Density Based Clustering Applied to Image Segmentation

Marilyn Vazquez

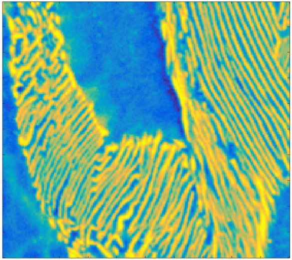
George Mason University
4400 University Drive
Fairfax, VA 22030

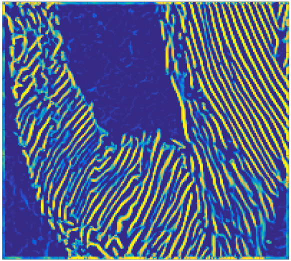
2017 Annual SIAM Meeting

Outline

1. Motivation
2. Cut-Cluster-Classify
3. Results
4. On Going Research
5. References

Classic Segmentation Scheme





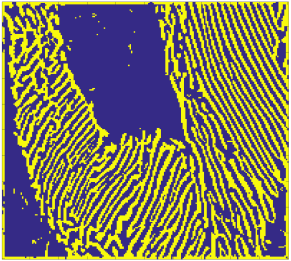
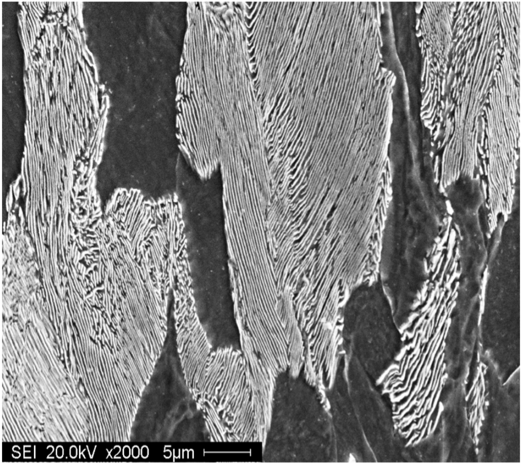


Image in false colors $\Longrightarrow$ Extract feature(s) $\Longrightarrow$ Segmentation according to the selected features

Challenges with Images

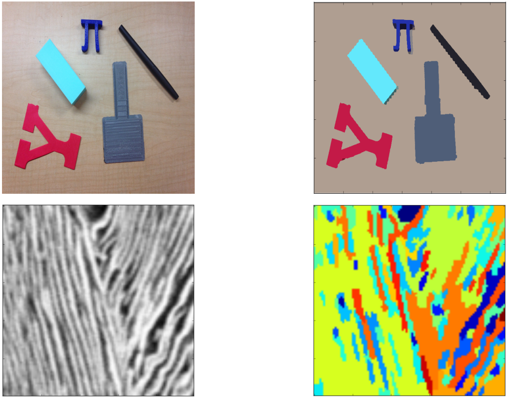


Example: Note that the objects of interest can be at different scales.

Challenges with Features

Depending on the choice of feature, results can vary.

Original image | Segmentation via "color feature"



How do you choose and represent a feature?

Segmentation

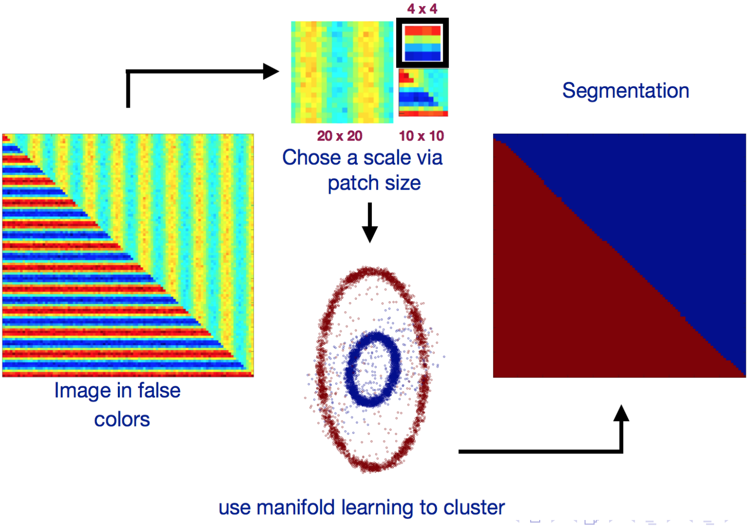
There are many ways to do this, for example:

- Region Based
- Neural Networks
- Histogram
- Threshold
- Clustering
- Optimization
- etc.

