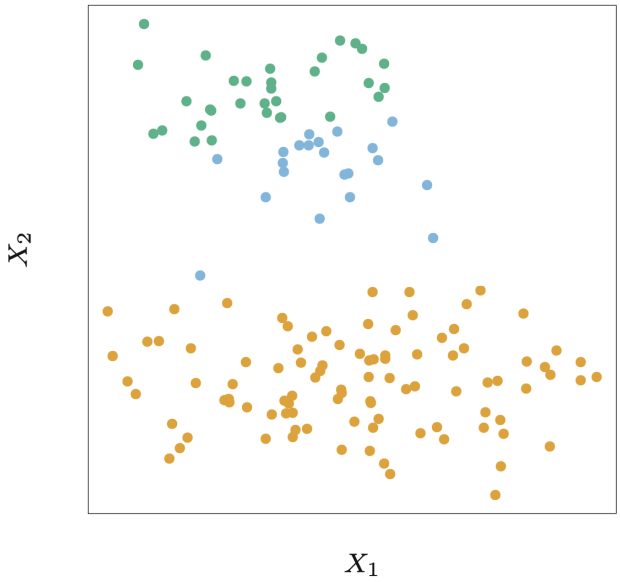
14 Clustering

Clustering is the problem of finding/making groups in the data. Unlike regression and classification, outcomes/labels are not available, so clustering is an example of an "unsupervised learning" problem (density estimation is another example). Clustering is often used as part of exploratory data analysis, or as a first stage of some later downstream estimation task (e.g., for dimension reduction), or as an end in and of itself. Here we study some popular clustering methods, following these notes. The data are $(X_1, ..., X_n) \sim \mathbb{P}$ with each $X_i \in \mathbb{R}^d$, and we want to find some grouping/partition assigning each point to a cluster. The figure below (Figure 14.4 from Hastie et al.) illustrates a clustering into three groups when $d = 2$.



1 k-means clustering

The k-means clustering algorithm (also called vector quantization) was apparently first proposed by Stuart Lloyd of Bell Labs back in 1957, but was not published for another quarter century, in Lloyd [1982]:

> Manuscript received May 1, 1981. The material in this paper was presented in part at the Institute of Mathematical Statistics Meeting, Atlantic City, NJ, September 10–13, 1957.

k-means is an example of *centroid-based clustering*, where each cluster j is represented by a specific central point $x_j^* \in \mathbb{R}^d$, called a *prototype* or *cluster center*. Once the cluster centers are defined, we group the data based on which cluster center they are closest to. It works as follows.

Remark 1. For now we take the number of clusters k as given. The problem of how to choose k in practice is fundamental and somewhat unsettled; one approach is to view k as the true number of clusters, which could be known a priori or estimated under strong assumptions, while another approach views k as a kind of scale parameter, with results displayed across a range of choices of k.

1.1 The algorithm

Let $C = \{c_1, ..., c_k\}$ denote a set of k points with each $c_j \in \mathbb{R}^d$; this is called a *codebook* and will represent the cluster centers. The projection $\Pi_C(X)$ of an observation $X \in \mathbb{R}^d$ onto particular cluster centers C is just the center closest to X, i.e.,

$$\Pi_C(X) = \arg\min_{c \in C} \|c - X\|^2$$

where $\|c - X\|^2 = \sum_j (c_j - X_j)^2$ denotes the Euclidean distance. Then the empirical clustering risk (also called *distortion* in vector quantization) for a given codebook of cluster centers C is

$$\widehat{R}(C) = \frac{1}{n}\sum_{i=1}^{n} \|X_i - \Pi_C(X_i)\|^2 = \frac{1}{n}\sum_{i=1}^{n} \min_{j \in \{1,...,k\}} \|X_i - c_j\|^2,$$

i.e., the average distance between each observation and its closest cluster center. Now k-means clustering amounts to finding the best codebook of cluster centers that minimizes this empirical risk, i.e., letting $\mathcal{C}_k$ denote the set of all codebooks of length k, the estimated cluster centers are

