi_r \quad (11) \qquad L_{ij}^t = \sum_{r=0}^{N-1} \lambda_r^t \phi_r(i) \psi_r(j) \quad (12)$$

- in matrix form: $L^t = \Phi \Lambda^t \Psi$
- the probability distribution of a random walk landing at location x_j after exactly t steps, starting at x_i

$$L_{ij}^t = Pr\{x(t) = x_j | x(0) = x_i\} \quad (13)$$

- given the random walk interpretation, *quantify the similarity between two points* according to the *evolution of their probability distributions*; (Weighted ℓ_2 distance btw the probability clouds)

$$D_t^2(i,j) = \sum_{k=1}^{N} (L_{ik}^t - L_{jk}^t)^2 \frac{1}{d_k}, \quad (14)$$

where the weight $\frac{1}{d_k}$ takes into account the empirical local density of the points by giving larger weight to vertices of lower degree

- $D_t(i,j)$ is the *diffusion distance* at time t.

Diffusion Maps

- a matter of choice to tune the parameter t corresponding to the number of time steps of the random walk (used $t = 1$)
- using different values of t corresponds to rescaling the axis
- the Euclidean distance between two points in the diffusion map space introduced in (8) is given by

$$||\mathcal{L}(x_i) - \mathcal{L}(x_j)||^2 = \sum_{r=1}^{N-1} \left(\lambda_r^t \psi_r(i) - \lambda_r^t \psi_r(j)\right)^2 \tag{15}$$

$$= \sum_{r=1}^{N-1} \lambda_r^{2t} \left(\psi_r(i) - \psi_r(j)\right)^2 \tag{16}$$

- Nadler et al. (2005) have shown that the expression (16) equals the diffusion distance $D_t^2(i,j)$ in (14), when $k = N - 1$ (when using all $N - 1$ (nontrivial) eigenvectors)
- for ease of visualization, use the top $k = 2$ eigenvectors for the projections

Diffusion distance

Weighted ℓ_2 distance between the probability clouds; for the weights we choose $1/d_k$, that is, inversely proportional to the vertex degrees.

$$\left\| L^t_{i,.} - L^t_{j,.} \right\|^2_{\ell^2(\mathbb{R}^N, 1/d)} = \sum_{k=1}^{N} \left(L^t_{ik} - L^t_{jk} \right)^2 \frac{1}{d_k}$$

$$= \sum_{k=1}^{N} \left[\sum_{l=1}^{N} \lambda_l^t \phi_l(i) \psi_l(k) - \lambda_l^t \phi_l(j) \psi_l(k) \right]^2 \frac{1}{d_k}$$

$$= \sum_{k=1}^{N} \sum_{l,r=1}^{N} \lambda_l^t \lambda_r^t \left(\phi_l(i) - \phi_l(j) \right) \left(\phi_r(i) - \phi_r(j) \right) \frac{\psi_l(k) \psi_r(k)}{d_k}$$

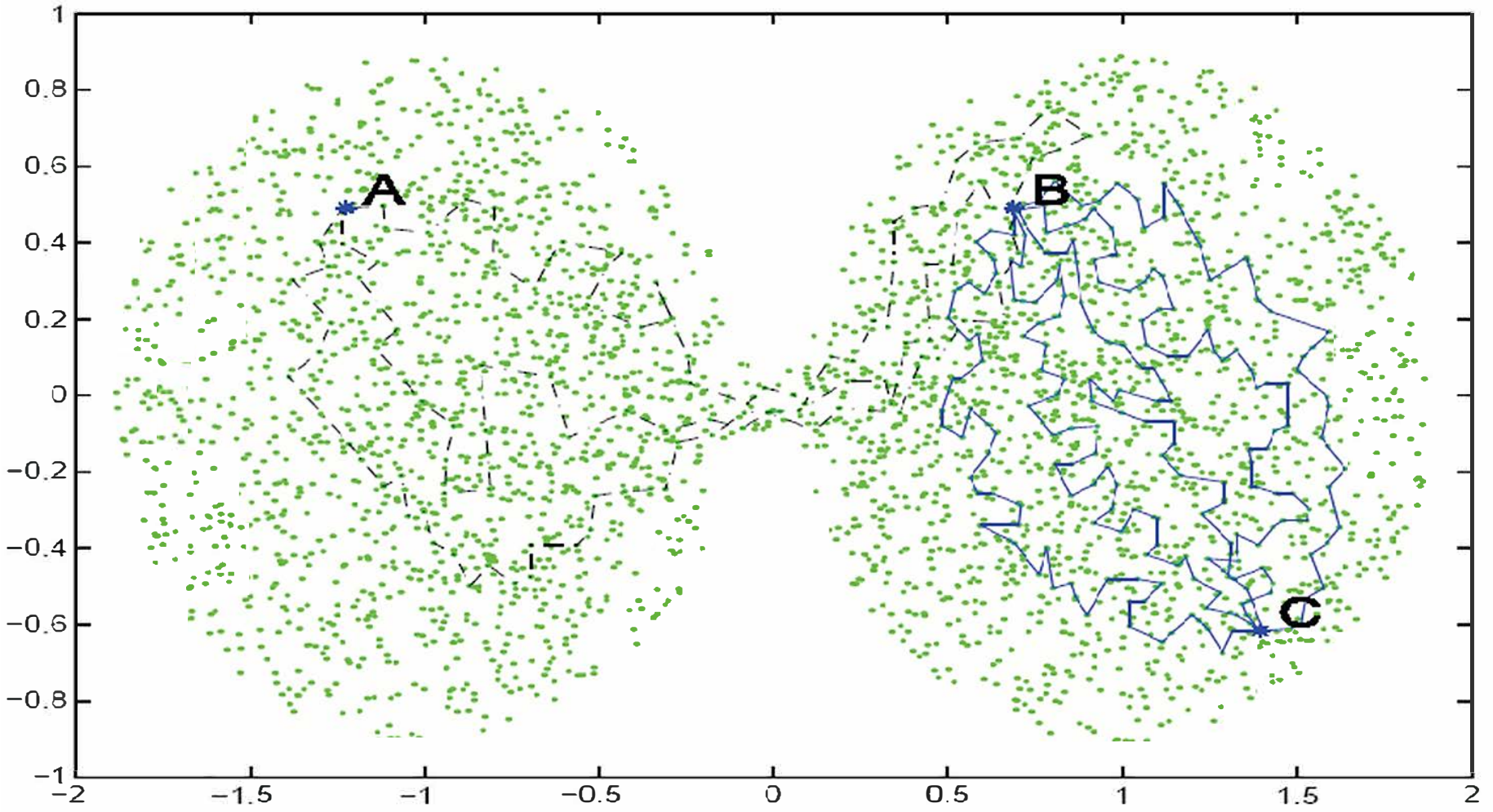
$$= \sum_{l,r=1}^{N} \lambda_l^t \lambda_r^t \left(\phi_l(i) - \phi_l(j) \right) \left(\phi_r(i) - \phi_r(j) \right) \sum_{k=1}^{N} \frac{\psi_l(k) \psi_r(k)}{d_k}$$

$$= \sum_{l,r=1}^{N} \lambda_l^t \lambda_r^t \left(\phi_l(i) - \phi_l(j) \right) \left(\phi_r(i) - \phi_r(j) \right) \delta_{lr}$$

$$= \sum_{l=1}^{N} \lambda_l^{2t} \left(\phi_l(i) - \phi_l(j) \right)^2 = D_t^2 \left(x_i, x_j \right).$$

Diffusion distance vs Euclidean distance

Why bother?



