The K-modes and Laplacian K-modes algorithms for clustering

Miguel Á. Carreira-Perpiñán
Electrical Engineering and Computer Science
University of California, Merced
http://faculty.ucmerced.edu/mcarreira-perpinan

work with **Weiran Wang** (Amazon)

Centroids-based clustering and centroid validity

- Given a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$, centroids-based clustering:
  - partition data points into clusters $k = 1, \ldots, K$
  - estimate a centroid $\mathbf{c}_k \in \mathbb{R}^D$ of each cluster k.
- Three widely used algorithms of this type:
  - K-means
  - K-medoids
  - mean-shift.

  No K-modes algorithm exists that clusters data by returning exactly K meaningful modes.
- Also, we want centroids that are valid patterns (even with data having nonconvex or manifold structure) and representative of their cluster.

K-means and K-medoids clustering

K-means minimizes the point-to-centroid squared Euclidean distances:

$$\min_{\mathbf{Z},\mathbf{C}} E(\mathbf{Z},\mathbf{C}) = \sum_{k=1}^{K}\sum_{n=1}^{N} z_{nk}\|\mathbf{x}_n - \mathbf{c}_k\|^2 \quad \text{s.t.} \quad \begin{cases} z_{nk} \in \{0,1\},\ \sum_{k=1}^{K} z_{nk} = 1, \\ n = 1,\dots,N,\ k = 1,\dots,K \end{cases}$$

- Alternating optimization over centroids and assignments.
- Produces exactly K clusters.
- Produces convex clusters (Voronoi tessellation: each point is assigned to its closest centroid in Euclidean distance).
  In practice, tends to define round clusters whether or not the data has such structure.
- Computationally $\mathcal{O}(KND)$ per iteration.

K-medoids: centroids must be training points (exemplars).

$\mathbf{c}_k \in \{\mathbf{x}_1,\dots,\mathbf{x}_N\}$ for $k = 1,\dots,K$.

Mean-shift clustering

Find modes of kernel density estimate with kernel G and bandwidth σ:

$$p(\mathbf{x};\sigma) = \frac{1}{N}\sum_{n=1}^{N} G\left(\left\|\frac{\mathbf{x}-\mathbf{x}_n}{\sigma}\right\|^2\right) \qquad \mathbf{x}\in\mathbb{R}^D$$

and assign point $\mathbf{x}_n$ to the mode it converges to under the mean-shift algorithm: a fixed-point iteration (weighted average of data points).

For Gaussian kernel G: $\mathbf{x} \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x})\mathbf{x}_n$.

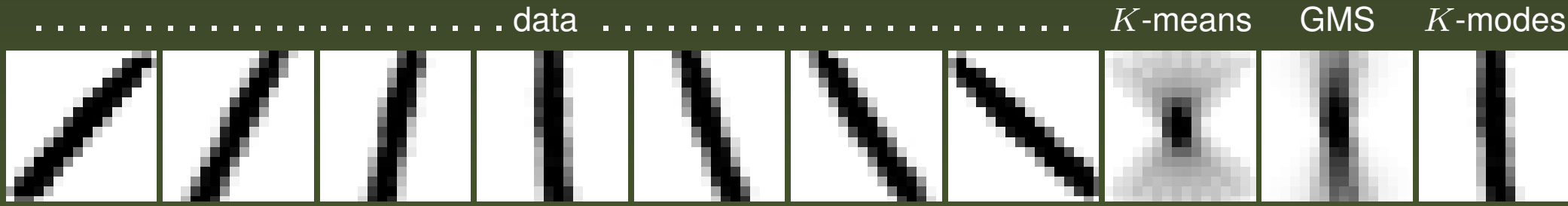
- Number of clusters depends on bandwidth σ.
  Not straightforward to find exactly K clusters.
- Can obtain clusters of arbitrary shapes.
  Very popular in low-dimensional clustering applications such as image segmentation.
- Does not work well in high dimension.
  For most bandwidth values σ, the KDE has either just one mode or too many modes.
- Computationally $\mathcal{O}(N^2D)$ per iteration (very slow).

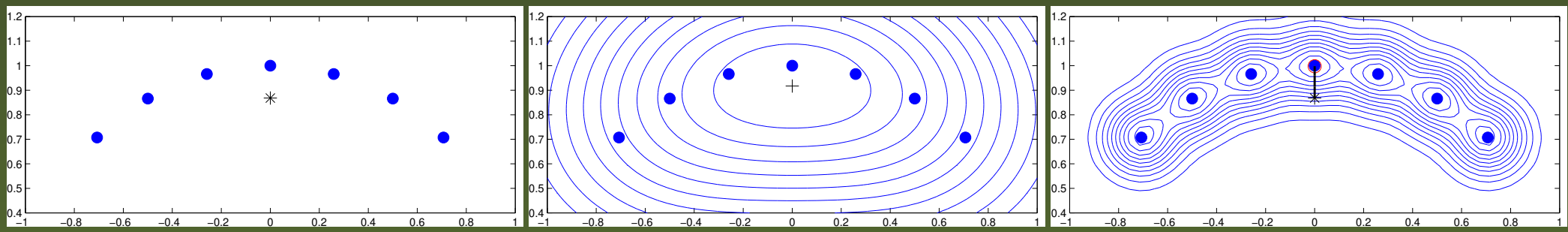
Validity of the centroids

The centroids are patterns in the input space $\mathbb{R}^D$.

- Are they valid?
- If so, how representative are they of their cluster?

Some applications define clusters with nonconvex or manifold structure (e.g. images, shapes).

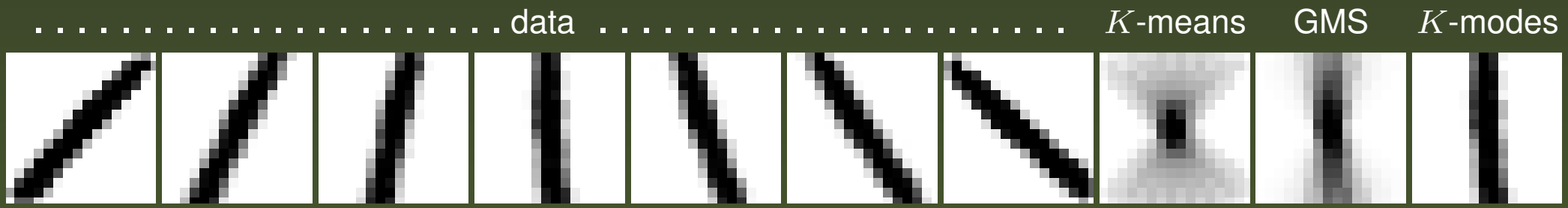


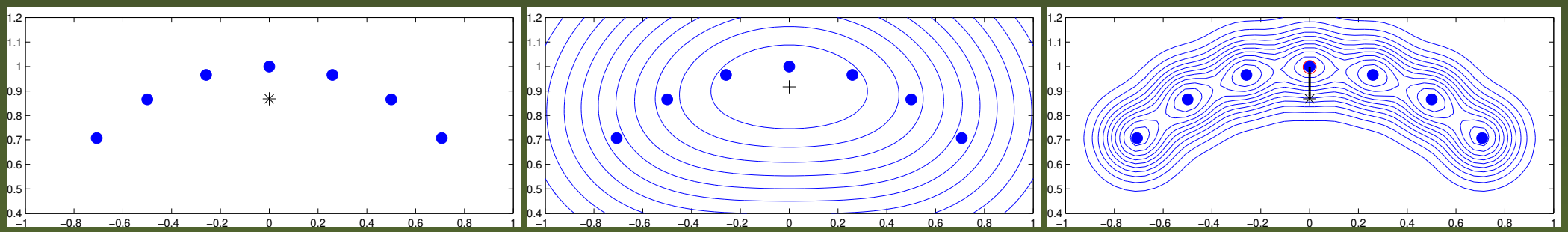


Validity of the centroids (cont.)