$$\widehat{C} = \arg\min_{C \in \mathcal{C}_k} \widehat{R}(C).$$

The population version of k-means can be defined via the population risk, i.e., the true cluster centers are the minimizers

$$C^* = \arg\min_{C \in \mathcal{C}_k} R(C)$$

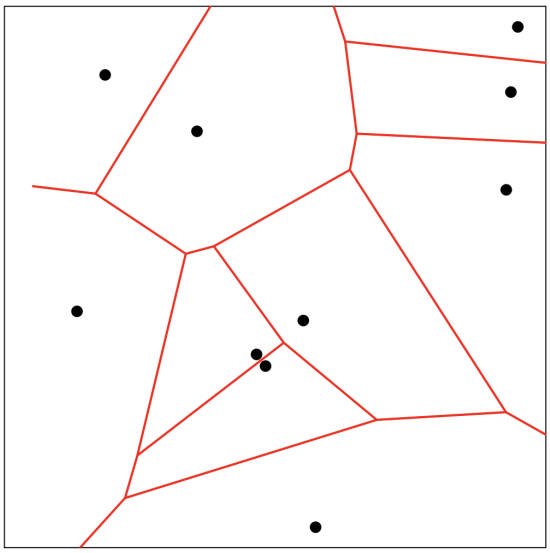
of the true risk

$$R(C) = \mathbb{E}\Big\{\|X - \Pi_C(X)\|^2\Big\} = \mathbb{E}\left\{\min_{j \in \{1,...,k\}} \|X - c_j\|^2\right\}.$$

Given a codebook C of k cluster centers, we can partition the support of X into k groups based on which cluster center is closest; this is called a *Voronoi tesselation*. For example the set closest to c_j is

$$V_j(C) = \Big\{x : \|x - c_j\| \leq \|x - c_\ell\|, \quad \text{for all } \ell \neq j\Big\}$$

and is called a Voronoi cell. The figure below shows an example where $k = 10$.



