K-Means and Mixture Models

COS 424/524, SML 302: Fundamentals of Machine Learning

Professor Engelhardt

COS 424/524, SML 302

Lecture 11

Unsupervised learning

Unsupervised learning is about finding structure in data.

Unsupervised learning is about uncovering patterns without knowing what we are looking for a priori.

We can use these patterns to *predict future samples* and to *discover meaningful structure* in the data.

Supervised versus unsupervised data

Machine learning was originally focused on classification and prediction.

Supervised learning is important and in the machine learning spotlight, but unsupervised learning plays an equally important, but often under-appreciated, role.

Why?

The INFORMATION EXPLOSION

Unsupervised learning problems

- Attorneys used to get file boxes of documents to read; Now they get gigabytes of emails. The trial is tomorrow. What is in these emails?
- We regularly search an indexed collection of documents. Queries might mean multiple things. Consider "Jaguar" as (a) an animal (b) a car and (c) an operating system. Given search results, can we identify these groups?
- Biologists run experiments, simultaneously testing many genes in cells after many different exposures. They want to recover sets of genes that respond differently across exposures. How can they do it?

The INFORMATION EXPLOSION

Unsupervised learning problems

- Neuroscientists run fMRI studies resulting in thousands of high-resolution images of brain activity, in a time series, while subjects are performing cognitive tasks. Which parts of the brain interact with each other?
- Historians collect reams of historical documents, scan them, and run them through OCR software. How can unsupervised learning help them with close reading and forming hypotheses?
- A reporter receives 5M emails from WikiLeaks. Where is the scoop?

The INFORMATION EXPLOSION

Unsupervised learning problems

- A physicist collects terabytes of measurements from the universe. What are the unexpected events? What should we examine?
- Bitcoin transactions are all recorded. Can patterns of activity be used to spot illegal trade?
- A new movie comes out on Netflix with no user ratings. Can we predict whether a user will like it or not?
- Using genotype data from many individuals, can I find regions of a genome that descend from distinct ancestral populations?
- Others?

Seeing observations in a different way

One way we use unsupervised learning is to help us see observations in a different way.

When observations happen at exceedingly large scales, we need methods for summarizing, grouping, and visualizing them, and we need methods for finding *unanticipated patterns*.

In this sense, it's like a microscope or, more accurately, a camera lens.

Unsupervised learning

Main idea: encode in the model a flexible structure that may exist in data, and let the algorithm find the particular instantiation of that structure.

Latent structure examples

- Latent patterns of brain activity corresponding to brain regions
- Groups of documents that are about a single event
- Collections of genes that respond similarly to a stimulus
- Groups of images that have the same person in them
- Sets of individuals with the same ancestry
- Users that share the same taste in movies

Unsupervised learning

We think such patterns exist, but we do not know what they are.

Much of machine learning research is about taking a new type of data, positing reasonable structures and patterns, and showing that the unsupervised learning algorithm does something that "makes sense" by finding appropriate labels and interpretations for those patterns.

What "makes sense" for each data domain is different, and tricky to quantify oftentimes.

Unsupervised learning framework

We have data $\mathcal{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$, or n samples each with values for p features. We do not have labels for these samples.

In *classification* and *regression*, we related sample features to a class label or a response.

In *clustering* and *dimension reduction* we find ways to relate sample features to each other.

Unsupervised learning as dimension reduction

Unsupervised learning may be thought of as "dimension reduction," taking high dimensional data and projecting it into a low dimensional space

Examples of patterns

- fMRI data: "typical" brain patterns
- Genotype data: shared genetic ancestry among individuals
- Google users: "types" of people
- Hospital data: previous patients with similar symptoms
- Music recommendations: music genres and moods

We can then work in the low dimensional space to, for example, make movie recommendations or find similar images.

Key point: we can only guess at what "type" of sample each dimension represents.

Dimension reduction

Dimension reduction has a number of desirable behaviors:

- summarizes the data in terms of patterns among the features, and how each sample expresses those patterns.
- smooths feature representations, e.g., “cat” and “feline” might be indistinguishable in the lower dimensional space
- compresses the data along representative dimensions.

Difficulties with unsupervised learning

- It is hard to measure success.
- Unsupervised learning objectives are often not well defined,
- ...but the goals are critically important.

Examples of unsupervised learning

- visualizing data;
- cataloging variation;
- feature selection and summarization;
- dimension reduction to remove noise or technical artifacts.

Unsupervised learning: Clustering

- *Idea:* partition data into groups of similar samples using ML methods
- Clustering is useful for:
  - Organizing data
  - Exploring hidden structure in data
  - Representing high-dimensional data in terms of one of K clusters

Examples of clustering applications

Clustering applications:

- Customers according to purchase histories
- Genes according to expression profile
- Search results according to topic
- Facebook users according to interests
- Images in a museum catalog according to image similarity
- Image pixels according to RGB color

Clustering: goal

Goal: partition data $\mathcal{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ into K groups.

- Each vector $\mathbf{x_i}$ is a vector of p features.
- Each vector $\mathbf{x_i}$ is associated with exactly one group.
- What should K be?
- Automatically choosing K can be complicated.

Clustering algorithm: K-means

Basic intuition behind the K-means algorithm:

- Associate each cluster with location in feature space (*a centroid*)
- Iterate until convergence:
  1. assign each sample to one cluster
  2. update each cluster centroid using only the assigned samples

Clustering algorithm: K-means

Practically, there are a few questions before this is possible:

- How do I assign a sample to a cluster?
- How do I update the cluster centroids?

K-means: distance functions

How do I assign a sample to a cluster?

For a p dimensional sample $\mathbf{x_i}$, assign it to the cluster whose p dimensional centroid η_k is the smallest distance from the sample.

Example distance functions

- Euclidean distance:

$$d(x_i, \eta_k) = \sqrt{\sum_{j=1}^{p} (x_{i,j} - \eta_{k,j})^2}$$

- A p-norm

$$d(x_i, \eta_k) = \left(\sum_{j=1}^{q} (x_{i,j} - \eta_{k,j})^p \right)^{1/p}$$

K-means: distance functions

How do I assign a sample to a cluster?