- K-means: centroids need not be valid (with nonconvex clusters, away from regions where data points lie).
- Mean-shift: a true cluster may be split in several modes (small σ) or the single mode need not be valid (large σ).

  This may happen no matter how we set σ.
- K-medoids: exemplars are generally valid but may be noisy or atypical. 





K-modes: objective function

For a given assignment $\mathbf{Z}$, L is the sum of a KDE defined separately for each cluster. Thus, a clustering must move centroids to local modes, but also define K separate KDEs.

We take $G =$ Gaussian with a common bandwidth σ for all points.

$$\max_{\mathbf{Z},\mathbf{C}} L(\mathbf{Z},\mathbf{C}) = \sum_{k=1}^{K}\sum_{n=1}^{N} z_{nk}\, G\left(\left\|\frac{\mathbf{x}_n - \mathbf{c}_k}{\sigma}\right\|^2\right) \quad \text{s.t.} \quad \begin{cases} z_{nk} \in \{0,1\},\ \sum_{k=1}^{K} z_{nk} = 1 \\ n = 1,\dots,N,\ k = 1,\dots,K. \end{cases}$$

This naturally combines the K-means idea of clustering through binary assignment variables with the mean-shift idea that high-density points are representative of a cluster (for suitable bandwidth values).

Two special cases: $\begin{cases} \sigma \to \infty & K\text{-means} \\ \sigma \to 0 & \text{a form of } K\text{-medoids.} \end{cases}$

K-modes interpolates smoothly between these two cases, creating a continuous path that links a K-mean to a K-medoid. However, its most interesting behavior is for intermediate σ, where the centroids are denoised, valid patterns and typical representatives of their cluster.

K-modes: optimization

For fixed σ (and K): alternating optimization over $(\mathbf{Z}, \mathbf{C})$:

- Assignment step: given the centroids $\mathbf{C}$, assign each point $\mathbf{x}_n$ to its closest centroid in Euclidean distance. Like K-means.
- Centroid (mode-finding) step: given the assignments $\mathbf{Z}$, run mean-shift over each centroid on its current KDE. Like mean-shift but for just one point. Note this step need not be exact (we may do just a few mean-shift iterations).

Like K-means but finding modes instead of means: it interleaves a hard assignment step of data points to centroids with a mode-finding step that moves each centroid to a mode of the KDE defined by the points currently assigned to it.

Computational cost per outer-loop iteration $\mathcal{O}(KND)$, slightly slower than K-means but much faster than mean-shift or K-medoids.

Homotopy algorithm: start at $\sigma = \infty$ (K-means) and reduce σ until we reach a target value σ^*. This helps to find a better local optimum.

K-modes vs K-means and mean-shift clustering

- Basic user parameter in K-modes: desired number of clusters K.

  The bandwidth σ is a scaling device to refine the centroids. We find that representative, valid centroids are obtained for a wide range of intermediate σ values.

- Like K-means, K-modes defines convex clusters. Unlike K-means it defines a KDE per cluster and valid centroids, and is more robust to outliers. Compare the effect of $\|\mathbf{x}_n - \mathbf{c}_k\|^2$ in K-means vs $G(\|(\mathbf{x}_n - \mathbf{c}_k)/\sigma\|^2)$ in K-modes.

- Mean-shift equates modes with clusters. This can be problematic:
  - The true density of a cluster may be multimodal to start with.
  - KDEs are bumpy unless σ is unreasonably large, particularly so with outliers (which create small modes) or in high dimensions.

  K-modes provides one approach to this problem, by separating the roles of cluster assignment and of centroids as high-density points.

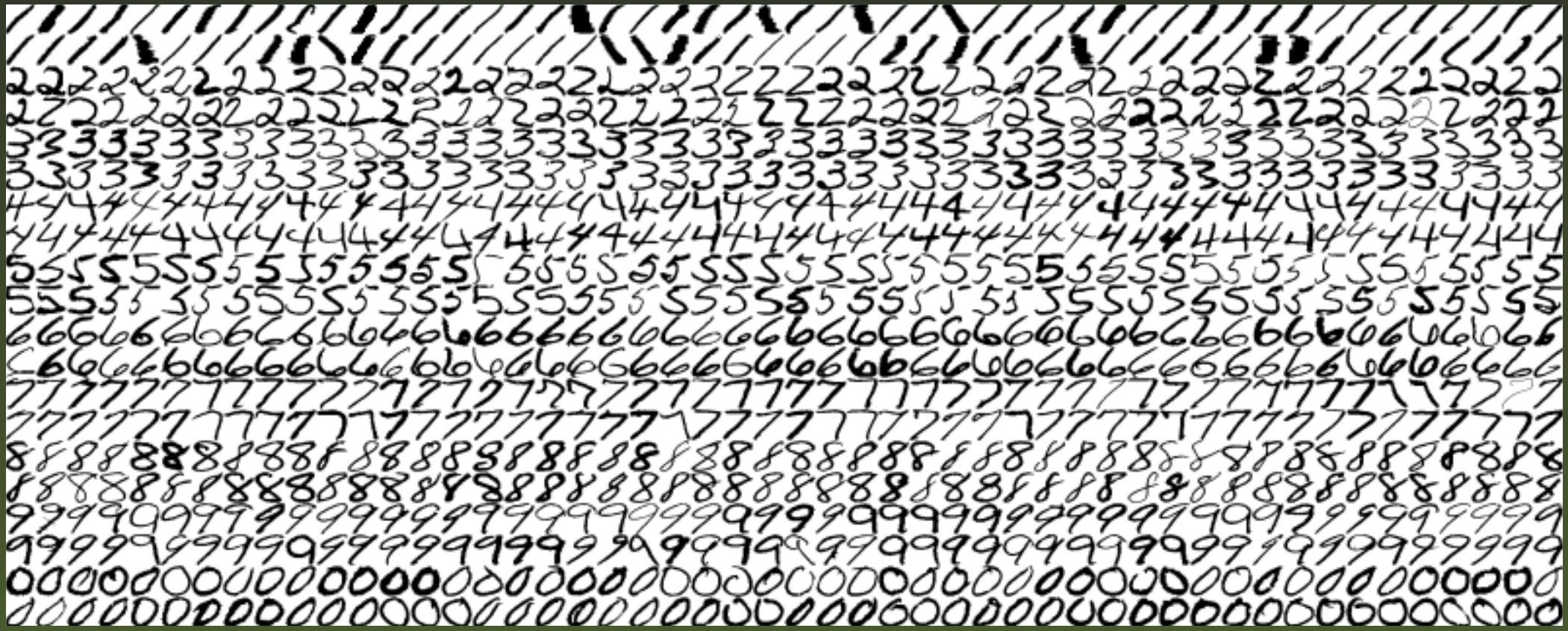
  Each cluster has its own KDE, which can be multimodal, and the homotopy algorithm tends to select an important mode among the modes within each cluster.