Limitations of the Euclidean distance

Figure: Euclidean distance may not be relevant to properly understand the distance (or similarity) between two points.

Is C is more similar to point B or to point A?

- (left) the natural answer is: C is more similar to B.
- (right) less obvious given the other observed data points... C should be more similar to A.
  - need a new metric for which C and A are closer than C and B given the geometry of the observed data.

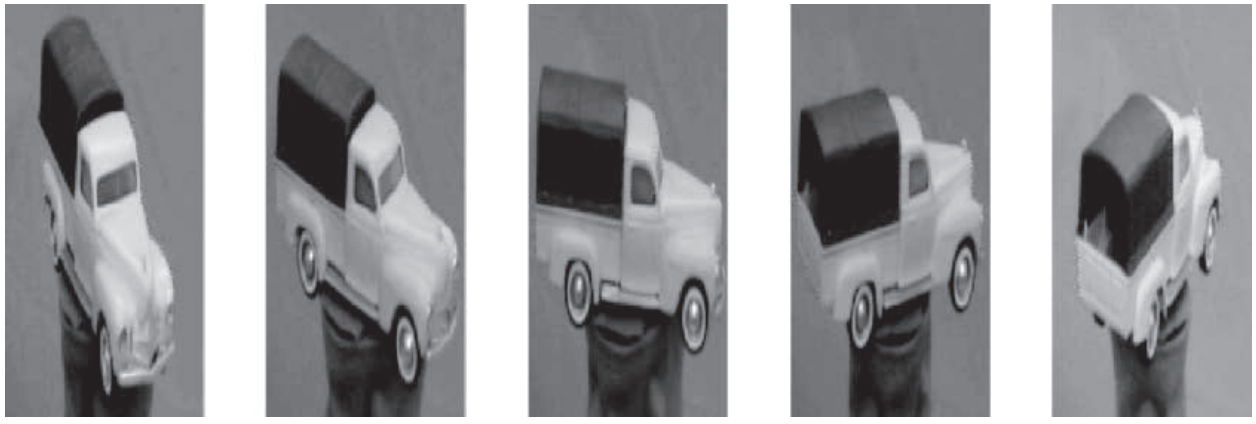
Eigenvector colourings

- denote by $\mathcal{C}_k$ the colouring of the N data points given by the eigenvector ψ_k
- colour of point $x_i \in V$ is given by the i-th entry in ψ_k, i.e.

$$\mathcal{C}_k(x_i) = \psi_k(i), \text{ for all } k = 0, \ldots, N-1 \text{ and } i = 1, \ldots, N.$$

- $\mathcal{C}_k$: *eigenvector colouring* of order k
- do not confuse with the "graph colouring" terminology
- colorbar: red denotes high values and blue denotes low values, in the eigenvector entries
- in practice, only the first k eigenvectors are used in the diffusion map introduced in (8), with $k << N-1$ chosen such that $\lambda_1^t \geq \lambda_2^t \ldots \geq \lambda_k^t > \delta$, but $\lambda_{k+1}^t < \delta$, where δ is a chosen tolerance
- show how one can extract relevant information from eigenvectors of much lower order.

Example



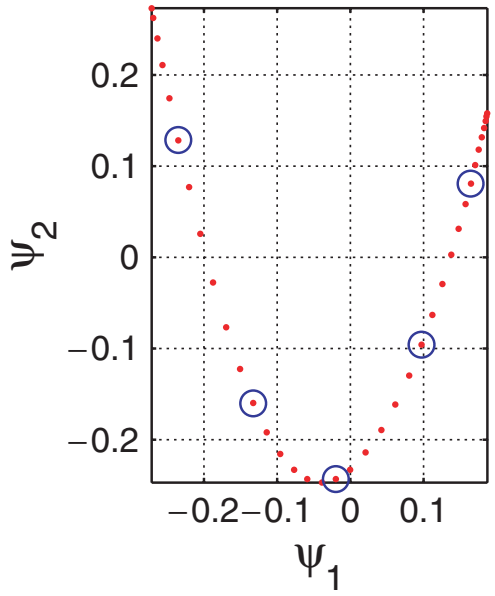


Figure: Figures of a truck taken at five different horizontal angles (top). The mapping of the 37 images into the first two eigenvectors, based on a Gaussian kernel with standard Euclidean distance between the images as the underlying metric (bottom). The blue circles correspond to the five specific images shown above.

Source: Nadler, Boaz, et al. "Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms."

Swiss roll

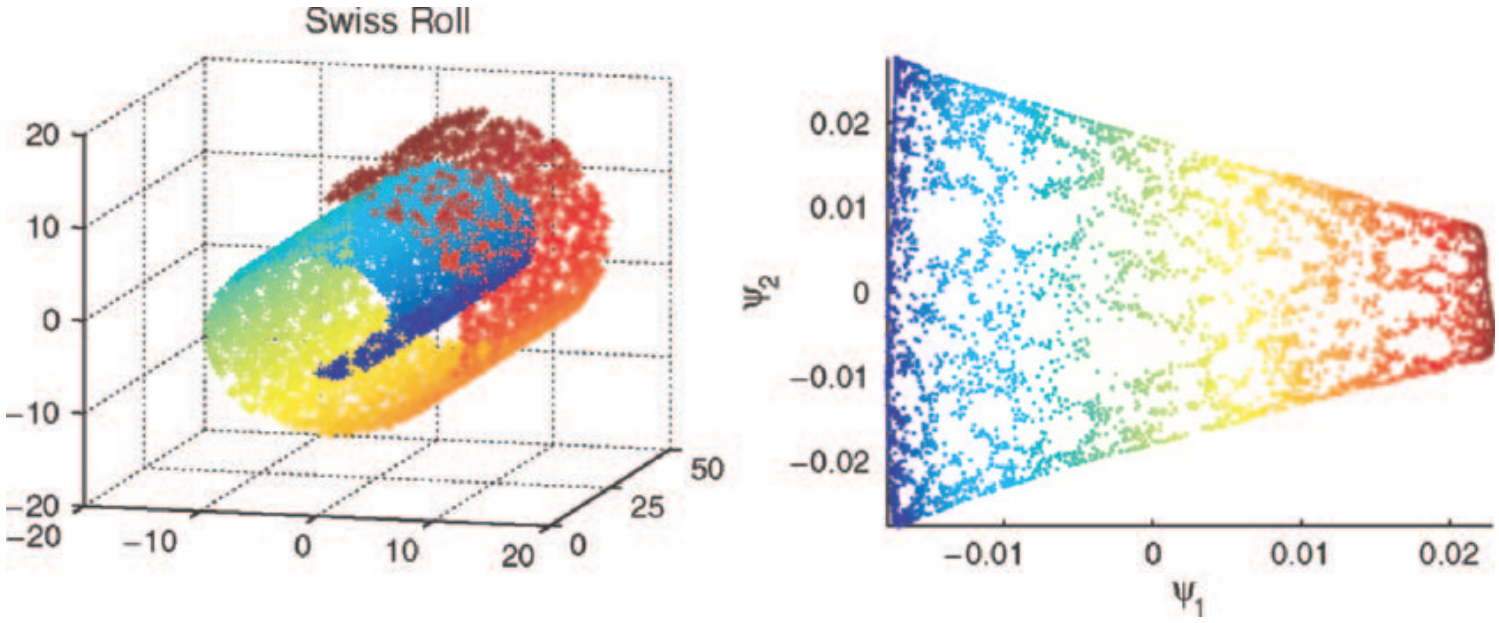


Figure: 5000 points sampled from a wide Swiss roll and embedding into various diffusion map coordinates. The length and width of the roll are similar. The spectral embedding via the first two diffusion map coordinates gives a reasonably nice parametrization of the manifold, uncovering its 2-d nature.

