# Unsupervised Learning: Clustering Analysis

Héctor Corrada Bravo

University of Maryland, College Park, USA

CMSC 643: 2018-10-29

# Unsupervised Learning

So far we have seen "Supervised Methods" where our goal is to analyze a *response* (or outcome) based on various *predictors*.

In many cases, especially for Exploratory Data Analysis, we want methods to extract patterns on variables without analyzing a specific *response*.

Methods for the latter case are called "Unsupervised Methods". Examples are *Principal Component Analysis* and *Clustering*.

# Unsupervised Learning

Interpretation of these methods is much more *subjective* than in Supervised Learning.

For example: if we want to know if a given *predictor* is related to *response*, we can perform statistical inference using hypothesis testing.

# Unsupervised Learning

If we want to know which predictors are useful for prediction: use cross-validation to do model selection.

Finally, if we want to see how well we can predict a specific response, we can use cross-validation to report on test error.

# Unsupervised Learning

In unsupervised methods, there is no similar clean evaluation methodology.

Nonetheless, they can be very useful methods to understand data at hand.

# Motivating Example

Time series dataset of mortgage affordability as calculated and distributed by Zillow: https://www.zillow.com/research/data/.

The dataset consists of monthly mortgage affordability values for 76 counties with data from 1979 to 2017.

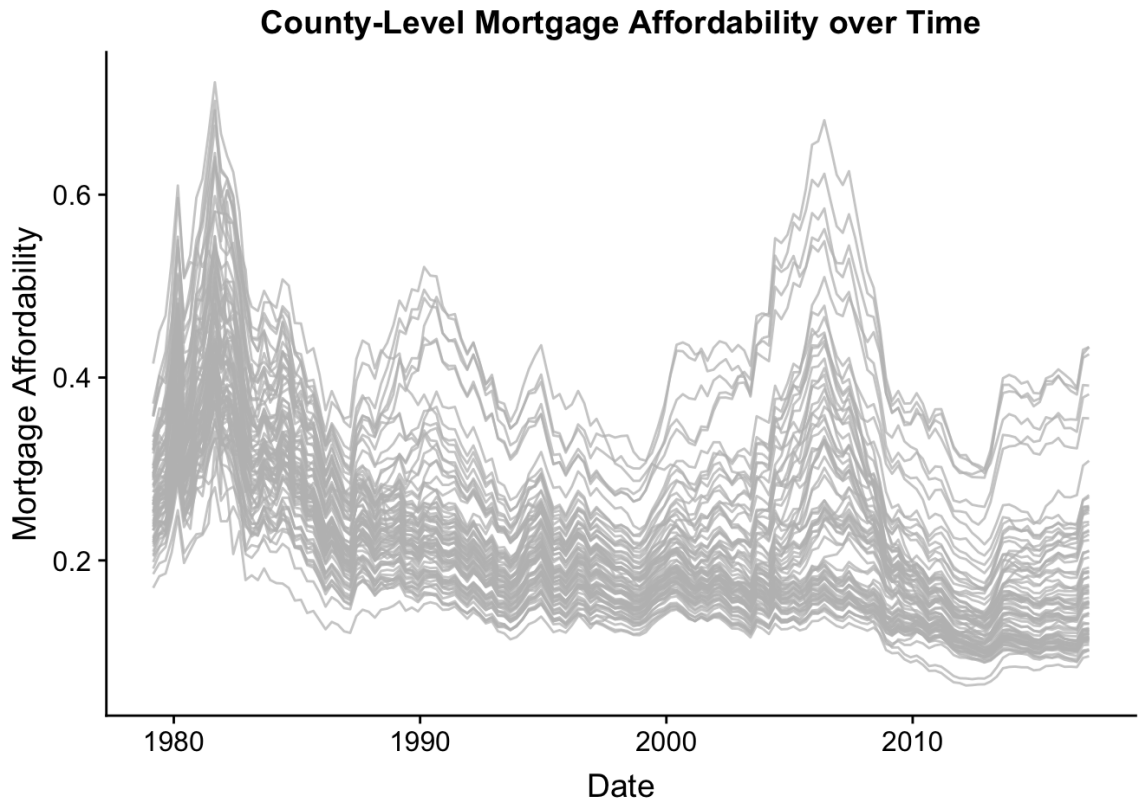
# Motivating Example

> "To calculate mortgage affordability, we first calculate the mortgage payment for the median-valued home in a metropolitan area by using the metro-level Zillow Home Value Index for a given quarter and the 30-year fixed mortgage interest rate during that time period, provided by the Freddie Mac Primary Mortgage Market Survey (based on a 20 percent down payment)."