  This makes K-modes do well even in high-dimensional problems, where mean-shift fails.

  It is also much faster (the centroid step is like running mean-shift over just one point).

Experiment: handwritten digit images

USPS data set: $N = 1\,000$ grayscale images of 16 $\times$ 16, so $\mathbf{x} \in \mathbb{R}^{256}$

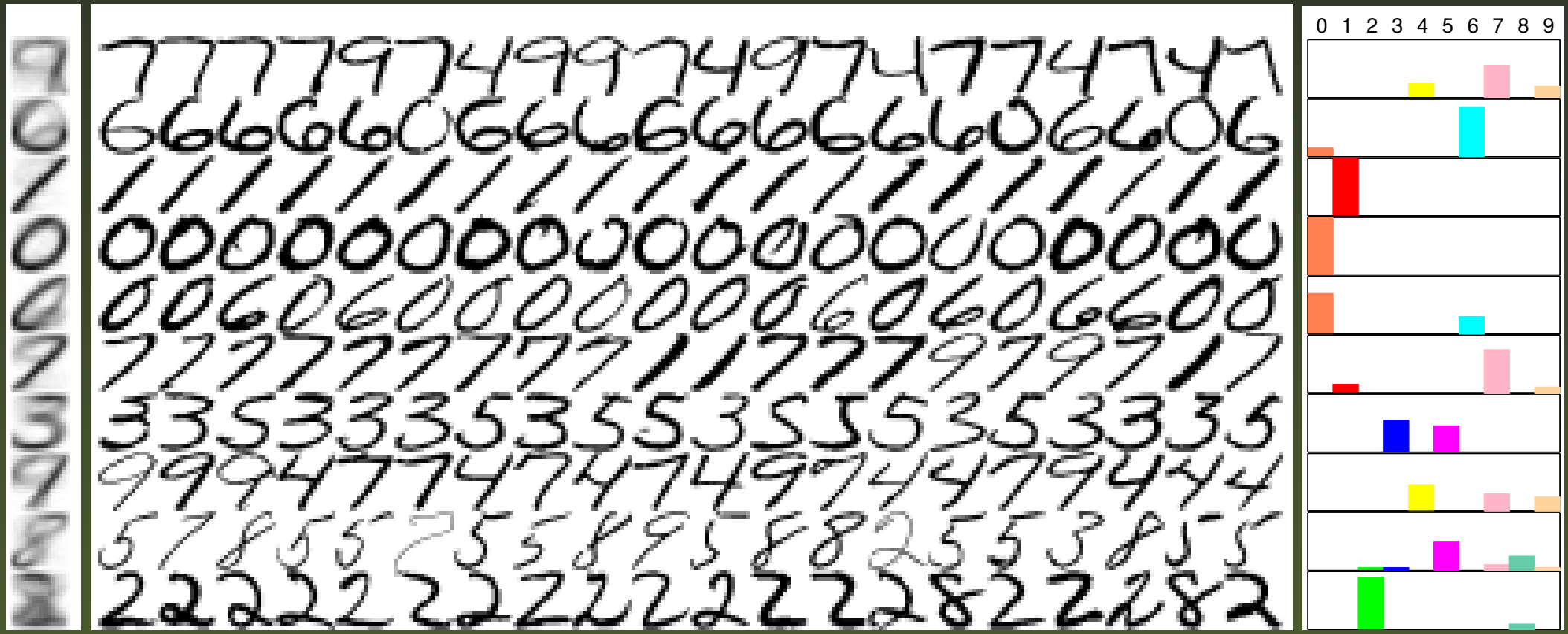


Experiment: handwritten digit images (cont.)

K-means result ($K = 10$, $\sigma = \infty$)

cent. 20 nearest neighbors of the centroid in its cluster

histogram of class labels for the neighbors

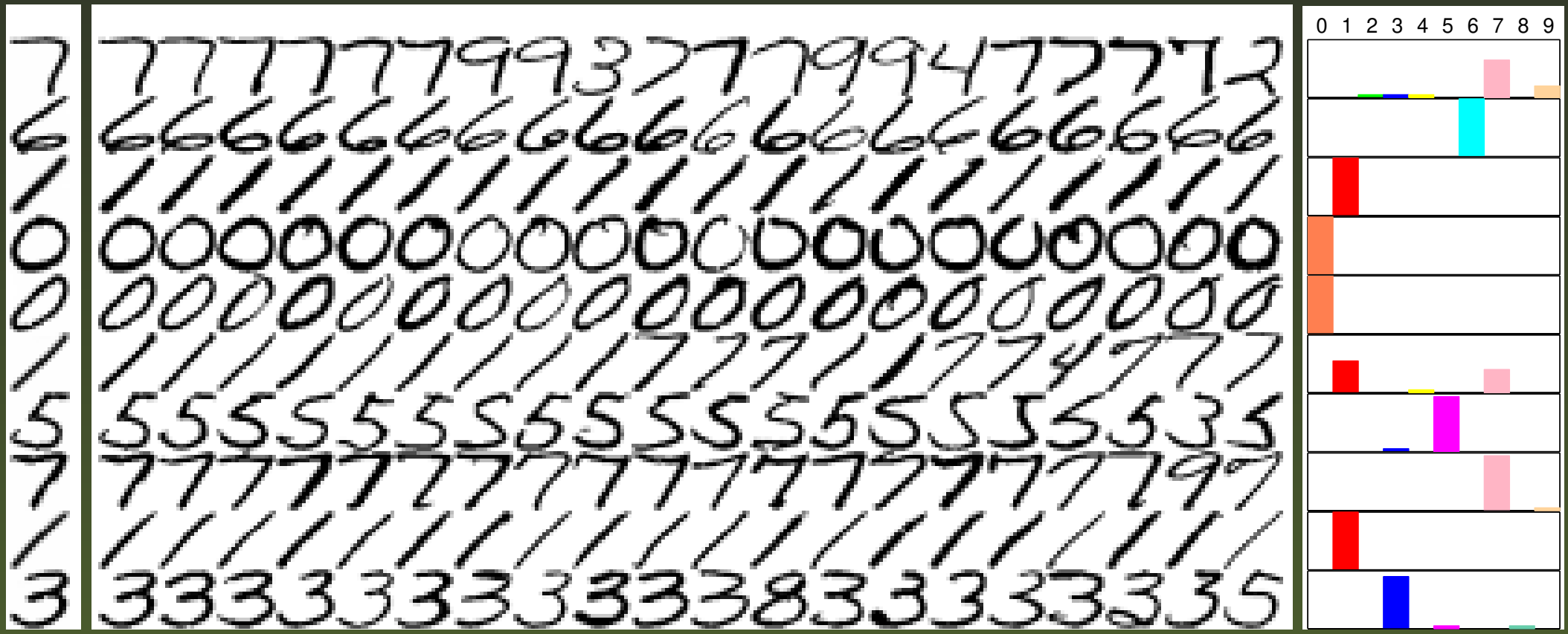


- Centroids average digits of different identity and style.
- Centroid neighborhoods are not homogeneous.

