Machine Learning I

Lecture 15: Cluster Analysis Part I

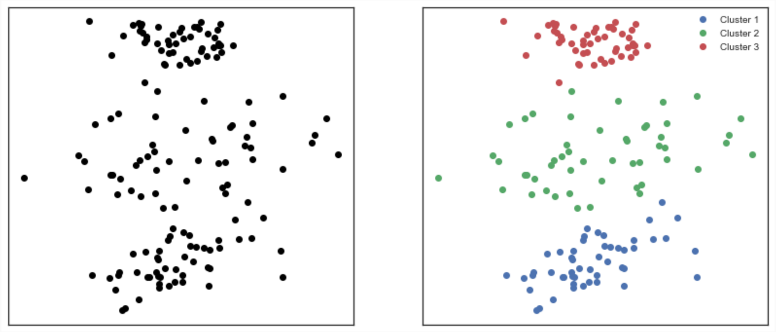
Nathaniel Bade

Northeastern University Department of Mathematics

Table of contents

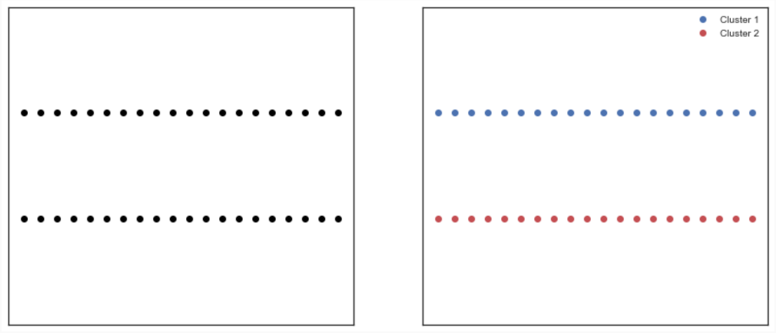
Cluster Analysis

Cluster Analysis



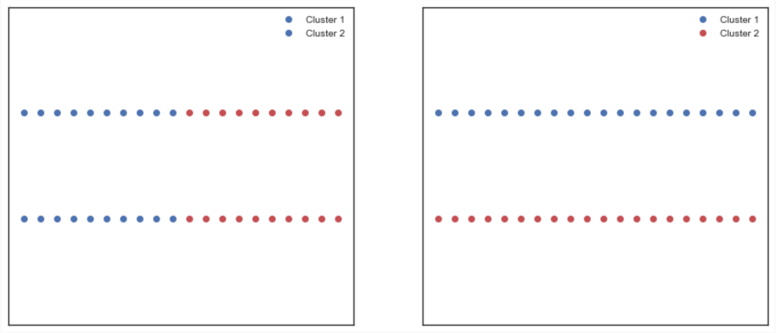
Clustering is one of the most widely used techniques in exploratory data analysis. Clustering is the task of organizing data into groups so that “similar” objects end up in the same **cluster** and “different” objects end up in different clusters.

Cluster Analysis



The basic problem for clustering is a problem for all unsupervised learning: the problem of ground truth. Since there are no labels from which to learn what possible clusters might look like, we have to make assumptions about how the data are grouped.

Cluster Analysis



The basic problem for clustering is a problem for all unsupervised learning: the problem of ground truth. Since there are no labels from which to learn what possible clusters might look like, we have to make assumptions about how the data are grouped.

A clustering algorithm that emphasizes similarly labeling close by points will not be as aware of the total distance in feature space, while a method that emphasizes not having far away points will ignore local information.

Clustering Paradigms

There are five basic kinds of clustering paradigms:

Centroid based clustering is governed by overall distance to closest centroid.

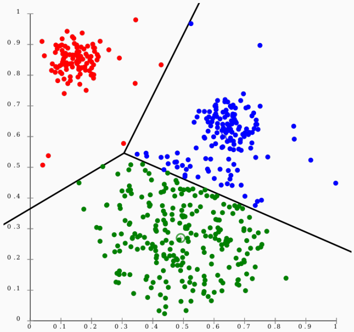
Hierarchical clustering build nested clusters by merging or splitting, forming a cluster **dendrogram**.

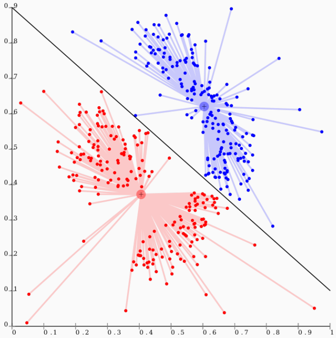
Density based methods are determined by contiguous regions with density above some threshold.

Spectral clustering decomposes the similarity graph $d(x_i, x_j)$ as in factor analysis and PCA.

Distribution models/mixing models assume clusters provide a underlying distribution on the domain.

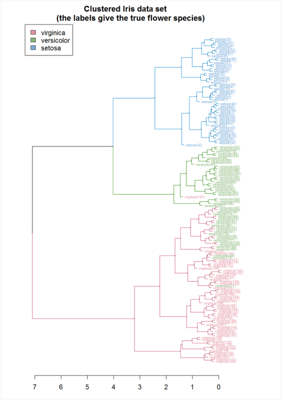
Centroid Based Clustering

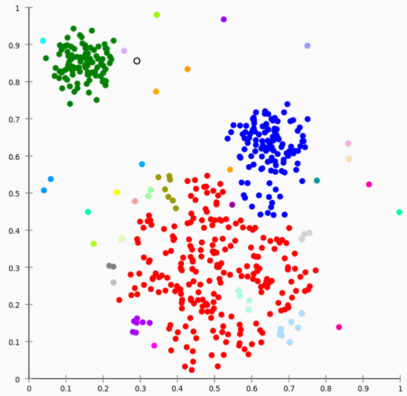




In centroid based clustering, clusters are determined by distance from a centroid μ_i, under a metric $d(x, y)$ of similarity matrix D. Examples include k-`means`, k-`mediods`, and the `meanshift` algorithm.

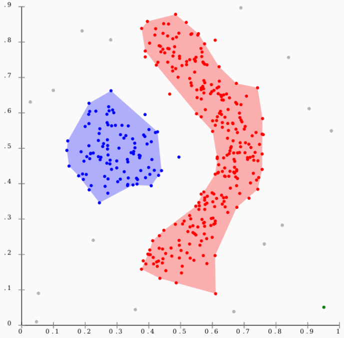
Hierarchical Clustering

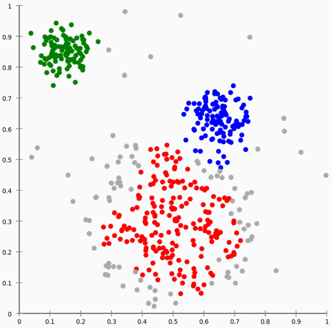




In hierarchical (connectivity, linkage) clustering, nested clusters are formed using greedy algorithms but either merging or splitting sets. Algorithms include `Ward` and `AgglomerativeClustering`.