# Motivating Example

> "Then, we consider what portion of the monthly median household income (U.S. Census) goes toward this monthly mortgage payment. Median household income is available with a lag. "

# Motivating Example




Can we partition counties into groups of counties with similar value trends across time?

# Preliminaries

In "Supervised Learning" we were concerned with estimates that minimize some error function relative to the outcome of interest $Y$:

$$\mu(x) = \arg\min_{\theta} E_{Y|X} L(Y, \theta)$$

# Preliminaries

In order to do this, explicitly or not, the methods we were using would be concerned with properties of the conditional probability distribution $Pr(Y|X)$,

without concerning itself with probability distribution $Pr(X)$ of the covariates themselves.

# Preliminaries

In unsupervised learning, we are interested in properties of $Pr(X)$.

In our example, what can we say about the distribution of home value time series?

Since the dimensionality of $Pr(X)$ can be large, unsupervised learning methods seek to find structured representations of $Pr(X)$ that would be possible to estimate.

# Preliminaries

In *clustering* we assume that predictor space is partitioned and that $Pr(X)$ is defined over those partitions.

In *dimensionality reduction* we assume that $Pr(X)$ is really defined over a space (manifold) of smaller dimension. We will start studying clustering first.

# Cluster Analysis

The high-level goal of cluster analysis is to organize objects (observations) that are *similar* to each other into groups.

We want objects within a group to be more *similar* to each other than objects in different groups.

# Cluster Analysis

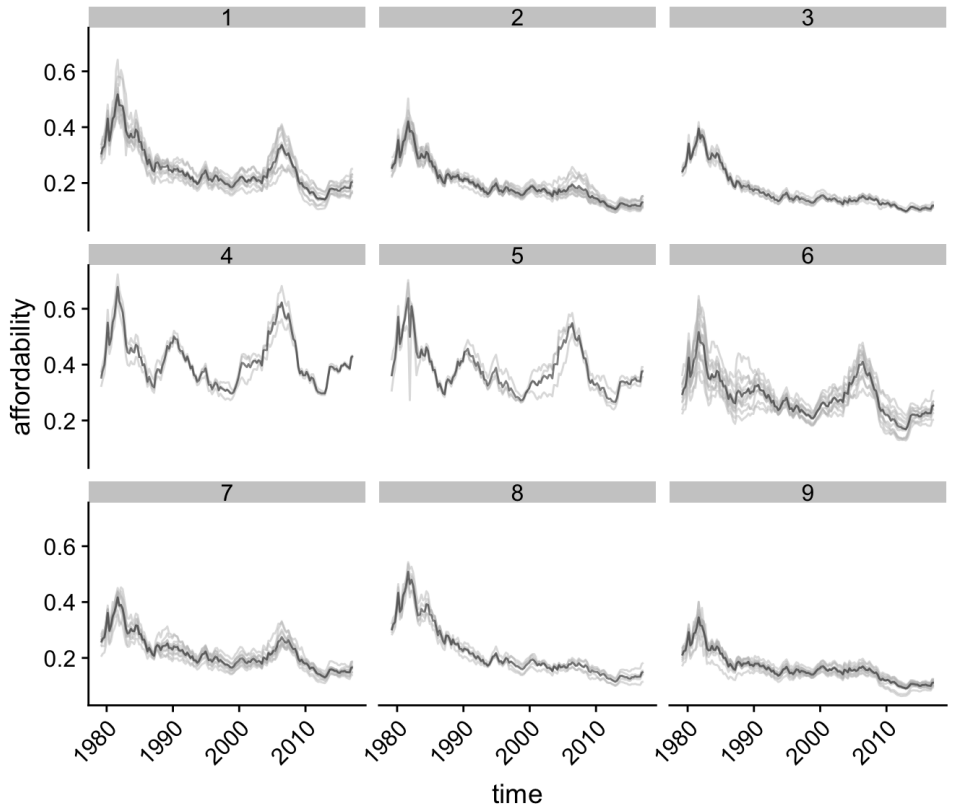
The high-level goal of cluster analysis is to organize objects (observations) that are *similar* to each other into groups.

We want objects within a group to be more *similar* to each other than objects in different groups.

Central to this high-level goal is how to measure the degree of *similarity* between objects.

A clustering method then uses the *similarity* measure provided to it to group objects into clusters.

# Cluster Analysis




Result of the k-means algorithm partitioning the data into 9 clusters.

The darker series within each cluster shows the average time series within the cluster.

