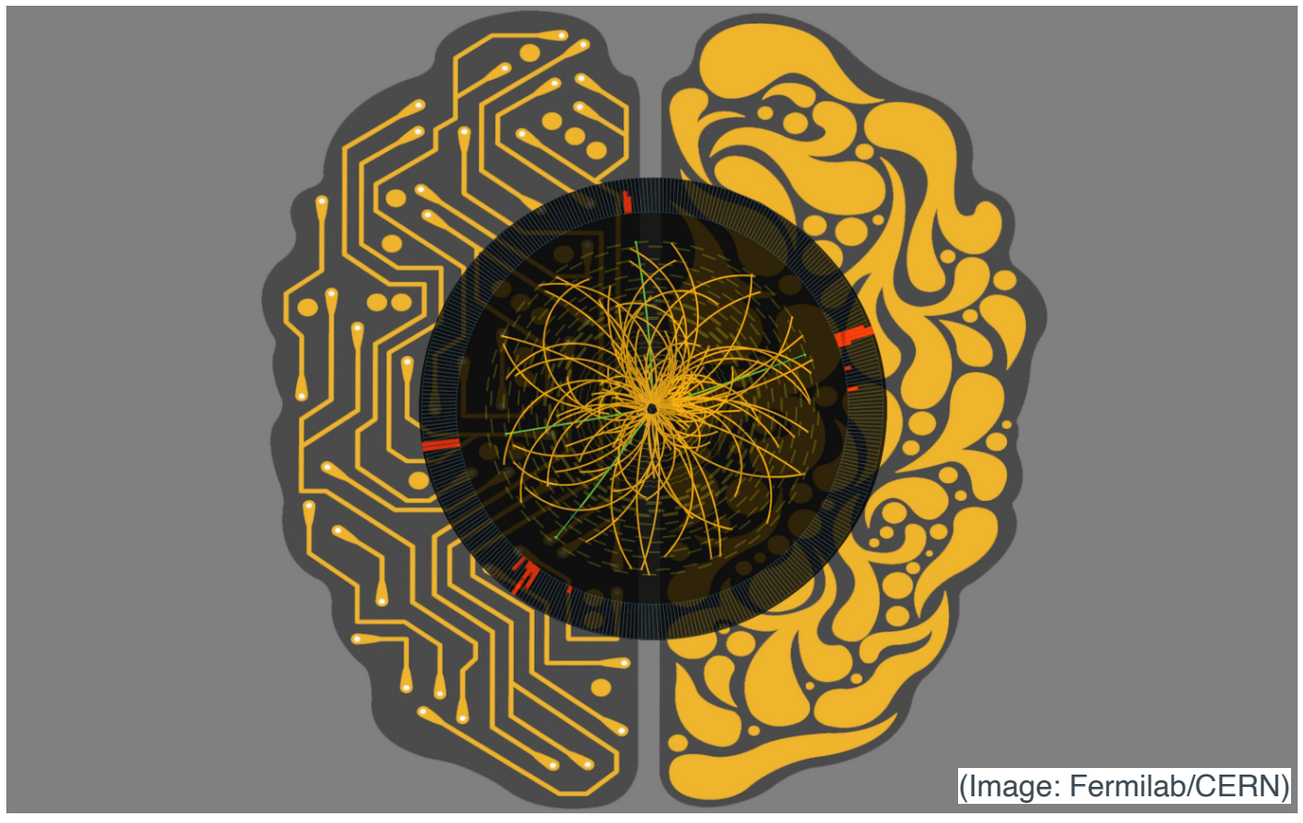
PHY 835: Collider Physics Phenomenology

Machine Learning in Fundamental Physics

Gary Shiu, UW-Madison



Lecture 14: Unsupervised Learning

Recap of Lecture 13

- Unsupervised learning
- Challenges of High-dimensional data
- Principal component analysis (PCA)
- Multi-dimensional scaling (MDS)
- t-stochastic neighbor embedding (t-SNE)

Outline for today

- K-means clustering
- Agglomerative clustering
- Density-based (DB) clustering
- Gaussian mixture models

References: 1803.08823, Deep Learning Book

Clustering

- Think of it as a simple way to look for hidden structure in high dimensions (coarse features or high-level structures in unlabelled data).