Experiment: handwritten digit images (cont.)

K-modes result ($K = 10$, $\sigma = 1$)

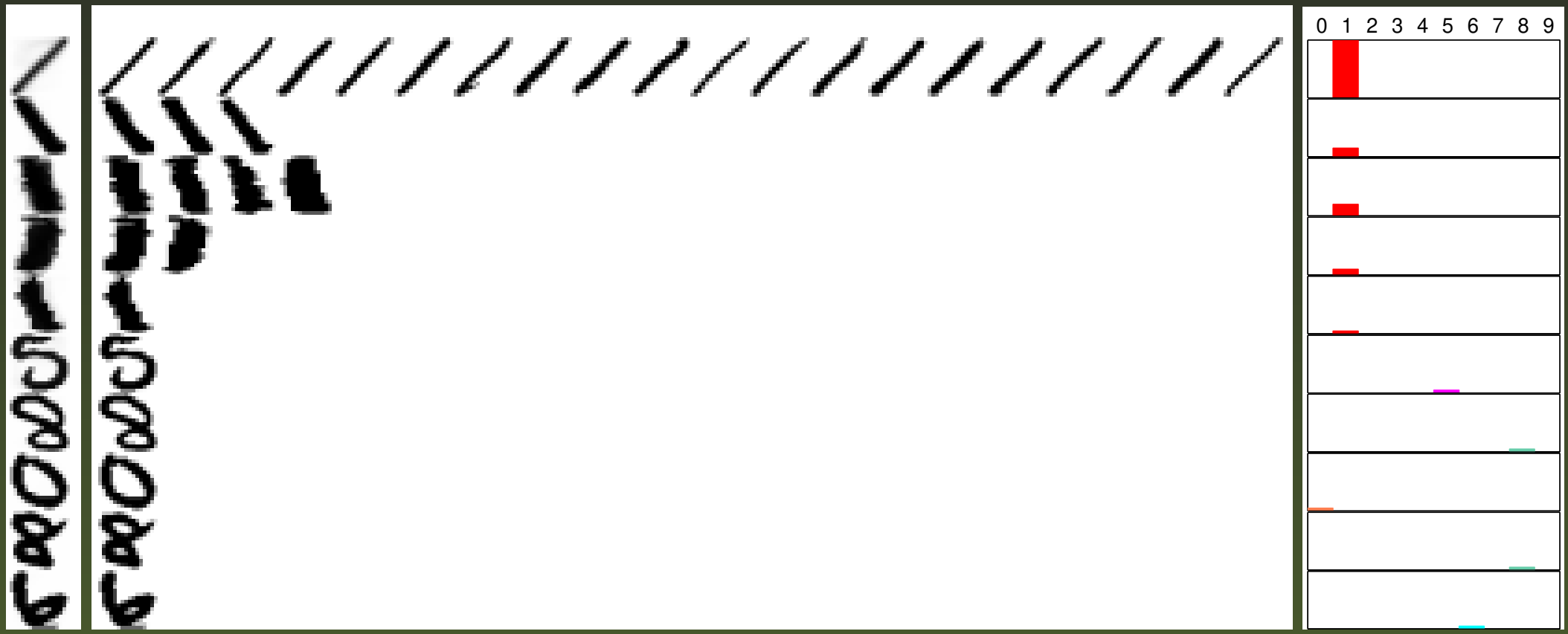
cent. 20 nearest neighbors of the centroid in its cluster histogram of class labels for the neighbors



- The centroids move to denser, more homogeneous regions.
- Representative of their neighborhood: they look like valid digits.

Experiment: handwritten digit images (cont.)

Mean-shift result ($\sigma = 1.8369$ so $K = 10$)



- Very hard to tune σ in order to achieve $K = 10$ clusters.
  Most values of σ give either one or many modes.
- In high dimensions, many modes have very few associated points and correspond to outliers.

K-modes: conclusion

- It allows the user to work with a KDE of bandwidth σ (like mean-shift clustering) but produce exactly K clusters (like K-means).
- It finds centroids that are valid patterns and lie in high-density areas (unlike K-means), are representative of their cluster and neighborhood, yet they average out noise or idiosyncrasies that exist in individual data points.

  An adequate smoothing can be achieved for a range of intermediate values of the bandwidth σ.
- Computationally, it is somewhat slower than K-means but far faster than mean-shift.

One disadvantage: like K-means, K-modes defines convex clusters (a Voronoi tessellation). This is solved by Laplacian K-modes.

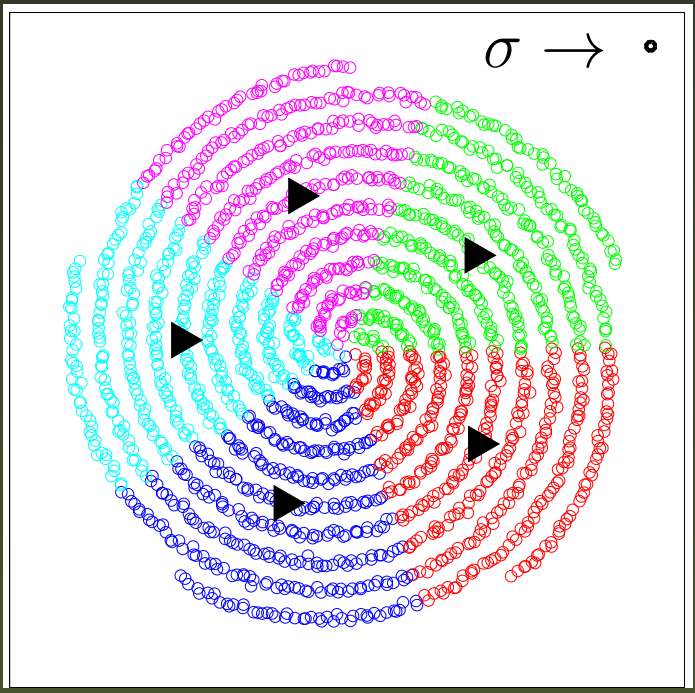
K-modes vs Laplacian K-modes: 5 spirals

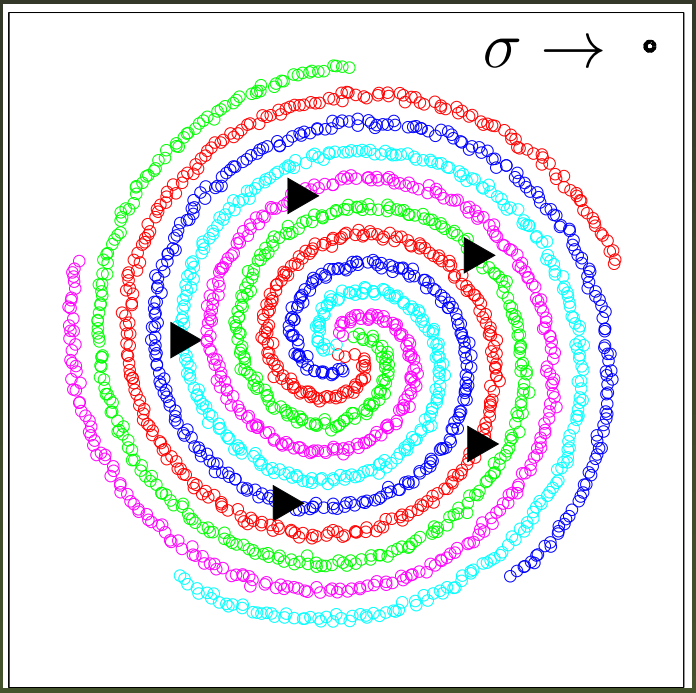
K-modes ($\lambda = 0$)