# Dissimilarity-based Clustering

For certain algorithms, instead of similarity we work with dissimilarity, often represented as distances.

When we have observations defined over attributes, or predictors, we define dissimilarity based on these attributes.

# Dissimilarity-based Clustering

Given measurements $x_{ij}$ for $i = 1, \ldots, N$ observations over $j = 1, \ldots, p$ predictors.

Suppose we define a dissimilarity $d_j(x_{ij}, x_{i'j})$, we can then define a dissimilarity between objects as

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$$

# Dissimilarity-based Clustering

In the k-means algorithm, and many other algorithms, the most common usage is squared distance

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

We can use different dissimilarities, for example

$$d_j(x_{ij}, x_{i'j}) = |x_{ij} - x_{i'j}|$$

which may affect our choice of clustering algorithm later on.

# Dissimilarity-based Clustering

For categorical variables, we could set

$$d_j(x_{ij}, x_{i'j}) = \begin{cases} 0 \text{ if } x_{ij} = x_{i'j} \\ \\ 1 \text{ o.w.} \end{cases}$$

# Dissimilarity-based Clustering

If the values the categorical variable have an intrinsic similarity

Generalize using symmetric matrix $L$ with elements

$L_{rr'} = L_{r'r}$,

$L_{rr} = 0$ and

$L_{rr'} \geq 0$ otherwise.

This may of course lead to a dissimilarity that is not a proper distance.

# K-means Clustering

A commonly used algorithm to perform clustering is the K-means algorithm.

It is appropriate when using squared Euclidean distance as the measure of object dissimilarity.

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

$$= \|x_i - x_{i'}\|^2$$

# K-means Clustering

K-means partitions observations into $K$ clusters, with $K$ provided as a parameter.

Given some clustering, or partition, $C$, denote cluster assignment of observation $x_i$ to cluster $k \in \{1, \ldots, K\}$ is denoted as $C(i) = k$.

# K-means Clustering

K-means partitions observations into $K$ clusters, with $K$ provided as a parameter.

Given some clustering, or partition, $C$, denote cluster assignment of observation $x_i$ to cluster $k \in \{1, \ldots, K\}$ is denoted as $C(i) = k$.

K-means minimizes this clustering criterion:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i:\, C(i)=k} \sum_{i':\, C(i')=k} \|x_i - x_{i'}\|^2$$

# K-means Clustering

This is equivalent to minimizing

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} N_k \sum_{i:\, C(i)=k} \|x_i - \bar{x}_k\|^2$$

with:

- $\bar{x}_k = (\bar{x}_{k1}, \ldots, \bar{x}_{kp})$
- $\bar{x}_{kj}$ is the average of predictor $j$ over the observations assigned to cluster $k$,
- $N_k$ is the number of observations assigned to cluster $k$

# K-means Clustering

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} N_k \sum_{i:\, C(i)=k} \|x_i - \bar{x}_k\|^2$$

Minimize the total distance given by each observation to the mean (centroid) of the cluster to which the observation is assigned.