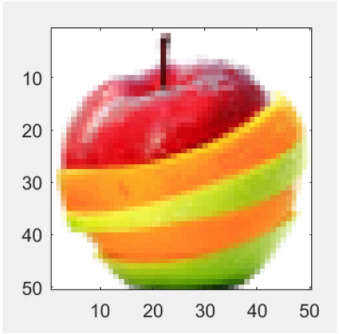
Density Based Clustering

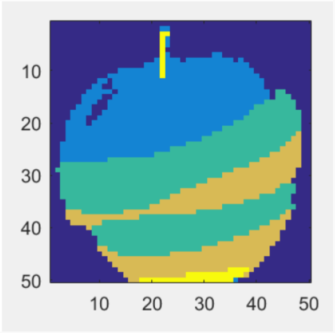




Density based clustering algorithms determine regions by applying a local density threshold (with a dynamically determined or fixed density) and then extracting the connected regions. Algorithms like DBSCAN on the right assume clusters are of similar density and have trouble separating nearby clusters. Algorithms like OPTICS improve upon this but are not currently implemented.

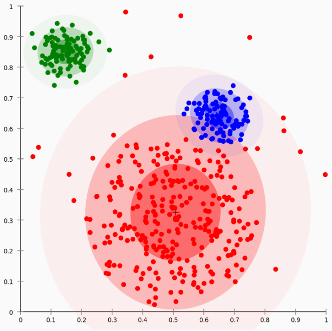
Spectral Clustering

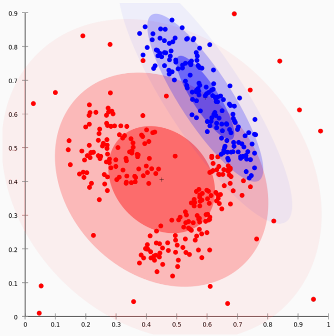




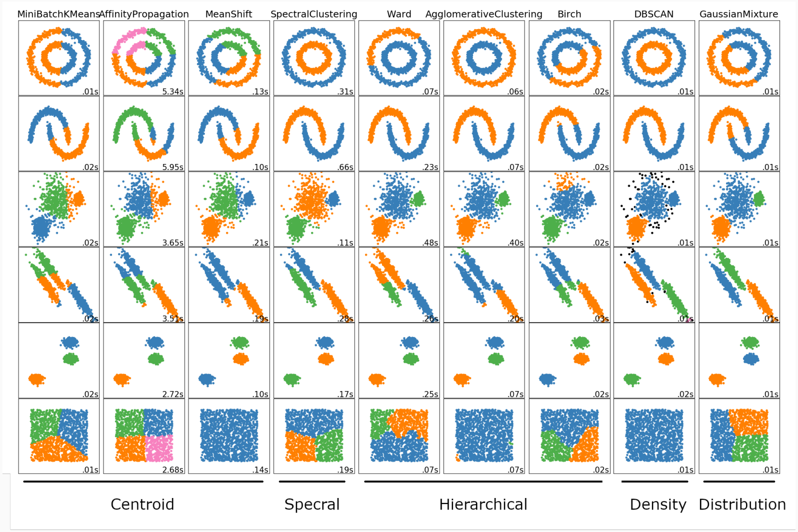
Spectral clustering forms a similarity matrix S_{ij} where each entry encode the similarity between datapoints i and j. We then project onto the first k eigenvectors of S_{ij} and perform the clustering there. Equivalently, we can construct a similarity graph and thing of pruning away low relevance connections until the graph disconnects. Sci-kit learn implements this as SpectralClustering.

Distribution Based Clustering





Distribution based, or mixing models, try to fit an underlying distribution to the data, often using maximum likelihood estimation. This is implemented in sci-kit learn as GaussianMixture.



Sci-kit learn's clustering algorithms compared: `https://scikit-learn.org/stable/modules/clustering.html`

Formalism

Proximity Matrices

Although the intuition behind clustering is geometric sometimes we want to talk about proximity between pairs of objects directly. For example

- Categorical data: we may want to say a horse is closer to a pig than an ant. We could assign a number based on whether the change was species, phylum, kingdom, etc.
- In numerical data where we do not expect dimensions to respect the triangle inequality, for example unnormalized body temperature and yearly salary. Or, if the distance explicitly uses another metric, like driving time between points on a grid.

This data can be represent as an $N \times N$ (dis)-similarity matrix $\mathbf{D}$. Most algorithms assume $\mathbf{D}$ is symmetric, non-negative with zero diagonal. If $\mathbf{D}$ is not symmetric, $(\mathbf{D} + \mathbf{D}^T)/2$ is often used.

Dissimilarity

Most often we have measurements x_{ij} for samples $i = 1, \ldots, N$ on variables $j = 1, \ldots, p$ and a dissimilarity matrix $\mathbf{D}$ described by a dissimilarity $d_j(x_{ij}, x_{i'j})$ on the j'th feature via

$$\mathbf{D}_{i,i'} = \mathbf{D}(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})\,.$$

The most common dissimilarity for quantitative variables is $d_j(x_{ij}, x_{i'j}) = \ell(|x_{ij} - x_{i'j}|)$, where ℓ is a monotonically increasing function. Alternatively, the correlation $\rho(x_i, x_{i'})$ can also be used.

Dissimilarity

For categorical variables it is common to use the number of changes that need to be made to transform one datum into another. For example, ‘CAT’ must change three letters to become ‘TED‘, but only one letters to become ‘RAT‘. Here, each $d_j = 1$ iff $x_{ij} = x_{i'j}$.

On the other hand, for ordinal variables is common to use the ordinal difference between variables. That is

$$d_j(x_i, x_{i'}) = \#\text{of elements between } x_i \text{ and } x_j\,.$$

The species, phylum, kingdom example from before is an example of this.

Dissimilarity Matrices

Different feature dissimilarities can be combined via a weighted sum into a dissimilarity matrix

$$\mathbf{D}_{i,i'} = \sum_{j=1}^{p} w_j d_j(x_{ij}, x_{i'j})\,, \qquad \sum_{j=1}^{p} w_j = 1\,.$$

The mean dissimilarity in each variable is then

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j})\,,$$

and the total mean dissimilarity is

$$\bar{D} = \frac{1}{N^2} \sum_{i,i'=1}^{N} \sum_{j=1}^{p} D(x_{ij}, x_{i'j}) = \sum_{j=1}^{p} \bar{d}_j\,.$$

Thus, setting $w_j = 1/\bar{d}_j$ gives all features equal influence. Many algorithms will take a dissimilarity matrix instead of a metric.