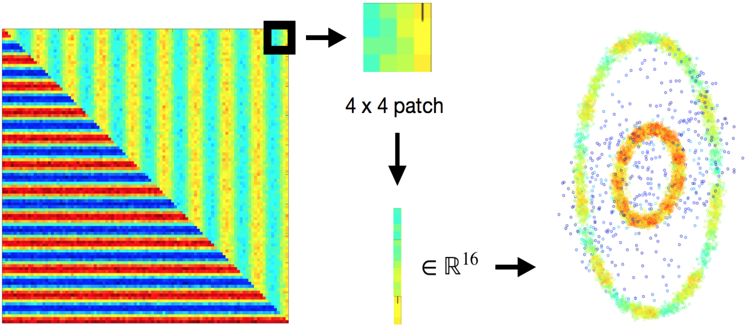
Which method is best to segment according to the feature of interest?

Our Proposed Scheme

Main Idea: Introduce manifold learning to achieve image segmentation

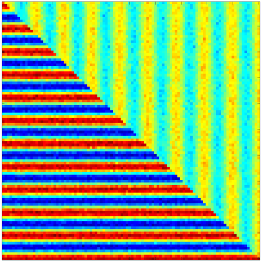


Data Driven Feature Space

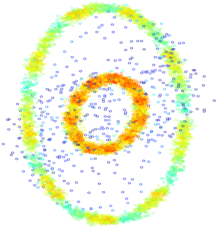


Each 4 × 4 patch is a point in a 16 dimensional data cloud, whose PCA representation is on the left.