Source: Nadler, Boaz, et al. ”Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms.”

Swiss roll

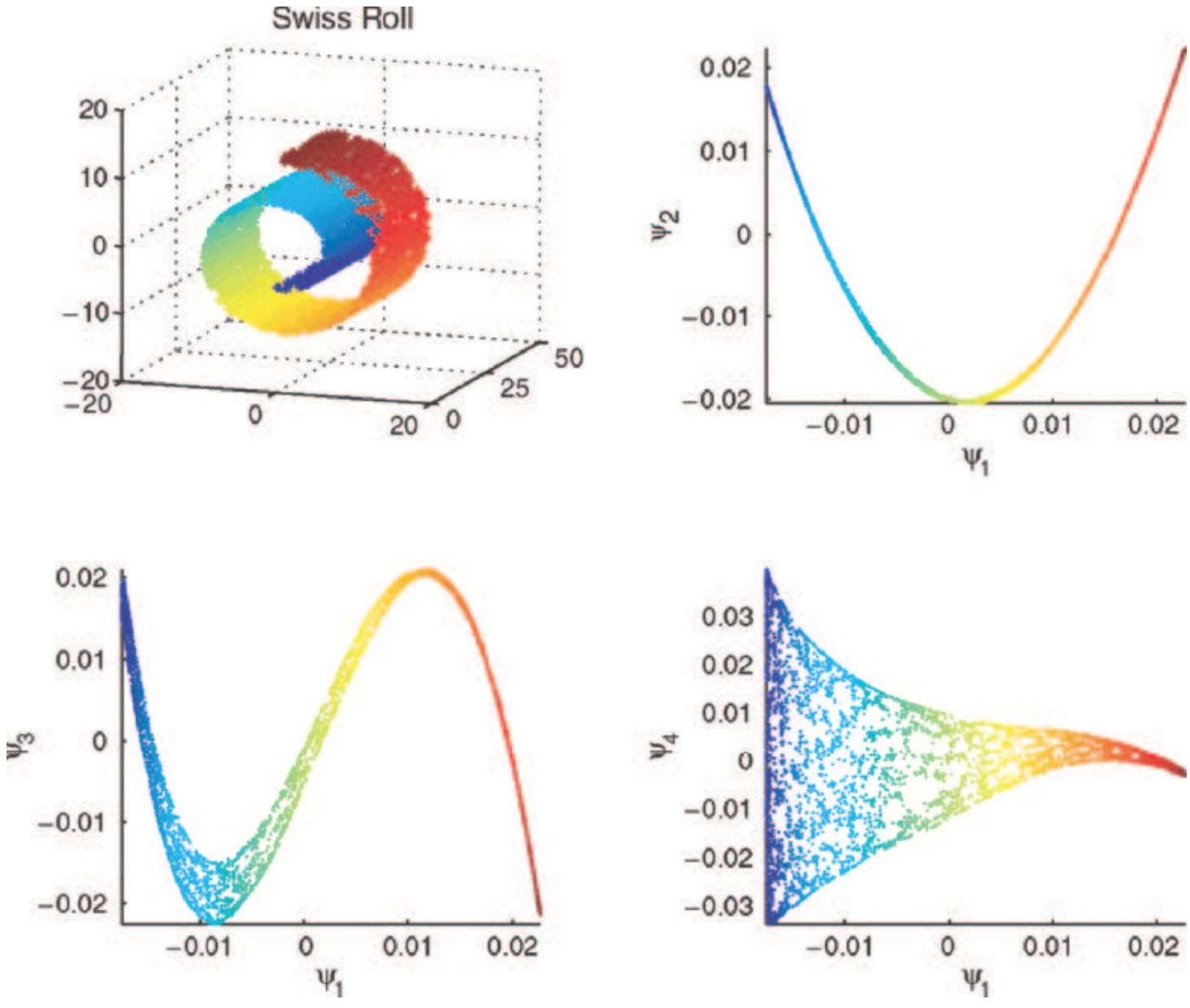


Figure: 5000 points sampled from a narrow Swiss roll and embedding into the first two diffusion map coordinates.

Data set of faces

Figure: **Left**: set of images randomly permuted. This is the input of the algorithm. **Right**: output of the algorithm, the sequence is recorded with respect to the angle of rotation of the head (the sequence is to be read from left to right, and top down).

Every image is 112×92 pixels, and can be viewed as a point in $\mathbb{R}^{112\times 92}$ However, face images are far from being randomly distributed in that high dimensional Euclidean space. There is only one physically meaningful parameter describing the images. We will say that the intrinsic dimension of the data set = 1.

Eigenvector localization

- The phenomenon of *eigenvector localization* occurs when most of the components of an eigenvector are zero or close to zero, and almost all the mass is localized on a relatively small subset of nodes.

- On the contrary, *delocalized eigenvectors* have most of their components small and of roughly the same magnitude.

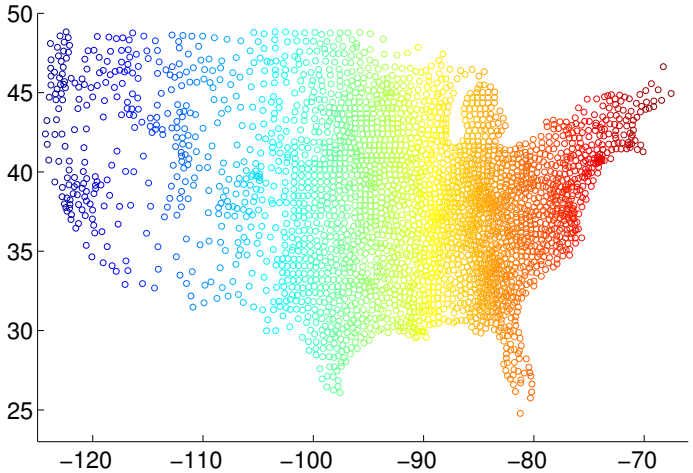
2000 US Census data set

- reports the number of people that migrated from every county to every other county within US during 1995-2000
- $M = (M_{ij})_{1 \le i,j \le N}$ the **total** number of people that migrated between county i and county j (so undirected graph $M_{ij} = M_{ji}$) (cannot handle directed graphs in the current framework)
- $N = 3107$ denotes the number of counties in mainland US
- let P_i denote the population of county i
- different similarity measures

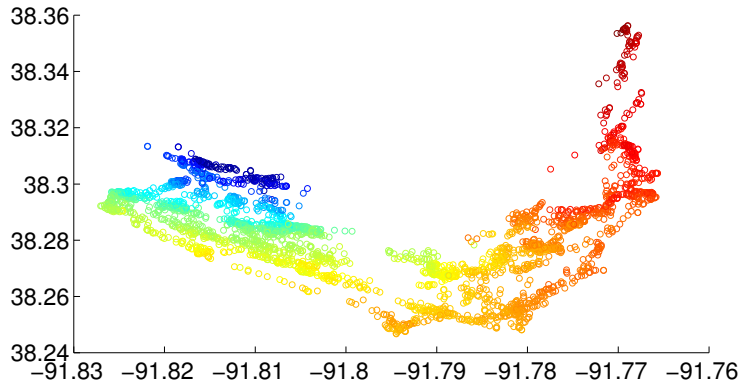
$$W_{ij}^{(1)} = \frac{M_{ij}^2}{P_i P_j}; \qquad W_{ij}^{(2)} = \frac{M_{ij}}{P_i + P_j}; \qquad W_{ij}^{(3)} = 5500 \frac{M_{ij}}{P_i P_j}$$