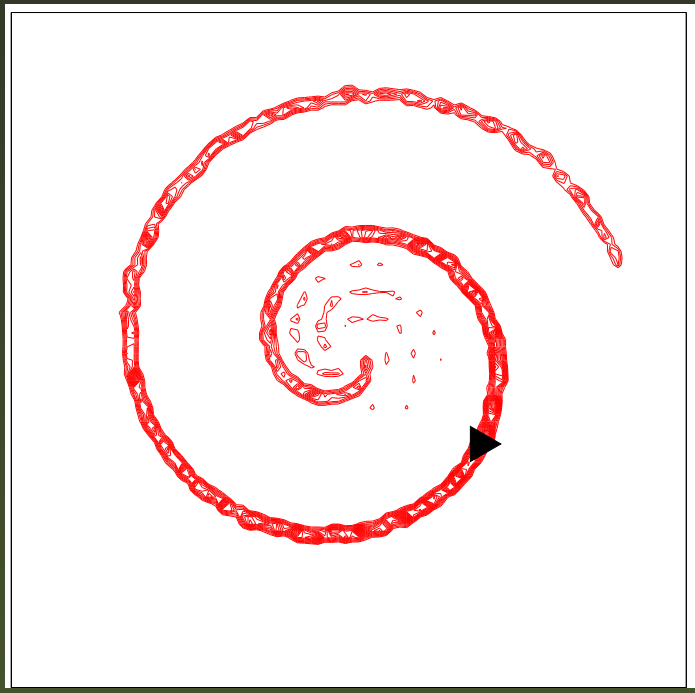
Laplacian K-modes ($\lambda = 100$)

centroids

KDE for red cluster







- Like K-modes: a KDE per cluster and valid, representative centroids (each a cluster mode)
- Beyond K-modes: nonconvex clusters and point-to-cluster soft assignments (nonparametric posterior probabilities $p(k|\mathbf{x})$).

Laplacian K-modes: objective function

We change the assignment rule of K-modes to handle more complex shaped clusters based on two ideas:

- Nearby data points should have similar assignments.
- Soft assignments $z_{nk} \in [0, 1]$: flexible clusters, simpler optimization.

Then:

- We first build a graph (e.g. k-nearest-neighbor graph) on the training set, and let $w_{mn} \geq 0$ be a similarity (e.g. binary, heat kernel) between points $\mathbf{x}_m$ and $\mathbf{x}_n$ for $n, m = 1, \ldots, N$.
- We add to the K-modes objective function a Laplacian smoothing term $\frac{\lambda}{2} \sum_{m,n=1}^{N} w_{mn} \|\mathbf{z}_m - \mathbf{z}_n\|^2$ to be minimized, where $\mathbf{z}_n \in \mathbb{R}^K$ is the assignment vector to each of the K clusters of $\mathbf{x}_n$, $n = 1, \ldots, N$, and $\lambda \geq 0$ is a trade-off parameter. Widely used in clustering (spectral clustering), dimensionality reduction (Laplacian eigenmaps), semi-supervised learning, etc.
- Soft assignments: make $\mathbf{z}_n$ continuous but nonnegative, unit-sum. $z_{nk} \approx p(k|\mathbf{x}_n)$, the probability of assigning $\mathbf{x}_n$ to cluster k.

Laplacian K-modes: objective function (cont.)

$$\min_{\mathbf{Z},\mathbf{C}} \quad \frac{\lambda}{2} \sum_{m,n=1}^{N} w_{mn} \|\mathbf{z}_m - \mathbf{z}_n\|^2 - \sum_{k=1}^{K} \sum_{n=1}^{N} z_{nk}\, G\left(\left\| \frac{\mathbf{x}_n - \mathbf{c}_k}{\sigma} \right\|^2 \right)$$

$$\text{s.t.} \quad z_{nk} \geq 0, \quad \sum_{k=1}^{K} z_{nk} = 1, \quad n = 1, \dots, N, \quad k = 1, \dots, K.$$

This naturally combines three powerful ideas in clustering:

- The explicit use of assignment variables (as in K-means).
- The estimation of cluster centroids which are modes of each cluster's density estimate (as in mean-shift).
- The smoothing effect of the graph Laplacian, which encourages similar assignments for nearby points (as in spectral clustering).

Interesting special case: $\lambda > 0$ and $\sigma \to \infty$ (Laplacian K-means).

Laplacian K-modes: optimization

For fixed (σ, λ) (and K): alternating optimization over $(\mathbf{Z}, \mathbf{C})$:

- Assignment step (over $\mathbf{Z}$ given the centroids $\mathbf{C}$): a convex quadratic program. Solvable in various ways, see paper.
- Centroid (mode-finding) step: as for K-modes (run mean-shift over each centroid on its current KDE).

Computational cost per outer-loop iteration $\mathcal{O}(KND)$, as for K-modes.

Homotopy algorithm: start at $(\sigma = \infty,\ \lambda = 0)$ (K-means) and reduce σ and increase λ until we reach a target value (σ^*, λ^*). This helps to find a better local optimum.

Out-of-sample problem

Given a test point $\mathbf{x} \in \mathbb{R}^D$, find its soft assignment $\mathbf{z}(\mathbf{x})$ to the clusters found during training:

- Solve the Laplacian K-modes problem with a dataset consisting of the original training set augmented with $\mathbf{x}$, but keeping $\mathbf{Z}$ and $\mathbf{C}$ fixed to the values obtained during training. This reduces to the following quadratic program (for fixed $\bar{\mathbf{z}}$, $\mathbf{q}$ and γ):

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z}} \tfrac{1}{2}\|\mathbf{z} - (\bar{\mathbf{z}} + \gamma\mathbf{q})\|^2 \qquad \text{s.t.} \qquad \mathbf{1}_K^T\mathbf{z} = 1,\ \mathbf{z} \geq \mathbf{0}$$

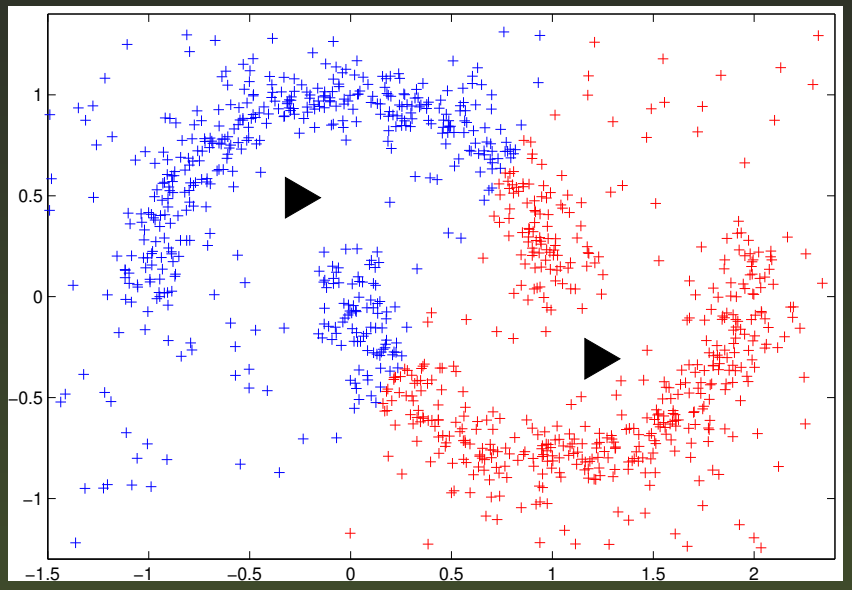
(projection of vector $\bar{\mathbf{z}} + \gamma\mathbf{q} \in \mathbb{R}^K$ onto the probability simplex).