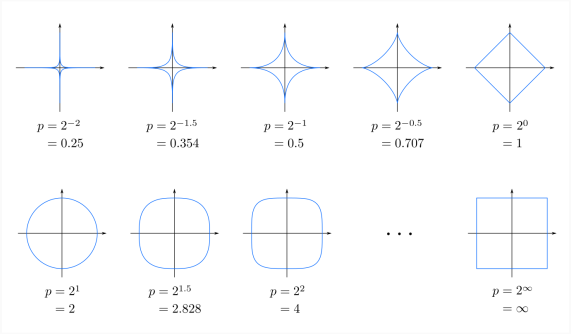
Dissimilarity Matrices

Notice that for $d_j = (x_{ij} - x_{i'j})^2$, this prescription for the weights is

$$\bar{d}_j = \frac{1}{N^2} \sum_{i,i'=1}^{N} (x_{ij} - x_{i'j})^2 = 2\mathrm{var}_j \, ,$$

where var_j is the sample variance. Therefore, the relative importance in of the features in the dissimilarity matrix is given by the variance.

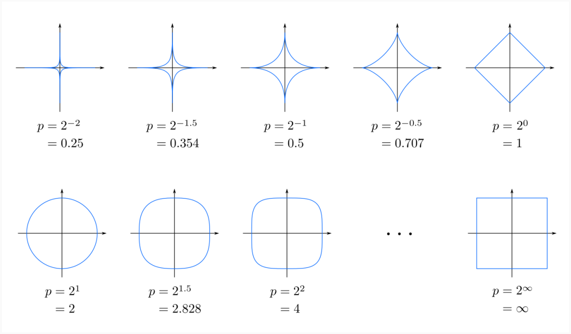
Dissimilarity From Distance



The dissimilarity matrix $D_{i,i'}$ can also be defined by a metric d on a metric space (or a combination of both):

$$D_{i,i'} = d(x_i, x_{i'}) \,.$$

Dissimilarity From Distance

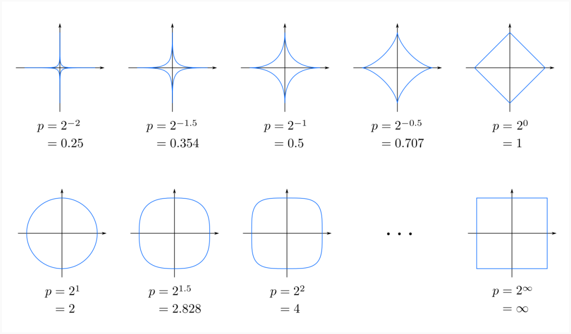


The most common metrics are the Minkowski L_q norms

$$d(x, y) = \left(\sum_{i=1}^{p} (|x_i - y_i|)^q \right)^{\frac{1}{q}} .$$

Here, L_2 is the standard Euclidean norm and L_1 is the absolute value norm.

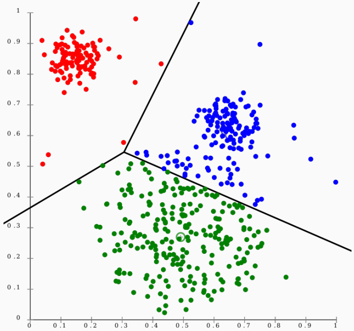
Dissimilarity From Distance

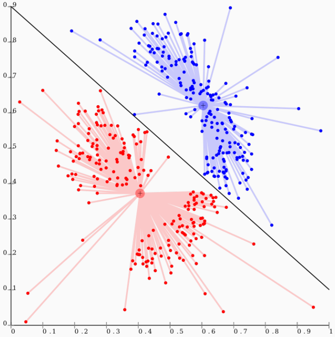


The dissimilarity matrix can replace the Euclidean distance in most algorithms and many implementations. It is the first piece of data processing that must be performed when discussing clustering, and can drastically change the outcome of your algorithm.

K-Means Clusters

K-Means Clusters

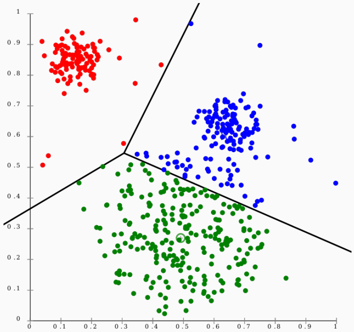


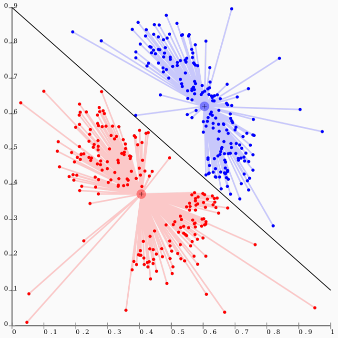


The K-means clustering algorithm constructs K clusters C_j by their proximity to a centroid μ_j. The algorithm attempts to minimize the distance to the centroid μ_j:

$$L(\mu) = \sum_{j=1}^{K} \sum_{x_i \in C_=j} ||x_i - \mu_j||^2 \,.$$

K-Means Clusters





For d given by the Euclidean distance, one can see that this is roughly equivalent to minimizing the **within group scatter**:

$$W(C) = \sum_{j=1}^{K} \sum_{x_i, x_{i'} \in C_= j} ||x_i - x_{i'}||^2 = \sum_{j=1}^{K} N_j \sum_{x_i \in C_j} ||x_i - \mu_j||^2 \,.$$

K-Means Algorithm

A brute force search is NP hard. However, there exists an iterative procedure to find a local minimum. Since

$$\text{argmin}_\mu \sum_{i \in S} ||x_i - \mu||^2 = \bar{x}_S \,,$$

where $\bar{x}_S$ is the mean of the x's in S, we can solve the enlarged problem by the following algorithm:

Initialize μ_j to random values. Then

For each j, set $C_j^{(t)} = \{x \in \mathcal{X} : \, i = \text{argmin}_j ||x - \mu^{(t-1)}||^2\}$.

For each j, update $\mu_j^{(t)} = \bar{x}_{C_j^{(t)}}$.

Repeat until convergence.

K-Means Algorithm

Lemma: The K-Means algorithm is non-increasing.