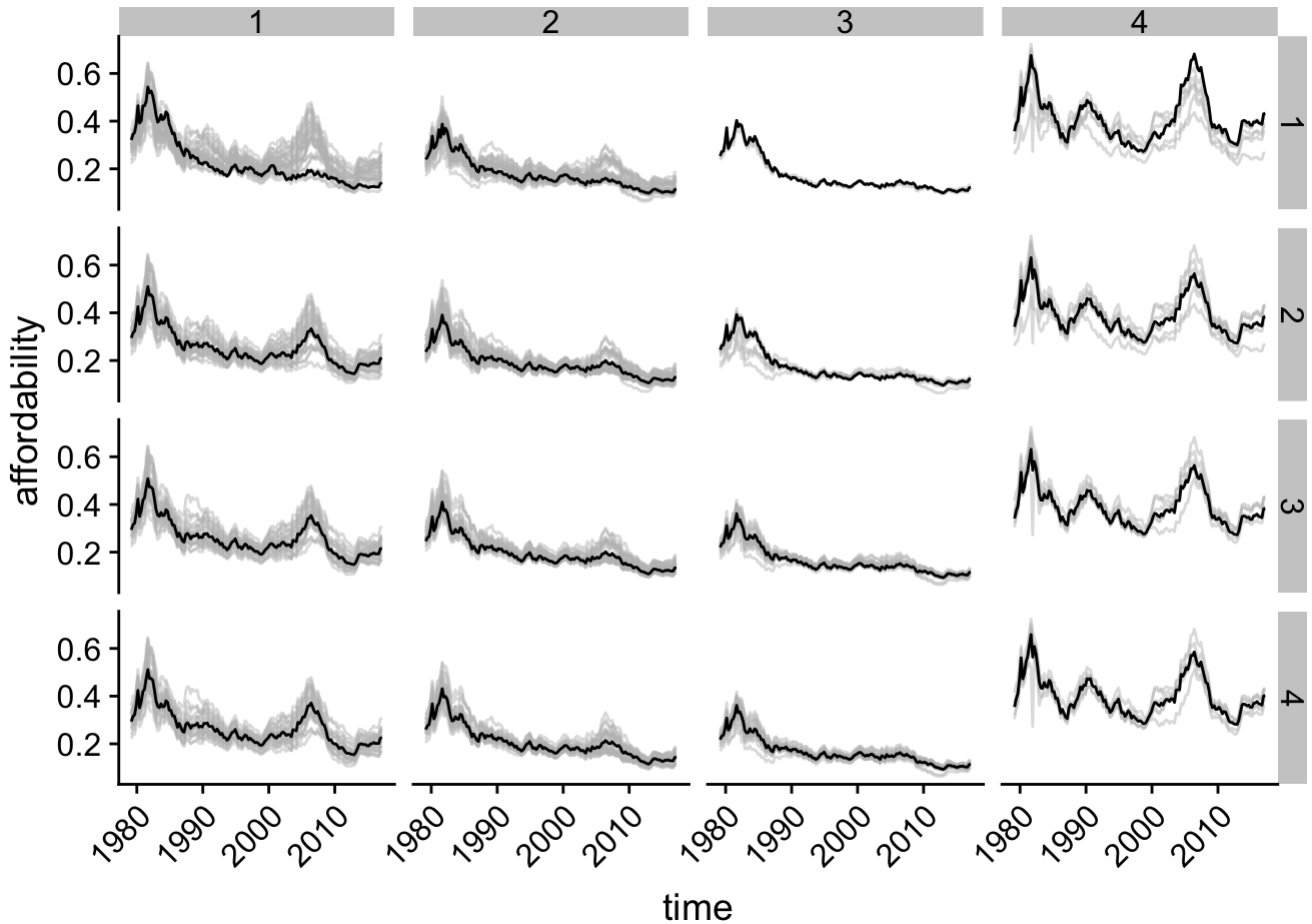
# K-means Clustering

An iterative algorithm is used to minimize this criterion

1. Initialize by choosing $K$ observations as centroids $m_1, m_2, \ldots, m_k$
2. Assign each observation $i$ to the cluster with the nearest centroid, i.e., set $C(i) = \arg\min_{1 \le k \le K} \|x_i - m_k\|^2$
3. Update centroids $m_k = \bar{x}_k$
4. Iterate steps 1 and 2 until convergence

# K-means Clustering

Here we illustrate the k-means algorithm over four iterations on our example data with $K = 4$.



# K-means Clustering

Criterion $W(C)$ is reduced in each iteration so the algorithm is assured to converge.

Not a convex criterion, the clustering we obtain may not be globally optimal.

In practice, the algorithm is run with multiple initializations (step 0) and the best clustering achieved is used.

# K-means Clustering

Also, selection of observations as centroids can be improved using the K-means++ algorithm:

1. Choose an observation as centroid $m_1$ uniformly at random
2. To choose centroid $m_k$, compute for each observation $i$ not chosen as a centroid the distance to the nearest centroid $d_i = \min_{1 \leq l < k} \|x_i - m_l\|^2$
3. Set centroid $m_k$ to an observation randomly chosen with probability $\frac{e_i^d}{\sum_{i'} e_{i'}^d}$
4. Iterate steps 1 and 2 until $K$ centroids are chosen