- $\mathbf{z}(\mathbf{x})$ results from a mixture of the two assignment rules:
  - $\bar{\mathbf{z}}$: average of training points' assignments weighted by $w_n(\mathbf{x})$.
  - $\mathbf{q}$: Gaussian posterior probability of point $\mathbf{x}$ to centroid k.
- Computational cost $\mathcal{O}(ND)$ (dominated by the cost of $\bar{\mathbf{z}}$).

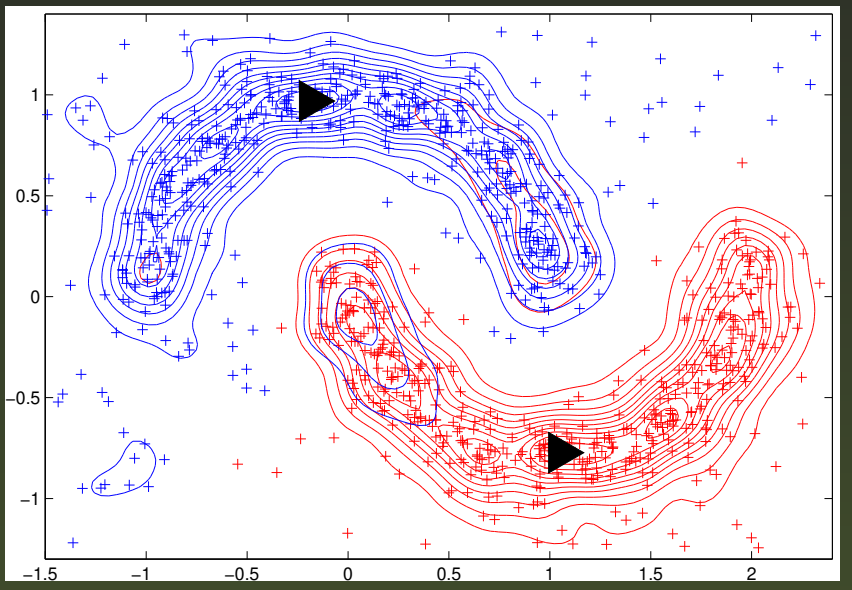
$\mathbf{z}(\mathbf{x})$ gives a nonparametric model for the cluster posterior probabilities $p(k|\mathbf{x})$ that can be applied to any data point $\mathbf{x}$ (training or test set).

Out-of-sample problem (cont.)

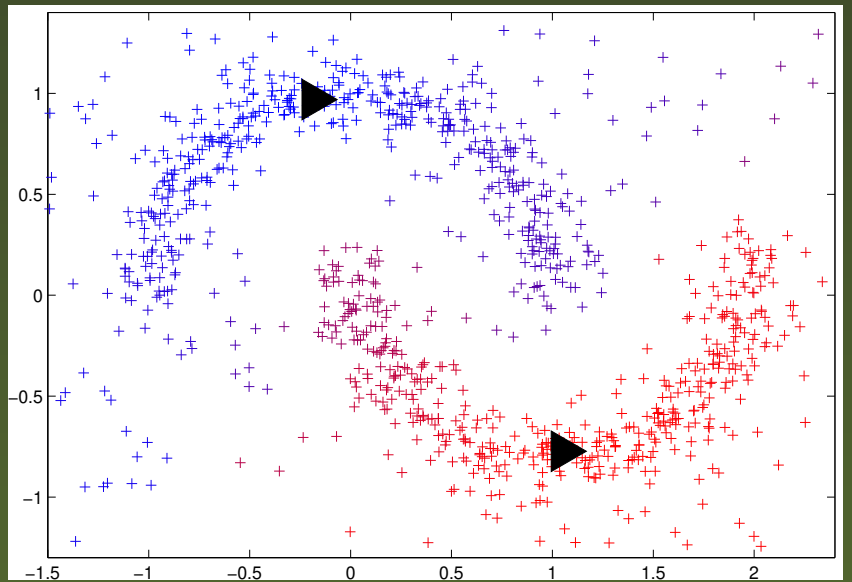
K-means



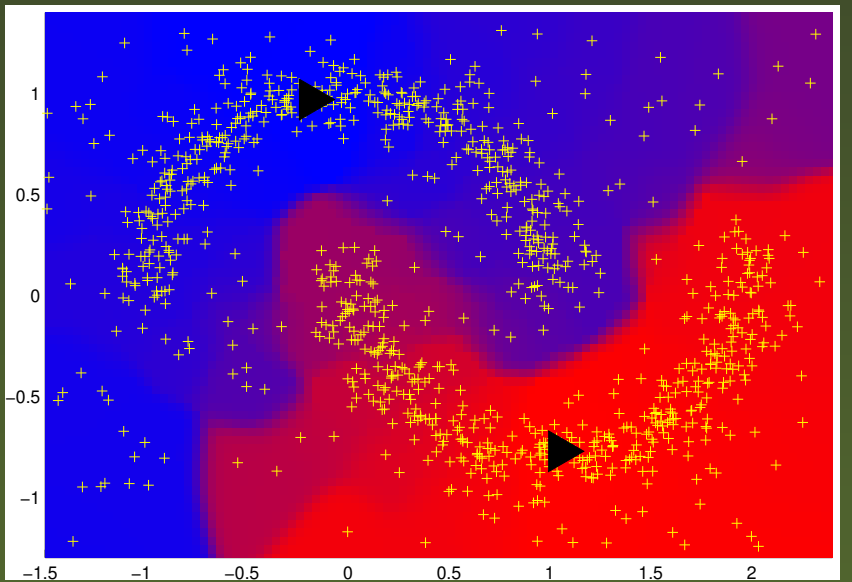
Laplacian K-modes



Soft assignments $\mathbf{z}(\mathbf{x}_n)$



Out-of-sample $\mathbf{z}(\mathbf{x})$



Parametric vs nonparametric cluster probabilities

Cluster posterior probabilities are helpful to estimate the uncertainty in clustering. Traditionally they are obtained using parametric models such as Gaussian mixtures trained to maximize the likelihood using EM.

Laplacian K-modes has two important advantages here:

- Its assignments optimize an objective designed specifically for clustering, unlike the likelihood.
- It produces a nonparametric model for clusters and assignments, which is more flexible.