Example distance functions

- Mahalanobis distance

$$d(x_i, \eta_k) = \sqrt{\sum_{j=1}^{p} \frac{(x_{i,j} - \eta_{k,j})^2}{\sigma_j^2}}$$

- Any kernel function (similarity)

For the rest of this lecture, let's choose Euclidean distance.

K-means: centroid updates

How do I update the cluster centroids?

Set each cluster centroid to the empirical mean for all points assigned to that cluster.

Does this step remind you of a previous method we have studied?

K-means: pick the initial locations of the cluster centroids

Many ways to do this too.

To choose an initial centroid, you might

- Pick K samples at random from your data set.
- Sample from the data space uniformly at random.
- Average two or more samples from your data set chosen at random.

Wise to restart this process multiple times to assess robustness of clustering to starting point.

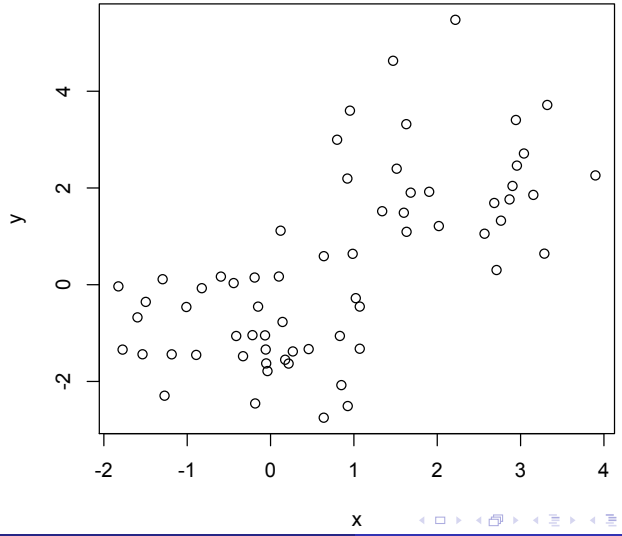
K-means algorithm

K-means clustering algorithm

- Given samples $\mathcal{D} \in \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$, and a number of clusters K
- Initialize K p-dimensional centroids
- Iterate until sample assignments do not change:
  1. Assign each point to closest of K cluster centroids according to distance function $d(\cdot, \cdot)$
  2. Recalculate cluster centroids as empirical mean of only samples assigned to that cluster.

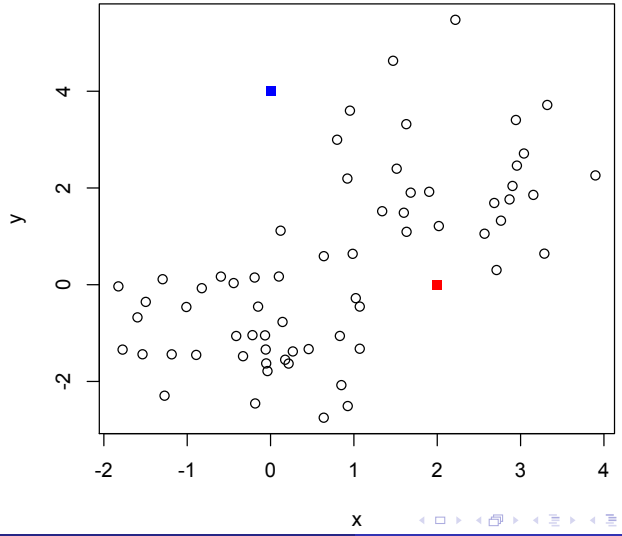
K-means clustering example

Start with unlabeled samples ($p = 2$)



K-means clustering example

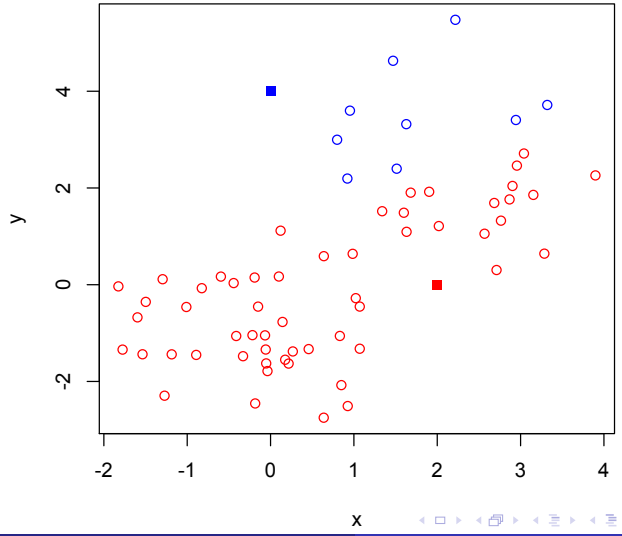
Initialize two random centroids for two clusters in this space.



K-means clustering example

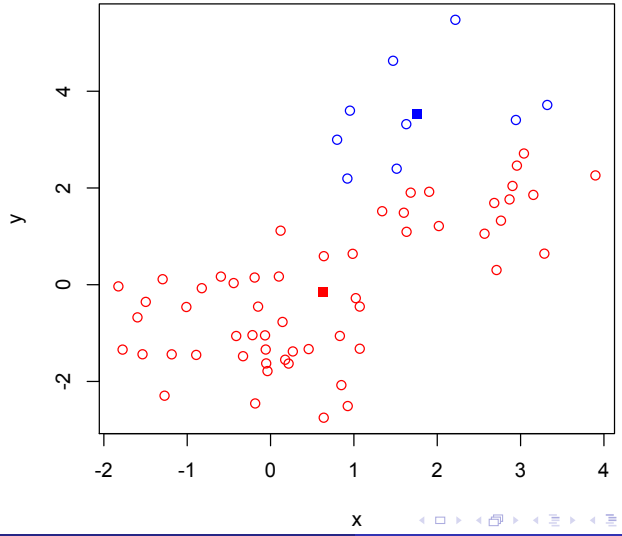
Assign each point to one of the two clusters by finding the centroid with the smallest Euclidean distance.

In this figure, the colors represent the cluster assignments.



K-means clustering example

Next, set the new cluster centroids to the mean of the locations of only those points assigned to that cluster.