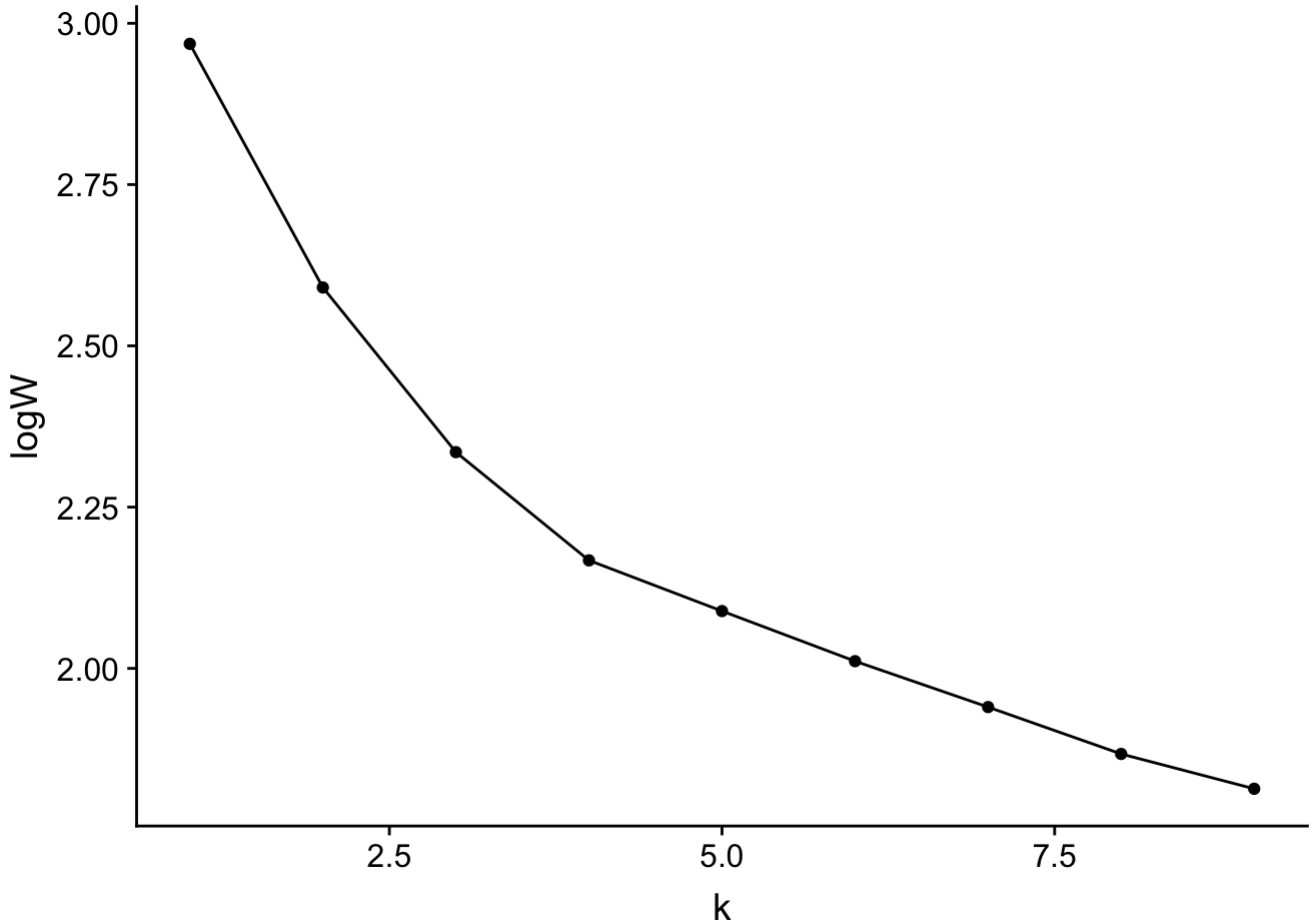
# Choosing the number of clusters

The number of parameters must be determined before running the K-means algorithm.

There is no clean direct method for choosing the number of clusters to use in the K-means algorithm (e.g. no cross-validation method)

# Choosing the number of clusters

Looking at criterion $W(C)$ alone is not sufficient as the criterion will become smaller as the value of $K$ is reduced.



# Choosing the number of clusters

We can use properties of this plot for ad-hoc selection.

Suppose there is a true underlying number $K^*$ of clusters in the data,

- improvement in the $W_K(C)$ statistic will be fast for values of $K \leq K^*$
- slower for values of $K > K^*$.

# Choosing the number of clusters

*Improvement in the $W_K(C)$ statistic will be fast for values of $K \leq K^*$*

In this case, there will be a cluster which will contain observations belonging to two of the true underlying clusters, and therefore will have poor within cluster similarity.

As $K$ is increased, observations may then be separated into separate clusters, providing a sharp improvement in the $W_K(C)$ statistic.

