Clustering

MIT 15.097 Course Notes

Cynthia Rudin and Şeyda Ertekin

Credit: Dasgupta, Hastie, Tibshirani, Friedman

Clustering (a.k.a. data segmentation) Let's segment a collection of examples into "clusters" so that objects within a cluster are more closely related to one another than objects assigned to different clusters. We want to assign each example x_i to a cluster $k \in \{1, ..., K\}$.

The K-Means algorithm is a very popular way to do this. It assumes points lie in Euclidean space.

Input: Finite set $\{\mathbf{x}_i\}_{1=1}^m, \mathbf{x}_i \in \mathbf{R}^n$
Output: $\mathbf{z}_1, ..., \mathbf{z}_K$ cluster centers
Goal: Minimize

$$\text{cost}(\mathbf{z}_1, ..., \mathbf{z}_K) := \sum_i \min_k \|\mathbf{x}_i - \mathbf{z}_k\|_2^2.$$

The choice of the squared norm is fortuitous, it really helps simplify the math!

If we're given points $\{\mathbf{z}_k\}_k$, they can induce a *Voronoi partition* of $\mathbf{R}^n$: they break the space into cells where each cell corresponds to one of the $\mathbf{z}_k$'s. That is, each cell contains the region of space whose nearest representative is $\mathbf{z}_k$.

Draw a picture

We can look at the examples in each of these regions of space, which are the clusters. Specifically,

$$C_k := \{\mathbf{x}_i : \text{ the closest representative to } \mathbf{x}_i \text{ is } \mathbf{z}_k\}.$$

Let's compute the cost another way. Before, we summed over examples, and then picked the right representative $\mathbf{z}_k$ for each example. This time, we'll sum over clusters, and look at all the examples in that cluster:

$$\text{cost}(\mathbf{z}_1, ..., \mathbf{z}_K) = \sum_k \sum_{\{i:\mathbf{x}_i \in C_k\}} \|\mathbf{x}_i - \mathbf{z}_k\|_2^2.$$

While we're analyzing, we'll need to consider suboptimal partitions of the data, where an example might not be assigned to the nearest representative. So we redefine the cost:

$$\text{cost}(C_1, ..., C_K; \mathbf{z}_1, ..., \mathbf{z}_K) = \sum_k \sum_{\{i:\mathbf{x}_i \in C_k\}} \|\mathbf{x}_i - \mathbf{z}_k\|_2^2. \tag{1}$$

Let's say we only have one cluster to deal with. Call it C. The representative is $\mathbf{z}$. The cost is then:

$$\text{cost}(C; \mathbf{z}) = \sum_{\{i:\mathbf{x}_i \in C\}} \|\mathbf{x}_i - \mathbf{z}\|_2^2.$$

Where should we place $\mathbf{z}$?

As you probably guessed, we would put it at the mean of the examples in C. But also, the additional cost incurred by picking $z \neq \text{mean}(C)$ can be characterized very simply:

Lemma 1. For any set $C \subset \mathbf{R}^n$ and any $\mathbf{z} \in \mathbf{R}^n$,

$$\text{cost}(C; \mathbf{z}) = \text{cost}(C, \text{mean}(C)) + |C| \cdot \|\mathbf{z} - \text{mean}(C)\|_2^2.$$

Let's go ahead and prove it. In order to do that, we need to do another bias-variance decomposition (this one's pretty much identical to one of the ones we did before).

Lemma 2. *Let $X \in \mathbf{R}^n$ be any random variable. For any $\mathbf{z} \in \mathbf{R}^n$, we have:*

$$\mathbf{E}_X\|X - \mathbf{z}\|_2^2 = \mathbf{E}_X\|X - \mathbf{E}_X X\|_2^2 + \|\mathbf{z} - \mathbf{E}_X X\|_2^2.$$

Proof. Let $\bar{\mathbf{x}} := \mathbf{E}_X X$.