- Points to take into account:

    - Distribution of clusters (overlapping/noisy clusters vs. well-separated clusters)

    - Geometry of the data (flat vs. non-flat)

    - Cluster size distribution (multiple vs. Uniform sizes)

    - Dimensionality of the data (low-dimensional vs. high-dimensional)

    - Computational efficiency of desired method

K-means Clustering

- Divide training set into k different clusters of data points which are *near* each-other.
- Consider a set of N unlabeled data points $\{\mathbf{x}_n\}_{n=1}^{N}$ where $\mathbf{x}_n \in \mathbb{R}^p$.
- K cluster centers called the cluster means: $\{\mu_k\}_{k=1}^{K}$ with $\mu_k \in \mathbb{R}^p$.
- Minimize the cost: $\mathcal{C}(\{x, \boldsymbol{\mu}\}) = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^2,$
- One-hot encoding: $r_{nk} = 1$ if $\mathbf{x}_n \in$ cluster k and 0 otherwise;

$$\sum_k r_{nk} = 1 \ \forall \ n \text{ and } \sum_n r_{nk} \equiv N_k,$$

- Find the best cluster means (center of mass) such that variance (moment of inertia) is minimized.

K-means Algorithm

- **Expectation:** Given $\{r_{nk}\}$, minimize C with respect to μ_k:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_n r_{nk} \mathbf{x}_n$$

- **Maximization:** Given $\{\mu_k\}$, find $\{r_{nk}\}$ which minimizes C:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_{k'} (\mathbf{x}_n - \boldsymbol{\mu}_{k'})^2 \\ 0 & \text{otherwise} \end{cases}$$

- Alternative between the above two steps until some convergence criterion is met (e.g., change in C is smaller than a threshold).

- Guaranteed to converge to local minimum (different initial random cluster center initializations and post-select). Complexity $\mathcal{O}(kN)$.

- Hard-assignment limit of the Gaussian mixture model, where all cluster variances are assumed to be the same.