- colourings based on latitude reveal the north-south separation
- $W^{(1)}$ does a better job at separating the east and west coasts
- $W^{(2)}$ highlights best the separation between north and south

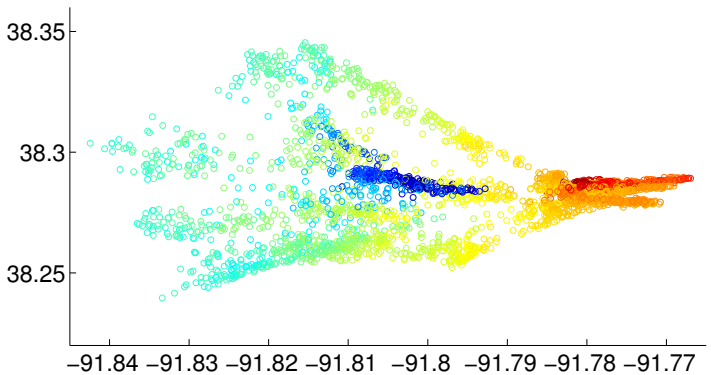
Colored by longitude



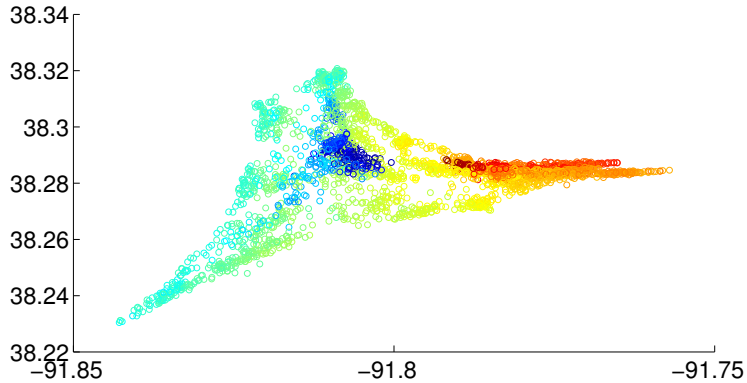
(a) USA map coloured by longitude



(b) $W_{ij}^{(1)} = \frac{M_{ij}^2}{P_i P_j}$



(c) $W_{ij}^{(2)} = \frac{M_{ij}}{P_i + P_j}$

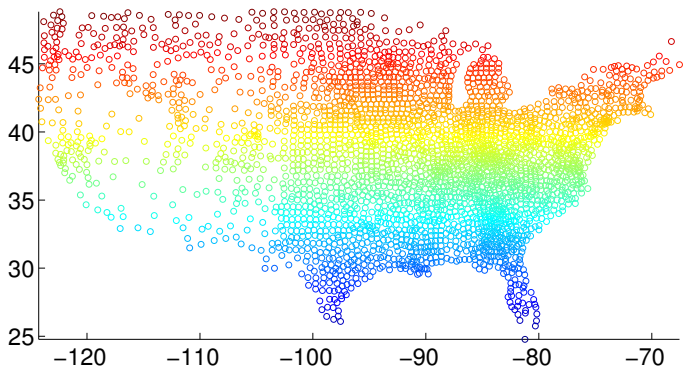


(d) $W_{ij}^{(3)} = \frac{M_{ij}}{P_i P_j}$

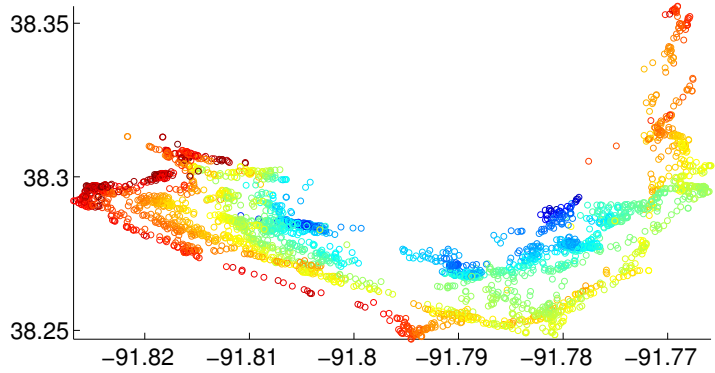
Figure: Diffusion map reconstructions from the top two eigenvectors, for various similarities, with nodes colored by longitude

$W^{(1)}$ does a better job at separating the east and west coasts.