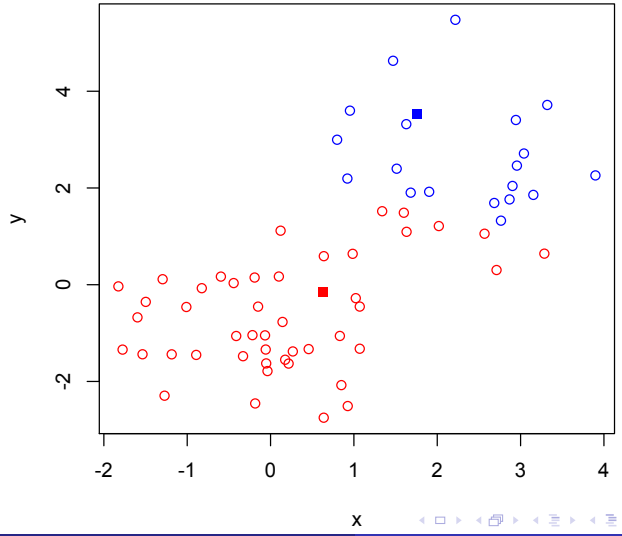


K-means clustering example

Repeat this process.

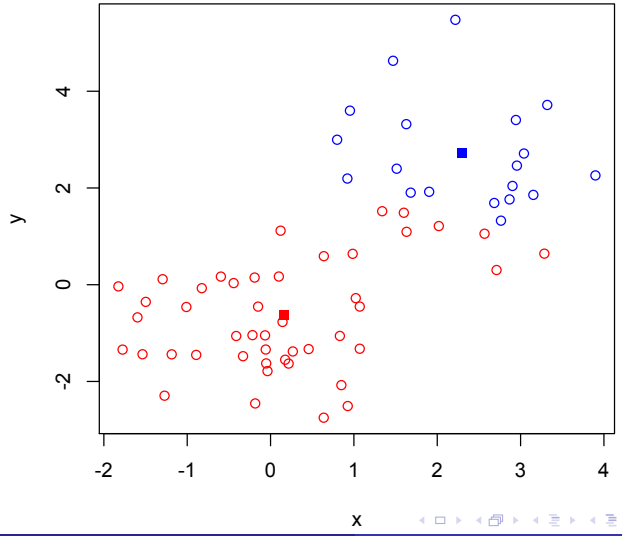
Assign each point to one of the two clusters by finding the centroid with the smallest Euclidean distance.

In this figure, the colors represent the cluster assignments.



K-means clustering example

Set the new cluster centroids to the mean of the locations of only those points assigned to that cluster.

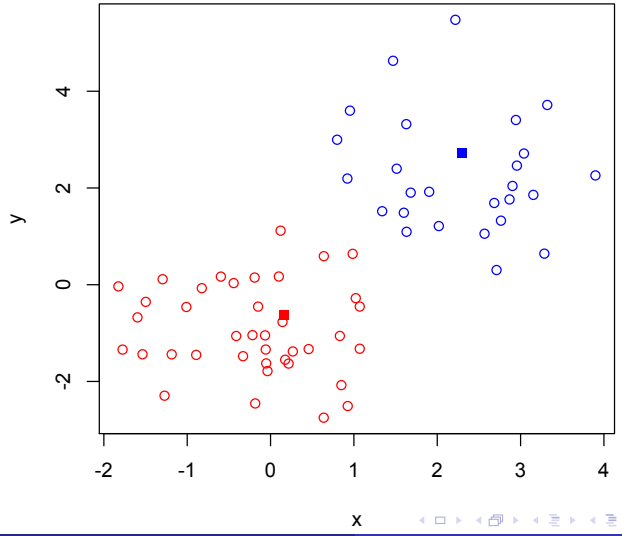


K-means clustering example

Repeat this process.

Assign each point to one of the two clusters by finding the centroid with the smallest Euclidean distance.

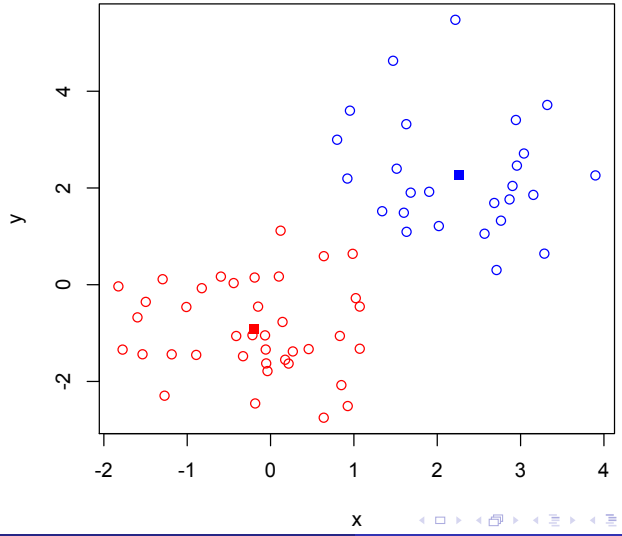
In this figure, the colors represent the cluster assignments.



K-means clustering example

Set the new cluster centroids to the mean of the locations of only those points assigned to that cluster.

Notice that the clusters centroids have not changed, and the same points will be assigned to each cluster. We can stop iterations.



K-means clustering example

Examples of K-means clustering examples at:
`www.cs.princeton.edu/~bee/demos.html`

K-means convergence

The K-means algorithm provably converges under some assumptions.

Sketch of proof

Two phases of K-means algorithm (updating cluster assignments, updating cluster centroids) is Newton's method for optimizing the *quantization measure* (or ℓ_2 norm) of each point from its (one) assigned cluster.

We are guaranteed that, for each update, the objective function decreases monotonically, so K-means converges to a local minimum.

See *Pattern Recognition and Machine Learning*, Chapter 9, or *[Bottou & Bengio 1994]*.