K-means Algorithm

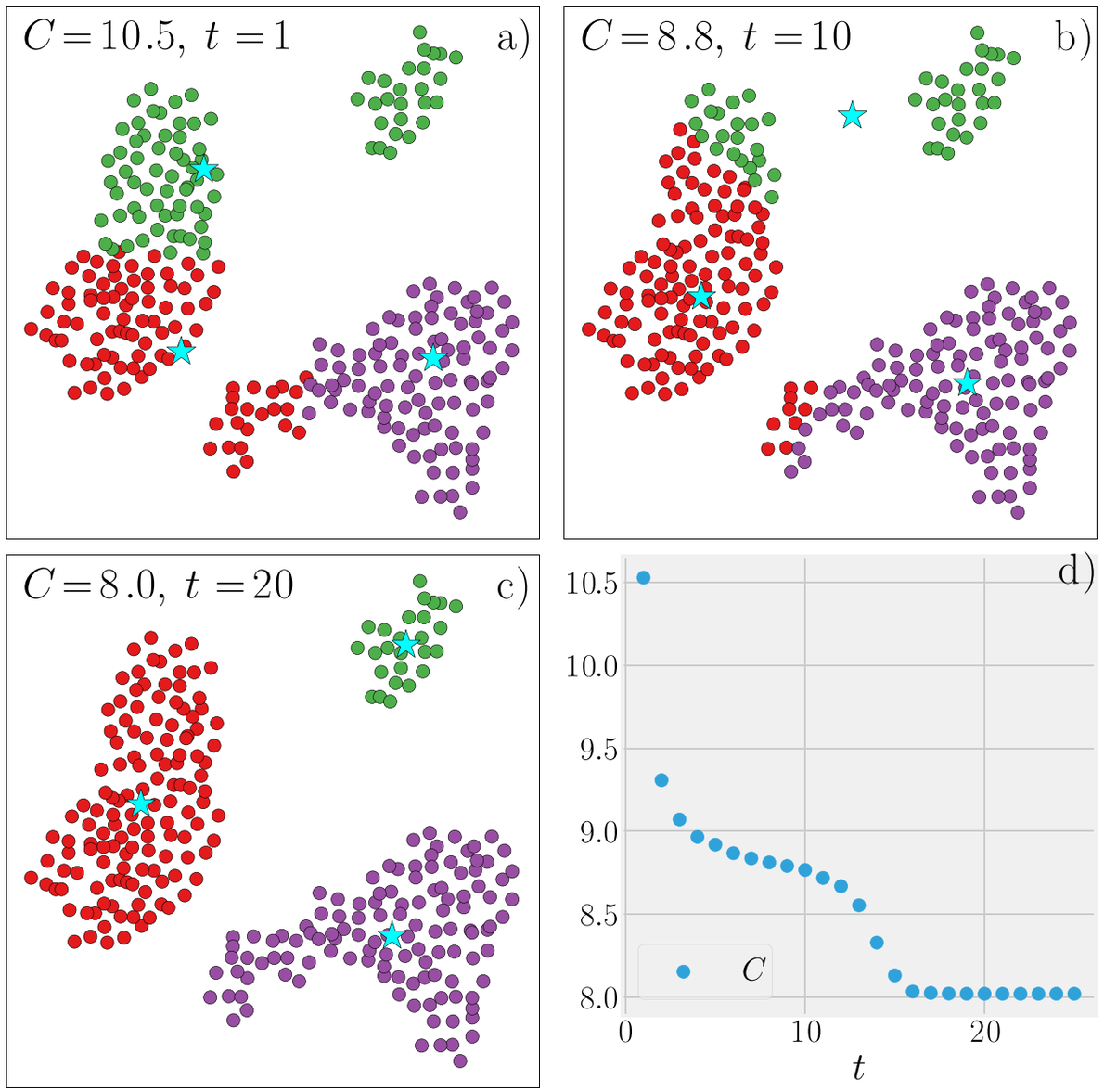


Fig. 55. K-means with $K = 3$ applied to an artificial two-dimensional dataset. The cluster means at each iteration are indicated by cyan star markers. t indicates the iteration number and $\mathcal{C}$ the value of the objective function. (a) The algorithm is initialized by randomly partitioning the space into 3 sectors to generate an initial assignment. (b)–(c) For well separated clusters, the algorithm converges rapidly to the true clusters. (d) The objective function as a function of the iteration. $\mathcal{C}$ converges after $t = 18$ iterations for this choice of random seed (for center initialization).