$$\begin{aligned}
\mathbf{E}_X\|X - \mathbf{z}\|_2^2 &= \mathbf{E}_X \sum_j (X^{(j)} - z^{(j)})^2 \\
&= \mathbf{E}_X \sum_j (X^{(j)} - \bar{x}^{(j)} + \bar{x}^{(j)} - z^j)^2 \\
&= \mathbf{E}_X \sum_j (X^{(j)} - \bar{x}^{(j)})^2 + \mathbf{E}_X \sum_j (\bar{x}^{(j)} - z^j)^2 \\
&\quad +2\mathbf{E}_X \sum_j (X^{(j)} - \bar{x}^{(j)})(\bar{x}^{(j)} - z^{(j)}) \\
&= \mathbf{E}_X\|X - \bar{\mathbf{x}}\|_2^2 + \mathbf{E}_X\|\bar{\mathbf{x}} - \mathbf{z}\|_2^2 + 0. \quad \blacksquare
\end{aligned}$$

To prove Lemma 1, pick a specific choice for X, namely X is a uniform random draw from the points $\mathbf{x}_i$ in set C. So X has a discrete distribution. What will happen with this choice of X is that the expectation will reduce to the cost we already defined above.

$$\begin{aligned}
\mathbf{E}_X \|X - \mathbf{z}\|_2^2 &= \sum_{\{i:\mathbf{x}_i \in C\}} (\text{prob. that point } i \text{ is chosen}) \|\mathbf{x}_i - \mathbf{z}\|_2^2 \\
&= \sum_{\{i:\mathbf{x}_i \in C\}} \frac{1}{|C|} \|\mathbf{x}_i - \mathbf{z}\|_2^2 = \frac{1}{|C|} \text{cost}(C, \mathbf{z}) \qquad (2)
\end{aligned}$$

and if we use Lemma 2 substituting $\mathbf{z}$ to be $\bar{\mathbf{x}}$ (a.k.a., $\mathbf{E}_X X$, or mean(C)) and simplify as in (2):

$$\mathbf{E}_X \|X - \bar{\mathbf{x}}\|_2^2 = \frac{1}{|C|} \text{cost}(C, \text{mean}(C)). \qquad (3)$$

We had already defined cost earlier, and the choice of X was nice because its expectation is just the cost. Let's recopy Lemma 2's statement here, using the $\bar{\mathbf{x}}$ notation:

$$\mathbf{E}_X \|X - \mathbf{z}\|_2^2 = \mathbf{E}_X \|X - \bar{\mathbf{x}}\|_2^2 + \|\mathbf{z} - \bar{\mathbf{x}}\|_2^2.$$

Plugging in (2) and (3),

$$\frac{1}{|C|} \text{cost}(C, \mathbf{z}) = \frac{1}{|C|} \text{cost}(C, \text{mean}(C)) + \|\mathbf{z} - \bar{\mathbf{x}}\|_2^2. \qquad (4)$$

Multiplying through,

$$\text{cost}(C; \mathbf{z}) = \text{cost}(C, \text{mean}(C)) + |C| \cdot \|\mathbf{z} - \text{mean}(C)\|_2^2.$$

And that's the statement of Lemma 1. ■

To really minimize the cost (1), you'd need to try all possible assignments of the m data points to K clusters. Uck! The number of distinct assignments is (Jain and Dubes 1988):

$$S(m, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^m$$

$$S(10, 4) = 34K, \quad S(19, 4) \approx 10^{10}, \ldots \quad \text{so not doable.}$$

Let's try some heuristic gradient-descent-ish method instead.

The K-Means Algorithm

Choose the value of K before you start.

```
Initialize centers $\mathbf{z}_1, ..., \mathbf{z}_K \in \mathbf{R}^n$ and clusters $C_1, ..., C_K$ in any way.
Repeat until there is no further change in cost:
      for each $k$:  $C_k \leftarrow \{\mathbf{x}_i :$ the closest representative is $\mathbf{z}_k\}$
      for each $k$:  $\mathbf{z}_k = \text{mean}(C_k)$
```

This is simple enough, and takes $O(Km)$ time per iteration.

