Topological Approaches to Unsupervised Learning

Leland McInnes

Unsupervised Learning

Unsupervised learning is the machine learning task of inferring structure in "unlabeled" data.

- Dimension Reduction
- Clustering
- Anomaly Detection

Dimension reduction

Given high dimensional data $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^n$ find a low dimensional representation of the data – find the "latent" variables that can describe the data.

Our working example will be the MNIST handwritten digits dataset.

28x28 pixel images of handwritten digits, converted to 784 dimensional vectors.

Dimension reduction

Principal Components Analysis

Principal Components Analysis

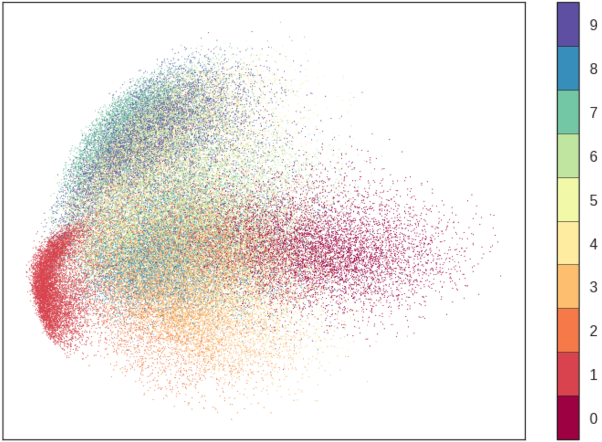
Project the data onto the d-dimensional hyperplane that minimizes the distance from points to the plane.

Principal Components Analysis

In practice this is solved by the top d eigenvectors of the covariance matrix of X.

Alternatively this is the top d singular vectors of the SVD of X.

Principal Components Analysis



Principal Components Analysis

This captures global structures of the data, but is a fundamentally linear projection and cannot capture non-linear manifold structure.

Laplacian Eigenmaps

Laplacian Eigenmaps

Assuming the data lies on a manifold, try to approximate the Laplace-Beltrami operator $\Delta = \nabla \cdot \nabla$ of the manifold.

Laplacian Eigenmaps

Select a kernel $\kappa(x, y)$, and construct a graph with vertices X and an edge (x_i, x_j) with weight $\kappa(x_i, x_j)$.

Laplacian Eigenmaps

The (symmetric) normalized Laplacian of the graph is a discrete approximation of the Laplace-Beltrami operator.

Laplacian Eigenmaps

Specifically, Belkin and Niyogi (2002) demonstrate that, under certain assumptions, in the limit as the bandwidth of the kernel tends to 0 and N tends to ∞, the normalized Laplcian converges to the Laplace-Beltrami operator.

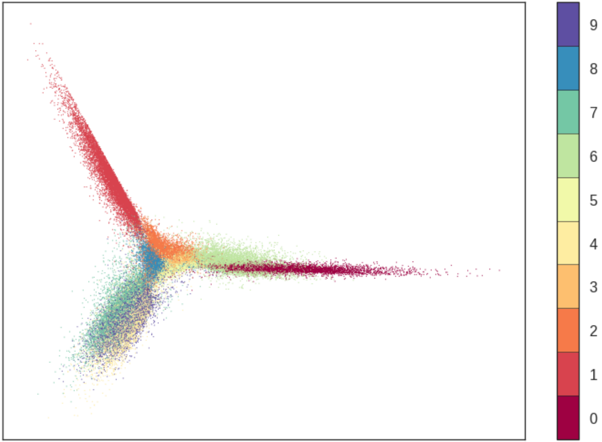
Laplacian Eigenmaps

A low dimensional embedding is obtained by considering the top eigenfunctions of the Laplace-Beltrami operator.

Laplacian Eigenmaps

This amounts to taking the top eigenvectors of the normalized Laplacian.

Laplacian Eigenmaps



Laplacian Eigenmaps

This understands manifold structure, but requires strong assumptions – specifically it requires that the data be uniformly distributed on the manifold.

Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection

Force the data to be (approximately) uniformly distributed by locally varying the Riemannian metric tensor to make it so.

Uniform Manifold Approximation and Projection

That is, we use the uniform distribution assumption to locally approximate the volume form and thence the metric tensor.

Uniform Manifold Approximation and Projection

This can be thought of as locally normalising distance relative to the local neighborhood.

Since we have finite data X we must locally approximate a different Riemannian metric for each point x_i.

Uniform Manifold Approximation and Projection

This provides us with N mutually incompatible metric spaces which we must somehow merge together.

Uniform Manifold Approximation and Projection

Since real world data has repeated points we actually only have pseudo-metric spaces.

Uniform Manifold Approximation and Projection

Since the metric local to x_i only knows about distances from x_i the distances between other points are not well defined…

…We can use extended-pseudo-metric spaces and set those distances to be ∞.

But how does one glue together different extended-pseudo-metric spaces?

Uniform Manifold Approximation and Projection

Fortunately we can modify the standard geometric realization and singular set functors from algebraic topology.

Uniform Manifold Approximation and Projection

This gives a pair of adjoint functors

$$Real : \mathbf{EPMet} \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \mathbf{sFuzz} : Sing$$

between extended-pseudo-metric spaces and fuzzy simplicial sets.

Uniform Manifold Approximation and Projection

Which means we can convert each local metric space into a fuzzy simplicial set and then take a fuzzy union to get a single fuzzy simplicial set representing the data.

Uniform Manifold Approximation and Projection

This can also be phrased in terms of colimits or pushouts.

Uniform Manifold Approximation and Projection

Now simply find a low dimensional representation $Y = \{y_1, \ldots, y_N\} \subset \mathbb{R}^d$ such that $Sing((Y, d_{\mathbb{R}^d}))$ approximates the fuzzy simplicial set for X.

Uniform Manifold Approximation and Projection

We can measure similarity of fuzzy simplicial sets using fuzzy set cross entropy.

Uniform Manifold Approximation and Projection

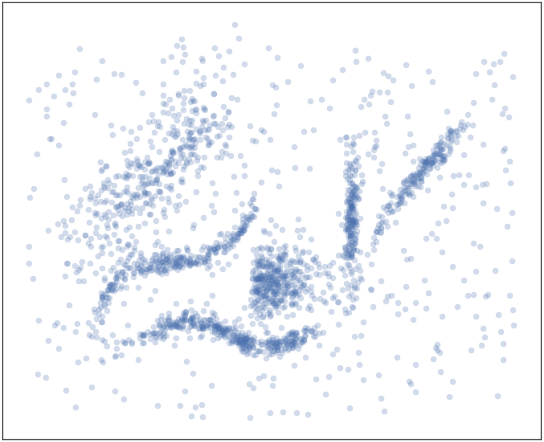
Clustering

Given a dataset $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^n$ find the groups or clumps of data that are similar.

Not necessarily a well posed problem – what constitutes a clump? What do we mean by similar?

For our example dataset we'll use some synthetic "hard to cluster" data in 2-dimensions
(so we can see what is going on).

Clustering



K-Means

K-Means

Assume we know how many clusters we want to find (call it k).

Project the data onto a k-dimensional hyperplane that minimizes the distance from points to the plane.

K-Means

One can think of this as finding k centroids, or archetypes, and we can instead minimize the distance to the closest archetype.

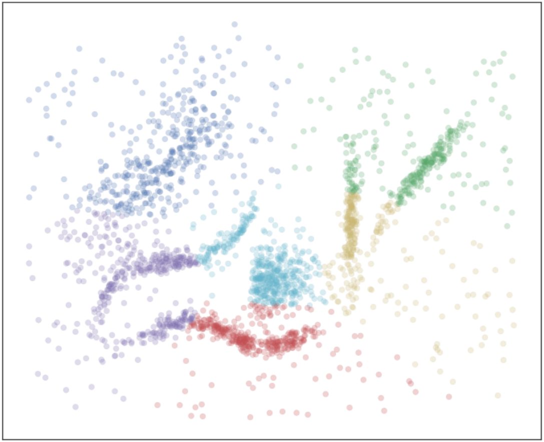
This is K-Means clustering.