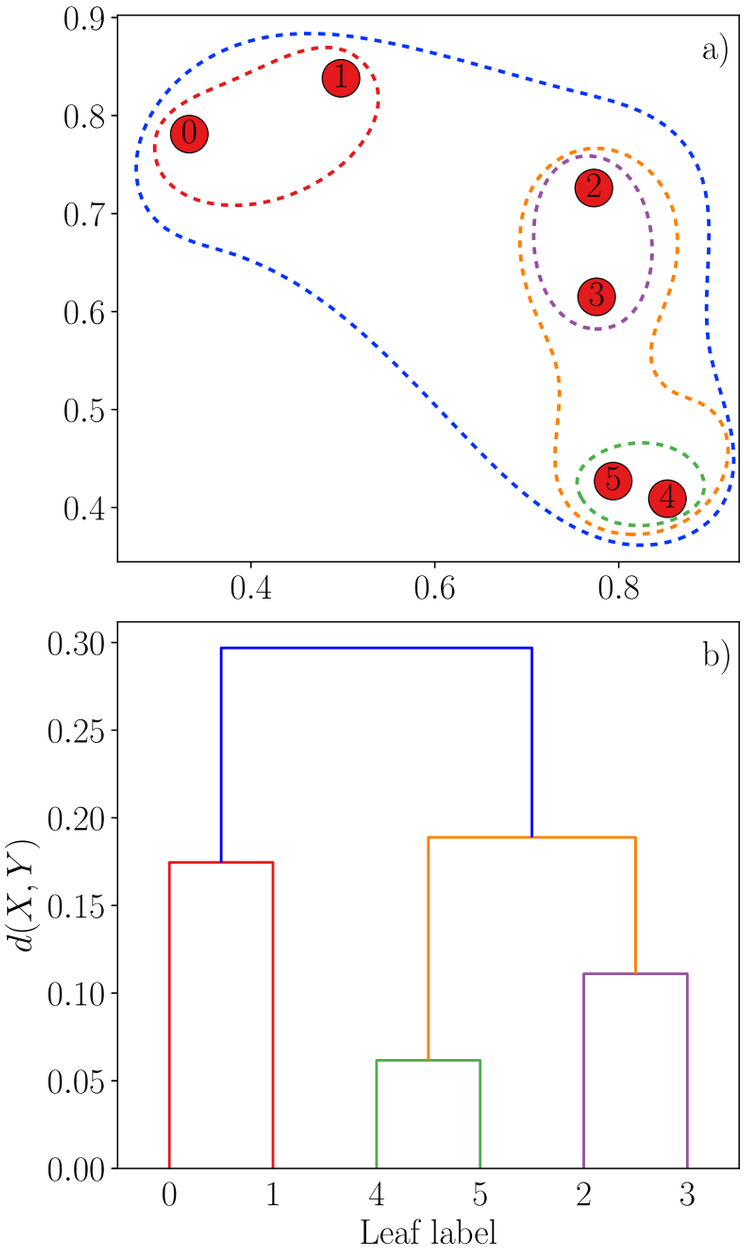
Agglomerative Method

- Start from small initial clusters, then progressively merged to form larger clusters.

- Hierarchy of cluster can be visualized in the form of a **dendrogram**.

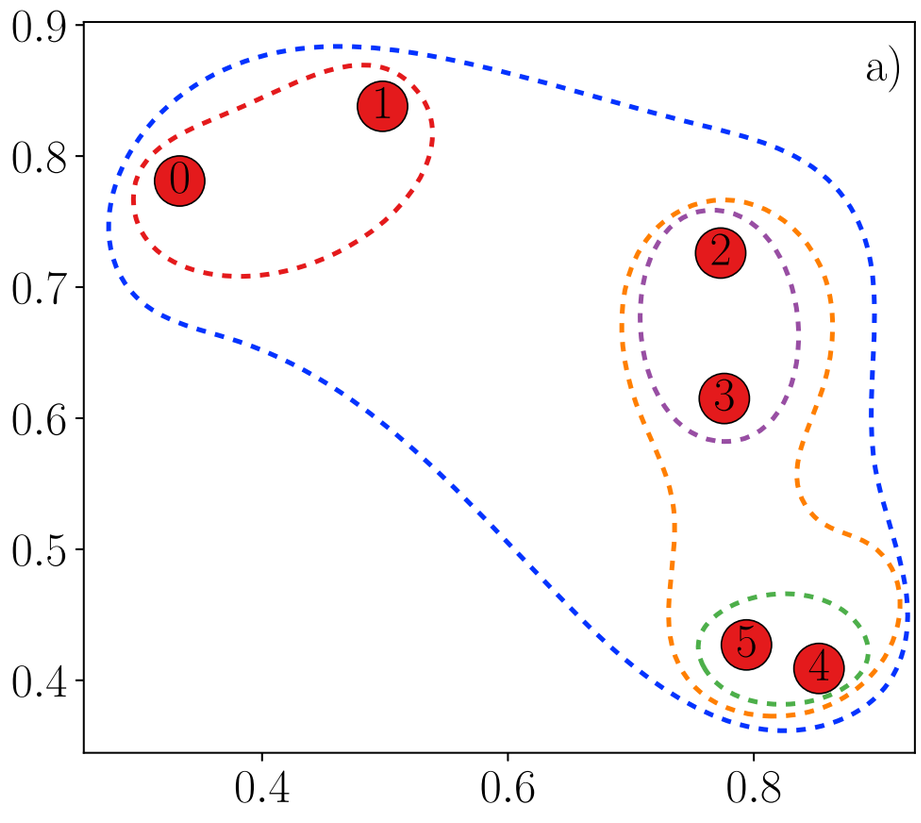
- Define a distance measure $d(X, Y)$ between clusters X and Y.

- Two distances that are closest with respect to $d(X, Y)$ are merged until a single cluster is left.