1.2 Approximation via Lloyd's algorithm

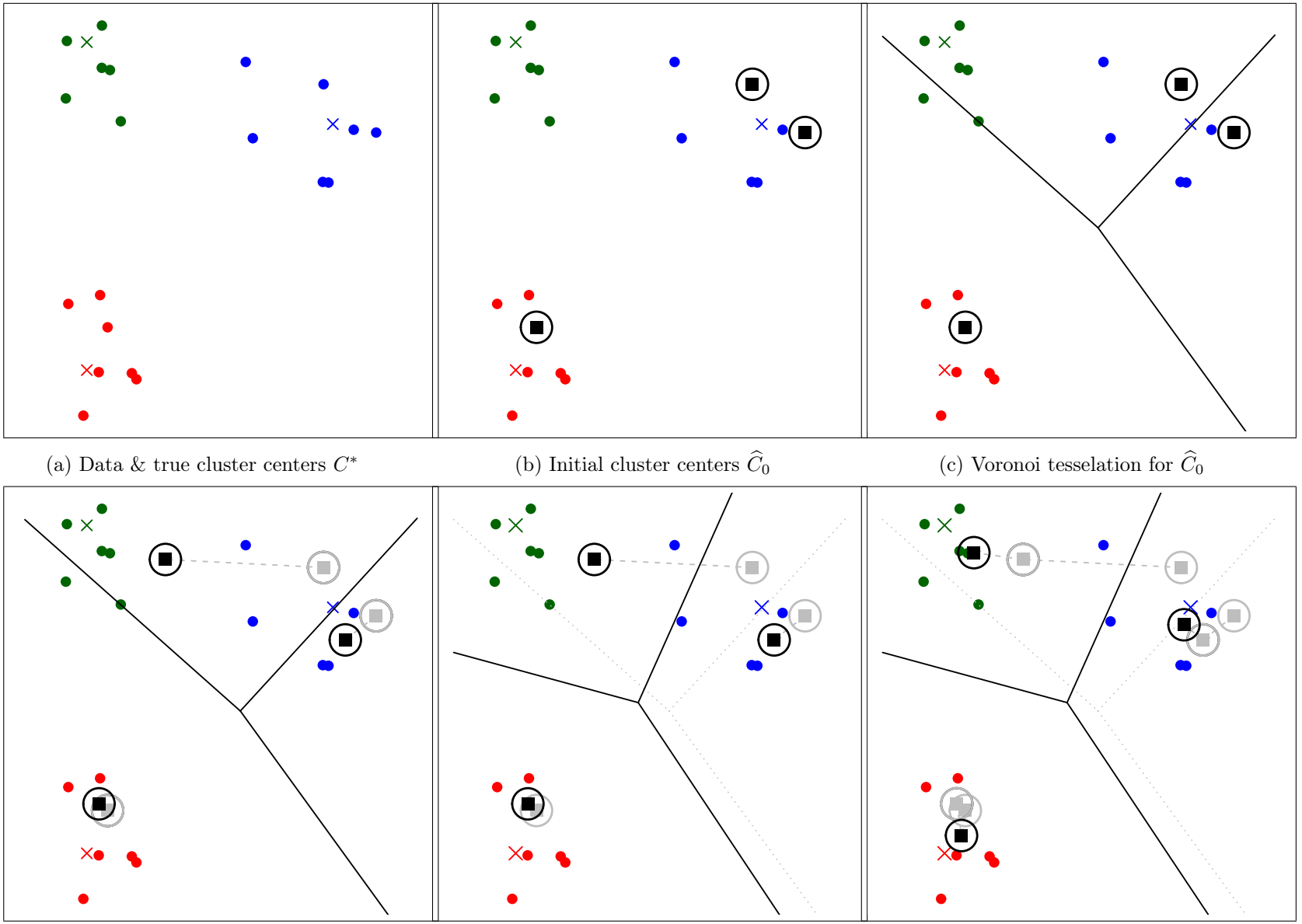
Unfortunately computing the global minimizer $\widehat{C}$ is computationally infeasible; it requires optimizing over combinatorial assignments of observations to clusters, and continuous cluster centers, and is in fact NP-hard [Aloise et al., 2009]. In practice we use approximations such as Lloyd's algorithm:

1. Choose a starting codebook of cluster centers $\widehat{C}_0 = \{c_{01}, ..., c_{0k}\}$.
2. Assign each observation to a cluster based on its closest cluster center, i.e., for each X_i, determine which Voronoi cell $V_j(\widehat{C}_0)$ it falls in.
3. Update the cluster centers to $\widehat{C}_1 = \{c_{11}, ..., c_{1k}\}$ defined as
$$c_{1j} = \frac{\sum_i X_i \mathbb{1}(X_i \in V_j(C))}{\sum_i \mathbb{1}(X_i \in V_j(C))},$$
i.e., the averages of the points in each of the Voronoi cells.
4. Repeat steps 2–3 until convergence, and output final cluster centers $\widehat{C}$.

This algorithm is illustrated in Figure 1 below. The first panel shows observations $X_i \in \mathbb{R}^2$, $i = 1, ..., n = 20$, from $k = 3$ clusters, along with the true cluster centers C^* marked with $\times$. The second panel shows three randomly chosen starting cluster centers C_0 (circled black squares). The third panel shows the Voronoi tesselation $V_j(C_0)$ based on these initial cluster centers. The fourth panel shows the averages of the points in each Voronoi cell (circled black squares), which are now the updated cluster centers C_1. The fifth panel shows the new Voronoi tesselation $V_j(C_1)$ based on these updated cluster centers. The sixth panel shows the averages of the points in each Voronoi cell (circled black squares), which are now the updated cluster centers C_2. The process repeats until the cluster centers stop changing by some ϵ.