K-Means

Computationally one randomly assigns k cluster centroids and then iterates:

1. Assign each data point to its closest centroid.
2. Set the new centroid locations to be the means of the data points assigned to them.
3. Repeat from step 1.

K-Means



K-Means

This captures global structures of the data, but is a fundamentally linear projection and cannot capture non-linear manifold structure.

K-Means

It also fails to deal well with noise in the data.

Spectral Clustering

Spectral Clustering

It would be good to extract some of the non-linear manifold structure of the data when clustering.

Spectral Clustering

We can do this using Laplacian Eigenmaps!

Spectral Clustering

By using the eigenvectors of the Laplacian of the appropriate weighted graph we “unfold” the manifold.

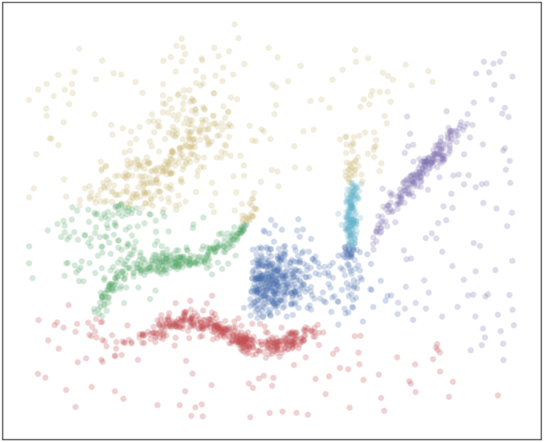
Spectral Clustering

Once we have linearised the manifold we can simply use K-Means to cluster.

Spectral Clustering

This is spectral clustering.

Spectral Clustering



Spectral Clustering

This does a better job, but still fails to deal with noise well.

co-UMAP

co-UMAP

UMAP analysed the non-linear manifold structure under a uniform distribution assumption.

We can do the opposite!

co-UMAP

Instead of normalising distances with respect to the local neighborhood we can exaggerate distances with respect to the local neighborhood.

co-UMAP

This denormalising of distances has the effect of downplaying noise.

co-UMAP

The same fuzzy simplicial set theory then goes through, but now instead of taking the fuzzy union of the local fuzzy simplicial sets we take the fuzzy intersection.

co-UMAP

Categorically this is equivalent to taking the pullback over all the local fuzzy simplicial sets with respect to the maximal fuzzy simplicial set on the given 0-simplices.

co-UMAP

The result is
a global fuzzy simplicial set
representing the data.

co-UMAP

A little bit of symbol pushing...

$$S : \Delta^{\mathrm{op}} \longrightarrow \mathbf{sFuzz}$$

$$S : \Delta^{\mathrm{op}} \longrightarrow \mathbf{Sets}^{\mathbb{I}^{\mathrm{op}}}$$

$$S : (\mathbb{I} \times \Delta)^{\mathrm{op}} \longrightarrow \mathbf{Sets}$$

$$S : \mathbb{I}^{\mathrm{op}} \longrightarrow \mathbf{Sets}^{\Delta^{\mathrm{op}}}$$

$$S : \mathbb{I}^{\mathrm{op}} \longrightarrow \mathbf{sSet}$$

co-UMAP

$$\mathbb{I}^{\mathrm{op}} \xrightarrow{\quad S \quad} \mathbf{sSet} \xrightarrow{\quad \pi_0 \quad} \mathbf{Sets}$$

co-UMAP

The composite functor $\pi_0 \circ S$ provides a fuzzy set of connected components.