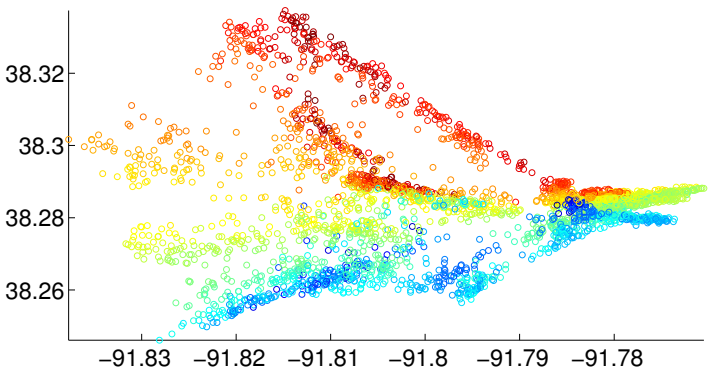
Colored by latitude



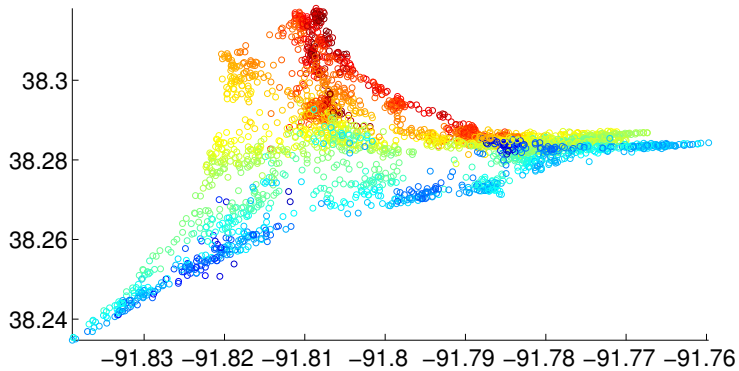
(a) USA map coloured by latitude



(b) $W_{ij}^{(1)} = \frac{M_{ij}^2}{P_i P_j}$



(c) $W_{ij}^{(2)} = \frac{M_{ij}}{P_i + P_j}$

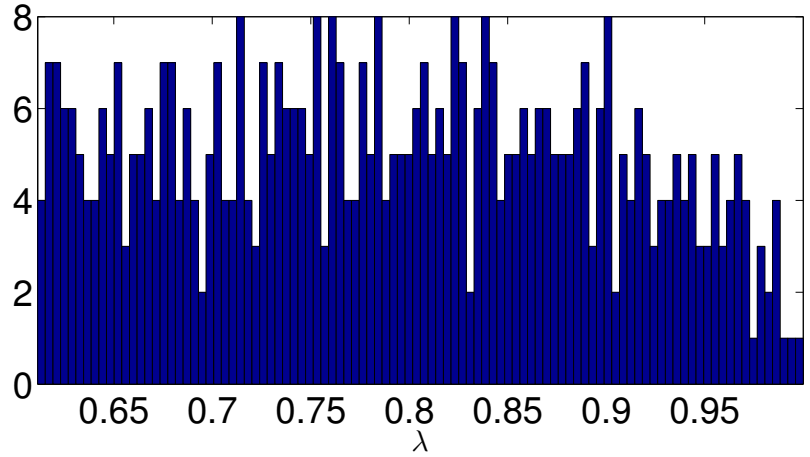


(d) $W_{ij}^{(3)} = 5500 \frac{M_{ij}}{P_i P_j}$

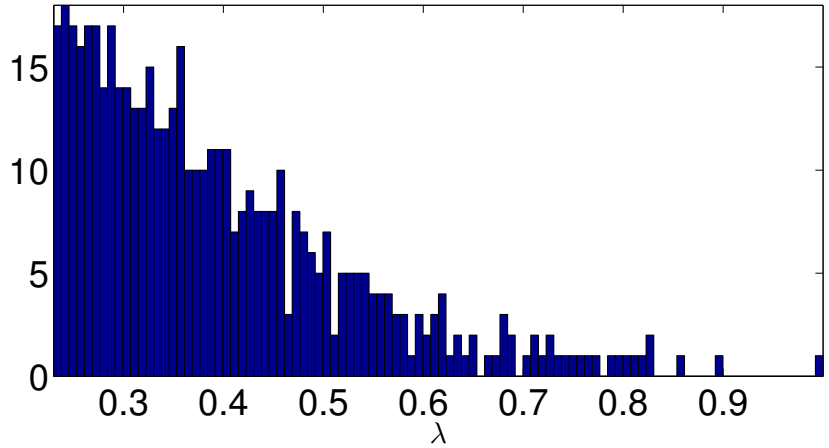
Figure: Diffusion map reconstructions from the top two eigenvectors, for various similarities, with nodes colored by longitude

$W^{(2)}$ highlights best the separation between north and south.

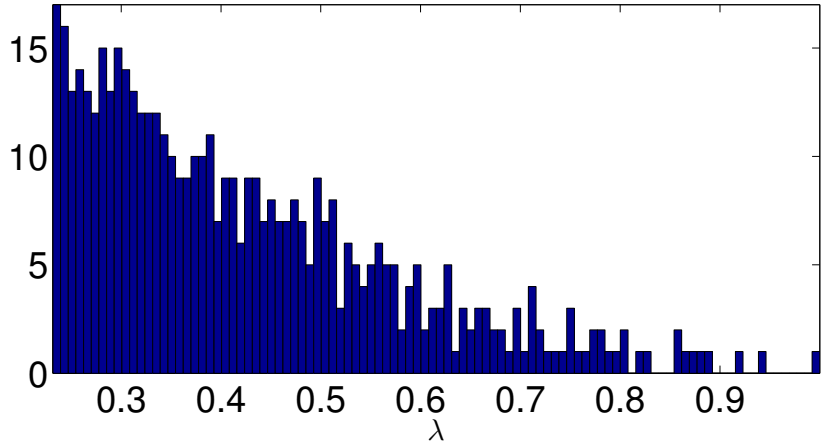
Spectrum of the graph Laplacian matrix



(a) $L = D^{-1}W^{(1)}$



(b) $L = D^{-1}W^{(2)}$



(c) $L = D^{-1}W^{(3)}$

Figure: Histogram of the top 500 eigenvalues of matrix L for different similarity matrices $W^{(i)}$.

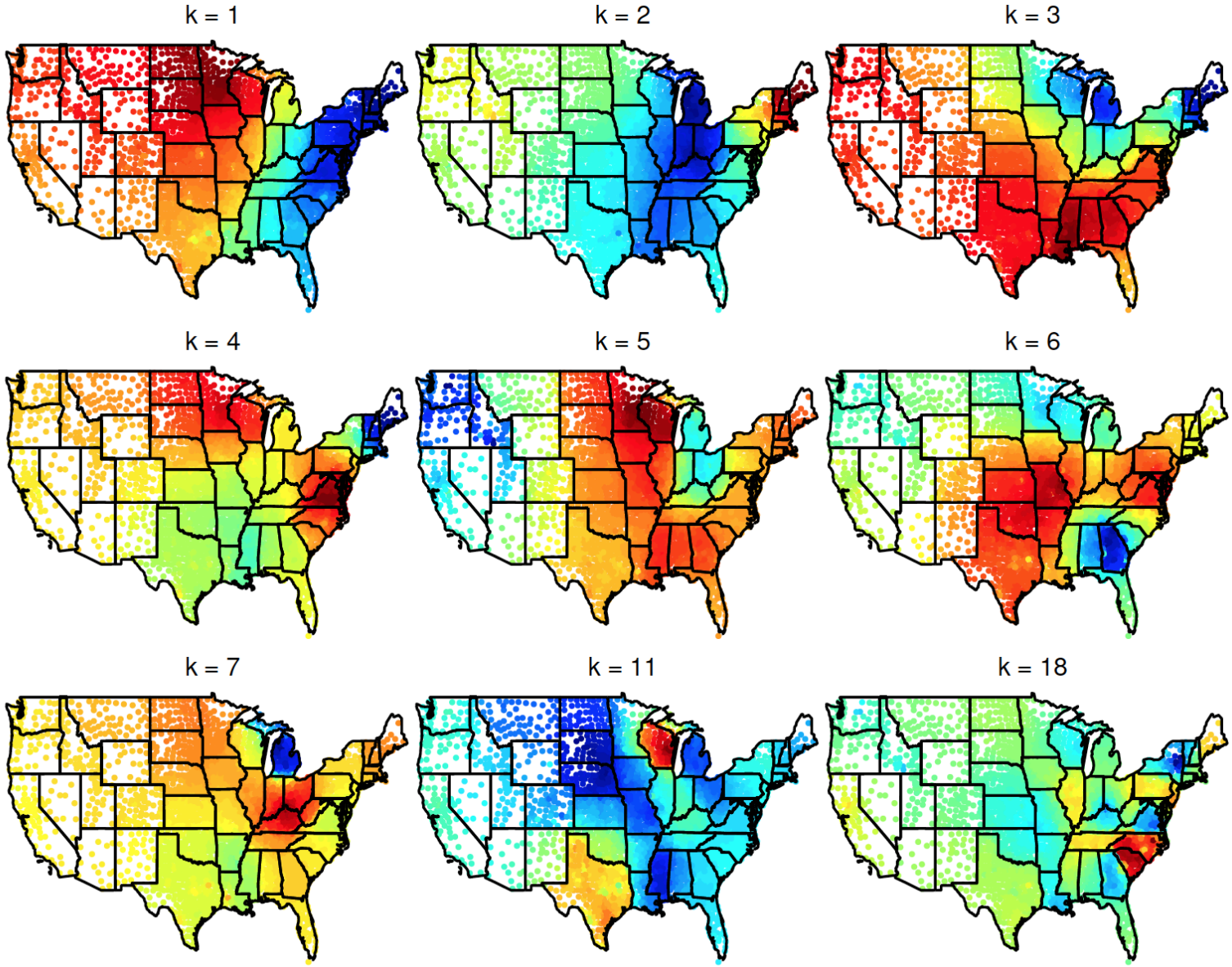


Figure: Eigenvector colourings for the similarity matrix $W_{ij} = \frac{M_{ij}^2}{P_i P_j}$.