K-means clustering analysis

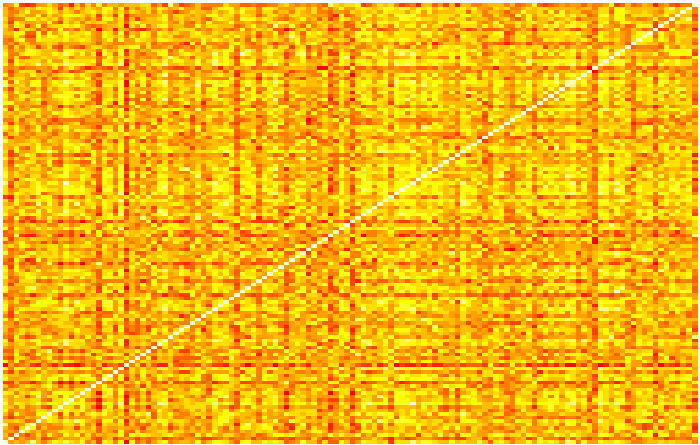
We can use K-means to cluster our course data.

- Cluster individuals based on their movie ratings
- Cluster movies based on their ratings

How do we assess the information contained in these clusters?

Cluster individuals by their movie ratings

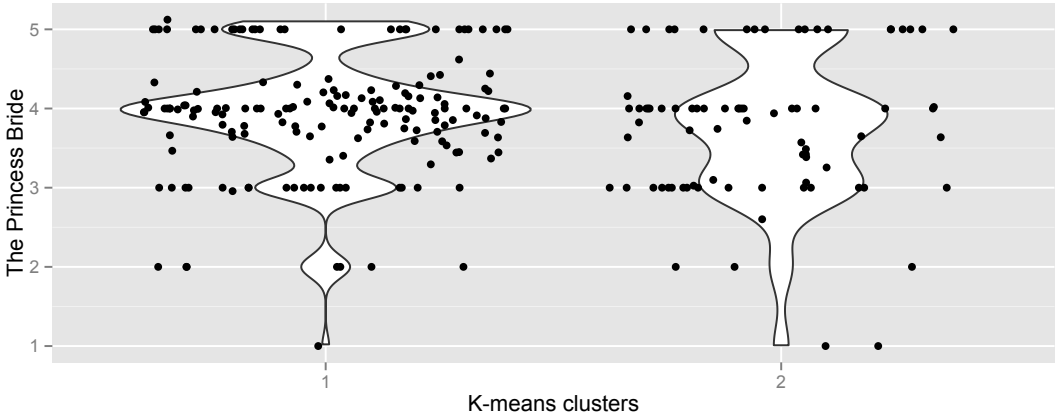
First, let's cluster individuals based on their (imputed) movie ratings.



Starting with $K = 2$, how can we understand the clusters?

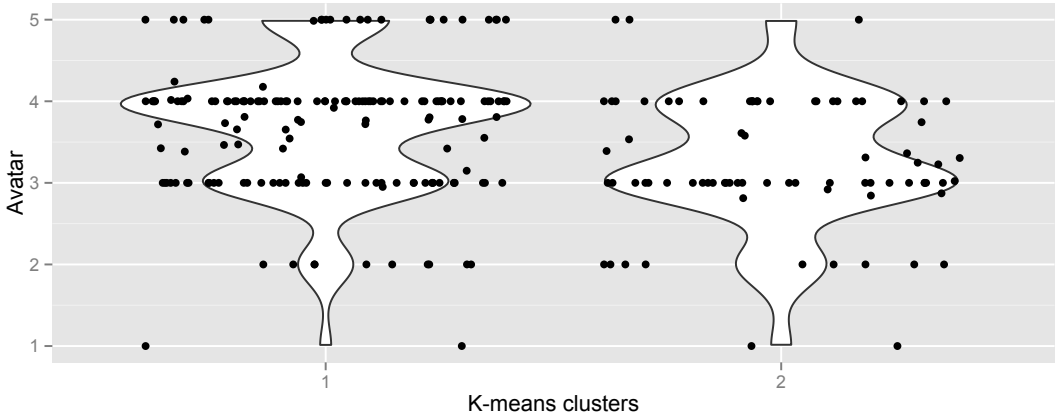
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, how does each cluster rate *The Princess Bride*?



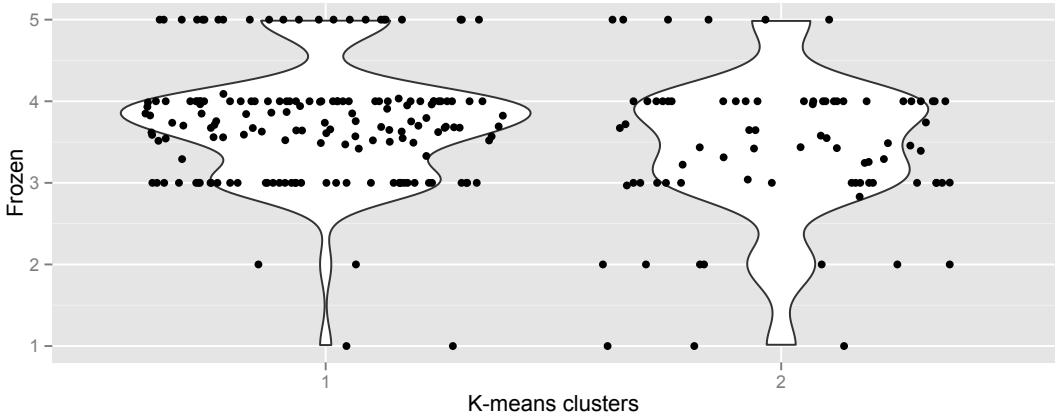
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, how does each cluster rate *Avatar*?



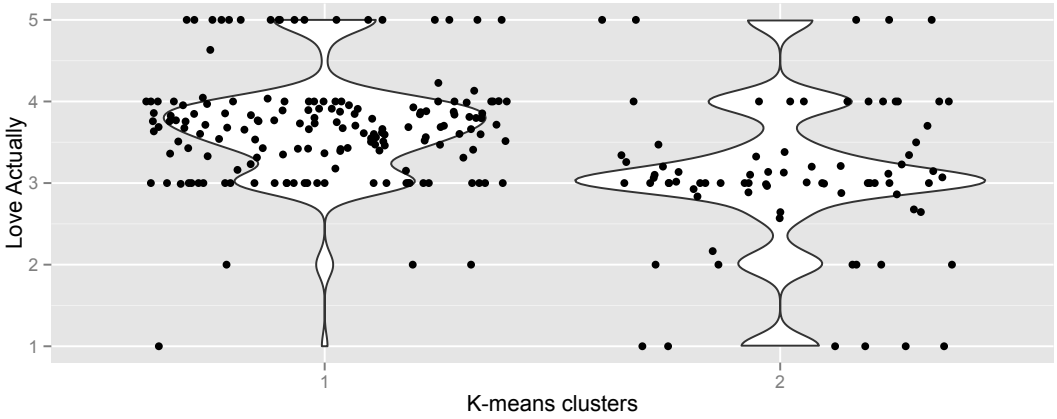
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, how does each cluster rate *Frozen*?