co-UMAP

We can make an explicit fuzzy set (A, μ) where A is the set of all connected components at any membership strength and

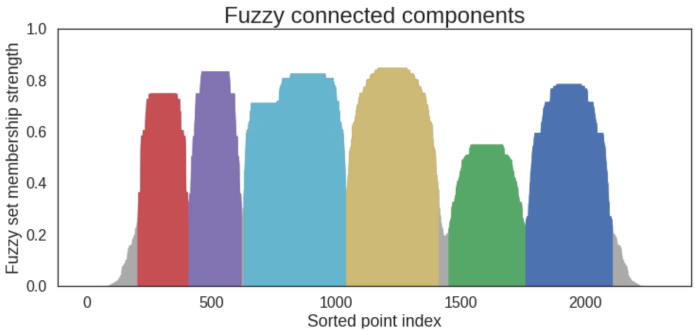
$$\mu(a) = \begin{cases} \sup\{i \in (0,1] \mid a \in \pi_0 \circ S(i)\} & \text{if } |a| \geq m \\ 0 & \text{otherwise} \end{cases}$$

co-UMAP

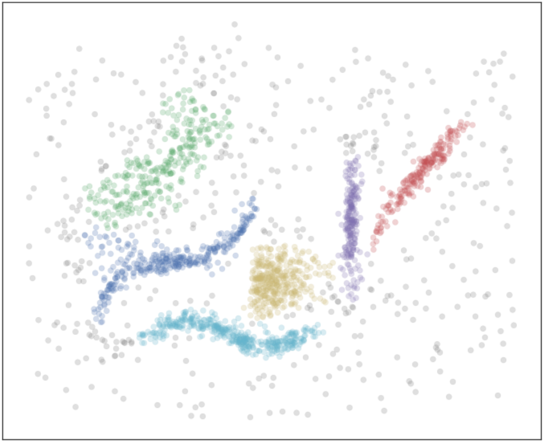
This effectively prunes out clusters that are "too small".

co-UMAP

A simple procedure can then select out clusters from this, leaving some points unclustered as "noise".



co-UMAP



Anomaly Detection

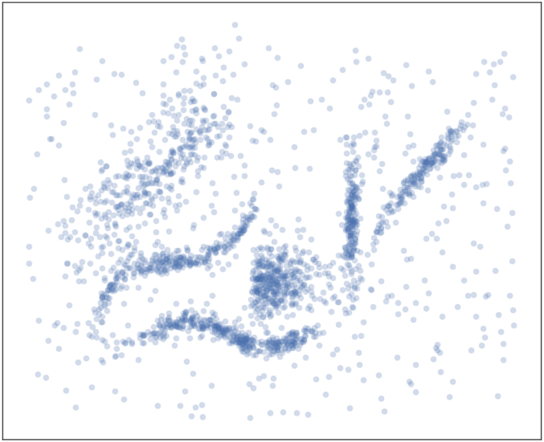
Anomaly detection is the task of "finding points that don't belong".

To determine if a point is unexpected, we first need to build a model of what we expect.

This has similarities to both dimension reduction and clustering.

We will use the same test dataset as for clustering.

Anomaly Detection



Gaussian Mixture Models

Gaussian Mixture Models

Suppose we want to model the data as a mixture of k multivariate Gaussian distributions.

Gaussian Mixture Models

We can measure the “error” of a given set of k Gaussians as the negative log likelihood of seeing the data under the distribution.

Gaussian Mixture Models

We then optimize to find parameters for k Gaussians that minimize this error.