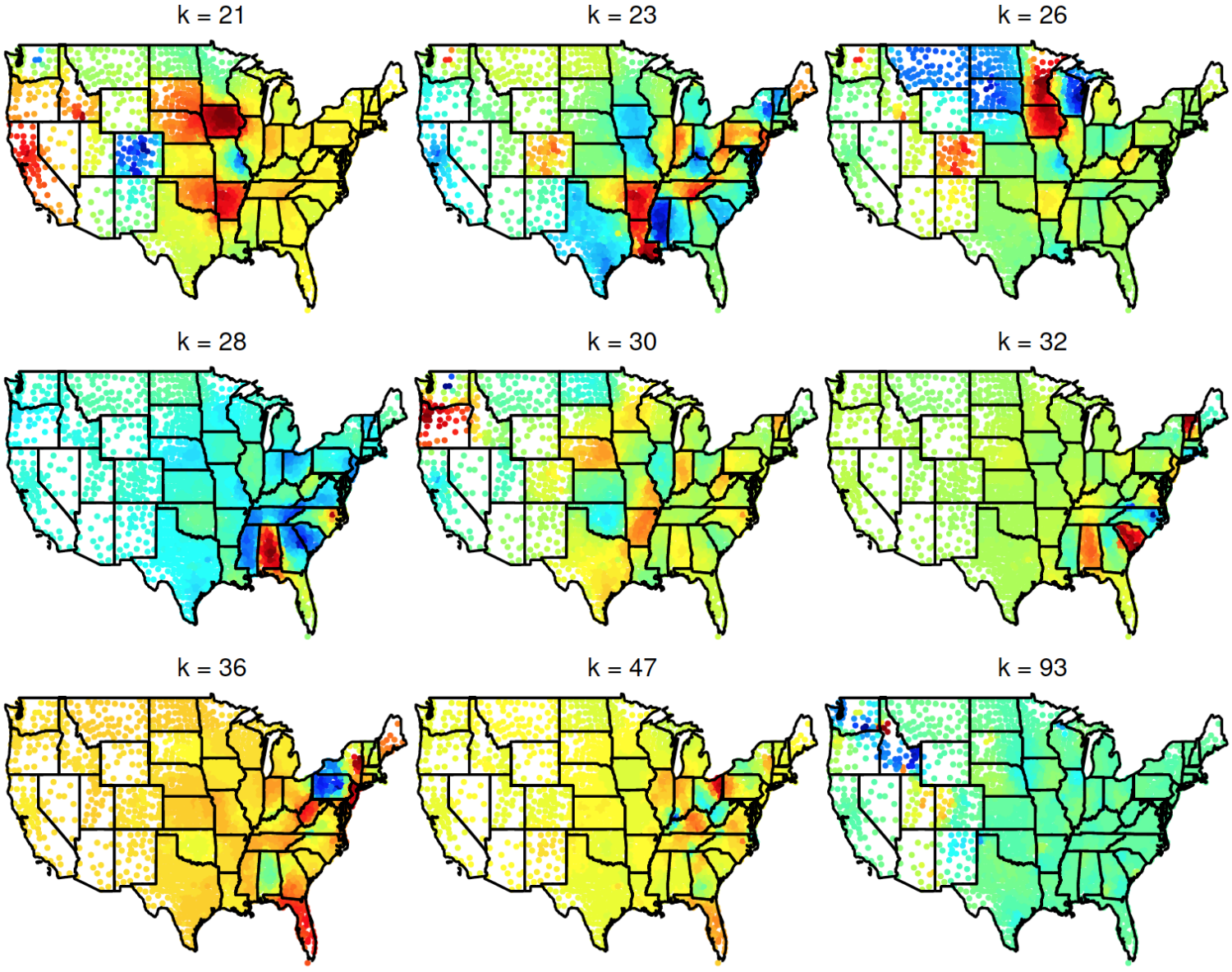
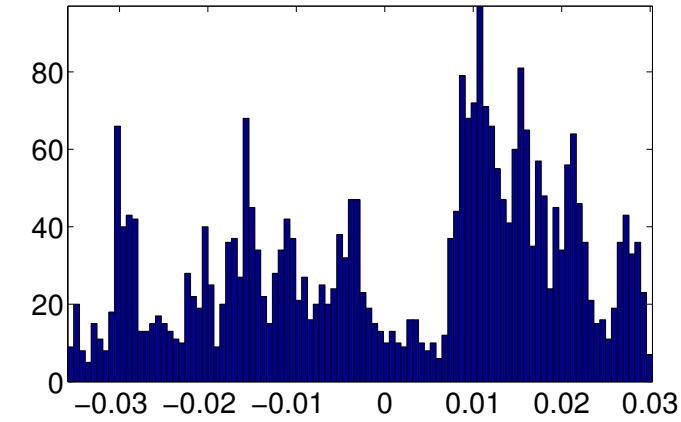
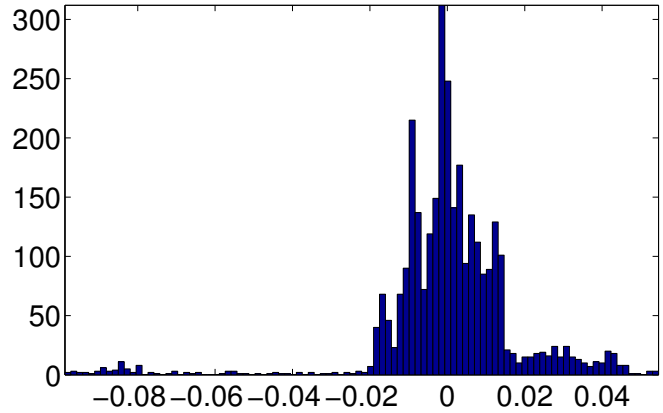


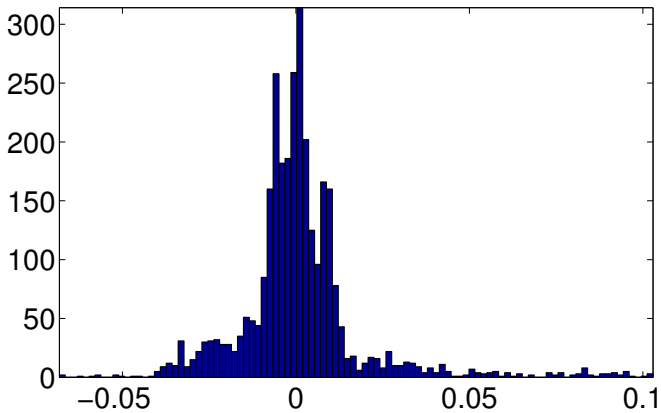
Figure: Further eigenvector colourings for the similarity matrix $W_{ij} = \frac{M_{ij}^2}{P_i P_j}$.



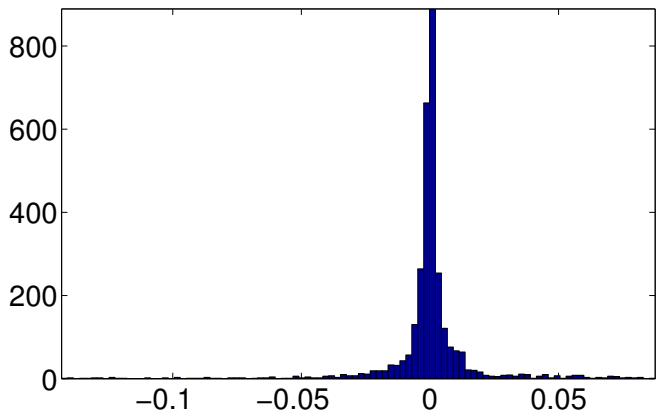
(a) ϕ_1



(b) ϕ_7



(c) ϕ_{28}



(d) ϕ_{83}

Figure: Histogram of eigenvectors $\phi_1, \phi_7, \phi_{28}, \phi_{83}$ of $L = D^{-1}W^{(1)}$

- ϕ_1 provides a meaningful partitioning that separates the East from the Midwest; entries in $[-0.03, 0.03]$ with few entries of zero magnitude.
- however, eigenvectors ϕ_7, ϕ_{28} and ϕ_{83} are *localized*: they have their larger entries localized on a specific subregion of the map (highlighted in blue or red in the colorings), while taking small values in magnitude on the rest of the domain.