$\boxed{\text{PPT demo}}$

Of course, it doesn't always converge to the optimal solution.

But does the cost converge?

Lemma 3. *During the course of the K-Means algorithm, the cost monotonically decreases.*

Proof. Let $\mathbf{z}_1^{(t)}, ..., \mathbf{z}_K^{(t)}, C_1^{(t)}, ..., C_K^{(t)}$ denote the centers and clusters at the start of the t^{th} iterate of K-Means. The first step of the iteration assigns each data point to its closest center, therefore, the cluster assignment is better:

$$\text{cost}(C_1^{(t+1)}, ..., C_K^{(t+1)}, \mathbf{z}_1^{(t)}, ..., \mathbf{z}_K^{(t)}) \leq \text{cost}(C_1^{(t)}, ..., C_K^{(t)}, \mathbf{z}_1^{(t)}, ..., \mathbf{z}_K^{(t)}).$$

On the second step, each cluster is re-centered at its mean, so the representatives are better. By Lemma 1,

$$\text{cost}(C_1^{(t+1)}, ..., C_K^{(t+1)}, \mathbf{z}_1^{(t+1)}, ..., \mathbf{z}_K^{(t+1)}) \leq \text{cost}(C_1^{(t+1)}, ..., C_K^{(t+1)}, \mathbf{z}_1^{(t)}, ..., \mathbf{z}_K^{(t)}).$$

■

$\boxed{\text{So does the cost converge?}}$

Example of how K-Means could converge to the wrong thing

How might you make K-Means more likely to converge to the optimal?

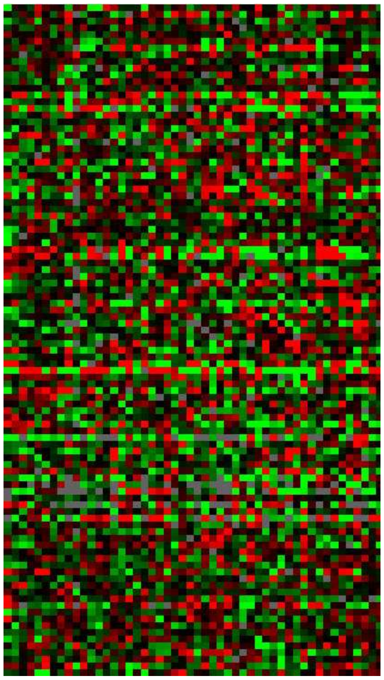
How might you choose K? (Why can't you measure test error?)

Other ways to evaluate clusters ("cluster validation")

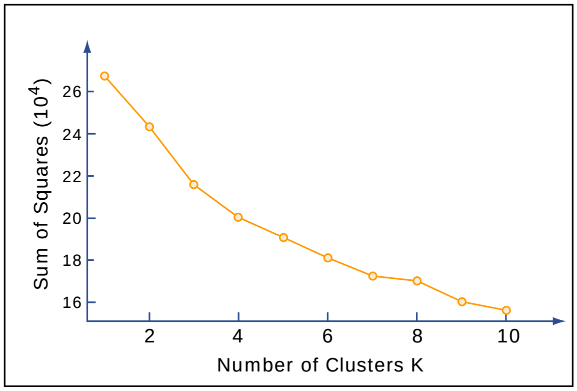
There are loads of cluster validity measures, alternatives to the cost. Draw a picture

- Davies-Baldwin Index - looks at average intracluster distance (within-cluster distance) to the centroid (want it to be small), and intercluster distances between centroids (want it to be large).
- Dunn Index - looks pairwise at minimal intercluster distance (want it to be large) and maximal intracluster distance (want it to be small).