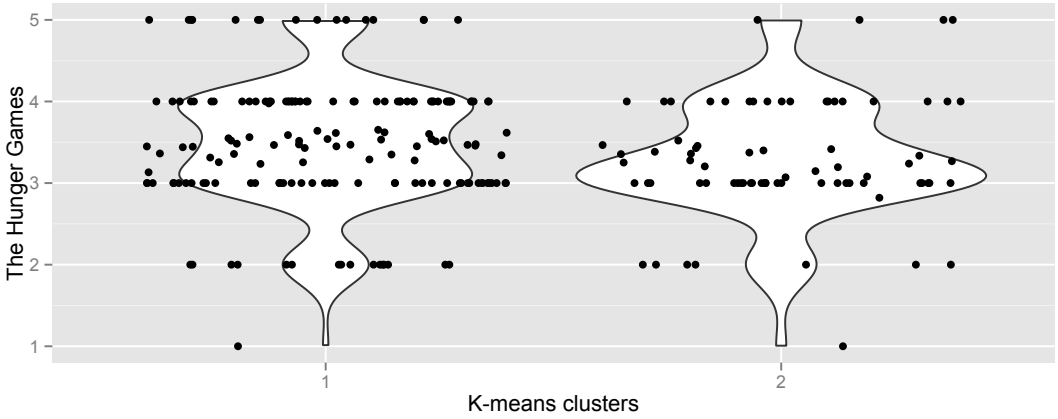
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, how does each cluster rate *Love Actually*?



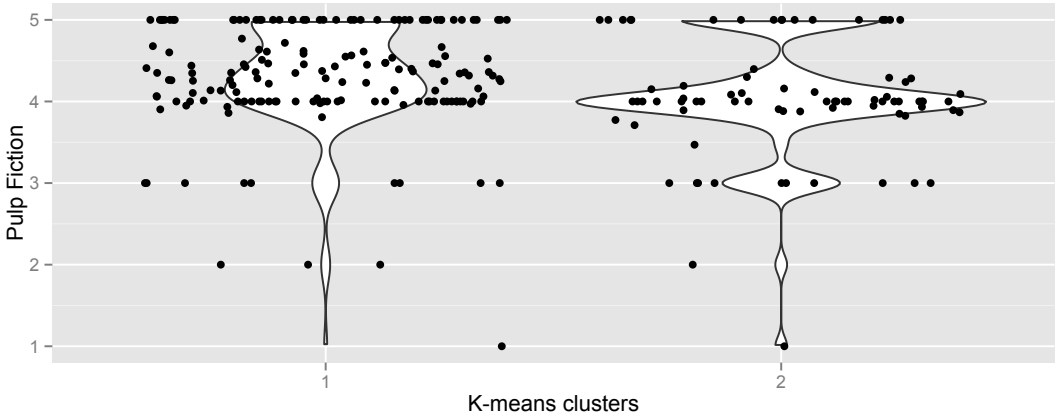
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, how does each cluster rate *Hunger Games*?



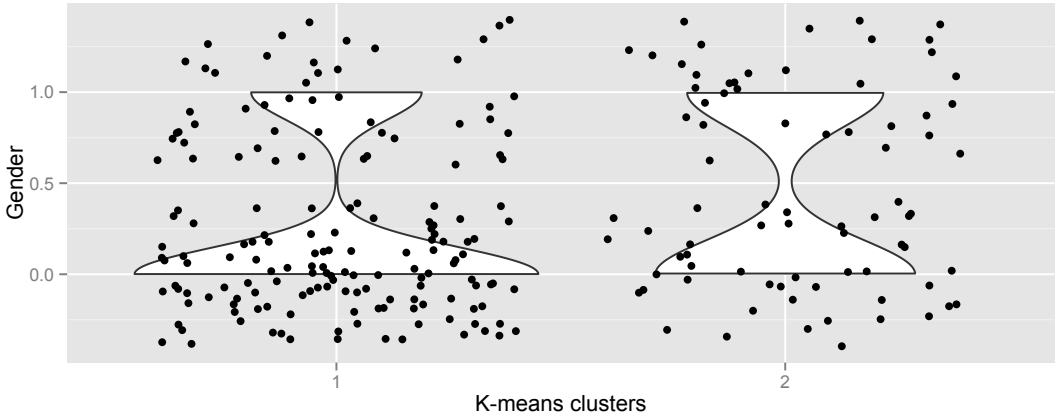
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, how does each cluster rate *Pulp Fiction*?



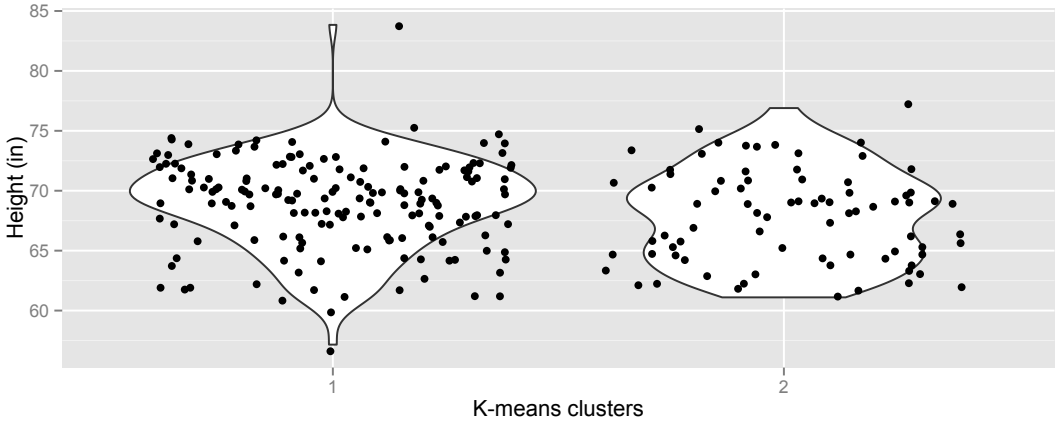
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, do we see clusters corresponding to males or females?