Agglomerative Clustering Algorithm

- Initialize each point to its own cluster.
- Given a set of K clusters $X_1, X_2, \ldots, X_K$, merge clusters until one cluster is left ($K = 1$):
  - Find the closest pair of clusters $(X_i, X_j) : (i, j) = \text{argmin}_{(i',j')} d(X_{i'}, X_{j'})$
  - Merge the pair. Update $K \leftarrow K - 1$.
- Different linkage methods (distances) result in different algorithms.



Agglomerative Clustering Algorithm

- Different linkage methods (distances):

Single linkage $d(X_i, X_j) = \min_{\mathbf{x}_i \in X_i, \mathbf{x}_j \in X_j} ||\mathbf{x}_i - \mathbf{x}_j||_2$

Complete linkage $d(X_i, X_j) = \max_{\mathbf{x}_i \in X_i, \mathbf{x}_j \in X_j} ||\mathbf{x}_i - \mathbf{x}_j||_2$

Average linkage $d(X_i, X_j) = \frac{1}{|X_i| \cdot |X_j|} \sum_{\mathbf{x}_i \in X_i, \mathbf{x}_j \in X_j} ||\mathbf{x}_i - \mathbf{x}_j||_2$

Ward linkage $d(X_i, X_j) = \frac{|X_i||X_j|}{|X_i \cup X_j|} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^2,$

- The Wald linkage is analogous to k-means in that it minimizes the moment of inertia.

- Problem: Calculation complexity $\mathcal{O}(N^2)$ (suitable for small datasets)

- Practical solution: start with k-means and then proceed hierarchical (agglomerative clustering).

Density-based (DB) Clustering

- Clusters are defined by regions with high density of data points.
- Noise or outliers are expected to form regions of low density.
- Unlike a distance-based approach, DB clustering considers clusters of multiple shapes and sizes while identifying outliers.
- Assumption: relative local density estimation is possible (normally inaccessible for high-dimensional data due to large sampling noise).
- Widely used algorithms: DBSCAN, DB Clustering, etc. See: https://pypi.org/project/fdc/

DBScan Algorithm

- Density-based spatial clustering of application with noise (Ester et al, 1996).

- Crude estimate of local density is the ϵ-neighborhood of point $\mathbf{x}_n$:

$$N_\varepsilon(\mathbf{x}_n) = \{\mathbf{x} \in X | d(\mathbf{x}, \mathbf{x}_n) < \varepsilon\}$$

- $\mathbf{x}_n$ is a **core-point** if at least **minPts** are in its ϵ-neighborhood. A point $\mathbf{x}_i$ is **density-reachable** if it's in a core-point's ϵ-neighborhood.

- $\rightarrow$ Until all points in X have been visited; **do**
  - Pick a point $\mathbf{x}_i$ that has not been visited
  - Mark $\mathbf{x}_i$ as a visited point
  - If $\mathbf{x}_i$ is a core point; **then**
    - Find the set $\mathcal{C}$ of all points that are *density reachable* from $\mathbf{x}_i$.
    - $\mathcal{C}$ now forms a cluster. Mark all points within that cluster as being visited.

$\rightarrow$ Return the cluster assignments $\mathcal{C}_1, \ldots, \mathcal{C}_k$, with k the number of clusters. Points that have not been assigned to a cluster are considered noise or outliers.