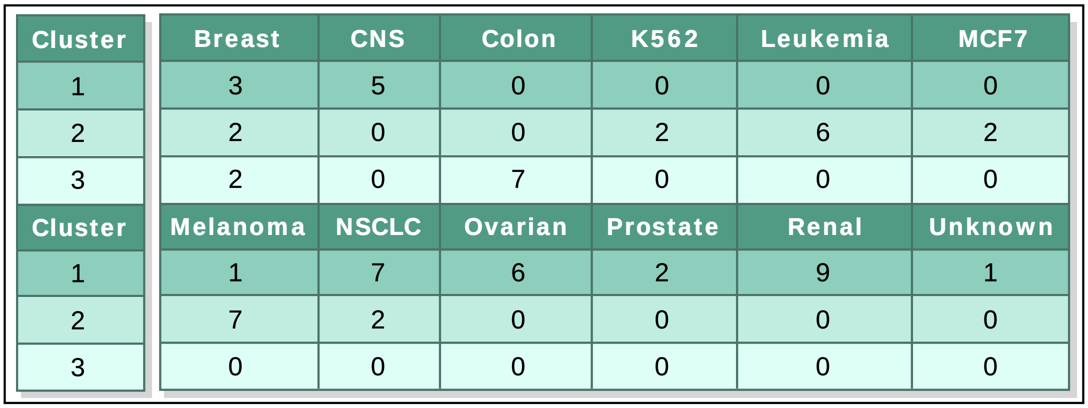
Example: Microarray data. Have 6830 genes (rows) and 64 patients (columns). The color of each box is a measurement of the expression level of a gene. The expression level of a gene is basically how much of its special protein it is producing. The physical chip itself doesn't actually measure protein levels, but a proxy for them (which is RNA, which sticks to the DNA on the chip). If the color is green, it means low expression levels, if the color is red, it means higher expression levels. Each patient is represented by a vector, which is the expression level of their genes. It's a column vector with values given in color:



Each patient (column) has some type of cancer. Want to cluster patients to see whether patients with the same types of cancers cluster together. So each cluster center is an "average" patient expression level vector for some type of cancer. It's also a column vector



Hm, there's no kink in this figure. Compare $K = 3$ solution with "true" clusters:

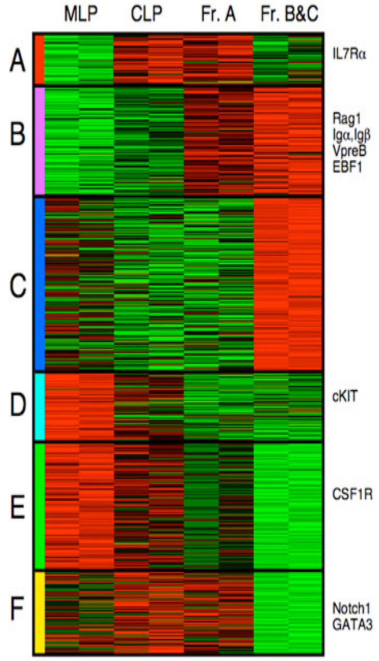


Cluster	Breast	CNS	Colon	K562	Leukemia	MCF7	Melanoma	NSCLC	Ovarian	Prostate	Renal	Unknown
1	3	5	0	0	0	0	1	7	6	2	9	1
2	2	0	0	2	6	2	7	2	0	0	0	0
3	2	0	7	0	0	0	0	0	0	0	0	0

Images by MIT OpenCourseWare, adapted from Hastie et al., *The Elements of Statistical Learning*, Springer, 2009.

It's pretty good at keeping the same cancers in the same cluster. The two breast cancers in the 2nd cluster were actually melanomas that metastasized.

Generally we cluster genes, not patients. Would really like to get something like this in practice:



Courtesy of the Rockefeller University Press. Used with permission.

Figure 7 from Rumfelt, Lynn, et al. "Lineage Specification and Plasticity in CD19- Early B cell Precursors." *Journal of Experimental Medicine* 203 (2006): 675-87.

where each row is a gene, and the columns are different immune cell types.

A major issue with K-means: as K changes, cluster membership can change arbitrarily. A solution is *Hierarchical Clustering*.

- clusters at the next level of the hierarchy are created by merging clusters at the next lowest level.
  - lowest level: each cluster has 1 example
  - highest level: there's only 1 cluster, containing all of the data.