Lloyd's algorithm is a greedy approximation and only finds a local optimum of $\widehat{R}(C)$; its performance can depend greatly on the initial cluster centers $\widehat{C}_0$. The simplest remedy is to re-run Lloyd's algorithm many times with different initializations; alternatively one could move to global mixed integer optimization approaches.

Another simple approach is to stick with Lloyd's algorithm but pick starting values in a smart way. Examples of this approach include k-means++ [Arthur and Vassilvitskii, 2007], which starts with a randomly chosen center, then picks the other $k - 1$ centers sequentially, picking points that are farther way with higher probability, so that the cluster centers are more likely to be spread out. These chosen cluster centers are then used as starting points $\widehat{C}_0$ in Lloyd's algorithm. Arthur and Vassilvitskii [2007], prove that the empirical risk of the k-means++ starting points $\widehat{R}(\widehat{C}_0)$ is within a constant factor of the minimum $\widehat{R}(\widehat{C})$ in expectation (over randomness in the k-means++ algorithm).



(a) Data & true cluster centers C^* (b) Initial cluster centers $\widehat{C}_0$ (c) Voronoi tesselation for $\widehat{C}_0$

(d) Updated cluster centers $\widehat{C}_1$ (e) Voronoi tesselation for $\widehat{C}_1$ (f) Updated cluster centers $\widehat{C}_2$

Figure 1: Two steps of Lloyd's algorithm.

1.3 Some theory

What kinds of guarantees does k-means clustering provide? The following result comes from Linder et al. [1994].

Theorem 1. *If $\|X\|^2 \leq M < \infty$ with probability one then*

$$\mathbb{E}R(\widehat{C}) - R(C^*) \leq B\sqrt{\frac{k(d+1)\log n}{n}}$$

for some constant $B > 0$.

Proof. Start by noting that

$$\begin{aligned} R(\widehat{C}) - R(C^*) &= R(\widehat{C}) - \widehat{R}(\widehat{C}) + \widehat{R}(\widehat{C}) - R(C^*) \\ &\leq R(\widehat{C}) - \widehat{R}(\widehat{C}) + \widehat{R}(C^*) - R(C^*) \leq 2 \sup_{C \in \mathcal{C}_k} |\widehat{R}(C) - R(C)| \\ &= 2 \sup_{C \in \mathcal{C}_k} \left| (\mathbb{P}_n - \mathbb{P}) \|X - \Pi_C(X)\|^2 \right| = 2 \sup_{c \in \mathbb{R}^k} \left| (\mathbb{P}_n - \mathbb{P}) \left(\min_j \|X - c_j\|^2 \right) \right| \end{aligned}$$

where the first inequality in the second line follows since $\widehat{R}(\widehat{C}) \leq \widehat{R}(C)$ for any C by definition of $\widehat{C}$ as a minimizer. We have bounded quantities like the above when the class over which the supremum is taken is finite using concentration inequalities; however, here the class $\mathcal{C}_k$ is infinite, and so requires new tools. In particular we can proceed by relating the above supremum to that over a Vapnik-Chervonenkis class $\mathcal{A}$ with shattering number $n^{k(d+1)}$, which gives a concentration inequality of the form

$$\begin{aligned} \mathbb{P}\left\{ 2 \sup_{c \in \mathbb{R}^k} \left| (\mathbb{P}_n - \mathbb{P}) \left(\min_j \|X - c_j\|^2 \right) \right| > \epsilon \right\} &\leq \mathbb{P}\left\{ 2B \sup_{A \in \mathcal{A}} |(\mathbb{P}_n - \mathbb{P})A| > \epsilon \right\} \\ &\leq B_0 n^{k(d+1)} \exp(-B_1 n \epsilon^2) \end{aligned}$$

by the VC theorem. Note the shattering number $n^{k(d+1)}$ plays a role analogous to the size of the set, from our earlier finite-class bounds. As before, this yields a bound on the expected value of the desired form, where instead of $\log |\mathcal{F}|$ we get a $\log n^{k(d+1)} = k(d+1)\log n$ factor. □

Remark 2. Note this result guarantees the risk of the k-means estimated cluster centers is close to the optimal risk, but this does not necessarily imply the estimates $\widehat{C}$ are close to C^*; there could be many labelings that result in near-oracle risk.