Proof: At step t, let $C_j^{(t-1)}$ be the previous clusters and $\mu_j^{(t-1)}$ the previous centroids. The total cluster variance

$$W(C^{(t-1)}, \mu^{(t-1)}) = \sum_{j=1}^{K} N_j \sum_{x_i \in C_j^{(t-1)}} ||x_i - \mu_j||^2$$

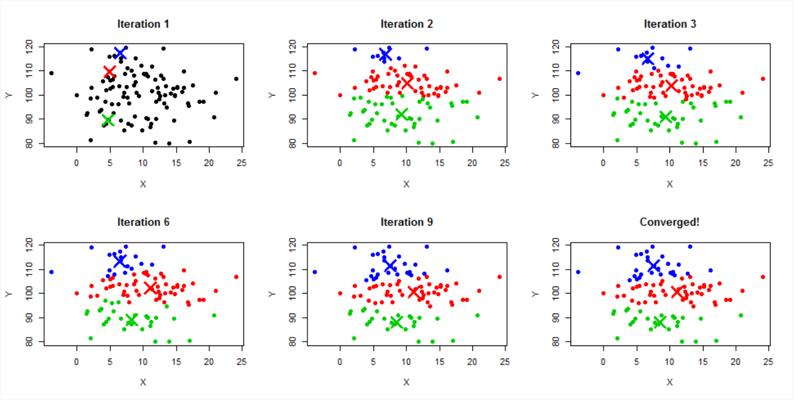
is minimized by setting the means at time step t to $\mu_j^{(t)} = \bar{x}_{C_j^{(t-1)}}$.
Therefore $W(C^{(t-1)}, \mu^{(t-1)}) \geq W(C^{(t-1)}, \mu^{(t)})$.

But, given this current set of means $\mu_j^{(t)}$, $W(C^{(t)})$ will be minimized by setting

$$C_j^{(t)} = \{x \in \mathcal{X} : \ i = \text{argmin}_j ||x - \mu||^2\}.$$

Therefore $W(C^{(t-1)}, \mu^{(t)}) \geq W(C^{(t)}, \mu^{(t)})$.

Example:



The efficiency of the algorithm come from it's greedy nature. Note in the example above that we have probably not found the global minimum, but that the clustering has found reasonable divisions given three labels.

Practical Issues: Initialization

Initialization is probably the greatest difficulty in most greedy algorithms. For example, assume we initialize by setting μ_j to be randomly selected data points x_j.

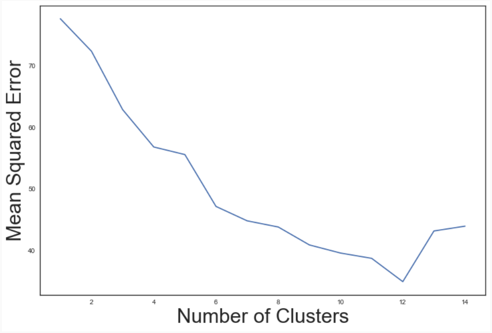
If the data in fact has K clusters, each of size N/K, then the probability of randomly selecting a point from each cluster is

$$\frac{K!(N/K)^K}{N^K} = \frac{K!}{K^K} \leq \left(\frac{1}{2}\right)^{\frac{K}{2}} .$$

So we will almost always have multiple initial centroids for K large.

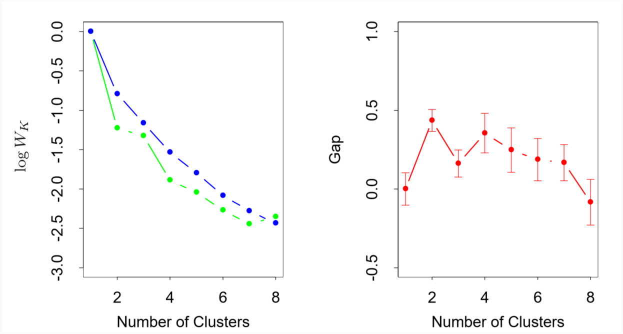
To get better results: Always take multiple randomized starts and consider ensambling them. Initialize with bottom up hierarchical methods. Alternatively, select more than k points and only keep the most widely separated.

Practical Issues: Choosing K



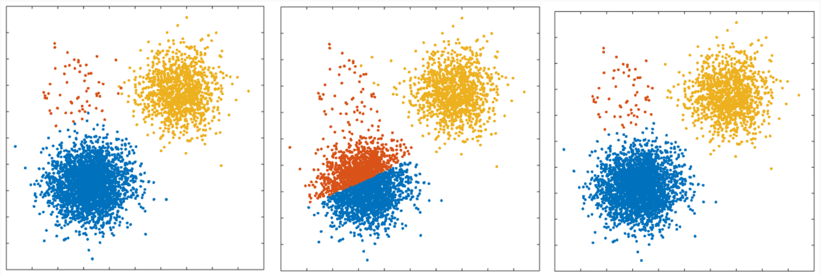
The choice of K is the other main practical issue in K-means clustering. A common criteria is to compute the clustering for many K, plot them, and set K to be just larger than the steepest "elbow," or "kink." In the example above, we might choose $K = 4$ or $K = 7$.

Practical Issues: Choosing K



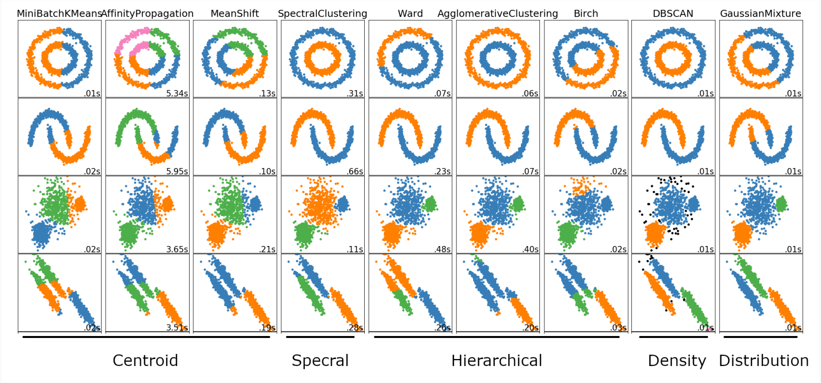
There are various purposed automatic methods for this, including the **gap statistic** which attempts to makes the idea of a “elbow” precise by comparing the within cluster dissimilarity W_i to the expected dissimilarity for a uniform distribution over a rectangle containing the data.

Practical Issues: Size and Density


(a) Generated synthetic data (b) K-means (c) MAP-DP

K-means can be thrown off by varying size and density. The squared term in the loss penalizes relatively large clusters, while the sum over all points weights denser patches more heavily.