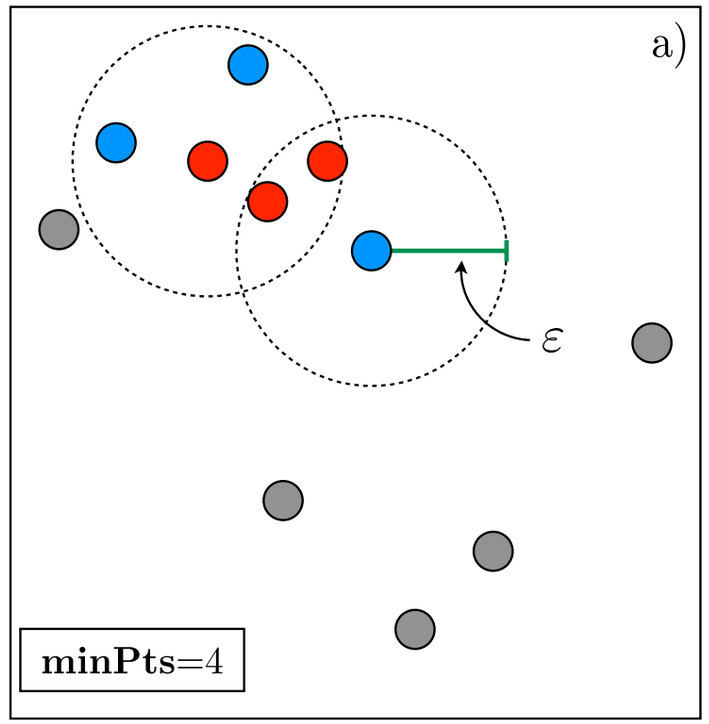
DBScan Algorithm

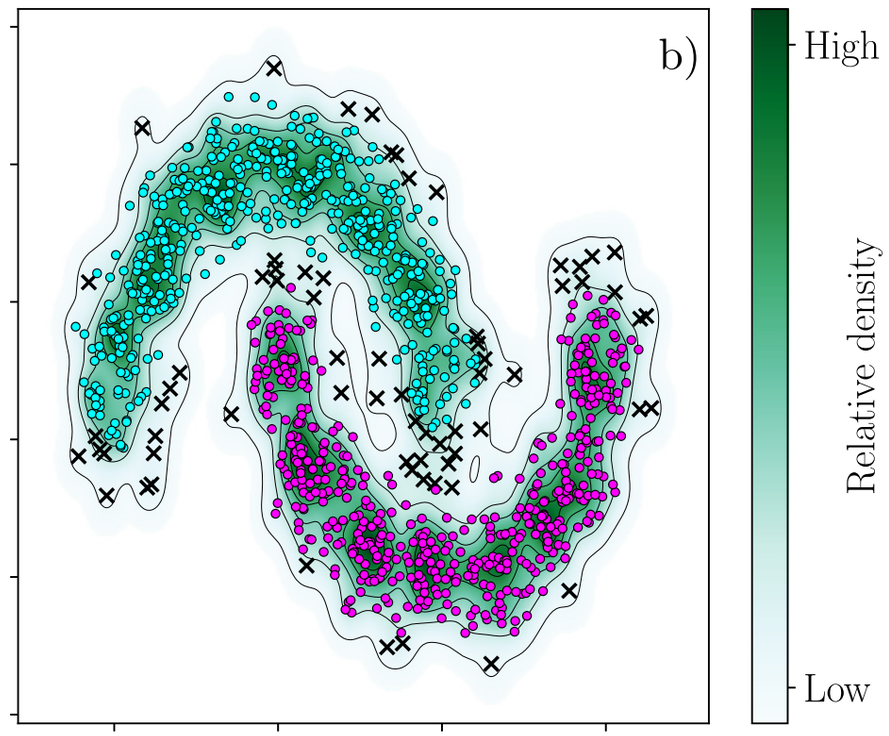
- Do not need to specify # clusters but only the hyperparameters ϵ and **minPts**.

- Scalable to large datasets as computational cost $\sim \mathcal{O}(N \log N)$.

- Note cluster with different shapes and sizes.

- Crosses are outliers.





Latent Variables

- Central to unsupervised learning is the idea of a latent or hidden variable (not directly observable; yet influence visible structure).

- The cluster identify of each datapoint is a latent variable. We cannot observe the label directly, but points in the same cluster are “close”.

- In this abstract language, clustering is an algorithm to learn the most probably value of a latent variable associated with each datapoint.

- Need to make assumption about the structure of data (common to unsupervised learning), e.g., underlying probability distribution from which the data was generated — generative model.

- E.g., in clustering, each cluster is characterized by some probability distribution (e.g. Gaussian distribution with some mean & variance). The latent variable is chosen to minimize some cost function.

Gaussian Mixture Model (GMM)

- Generative model often used in the context of clustering.
- Points are drawn from one of the K Gaussians, with its own μ_k & Σ_k:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})^T\right]$$

- π_k = Probability a pt is drawn from mixture k, the probability of generating a point $\mathbf{x}$ in a GMM is:

$$p(\boldsymbol{x}|\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\}) = \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\pi_k.$$

- Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots \mathbf{x}_N\}$, the likelihood of the dataset:

$$p(\boldsymbol{X}|\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\})$$

- Denote the set of parameters $\{\mu_k, \Sigma_k, \pi_k\}$ by θ.