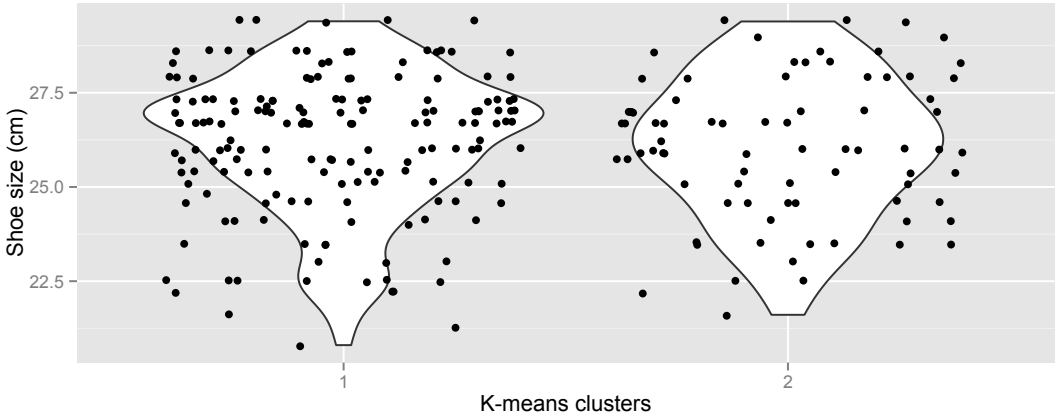
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, do we see clusters corresponding to height?



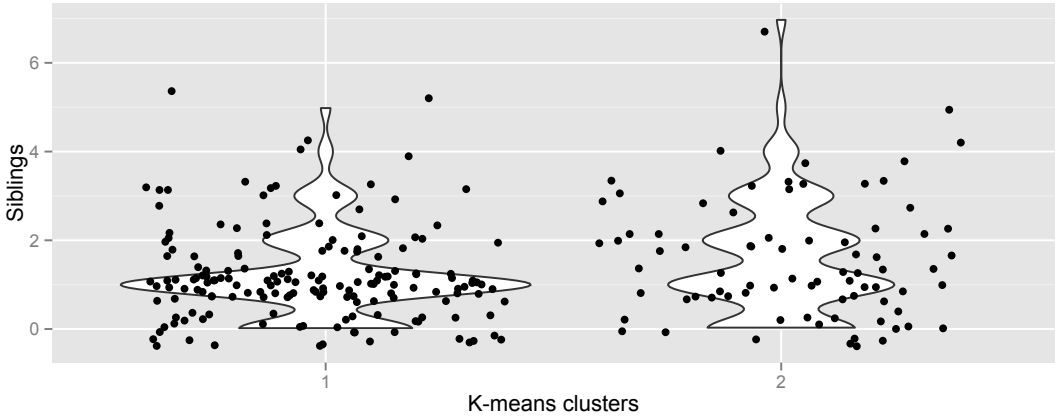
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, do we see clusters corresponding to shoe size?



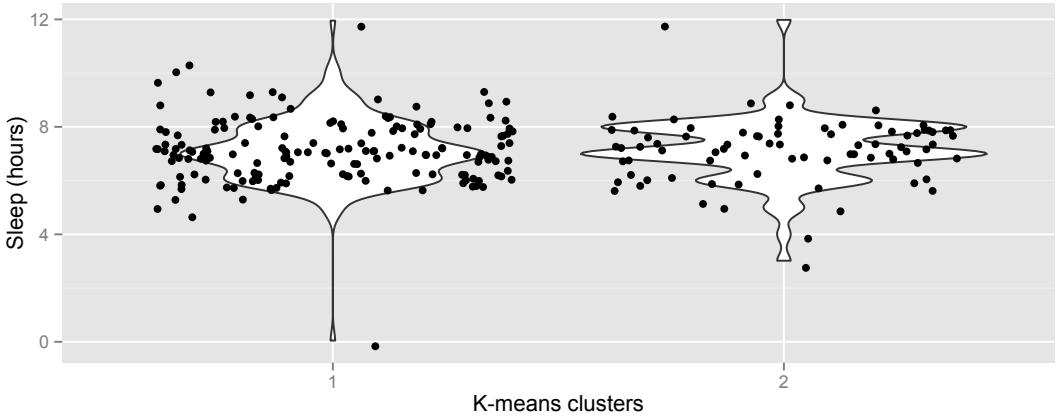
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, do we see clusters corresponding to number of siblings?



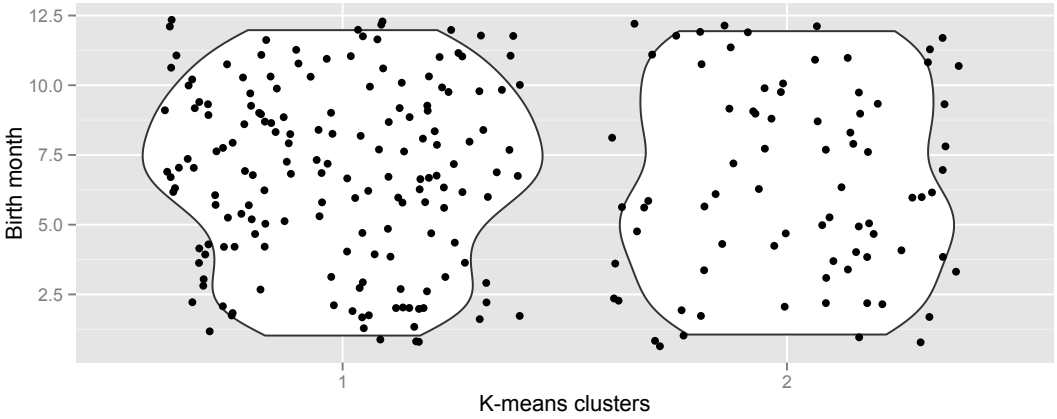
Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, do we see clusters corresponding to average sleep?



Cluster individuals by their movie ratings, $K = 2$

With $K = 2$, do we see clusters corresponding to birth month?