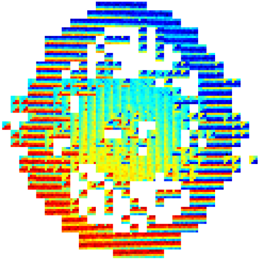
Patch Space



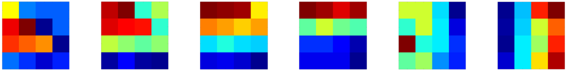
Original image
in false colors



PCA projection of
4 by 4 patches



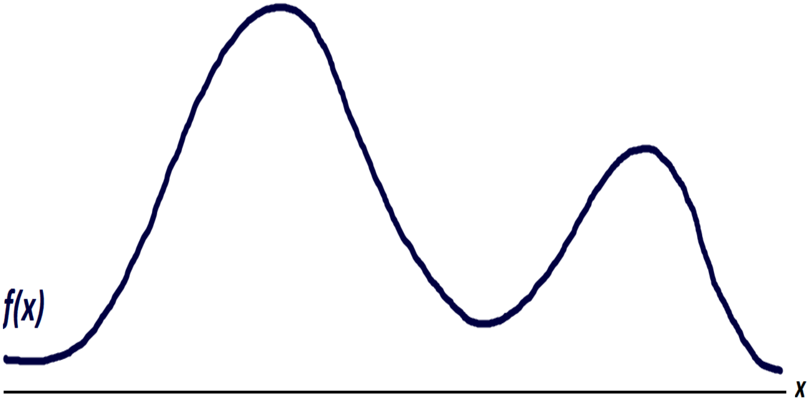
Patches on top
of their PCA projection



Sample 4 by 4 patches

Density Based Clustering

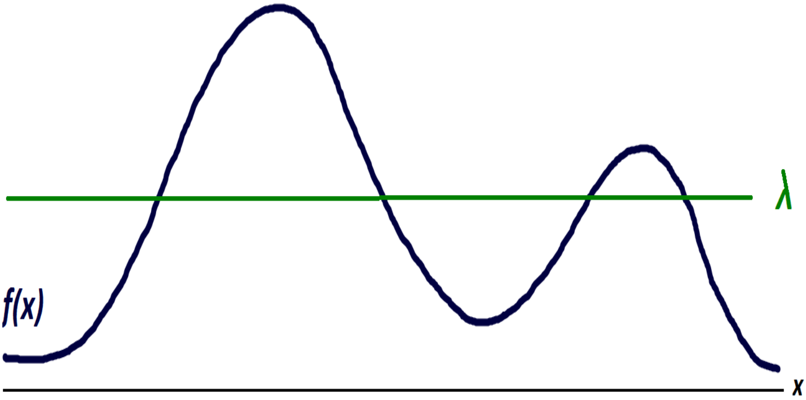
How would you cluster points with the following density f?



Challenges: (1) Points to be cluster are in m^2 dimensions. (2) We do not have access to the real density f, but an estimation $\hat{f}$.

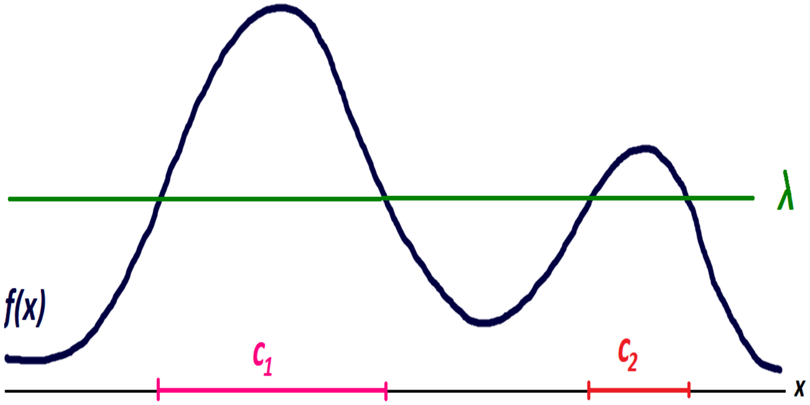
Cut: Super-level Set

Suppose we knew f, then a solution can be found looking at super-level sets, i.e. $\{x \in X_N : f(x) \geq \lambda\}$



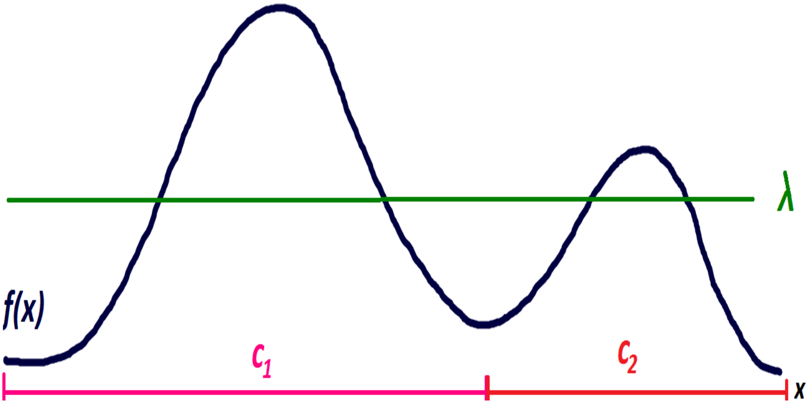
Cluster: Connected Components

Connected components of $\{x \in X_N : f(x) \geq \lambda\}$



Classify

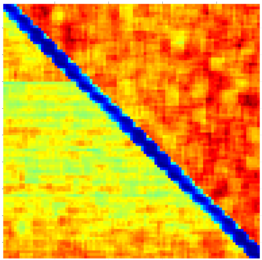
How do we label the rest of the points? Using a classifier!