# Choosing the number of clusters

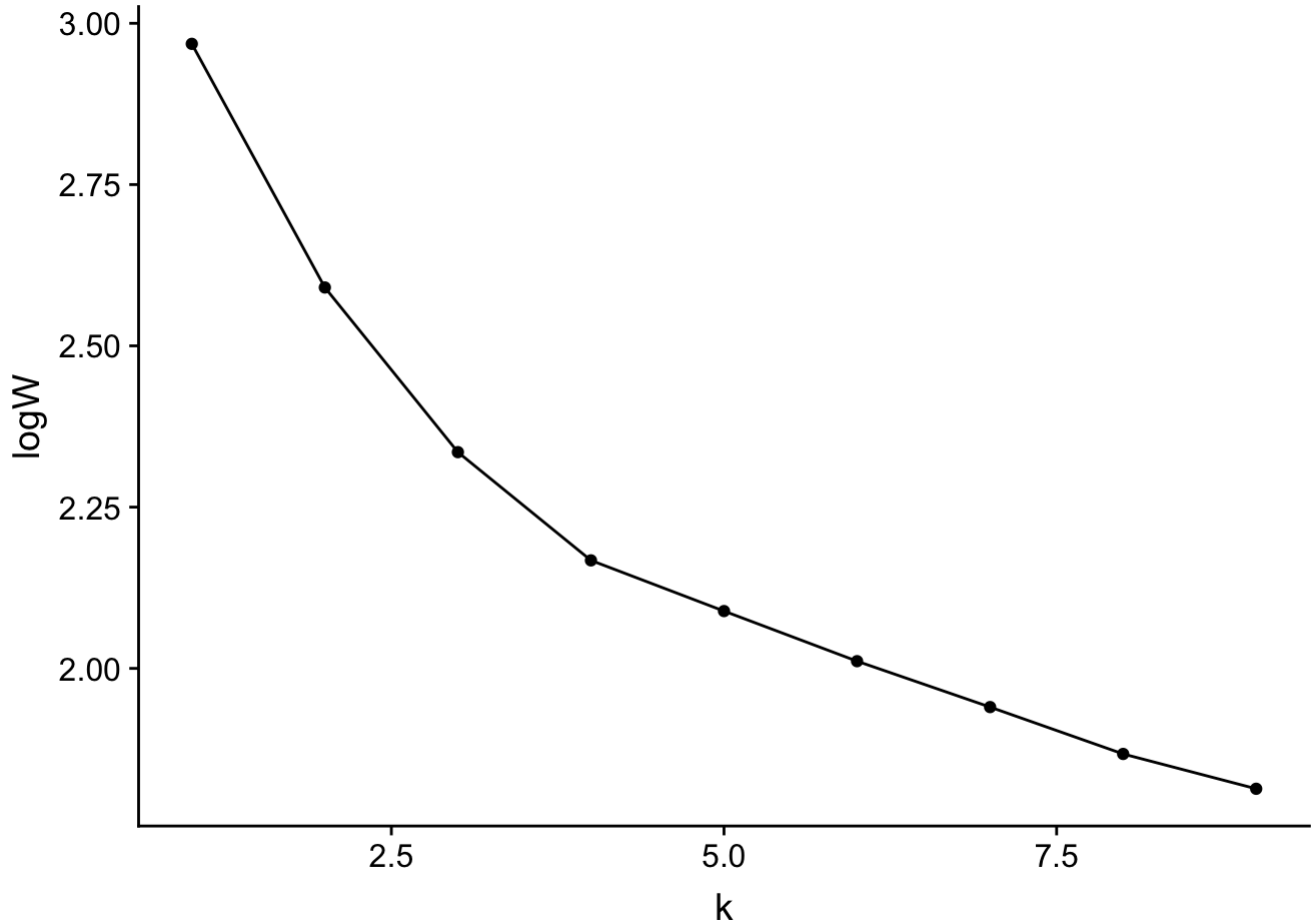
*Improvement in the $W_K(C)$ statistic will be slower for values of $K > K^*$*

In this case, observations belonging to a single true cluster are split into multiple cluster, all with generally high within-cluster similarity,

Splitting these clusters further will not improve the $W_K(C)$ statistic very sharply.

# Choosing the number of clusters

The curve will therefore have an inflection point around $K^*$.



# Choosing the number of clusters

The *gap statistic* is used to identify the inflection point in the curve.