Cluster individuals by their movie ratings, $K = 2$

What did we learn from this analysis?

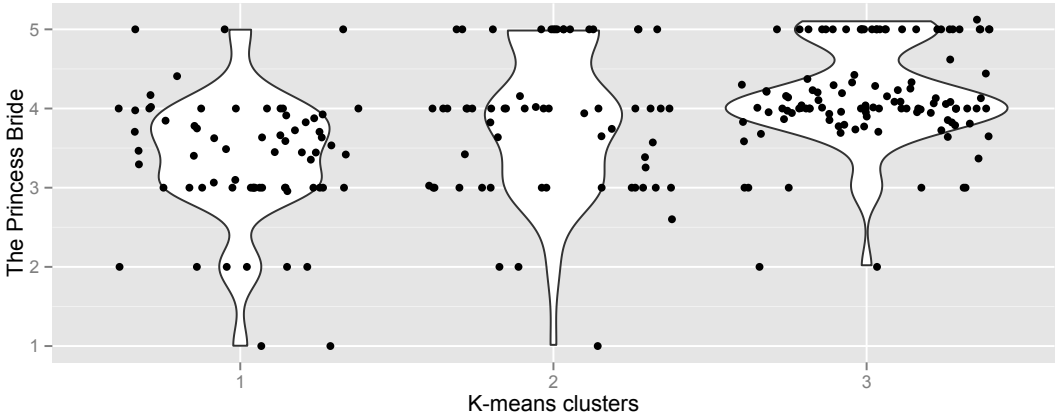
When $K = 2$, clusters of individuals based on movie ratings generally correspond to:

- Cluster 1: people who generally like mainstream movies, and really liked *Pulp Fiction*
- Cluster 2: people who don't like mainstream movies consistently, and didn't care for *Pulp Fiction*

It was hard to distinguish the clusters from other individual-specific characteristics.

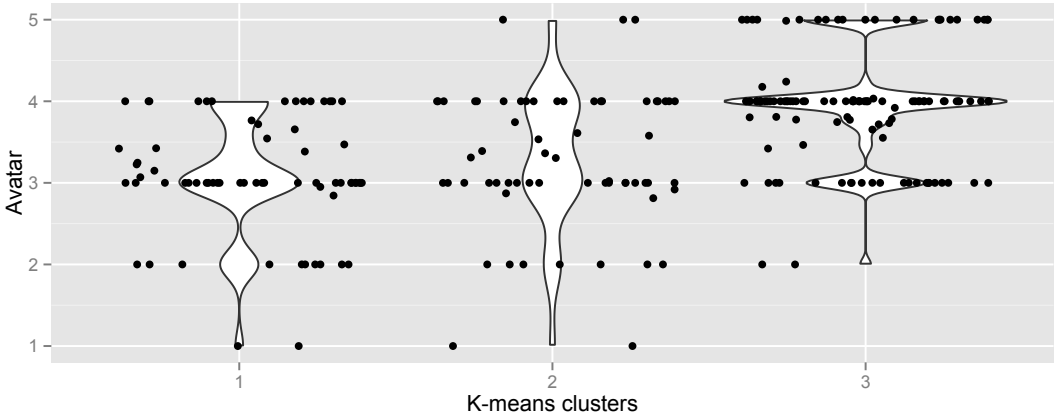
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, how does each cluster rate *The Princess Bride*?



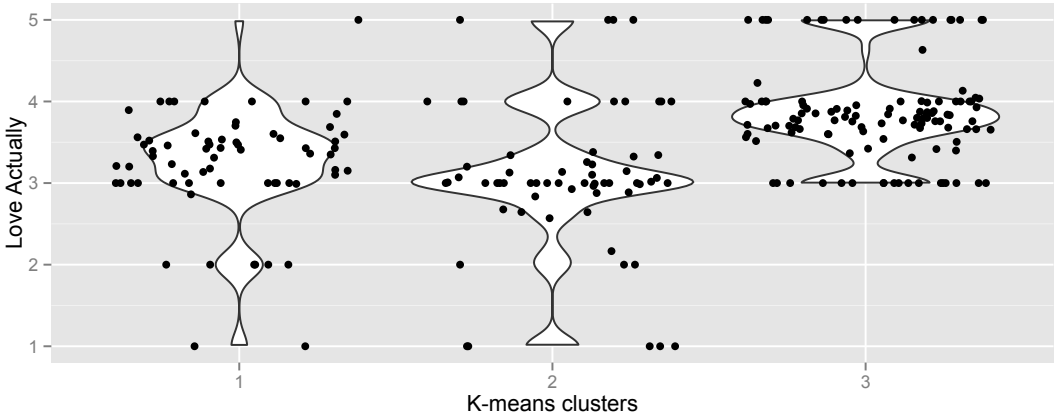
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, how does each cluster rate *Avatar*?



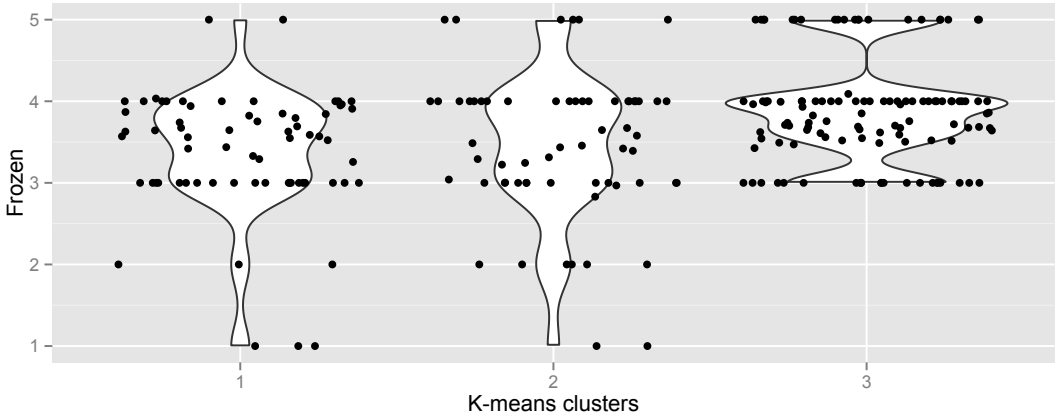
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, how does each cluster rate *Love Actually*?