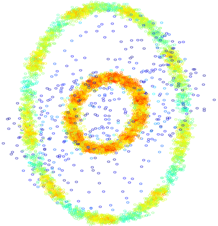
Cut

Step 1: Estimate sample density

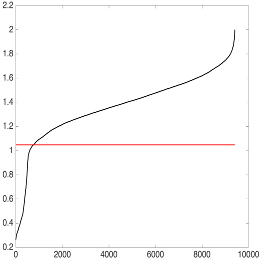
Step 2: Threshold, or cut, according to density



Patch density indexed by top left corner pixel



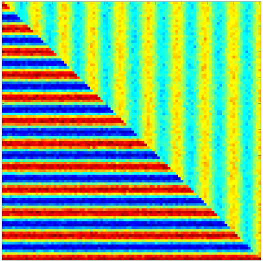
PCA projection colored by density



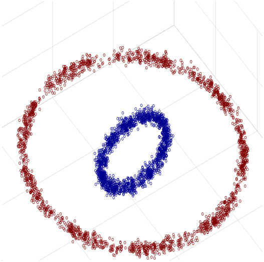
Sorted density with threshold drawn in red

Cluster

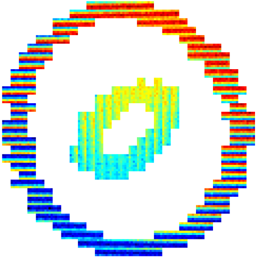
Step 3: Cluster points that passed the threshold



Original image
in false colors



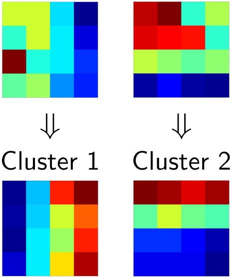
Sampled data clustered



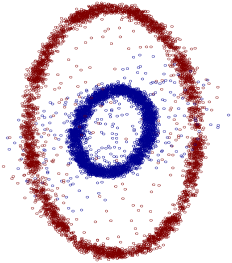
Patches on top of
the clustered projection

Classification

Step 4: Classify remaining points



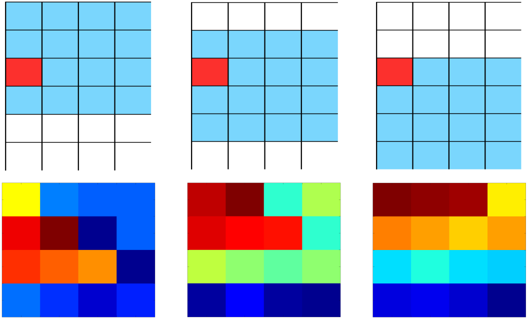
Sample patches being classified.



PCA projection of all 4 by 4 patches classified.

From patches to Pixels

Step 5: Label pixels using patch labels



To decide which cluster pixel (3,1) belongs to, we look at all the patches that it appears in. As illustrated in the top images, it only appears in 3 patches. The actual patches, shown on the bottom, were classified to cluster 1, 2, and 2, respectively. This means that cluster 2 gets 2 votes, and cluster 1 gets 1 vote. Therefore, this pixel gets placed in cluster 2.

Density Estimation

Our choice: $\hat{q}(x) = \frac{1}{||x - x_k||}$ i.e. one over the distance to the kth nearest neighbor (kNN).

Benefits of this estimator:

- Efficient algorithms to find $||x - x_k||$
- Consistent with $q(x)^{\frac{1}{d}}$

Consistent Estimator

Suppose $X_N \subset \mathbb{R}^d$ is the set of data points, $x \in \mathbb{R}^d, x_i \in X_N$. Let $k \geq 1$ be fixed, x_i be identically distributed, and q be the density function. Also, suppose x_i is the ith nearest neighbor of x and define $\epsilon = ||x - x_k||$. Then:

$$\frac{k}{N} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{d(x,x_i)\leq\epsilon\}} \implies \mathbb{E}\left[\frac{k}{N}\right] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{d(x,x_i)\leq\epsilon\}}\right]$$

$$\implies \frac{k}{N} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\left[\mathbb{1}_{\{d(x,x_i)\leq\epsilon\}}\right] = \frac{c_1\epsilon^d}{N}\sum_{i=1}^{N}(q(x) + \epsilon\mathcal{O}(\xi))$$

$$\implies \frac{k}{N} \propto \epsilon^d(q(x) + \mathcal{O}(\epsilon)) \implies ||x - x_k|| \propto q(x)^{-\frac{1}{d}} + \mathcal{O}(\epsilon)$$