The graph partitioning problem (GPP)

- Investigate the connection of such geographically cohesive coloured subgraphs with the (GPP)
- In general, the GPP seeks to decompose a graph into K disjoint subgraphs (clusters), while minimizing the sum of the weights of the "cut" edges, i.e., edges with endpoints in different clusters
- Given the number of clusters K, the Weighted-Min-Cut problem is an optimization problem that computes a partition $\mathcal{P}_1, \ldots, \mathcal{P}_K$ of the vertex set, by minimizing the weights of the cut edges

$$\text{Weighted Cut}(\mathcal{P}_1, \ldots, \mathcal{P}_k) = \sum_{i=1}^{k} E_w(\mathcal{P}_i, \overline{\mathcal{P}_i}), \tag{17}$$

where $E_w(X, Y) = \sum_{i \in X, j \in Y} W_{ij}$, and $\overline{X}$ denotes the complement of X.

Spectral clustering

- extensive literature survey on spectral clustering algorithms: *Von Luxburg, Ulrike. "A tutorial on spectral clustering." Statistics and computing 17.4 (2007): 395-416*

  `https://arxiv.org/abs/0711.0189`

- & the popular spectral relaxation introduced by Shi and Malik (early 2000s)
- When dividing a graph into two smaller subgraphs, one wishes to minimize the sum of the weights on the edges across two different subgraphs, and simultaneously, maximize the sum of the weights on the edges within the subgraphs.
- Alternatively, one tries to maximize the ratio between the latter quantity and the former, i.e., between the weights of the inside edges and the weights of the outside edges.
- We regard the US states as the clusters, and investigate the possibility that the isolated coloured regions that emerge correspond to local cuts in the weighted graph

Clustering

- denote by S the matrix of size $N \times N$ ($N = 49$ the number of mainland US states) that aggregates the similarities between counties at the level of states

- if state i has k counties with indices $x_1, \ldots, x_k$, and state j has l counties with indices $y_1, \ldots, y_l$, then we consider the $k \times l$ submatrix

$$\tilde{W}_{i,j} = W_{\{x_1,\ldots,x_k\},\{y_1,\ldots,y_l\}} \tag{18}$$

  and denote by S_{ij} the sum of the kl entries in $\tilde{W}_{i,j}$

- heatmap shows the components of the matrix S on a logarithmic scale, where the intensity of entry (i, j) denotes the aggregated similarity between states i and j

Cluster-Cluster Meta Adjacency Matrix

$S :=$

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
Cluster 2	.	.	.	.	.
Cluster 3	.				
Cluster 4	.				
Cluster 5	S_{51}	S_{52}	S_{53}	S_{54}	S_{55}

- S_{ii} is "inside degree" of state i, $d_i^{in} = S_{ii}$, which measures the internal similarity between the counties of state i
- denote by $d_i^{out} = \sum_{u=1, u \neq i}^{N} S_{i,u}$ (i.e., the sum of the non-diagonal elements in row i) the "outside degree" of node i, which measures the similarity/migration between the counties of state i and all other counties outside of state i
- denote by $d_i^{ratio} = \frac{d_i^{in}}{d_i^{out}}$, the "ratio degree" of node i which straddles the boundary between intra-state and inter-state migration
- a large ratio degree is a good indicative that a state is very well connected internally, and has little connectivity with the outside world, and thus is a good candidate for a cluster.
- the Table ranks the top 15 states within the US in terms of their ratio degree.

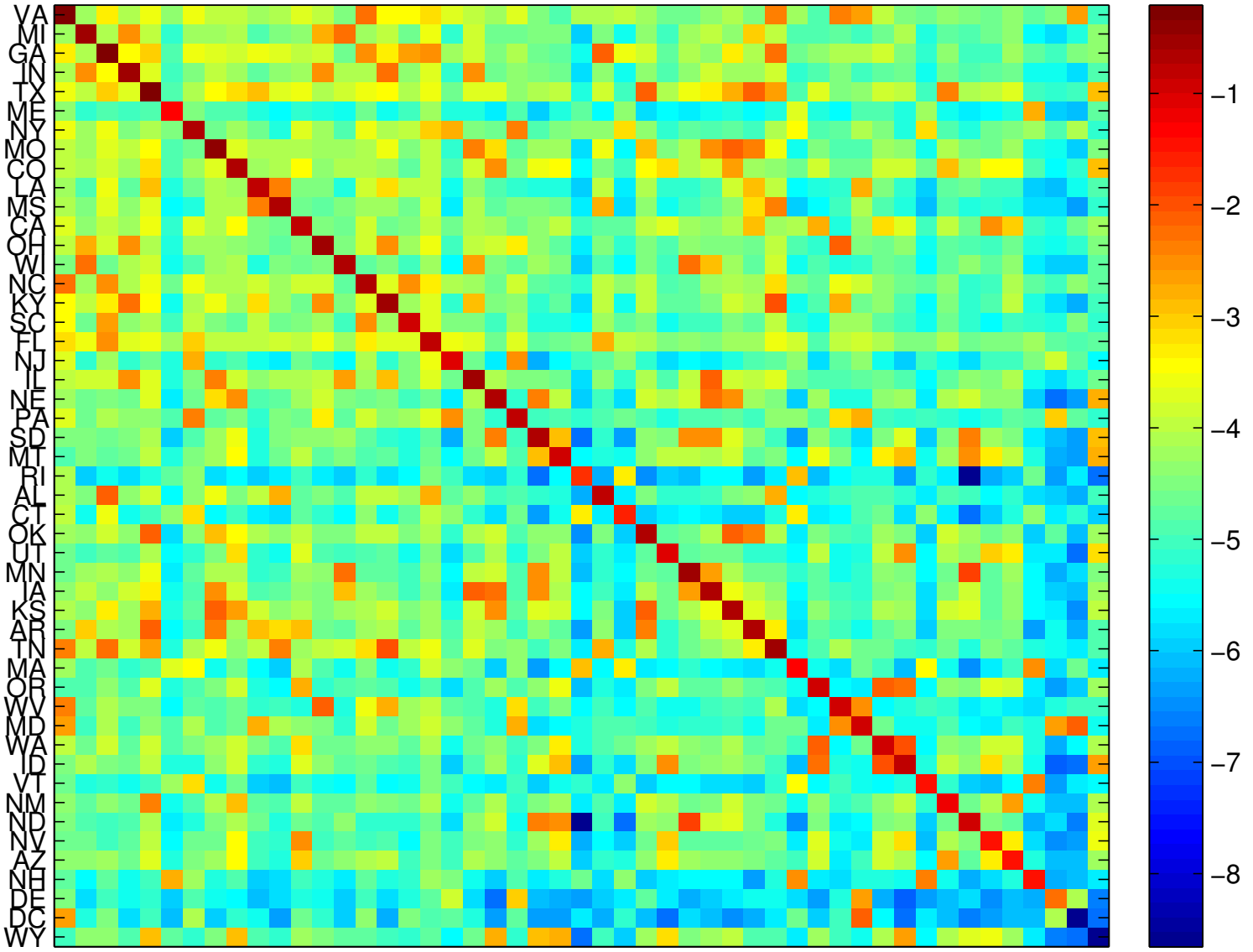


Figure: Heatmap of the inter-state migration flows. Rows (and columns) are sorted by the ratio degrees of the states. The intensity of entry (i, j) denotes, on a logarithmic scale, the similarity between states i and j, i.e., the sum of all entries in the submatrix $\tilde{W}_{i,j}$

rank	state	ratio degree
1.	VA	26.7
2.	MI	20.4
3.	GA	19.9
4.	IN	19.7
5.	TX	19.0
6.	ME	18.9
7.	NY	18.7
8.	MO	18.5
9.	CO	17.1
10.	LA	16.6
11.	MS	16.1
12.	CA	15.7
13.	OH	15.6
14.	WI	14.5

Table: Top 15 states within the US, ordered by ratio degree.