  Likewise, its optimization does not depend on the particular type of clusters—there is a unique Laplacian K-modes algorithm, while each mixture model requires its own, particular EM algorithm.

In Laplacian K-modes:

- The density of cluster k is a Gaussian KDE with adaptive weights $p(\mathbf{x}|k) = \sum_{n=1}^{N} z_{nk}\, G(\|(\mathbf{x}_n - \mathbf{x})/\sigma\|^2)$. Weights are exactly zero for many points.
- Posterior probabilities $p(k|\mathbf{x})$ given a point $\mathbf{x}$: not in closed form.

Experiments: clustering evaluation

Statistics of high-dimensional data sets (with known class labels)

dataset	size (N)	dimensionality (D)	# of classes (K)
MNIST (handwritten digit images)	2 000	784	10
COIL–20 (rotated object images)	1 440	1 024	20
TDT2 (documents of different topics)	9 394	36 771	30

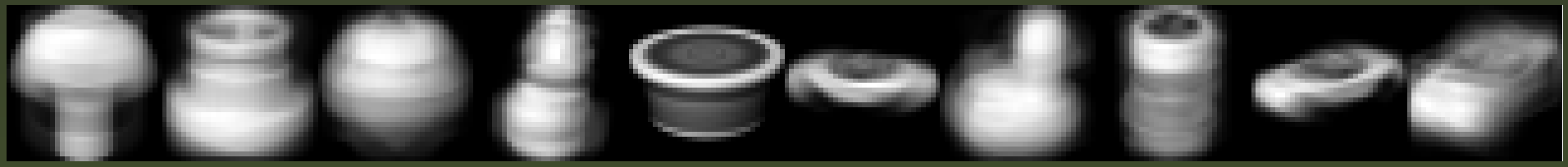
mean±std (20 runs): ACC: clustering accuracy (%), NMI: normalized mutual info. (%), T: runtime (")

	dataset	K-means	K-modes	GMS	DCD	NCut	GNMF	Lap. K-modes
ACC	MNIST	54.0±2.7	56.9±2.4	N/A	63.4±6.2	69.2±5.6	68.8±6.2	70.4±3.0
	COIL–20	54.4±4.8	61.6±3.4	27.2	65.8±2.3	64.6±5.4	67.4±5.5	76.0±3.1
	TDT2	58.0±4.5	61.8±3.9	N/A	53.1±3.0	66.9±3.5	78.0±4.4	87.3±3.4
NMI	MNIST	51.5±1.1	52.6±0.9	N/A	63.0±4.0	74.3±2.3	73.7±3.0	74.2±2.3
	COIL–20	70.5±2.7	73.6±1.6	38.9	74.8±1.0	81.8±3.2	84.0±3.0	85.3±1.5
	TDT2	71.3±1.7	72.8±1.6	N/A	66.6±0.9	74.5±1.4	80.7±2.2	85.8±1.8
T	MNIST	19.9	192.3	N/A	2 022.9	311.6	396.7	512.6

Experiments: COIL–20 dataset

$\mathbf{x}_n =$ greyscale image of 128×128, rotations of 20 objects every $5°$:

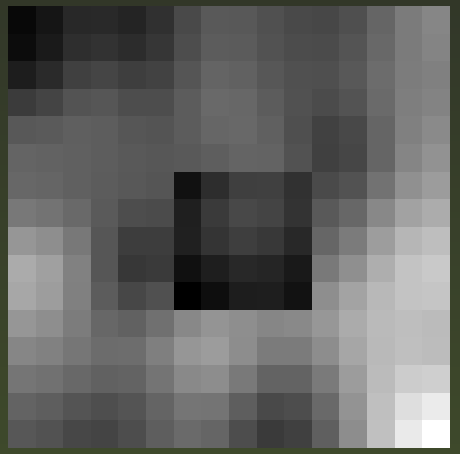
K-means ($K = 10$)



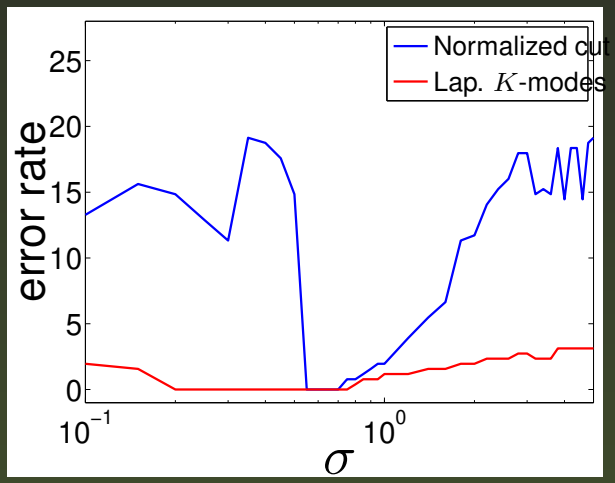
Laplacian K-modes ($K = 10$)

Experiments: figure-ground segmentation

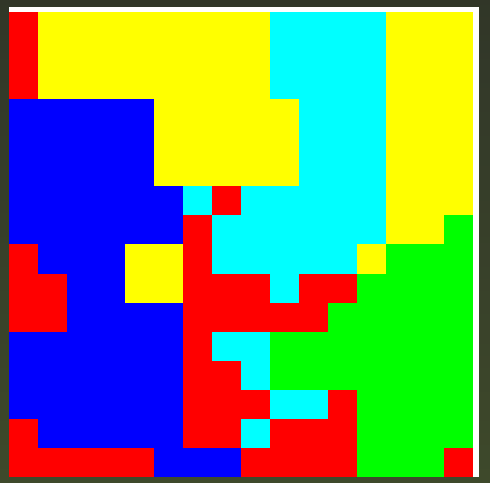
original image



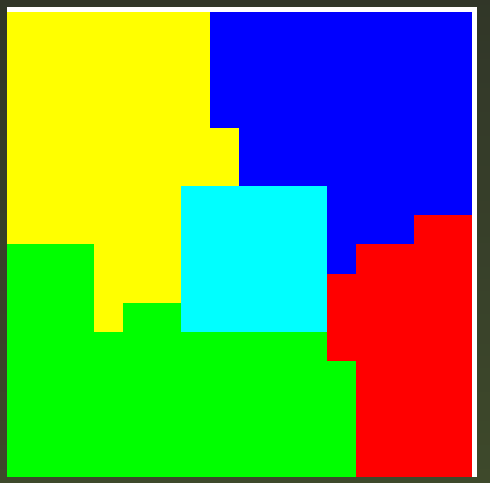
classification error



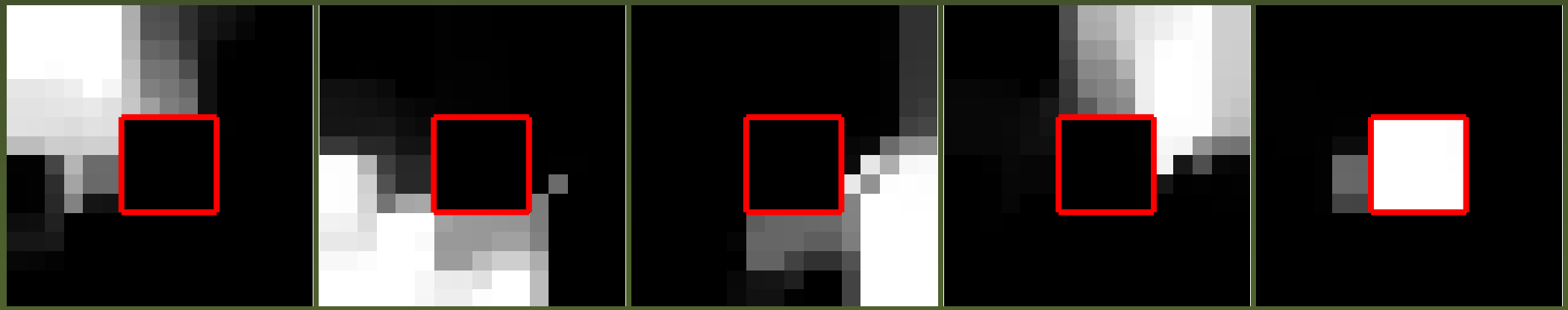
Normalized cut ($\sigma = 0.2$, $K = 5$)