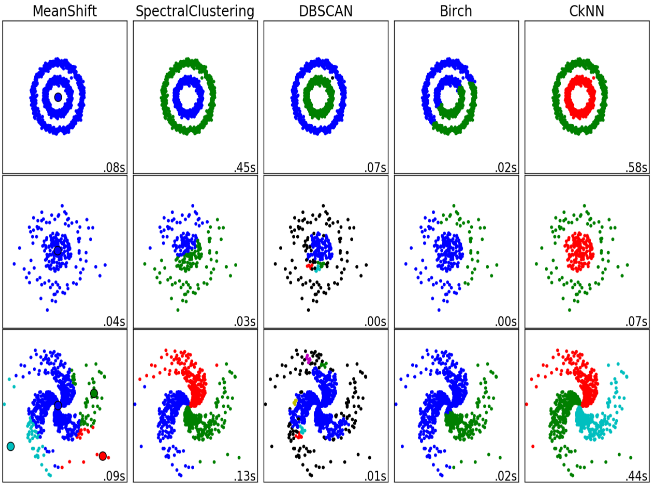
Cluster

Our choice: Continuous k Nearest Neighbor (CkNN) introduced by T. Berry and T. Sauer.

Facts:

- It is a very efficient and accurate method of clustering
- The unnormalized graph Laplacian spectrally converges to the Laplace-de Rham operator
- This implies that the connected components of the CkNN graph will converge to the connected components of the manifold

Cluster



Comparing 5 different clustering algorithms.

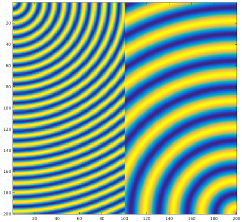
Classify

Our choice: kNN classifier, which is a very fast classifier but there could be better choices.

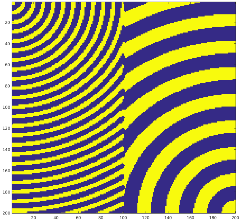
Steps

1. Find the k nearest neighbors that have a label
2. Pick the most frequent label among the neighbors

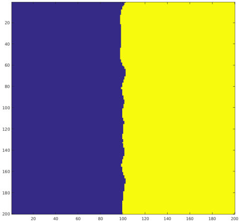
Synthetic Images: Multi-scale Features



Original image in false colors.

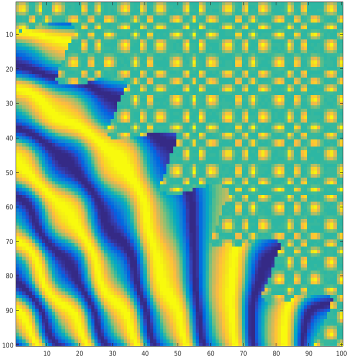


Segmentation from 2×2 patches.

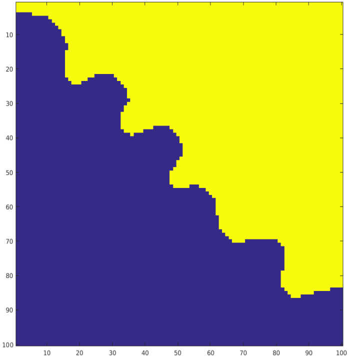


Segmentation from 12×12 patches.

Synthetic Images

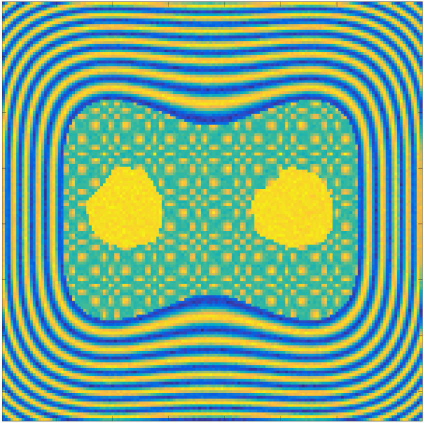


Original image in false colors.