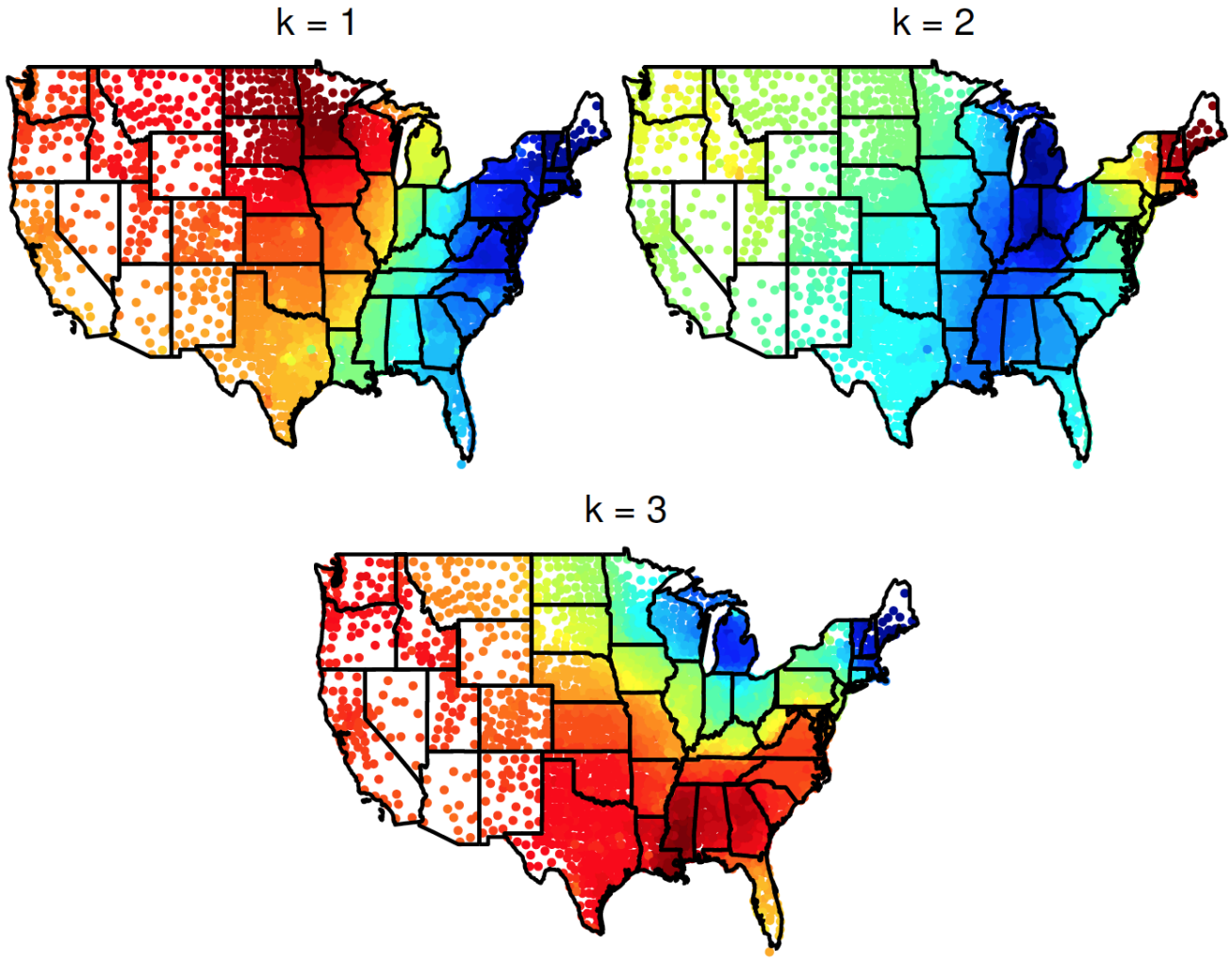
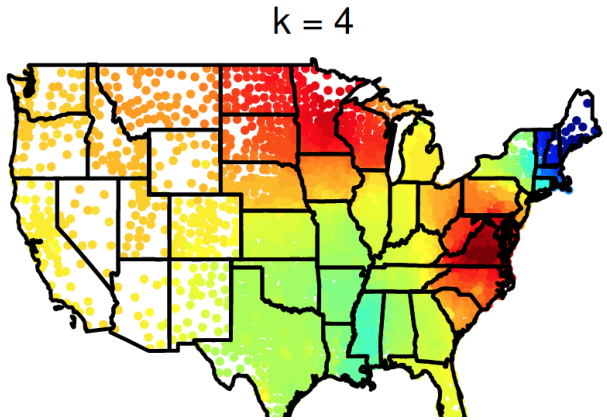
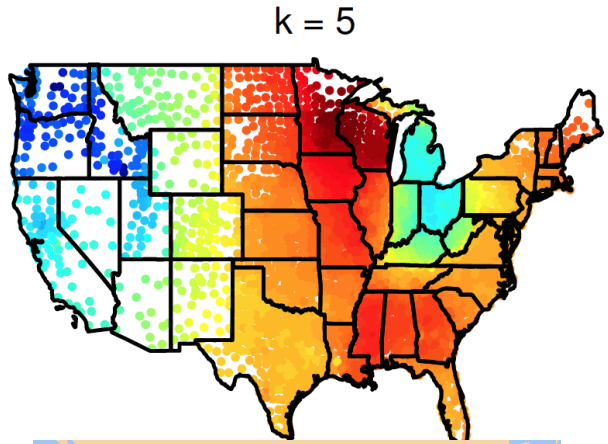
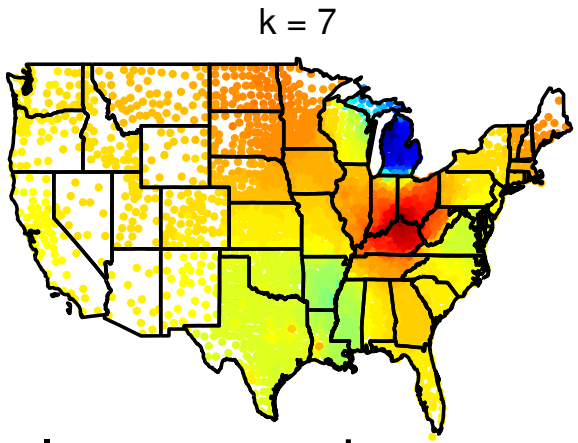


Figure: Top three eigenvectors correspond to **global cuts** between various coasts within the US. The only state that stands out individually is Michigan (MI) for $k = 3$, which has rank 2.

Eigenvector colorings vs Ratio Degree







- $k = 4$: the largest entries correspond to counties in Virginia (VA) which is also ranked 1^{st}
- $k = 5$: Wisconsin (WI) ranked 14
- $k = 6$: the states coloured in dark red and dark blue are Georgia (GA) with rank 3, and Missouri (MO) of rank 8
- $k = 7$: Michigan (MI), of rank 2, stands out as the only dark blue coloured state.