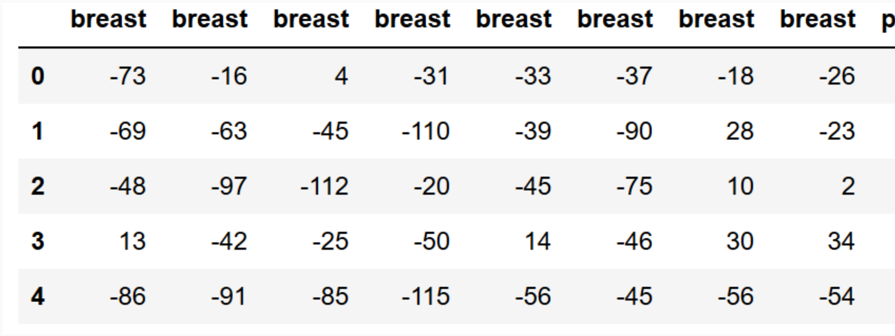
Practical Issues: Size and Density



Finally, K-means is a centroid based method and so is optimized on position in space, not position relative to other points. This means that contiguous clusters with a less symmetric geometry will not be found. By optimizing over balls, K-means is forming linear decision boundaries, and cannot find other kinds of boundaries without being embedded in a higher dimensional space.

Example: Gene Expression Clustering

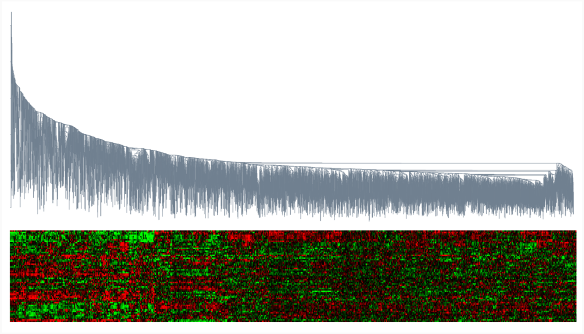
Example: Gene Expression Clustering



	breast	breast	breast	breast	breast	breast	breast	breast	p
0	-73	-16	4	-31	-33	-37	-18	-26	
1	-69	-63	-45	-110	-39	-90	28	-23	
2	-48	-97	-112	-20	-45	-75	10	2	
3	13	-42	-25	-50	14	-46	30	34	
4	-86	-91	-85	-115	-56	-45	-56	-54	

The dataset Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures lists 16063 genes for 144 patients with one of 14 types of cancer: breast, prostate, lung, collerectal, lymphoma, bladder, melanoma, uterus, leukemia, renal, pancreas, ovary, meso, and cns.

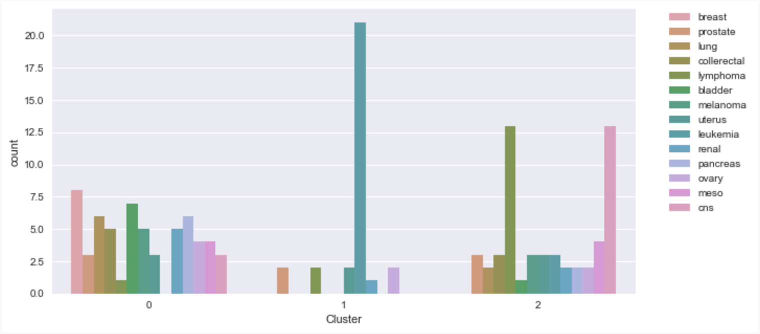
Example: Gene Expression Clustering



In the above we see a similar DNA microarray colored by expression levels. On the left we see an attempt at hierarchical clustering that we will expand on next week.

The goal is to see if certain genes tend to express certain types of cancers. We could of course treat this as a supervised learning problem, but it would be a much stronger result if the clusters can be learned unsupervised.

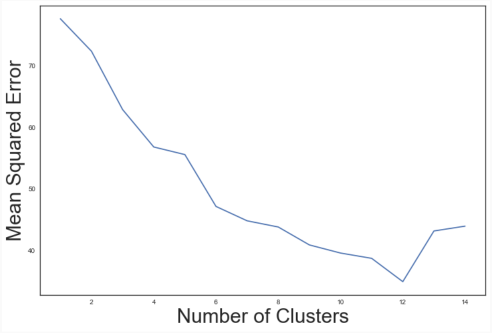
Example: $K = 3$



As a first pass we use K-Means clustering with $K = 3$. Notice that Cluster 1, accounts for almost all of the cases of leukemia, while lymphoma and cns are localized in cluster 2.

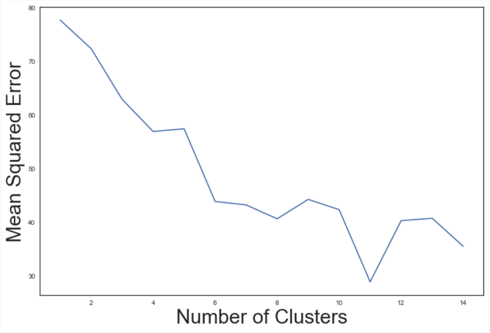
We are certainly seeing some differentiation between the cancers based solely on gene clustering, but so far we've just picked an arbitrary K, not even corresponding to our 14 labels.

Example: K vs MSE



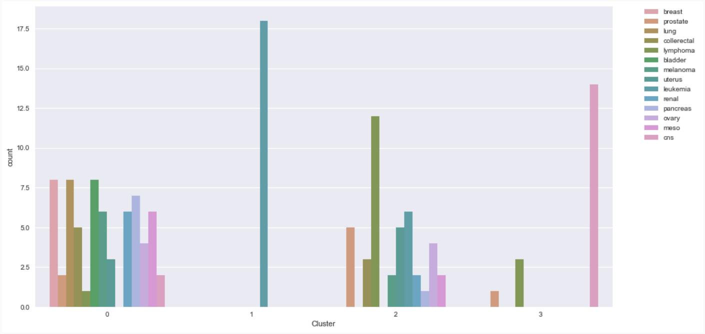
In the above, we plot the mean squared error vs the number of clusters K. The plot has a couple of elbows, the most significant being the ones at 4 and in the 6, 7 range. There is a bump at 12, but this may be due some other error. It isn't stable under multiple runs and is probably the result of over fitting.

Example: K vs MSE



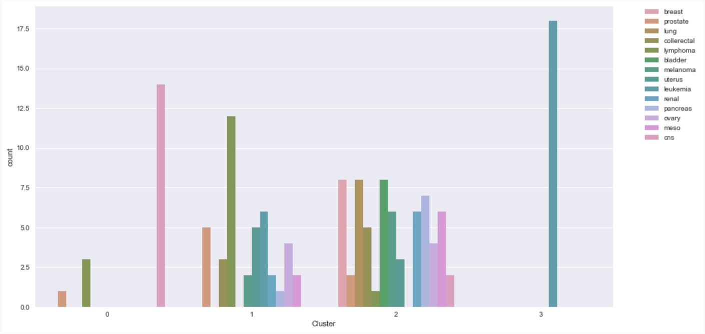
We plot the mean squared error vs the number of clusters K. The plot has a couple of elbows, the most significant being the ones at 4 and in the 6, 7 range. There is a bump at 12, but this may be due some other error. It isn't stable under multiple runs and is probably the result of over fitting.