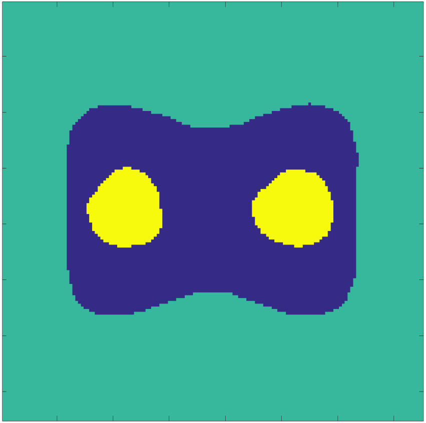


Segmentation from 10×10 patches.

Synthetic Images

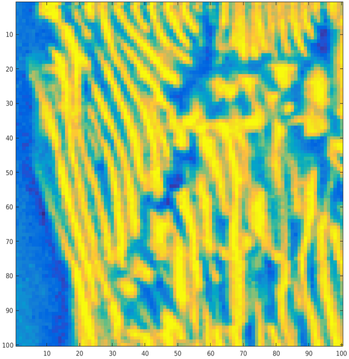


Original image in false colors.

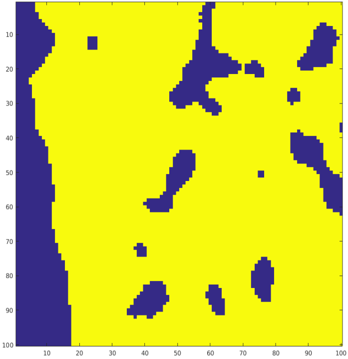


Segmentation from 6×6 patches.

Real Images

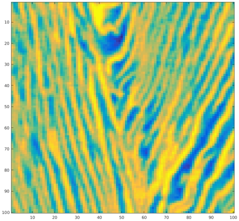


Original image in false colors.

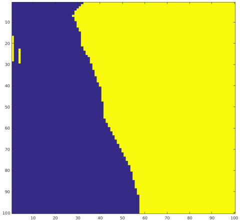


Segmentation from 5×5 patches.

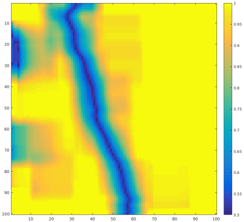
Real Images



Original image in false colors.

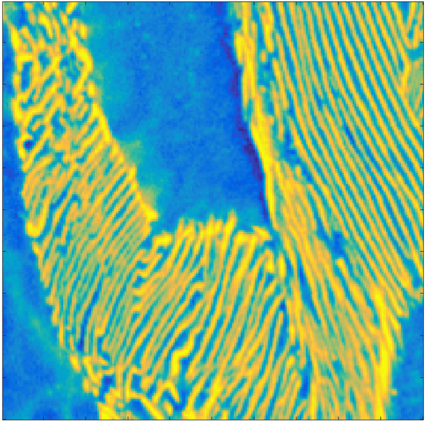


Segmentation from 8×8 patches.



Confidence level on the voting outcome.

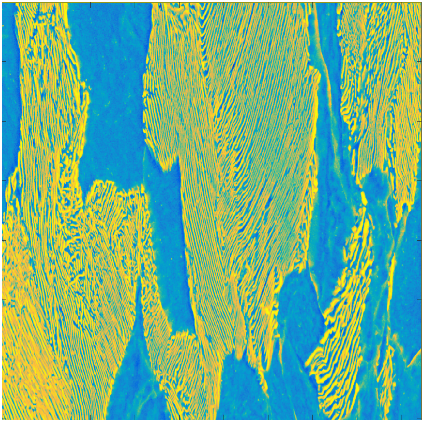
Real Images



Original image in false colors.

Segmentation from 20×20 patches.

Real Images



Original image in false colors.

Segmentation from 30×30 patches.

Future Directions

- Better metrics to measure similarities between patches
- Hierarchical approach to achieve multiscale segmentation: How do we incorporate and compare different patch sizes?
- Find a sense of consistency for classifiers

References

Berry, Tyrus and Sauer, Tim. *"Consistent Manifold Representation for Topological Data Analysis."* arXiv preprint arXiv:1606.02353, 2016