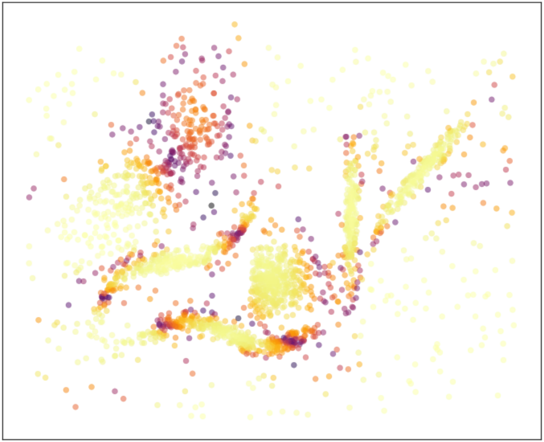
Gaussian Mixture Models

We can then express how anomalous a data point is as the negative log likelihood of observing that point.

Gaussian Mixture Models

In other words: “how unlikely is the data point under the model?”

Gaussian Mixture Models



Gaussian Mixture Models

This has a similar flavour to PCA and K-Means, and suffers from some of the same problems.

Gaussian Mixture Models

Gaussians can't follow non-linear manifold structure well.

Sufficient noise can corrupt the fit.

Local Outlier Factor

Local Outlier Factor

To better follow the manifold we need a non-parametric estimate of density.

Local Outlier Factor

The reciprocal of the distance to the k^{th} nearest neighbor provides an approximate density.

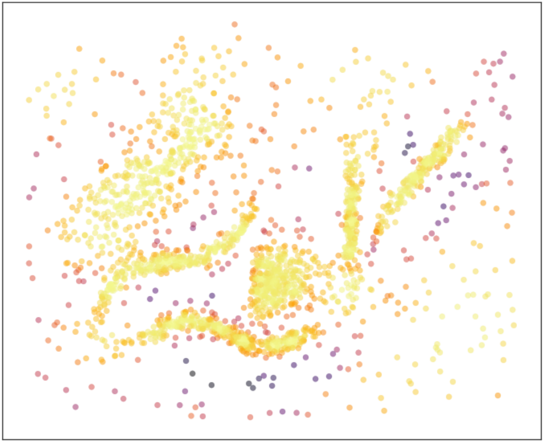
Local Outlier Factor

An anomaly is then a point that has signficantly different density than that of its nearest neighbors.

Local Outlier Factor

This provides the intuition for the Local Outlier Factor.

Local Outlier Factor



Local Outlier Factor

This is certainly better, but is heuristic, and scores are not as easily interpretable as one might like.

Dual co-UMAP

Dual co-UMAP

co-UMAP provided something similar to a non-parametric density estimate.

Dual co-UMAP

Given input X we can run co-UMAP and consider the fuzzy set of connected components (A, μ).

Dual co-UMAP

We can generate a density estimate using the fuzzy set (X, ν) where

$$\nu(x) = \sup\{\mu(a) \mid a \in A \text{ and } x \in a\}$$

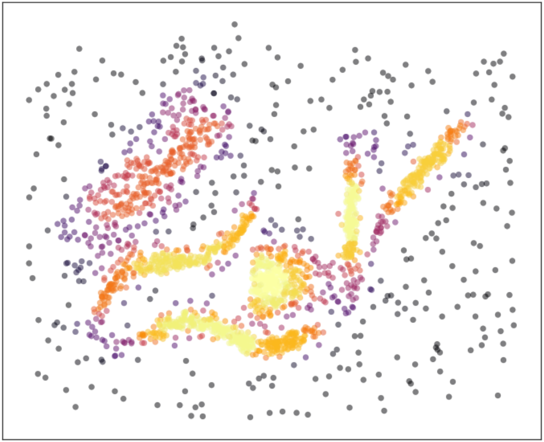
Dual co-UMAP

We can simply take the fuzzy set complement of this density estimate!

Dual co-UMAP

This is the fuzzy truth value that a point is not in any connected component.

Dual co-UMAP



Conclusion

With a little bit of topology and category theory for heavy lifting we can build a single powerful unified theory for unsupervised learning!

This is computationally tractable!
($O(N \log N)$ average case performance)

Implementations are available!

`https://github.com/lmcinnes/umap`

`https://github.com/scikit-learn-contrib/hdbscan`