Example: K vs MSE



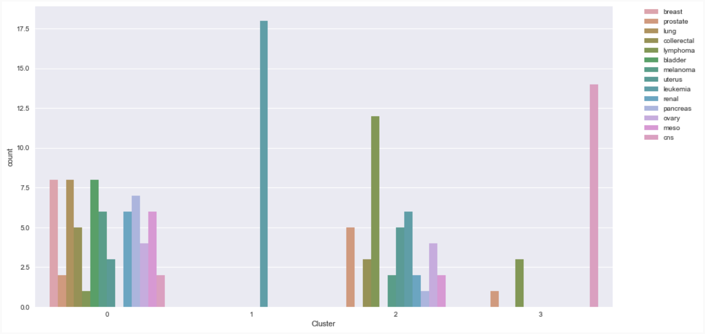
The histogram cluster labels for 4 clusters is very promising. We see at least leukemia and cns isolated with lymphoma expressing strongly in cluster 2.

Example: K vs MSE



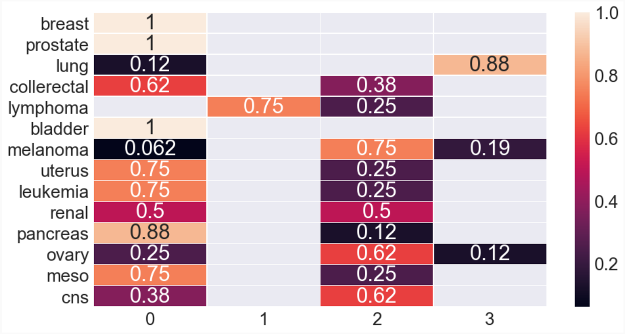
The histogram cluster labels for 4 clusters is very promising. We see at least leukemia and cns isolated with lymphoma expressing strongly in cluster 2. Whats more this division is fairly stable across runs.

Example: K vs MSE



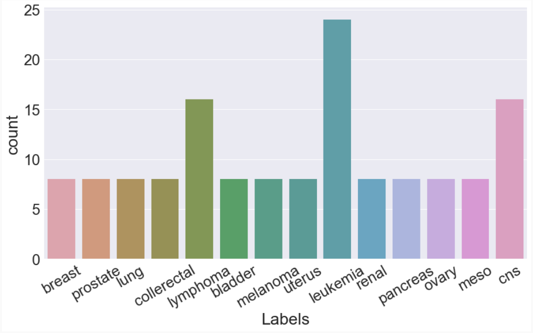
Concretely, lets look at the heatmap corresponding to our original example.

Example: K vs MSE



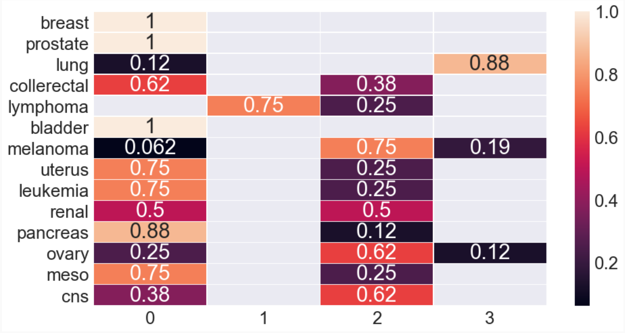
Concretely, lets look at the heatmap corresponding to our original example. We see that in fact we missed something: certain cancers were over represented in the data and in fact cluster one contains 100% of the breast, prostate, and bladder cancer.

Example: K vs MSE



Indeed, we see that lymphoma, leukemia and cns are over represented in this dataset, which almost certainly accounts for the clusters containing these types of cancer specifically. But on the other hand it means they are differential.

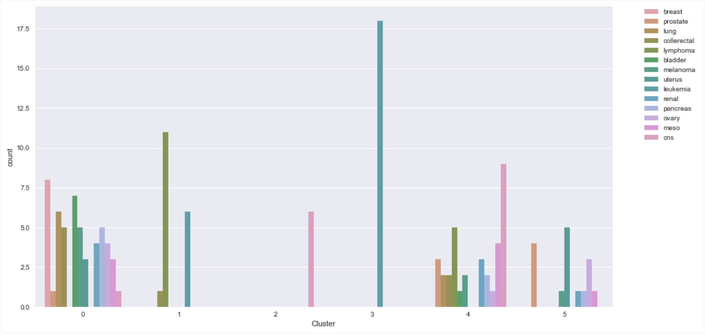
Example: K vs MSE



	0	1	2	3
breast	1			
prostate	1			
lung	0.12			0.88
collerectal	0.62		0.38	
lymphoma		0.75	0.25	
bladder	1			
melanoma	0.062		0.75	0.19
uterus	0.75		0.25	
leukemia	0.75		0.25	
renal	0.5		0.5	
pancreas	0.88		0.12	
ovary	0.25		0.62	0.12
meso	0.75		0.25	
cns	0.38		0.62	

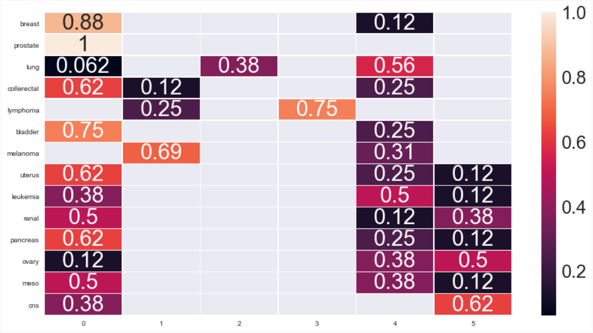
In fact, we see seem to be seeing here is a lot of cancers expressing with a similar gene cluster profile while two cancers seem to be very different, and possibly another cluster containing melanoma, ovary and cns.

Example: K vs MSE



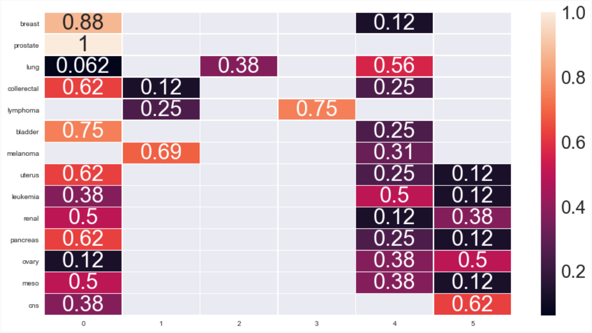
For six clusters, we see even more specialization.