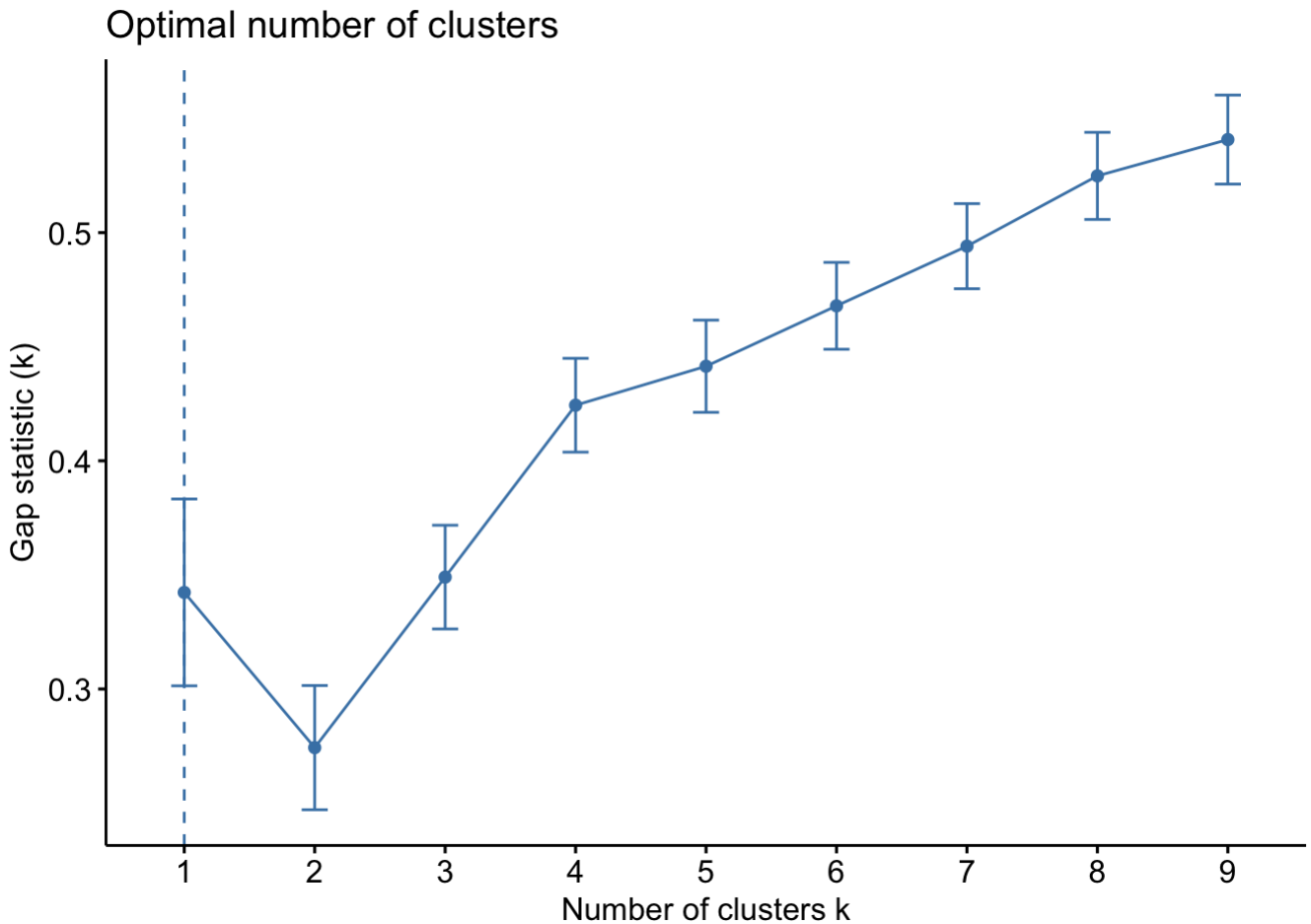
It compares the behavior of the $W_K(C)$ statistic based on the data with the behavior of the $W_K(C)$ statistic for data generated uniformly at random over the range of the data.

Chooses the $K$ that maximizes the gap between these two $W_K(C)$ curves.

# Choosing the number of clusters

For this dataset, the gap statistic suggests there is no clear cluster structure and therefore $K = 1$ is the best choice.

A choice of $K = 4$ is also appropriate.



# Soft K-means Clustering

Instead of the combinatorial approach of the $K$-means algorithm, take a more direct probabilistic approach to modeling distribution $Pr(X)$.

Assume each of the $K$ clusters corresponds to a multivariate distribution $Pr_k(X)$,

$Pr(X)$ is given by *mixture* of these distributions as $Pr(X) = \sum_{k=1}^{K} \pi_k Pr_k(X)$.

# Soft K-means Clustering

Specifically, take $Pr_k(X)$ as a multivariate normal distribution $f_k(X) = N(\mu_k, \sigma_k^2 I)$

and mixture density $f(X) = \sum_{k=1}^{K} \pi_k f_k(X)$.