Laplacian K-modes ($\sigma = 0.2$, $K = 5$)



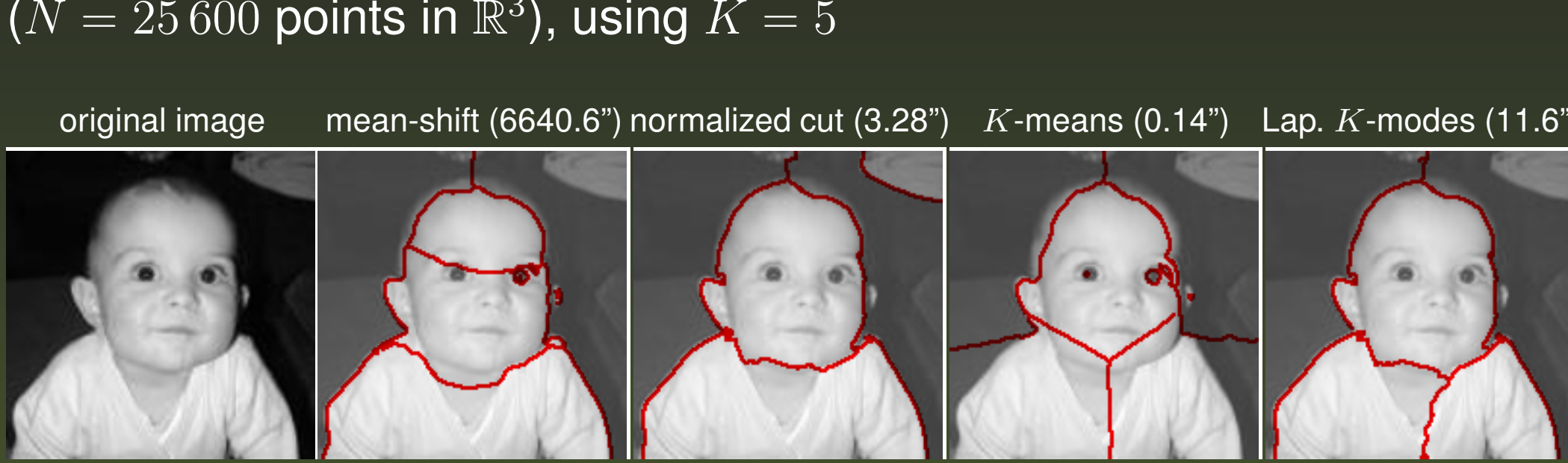
Soft assignments $z_{nk} \approx p(k|\mathbf{x}_n)$ for each cluster in Laplacian K-modes



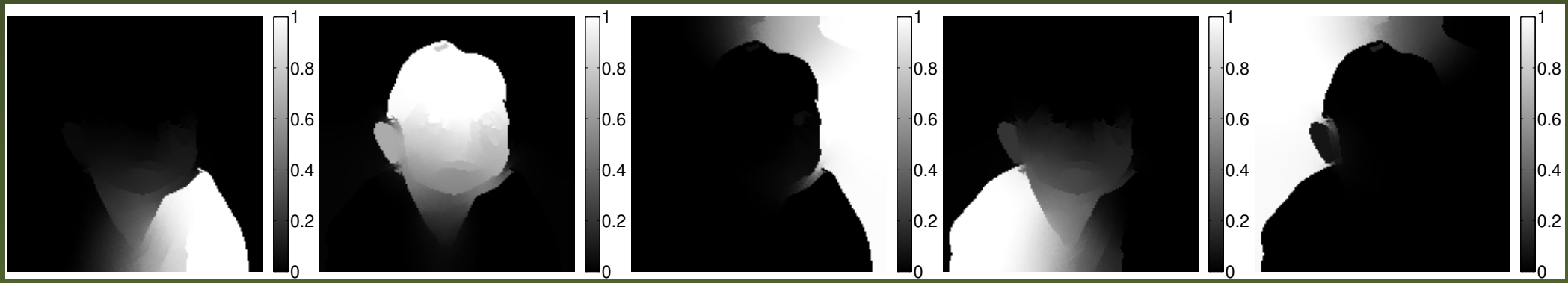
Experiments: image segmentation

Grayscale image of 160 $\times$ 160, feature vector = (location,intensity) ($N = 25\,600$ points in $\mathbb{R}^3$), using $K = 5$

original image | mean-shift (6640.6") | normalized cut (3.28") | K-means (0.14") | Lap. K-modes (11.6")



Soft assignments $z_{nk} \approx p(k|\mathbf{x}_n)$ for each cluster in Laplacian K-modes



Laplacian K-modes: conclusion

- It allows the user to work with a KDE of bandwidth σ (like mean-shift clustering) but produce exactly K clusters (like K-means).
- It finds centroids that are valid patterns and lie in high-density areas (unlike K-means), are representative of their cluster and neighborhood, yet they average out noise or idiosyncrasies that exist in individual data points.
- Computationally, it is slower than K-means but far faster than mean-shift.
- It finds density estimates for each cluster, even with challenging problems where the clusters have manifold structure, are highly nonconvex or in high dimension, as with images or text data.
- It provides a nonparametric model for the cluster posterior probabilities $p(k|\mathbf{x})$ for any test point $\mathbf{x}$.

References and code

http://faculty.ucmerced.edu/mcarreira-perpinan

The papers describing the (Laplacian) K-modes algorithm:

- M. Á. Carreira-Perpiñán and W. Wang:
  The K-modes algorithm for clustering.
  arXiv:1304.6478, Apr. 23, 2013.
- W. Wang and M. Á. Carreira-Perpiñán:
  The Laplacian K-modes algorithm for clustering.
  arXiv:1406.3895, Jun. 5, 2014.

A review paper on mean-shift algorithms, including (Laplacian) K-modes:

- M. Á. Carreira-Perpiñán:
  A review of mean-shift algorithms for clustering.
  arXiv:1503.00687, Mar. 2, 2015.
  Also as "Clustering methods based on kernel density estimators: mean-shift algorithms", in *Handbook of Cluster Analysis* (C. Hennig, M. Meila, F. Murtagh and R. Rocci, eds.), CRC/Chapman and Hall, pp. 383–418.

Work partly supported by NSF CAREER award IIS–0754089.