Example: K vs MSE



For six clusters, we see even more specialization. The heatmap in particular shows that breast and prostate cancer are still expressing similarly, that lymphoma, cns and melanoma have different expressions than other cancers.

Example: K vs MSE



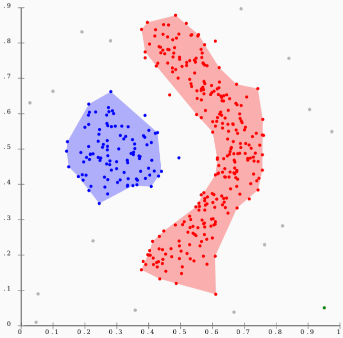
	0	1	2	3	4	5
breast	0.88				0.12	
prostate	1					
lung	0.062		0.38		0.56	
collerectal	0.62	0.12			0.25	
lymphoma		0.25		0.75		
bladder	0.75				0.25	
melanoma		0.69			0.31	
uterus	0.62				0.25	0.12
leukemia	0.38				0.5	0.12
renal	0.5				0.12	0.38
pancreas	0.62				0.25	0.12
ovary	0.12				0.38	0.5
meso	0.5				0.38	0.12
cns	0.38					0.62

I would stress that all of these "results" should be put in their proper context: they are indications that there is a connection, not evidence of one. The fact that just by clustering the features we can separate some different types of cancer is exciting, but it's the beginning of a lot of work not the end.

In fact, the data analysis isn't even done. Recall that this clustering is given by a linear decision boundary. We could now try to fit linear functions, or we could attempt to cluster using less linear methods.

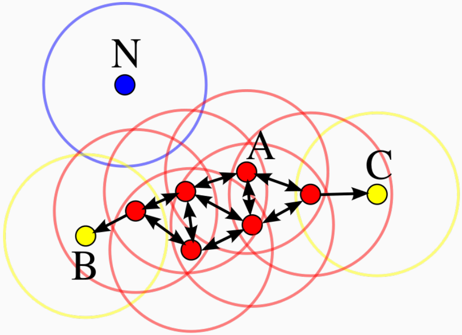
Density Based Clustering

Density Based Clustering



In density based clustering, we try to construct continuous neighborhoods of points based on being above some threshold density. For a fixed radius r, a point x_i is considered part of a cluster if the number of points in $B_r(x_i) = \{x_j : ||x_i - x_j||\}$ is sufficiently large. Fix n_r to be the number of points required to be in a cluster.

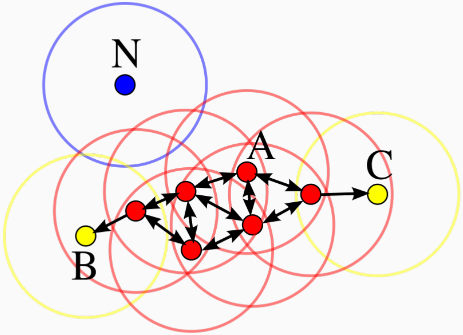
DBSCAN Definitions



The DBSCAN algorithm classifies points a **core points** like A, **boundary points** like B and C, and **noise points** like N.

A core point is one for which $B_r(p) \geq n_r$. A point q is **directly reachable** from p if $q \in B_r(p)$. If a point q is directly reachable from a core point then it is a boundary point. If a point is not directly reachable by any boundary point it is a noise point.

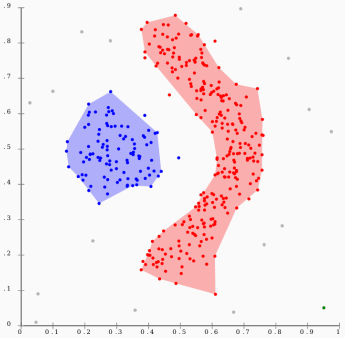
DBSCAN Definitions



A point p_k is **reachable** from $q = p_0$ if there exists a series of points $p_0, p_1, \ldots, p_k$ such that p_i is directly reachable by p_{i+1}. Finally, points p and q are **connected** if there is a point o such that p and q are reachable by o.

A **cluster** is a maximal set of connected points.

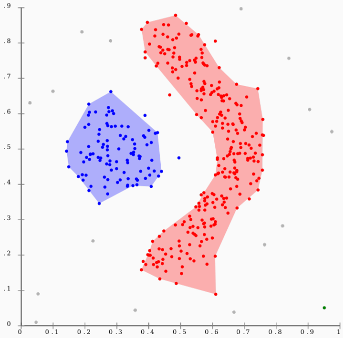
DBSCAN Definitions



A point p_k is **reachable** from $q = p_0$ if there exists a series of points $p_0, p_1, \ldots, p_k$ such that p_i is directly reachable by p_{i+1}. Finally, points p and q are **connected** if there is a point o such that p and q are reachable by o.

A **cluster** is a maximal set of connected points.

DBSCAN Algorithm



The DBSCAN algorithm is straightforward:

Pick a random point p and find all connected points C and remove them from $\mathcal{T}$.

If p is a core point, C is a cluster.

If p is not a core point do nothing.

These steps are repeated until the dataset $\mathcal{T}$ is exhausted. The leftover points are noise.

Practical Issues: Advantages

DBSCAN has quite a few advantages:

- It does not require the number of clusters K to be chosen beforehand.
- It can find non-circular data, and therefore nonlinear decision boundaries.
- It includes a criteria for outliers.
- It only requires setting two meaningful parameters that could be estimateable.

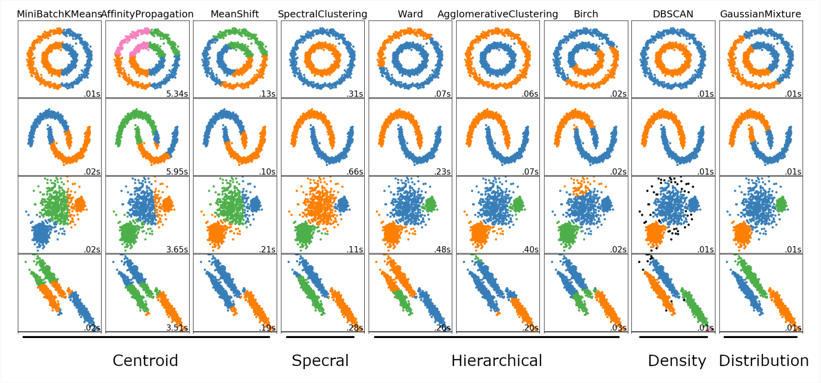
Practical Issues: Advantages

It also has a few problems:

- The robustness of the search depends heavily on the metric. In particular, if it is Euclidean it falls victim to the curse of dimensioanlity and may be almost useless.
- If there are large distances in cluster densities, the parameters in DBSCAN may only be tunable to a subset.
- If the data are poorly understood, picking r and n_d may be hard.