1.4 Choosing k

As mentioned earlier, there are many different approaches to choosing k. Note that k is not a tuning parameter like bandwidth, basis dimension, penalty parameters we have seen before; for those parameters the risk is minimized by balancing a bias/variance trade-off. However, for k in k-means, the risk always decreases with k.

Nonetheless there are many proposals for choosing k, including elbow methods (starting with small k and moving up until you see a sharp decrease in empirical risk), hypothesis testing (testing whether the number of clusters is k or greater, and picking the first k that is not rejected), etc. Displaying results across a range of k values is probably safest.

2 Density-based clustering

An alternative to centroid-based methods like k-means is *density-based clustering*, where clusters are based on finding areas of high density. Here we discuss one such method, level set clustering. The main idea originally comes from Hartigan [1975]; recent references that study the method in detail include for example Rigollet [2007] and Rinaldo and Wasserman [2010].

2.1 Level set clustering

The population version of level set clustering can be characterized as follows. The upper level set of a density p is defined as the set

$$L_\lambda = \{x : p(x) > \lambda\}$$

of x values for which the density is at least λ. Suppose L_λ can be decomposed into finitely many disjoint sets, so $L_\lambda = C_1 \cup ... \cup C_{k_\lambda}$. Then $\mathcal{C}_\lambda = \{C_1, ..., C_{k_\lambda}\}$ are called the *level set clusters* at level λ.

The estimated version follows analogously. The level sets can be estimated with plug-in estimators, e.g.,

$$\widehat{L}_\lambda = \{x : \widehat{p}(x) > \lambda\}$$

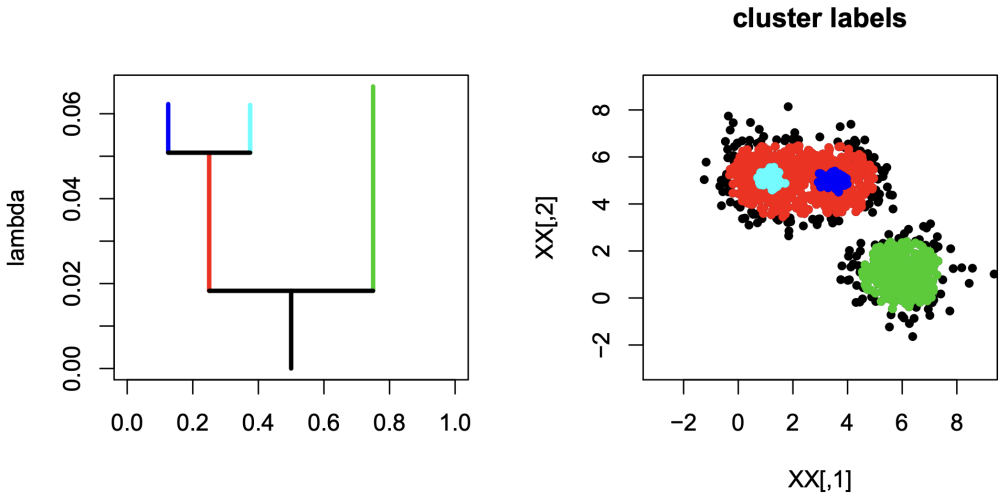
for some density estimator $\widehat{p}$. Then we find the clusters as follows. Let

$$\mathcal{X}_\lambda = \{X_i : \widehat{p}(X_i) > \lambda\}$$

denote the set of observations in the estimated level set. Now construct a graph G_λ with each $X_i \in \mathcal{X}_\lambda$ as a vertex, and where edges connect observations X_i and X_j only if $\|X_i - X_j\| \leq \epsilon$, for some tuning parameter $\epsilon > 0$. Finally the estimated clusters at level λ are taken to be the connected components of the graph. Therefore the steps are:

1. Estimate $\widehat{p}$ and its level set at level λ.
2. Build a graph over $\mathcal{X}_\lambda = \{X_i : \widehat{p}(X_i) > \lambda\}$ and define clusters as connected components.