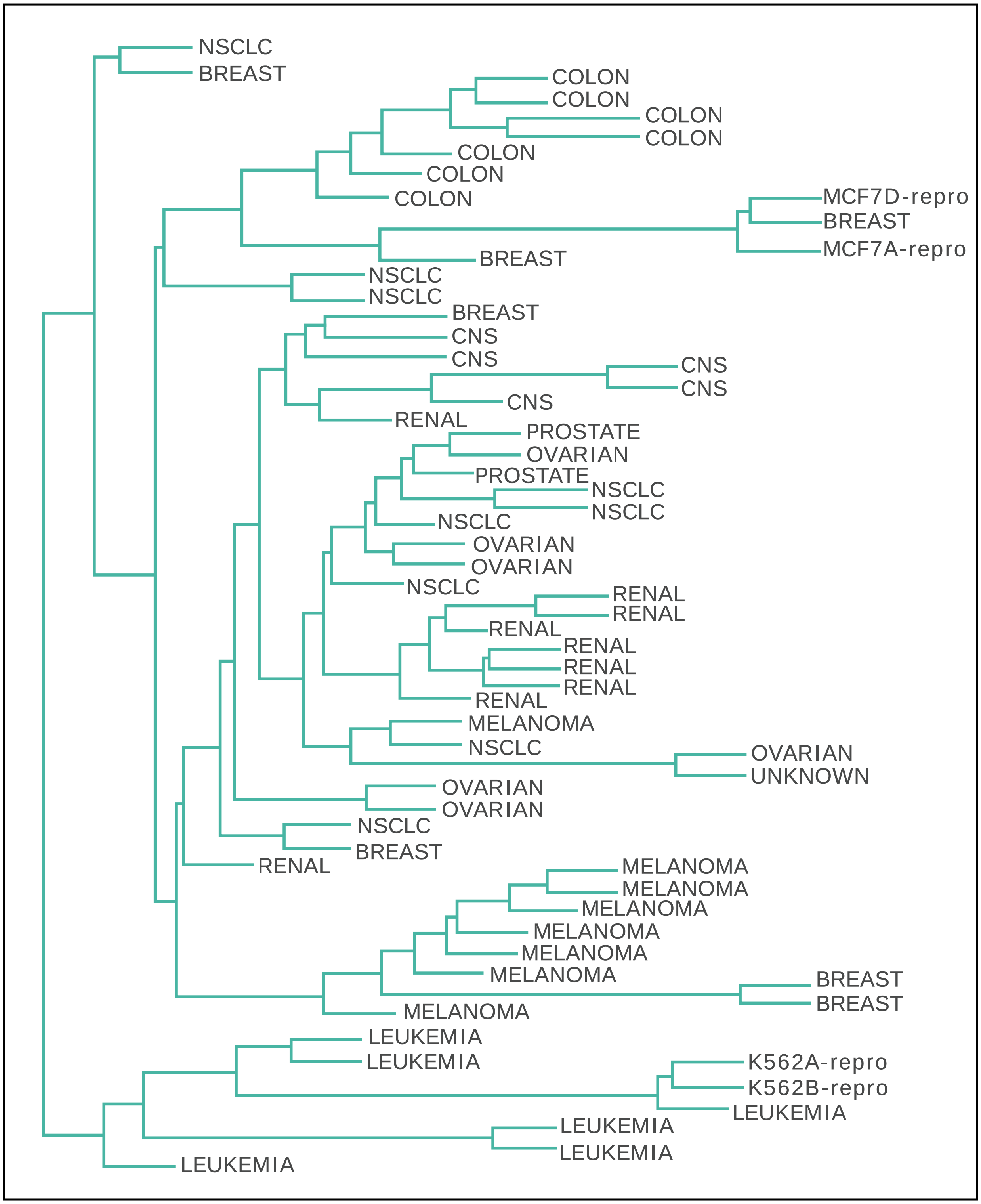


Image by MIT OpenCourseWare, adapted from Hastie et al., *The Elements of Statistical Learning*, Springer, 2009.

Application Slides