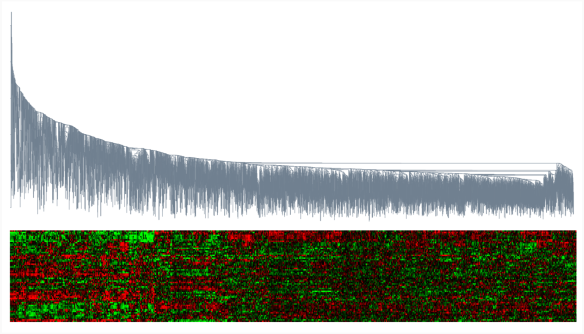
The problem of varying densities is addressed by the OPTICS algorithm, while ϵ fixing can be addressed by finding elbows in the k-distance graph.

Practical Issues: Size and Density



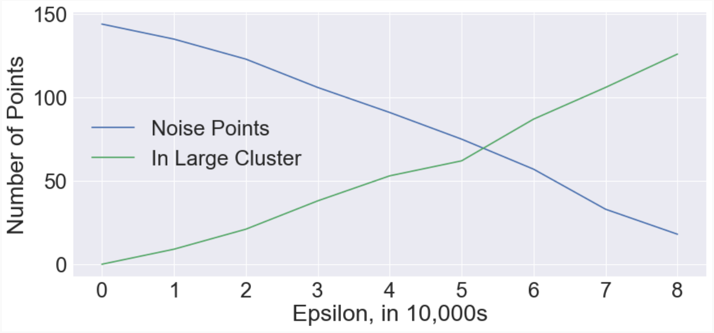
Example Revisited

DBSCAN



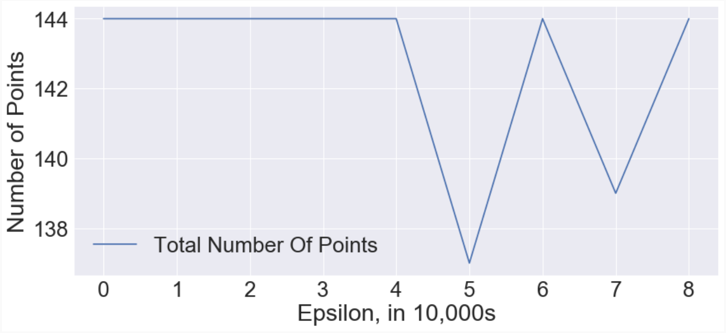
What happens if we try DBSCAN on the gene expression dataset? Gene expression has many features and so id ϵ is small every point is an outlier but if ϵ is large enough to not be an outlier it just contains all non-outlier points.

DBSCAN



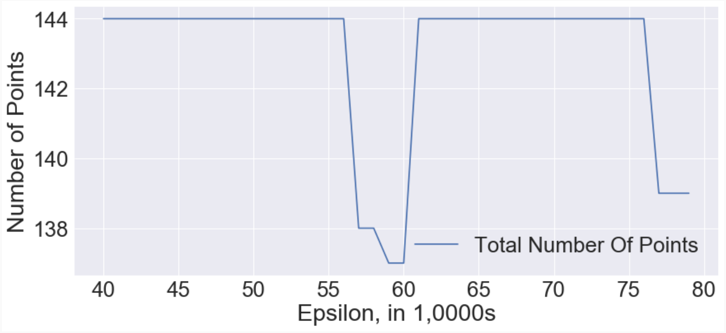
In the plot above, we plot the value of ϵ vs the number of points classified as noise and the number of points classified as in a single large cluster.

DBSCAN



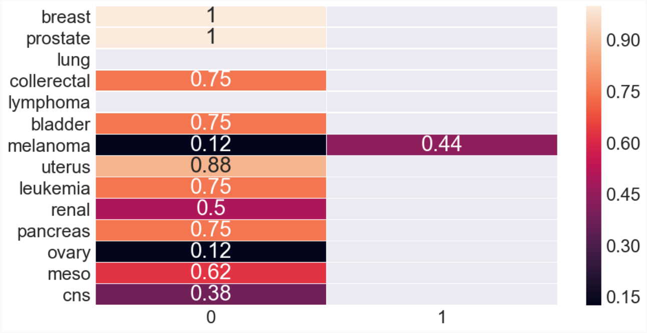
Looking at the total, we do see that there is a region where multiple clusters form.

DBSCAN



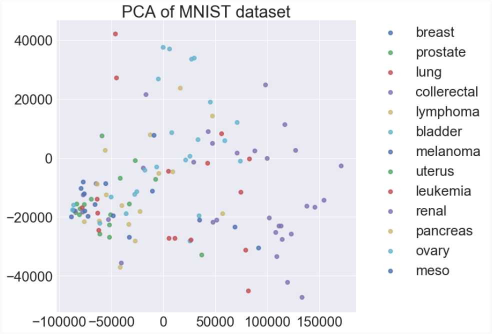
Looking at the total, we do see that there is a region where multiple clusters form. Zooming in we find a minimum roughly around $\epsilon = 60,000$.

DBSCAN



Looking at the total, we do see that there is a region where multiple clusters form. Zooming in we find a minimum roughly around $\epsilon = 60,000$. Forming the heat map a division between melanoma and other types of cancer, with lung and lymphoma curiously absent. While this does seem to agree with the results for K-means, it does add a bit to the analysis by implicating that the clusters for lymphoma and cns are probably distinct, but diffuse relative to the cluster for breast and prostate.

DBSCAN



Performing the PCA shows one reason this is so difficult, there doesn't to be a clear division of clusters. In fact, K-means may be doing better than we should expect.

It's important to note that it can be helpful to run algorithms that perform badly if it's well established when they perform badly. In this case, we are lead to believe that the data may have varying density clusters. This is a testable hypothesis.

References

For sci-kit learns comparison of clustering algorithms:

https://scikit-learn.org/stable/modules/clustering.html

http://www.learnbymarketing.com/methods/k-means-clustering/

When K-Means Clustering Fails. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0162259

Images courtesy of Wikimedia, and

https://towardsdatascience.com/spectral-clustering-for-beginners-d08b7d25b4d8