A two-dimensional example is shown in the figure below.



The level λ and distance ϵ control the number of clusters, akin to k in k-means clustering. Often λ is chosen as a quantile so that

$$\mathbb{P}_n(\mathcal{X}_\lambda) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(X_i \in \mathcal{X}_t) = q,$$

i.e., the level $\lambda = \lambda(q)$ of the level set is chosen to control the size q of the level set (e.g., to ensure it contains say exactly 20% of the sample). As with k in k-means, results should usually be displayed across a range of λ values, rather than picking one λ and hoping it aligns with some true number of clusters.

Remark 3. Another popular density-based clustering method is DBSCAN ("density-based spatial clustering of applications with noise"), developed by Ester et al. [1996]). It depends on tuning parameters M and ϵ, and works as follows: classify all observations into "core points" (observations with at least M points within distance ϵ), "border points" (fewer than M points within ϵ, but in ϵ-neighborhood of core point), and "noise points" (none of the above). Then form clusters by connecting core points to any points within ϵ. This can be viewed as a special case of level set clustering, with densities estimated using point counts within ϵ-balls.

Level set estimation is an interesting problem in its own right and has applications outside clustering, for example in anomaly detection (where lower level sets are the target), multimodality testing, nonlinear PCA, etc. There are many connections between level set estimation and classification, where a primary goal is to estimate the decision set $G^* = \{x : \mu(x) > 1/2\}$, i.e., a level set of a regression function rather than a density. In level set estimation it is common to measure performance via the symmetric difference

$$d_\Delta(\widehat{L}, L_\lambda) = \int_{\widehat{L}\Delta L_\lambda} d\mathbb{P}(x) = \mathbb{P}(X \in \widehat{L}\Delta L_\lambda)$$

for the set difference $A\Delta B = (A^C \cap B) \cup (A \cap B^C)$, or the weighted symmetric difference

$$d_H(\widehat{L}, L_\lambda) = \int_{\widehat{L}\Delta L_\lambda} |p(x) - \lambda| \; d\mathbb{P}(x),$$

which penalizes more when the true density is farther from the level λ. Many results from classification extend naturally to level set estimation, as illustrated below.

Theorem 2. *Assume the margin condition* $\mathbb{P}(0 < |p(X) - \lambda| \le t) \le Ct^\alpha$. *Then the plug-in level set estimator* $\widehat{L} = \{x : \widehat{p}(x) > \lambda\}$ *satisfies*

$$\mathbb{E}\{d_H(\widehat{L}, L_\lambda)\} \le C\|\widehat{p} - p\|_\infty^{1+\alpha}.$$

Proof. Using the same logic as for classification,

$$\begin{aligned}
\mathbb{E}\{d_H(\widehat{L}, L_\lambda)\} &= \int \Big|\mathbb{1}\{\widehat{p}(x) - \lambda > 0\} - \mathbb{1}\{p(x) - \lambda > 0\}\Big| |p(x) - \lambda| \; p(x) \; dx \\
&\le \int \mathbb{1}\{|p(x) - \lambda| \le |\widehat{p}(x) - p(x)|\} |\widehat{p}(x) - p(x)| \; p(x) \; dx \\
&\le \|\widehat{p} - p\|_\infty \mathbb{P}(0 < |p(X) - \lambda| \le \|\widehat{p} - p\|_\infty).
\end{aligned}$$

which together with the margin assumption yields the result. □

3 Other methods

There are many clustering methods available, all with their own advantages and disadvantages. Even within centroid-based and density-based clustering, there are other options beyond k-means and level set clustering. Most of the centroid-based methods are variants on k-means, similar to k-means++. Other variants of density-based clustering include mode and mean-shift clustering, which are based on estimating the modes of the density and then clustering based on which points are closest to which modes.