Gaussian Mixture Model (GMM)

- Common cost function is Maximum likelihood estimation (MLE).

- Latent variables are chosen to maximize the likelihood of the observed data under our generative model → Expectation-Maximization (EM) equations.

- Latent variable $\mathbf{z} = (z_1, \ldots, z_K)$ for point $\mathbf{x}$ has the property that $z_k = 1$ if $\mathbf{x}$ is drawn from the k-th Gaussian, and $z_{j \neq k} = 0$.

- Probability of observing a datapoint $\mathbf{x}$ given $\mathbf{z}$:

$$p(\boldsymbol{x}|\boldsymbol{z}; \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k)^{z_k}$$

- Probability of observing a given value of latent variable:

$$p(\boldsymbol{z}|\{\pi_k\}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Gaussian Mixture Model (GMM)

- Joint probability of a clustering assignment $\mathbf{z}$ and a datapoint $\mathbf{x}$:

$$p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{z}; \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\})p(\boldsymbol{z}|\{\pi_k\}).$$

- Conditional probability of the datapoint in the k-th cluster, $\gamma(z_k)$, given model parameters θ is

$$\gamma(z_k) \equiv p(z_k = 1|\boldsymbol{x}; \theta) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}.$$

  known as the "responsibility" that mixture k takes for explaining $\mathbf{x}$.

- This soft classifier can be made into a hard assignment by assigning each point to the cluster with the largest probability $\arg\max_k \gamma(z_k)$.

Gaussian Mixture Model (GMM)

- Choose the parameters that maximize the likelihood of the data:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{X}|\boldsymbol{\theta})$$

- Use the MLE $\hat{\theta}$ to calculate the optimal hard cluster assignment: $\arg\max_k \hat{\gamma}(z_k)$ where $\hat{\gamma}(z_k) = p(z_k = 1|\boldsymbol{x}; \hat{\boldsymbol{\theta}})$.

- Often impossible to find the global maximum; settle for a local maximum. One approach is to use SGD.

- An alternative approach is an iterative procedure called EM: given an initial guess $\theta^{(0)}$, EM iteratively generates new estimates $\theta^{(1)}, \theta^{(2)}, \ldots$ with non-decreasing likelihood.

Gaussian Mixture Model (GMM)

- Maximize the expected log likelihood given an assignment of the latent variables: $\theta^{(t+1)} = \arg\max_\theta E_{p(\mathbf{Z}|\mathbf{X};\theta^{(t)})}[\log p(\mathbf{X}, \mathbf{Z}; \theta)]$

$$\mathbb{E}_{\tilde{p}^{(t)}}[\log p(\mathbf{X}, \mathbf{Z}; \theta)] = \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} [\log \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \log \pi_k]$$

with the shorthand $\gamma_{ik}^{(t)} = p(z_{ik}|\mathbf{X}; \theta^{(t)})$ with z_{ik} the kth component of $\mathbf{z}_i$.

- Setting the derivatives w.r.t. θ to zero subjected to the constraint

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_i^N \gamma_{ik}^{(t)} \boldsymbol{x}_i}{\sum_i \gamma_{ik}^{(t)}}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{k}}^{(t+1)} = \frac{\sum_i^N \gamma_{ik}^{(t)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i \gamma_{ik}^{(t)}}$$
$$\boldsymbol{\pi}_{\boldsymbol{k}}^{(t+1)} = \frac{1}{N}\sum_k \gamma_{ik}^{(t)}$$

- These are the usual estimates for the mean and variances, with each datapoint weighted according to our current best guess for the probability that it belongs to cluster k.

Gaussian Mixture Model (GMM)

- Our new estimate $\theta^{(t+1)}$ is then used to calculate responsibility $\gamma_{ik}^{(t+1)}$ and repeat the process (c.f. K-means algorithm).