# Soft K-means Clustering

Use Maximum Likelihood to estimate parameters

$$\theta = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, \pi_1, \ldots, \pi_K)$$

based on their log-likelihood

$$\ell(\theta; X) = \sum_{i=1}^{N} \log \left[ \sum_{k=1}^{K} \pi_k f_k(x_i; \theta) \right]$$

# Soft K-means Clustering

$$\ell(\theta; X) = \sum_{i=1}^{N} \log \left[ \sum_{k=1}^{K} \pi_k f_k(x_i; \theta) \right]$$

Maximizing this likelihood directly is computationally difficult

Use Expectation Maximization algorithm (EM) instead.

# Soft K-means Clustering

Consider unobserved latent variables $\Delta_{ik}$ taking values 0 or 1,

$\Delta_{ij} = 1$ specifies observation $x_i$ was generated by component $k$ of the mixture distribution.

# Soft K-means Clustering

Now set $Pr(\Delta_{ik} = 1) = \pi_k$,and assume we *observed* values for indicator variables $\Delta_{ik}$.

We can write the log-likelihood of our parameters in this case as

$$\ell_0(\theta; X, \Delta) = \sum_{i=1}^{N} \sum_{k=1}^{K} \Delta_{ik} \log f_k(x_i; \theta) + \sum_{i=1}^{N} \sum_{k=1}^{K} \Delta_{ik} \log \pi_k$$

# Soft K-means Clustering

Maximum likelihood estimates:

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \Delta_{ik} x_i}{\sum_{i=1}^{N} \Delta_{ik}}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \Delta_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \Delta_{ik}}$$

$$\hat{\pi}_k = \frac{\sum_{i=1}^{K}}{N}.$$

# Soft K-means Clustering

Of course, this result depends on observing values for $\Delta_{ik}$ which we don't observe. Use an iterative approach as well:

- given current estimate of parameters $\theta$,
- maximize

$$E(\ell_0(\theta'; X, \Delta)|X, \theta)$$

.

# Soft K-means Clustering

Of course, this result depends on observing values for $\Delta_{ik}$ which we don't observe. Use an iterative approach as well:

- given current estimate of parameters $\theta$,
- maximize

$$E(\ell_0(\theta'; X, \Delta)|X, \theta)$$

.

We can prove that maximizing this quantity also maximizes the likelihood we need $\ell(\theta; X)$.

# Soft K-means Clustering

In the mixture case, what is the function we would maximize?

Define

$$\gamma_{ik}(\theta) = E(\Delta_{ik}|X_i, \theta) = Pr(\Delta_{ik} = 1|X_i, \theta)$$

# Soft K-means Clustering

Use Bayes' Rule to write this in terms of the multivariate normal densities with respect to current estimates $\theta$:

$$\begin{aligned}\gamma_{ik} &= \frac{Pr(X_i|\Delta_{ik}=1)Pr(\Delta_{ik}=1)}{Pr(X_i)} \\ &= \frac{f_k(x_i;\mu_k,\sigma_k^2)\pi_k}{\sum_{l=1}^{K} f_l(x_i;\mu_l,\sigma_l^2)\pi_l}\end{aligned}$$

# Soft K-means Clustering

Quantity $\gamma_{ik}(\theta)$ is referred to as the *responsibility* of cluster $k$ for observation $i$, according to current parameter estimate $\theta$.

# Soft K-means Clustering

Then the expectation we are maximizing is given by

$$E(\ell_0(\theta'; X, \Delta)|X, \theta) = \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}(\theta)\log f_k(x_i;\theta') + \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}(\theta)\log \pi'_k$$

# Soft K-means Clustering

We can now give a complete specification of the EM algorithm for mixture model clustering.

1. Take initial guesses for parameters $\theta$
2. *Expectation Step*: Compute responsibilities $\gamma_{ik}(\theta)$
3. *Maximization Step*: Estimate new parameters based on responsibilities as below.
4. Iterate steps 1 and 2 until convergence

# Soft K-means Algorithm

Estimates in the Maximization step are given by

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \gamma_{ik}(\theta) x_i}{\sum_{i=1}^{N} \gamma_{ik}}$$

$$\hat{\sigma}_k^2 = \frac{\sum i = 1^{N} \gamma_{ik}(\theta)(x_i - \mu_k)}{\sum_{i=1}^{N} \gamma_{ik}(\theta)}$$

and

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} \gamma_{ik}(\theta)}{N}$$

# Soft K-means Algorithm

The name "soft" K-means refers to the fact that parameter estimates for each cluster are obtained by weighted averages across all observations.

# Summary

Clustering methods are intuitive methods useful to understand structure within unlabeled observations.

Model-based methods, using the EM algorithm, provide a large amount of flexibililty over simpler methods like the K-means algorithm.

Kernel-based clustering methods (Kernel K-means) permit the clustering of observations based on non-linear similarity functions.