There are also other types of clustering methods that do not fall under the centroid-based or density-based umbrellas, but instead use *model-based clustering* or *hierarchical clustering*. For example, mixture models are a common way to discover simple model-based cluster structure. These models assume the density is a mix of groups of densities

$$p(x) = \sum_{j=1}^{k} \pi_j p_j(x; \theta_j)$$

where $\pi_j \geq 0$, $\sum_j \pi_j = 1$ and $(\pi_1, ..., \pi_k, \theta_1, ..., \theta_k)$ are the unknown parameters. In other words, you have k different clusters, each arising from a different distribution $p_j(x; \theta_j)$. Mixture models can be estimated by maximum likelihood, and in particular the EM algorithm. Although mixture models are conceptually simple, they can be surprisingly complicated in practice, due to computational difficulties, nonidentifiability, etc.

Another paradigm is based on hierarchical clustering. Hierarchical clustering can either be bottom-up ("agglomerative") or top-down ("divisive"). Agglomerative hierarchical clustering is most common. It starts with each observation in its own cluster; then progressively combines observations into clusters based on how close they are. Distance between clusters can be measured via single/complete/average linkage rules. For single linkage rules, distance between clusters is based on the minimum distance between points in each cluster; for complete linkage rules, distance is the maximum distance; and for average linkage rules, distance is the mean distance. The figure below illustrates hierarchical clustering.

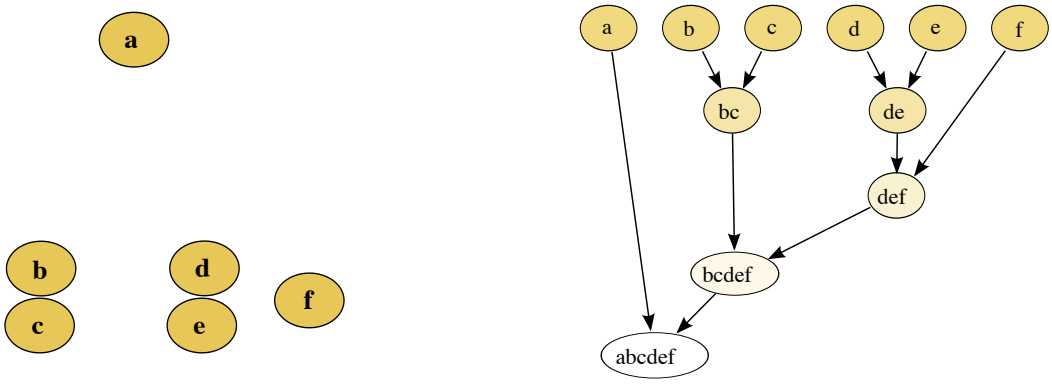


Figure 2: Hierarchical clustering example from wikipedia. (Left panel displays the original data, right panel displays the dendrogram.)

References

D. Aloise, A. Deshpande, P. Hansen, and P. Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75:245–248, 2009.

D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms*, pages 1027–1035, 2007.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd*, 96(34):226–231, 1996.

J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

T. Linder, G. Lugosi, and K. Zeger. Rates of convergence in the source coding theorem, in empirical quantizer design, and in universal lossy source coding. *IEEE Transactions on Information Theory*, 40(6):1728–1740, 1994.

S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

P. Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8(7), 2007.

A. Rinaldo and L. Wasserman. Generalized density clustering. *The Annals of Statistics*, 38 (5):2678–2722, 2010.