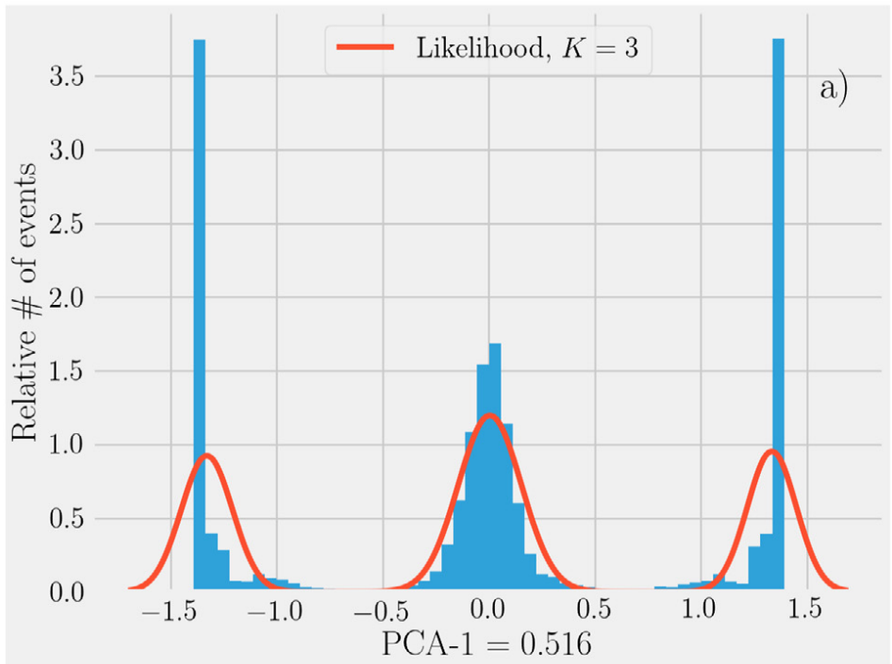
- Application of GMM to the Ising dataset.

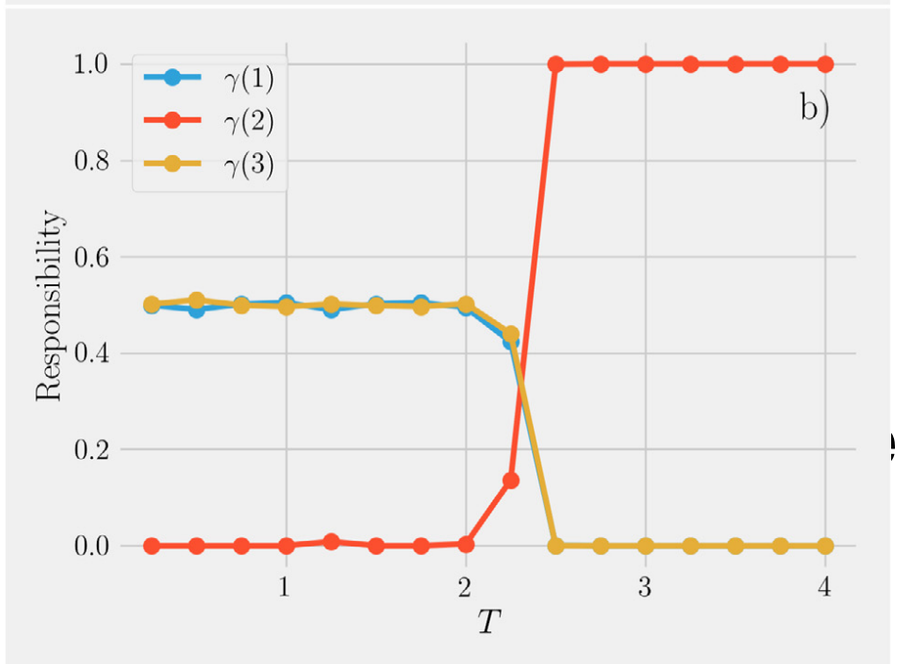
- General lessons:

  - Useful to think of the correlations between visible features as resulting from latent variables.

  - Posit a generative model and find parameters that maximize the likelihood of the observed data.

  - Instead of directly estimate the MLE, computational efficient methods e.g. EM equations.





Summary

- K-means clustering
- Agglomerative clustering
- Density-based (DB) clustering
- Gaussian mixture models