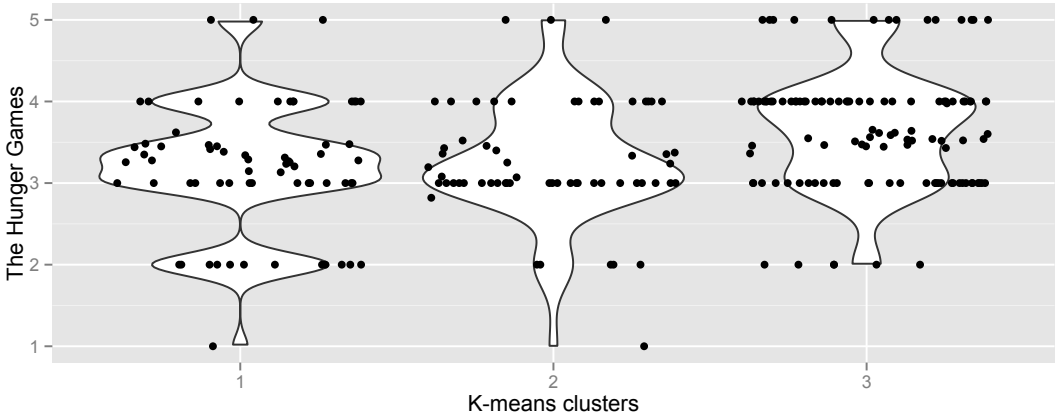
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, how does each cluster rate *Frozen*?



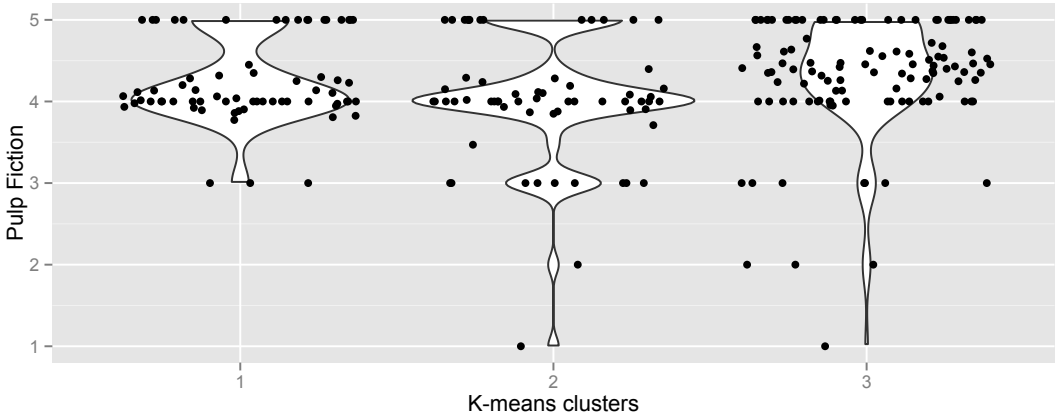
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, how does each cluster rate *Hunger Games*?



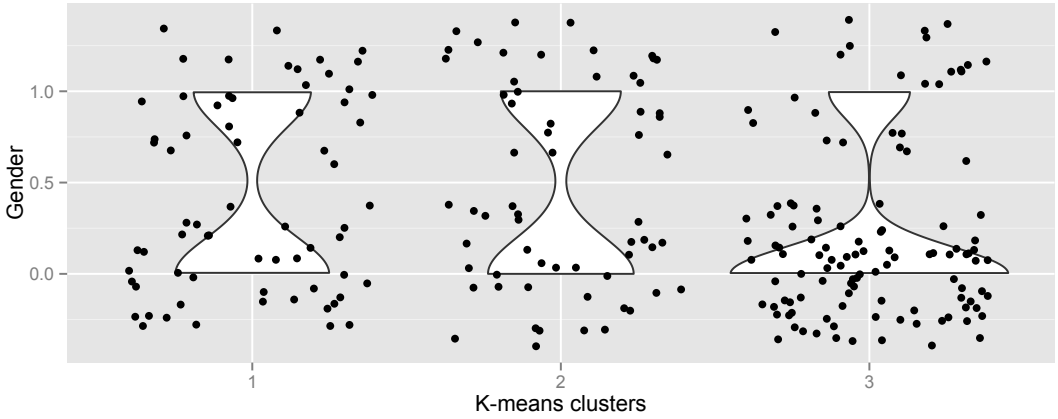
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, how does each cluster rate *Pulp Fiction*?



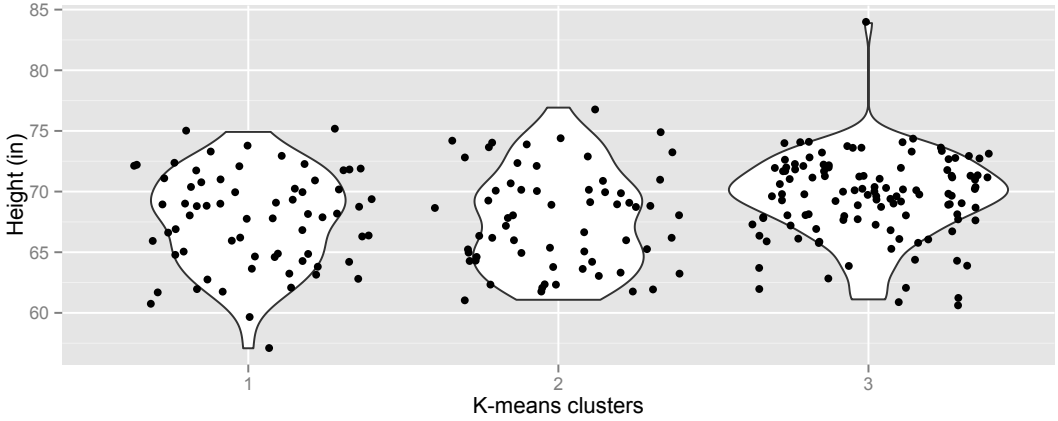
Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, do we see clusters corresponding to males or females?



Cluster individuals by their movie ratings, $K = 3$

With $K = 3$, do we see clusters corresponding to height?



Cluster individuals by their movie ratings, $K = 3$

What did we learn from this analysis?

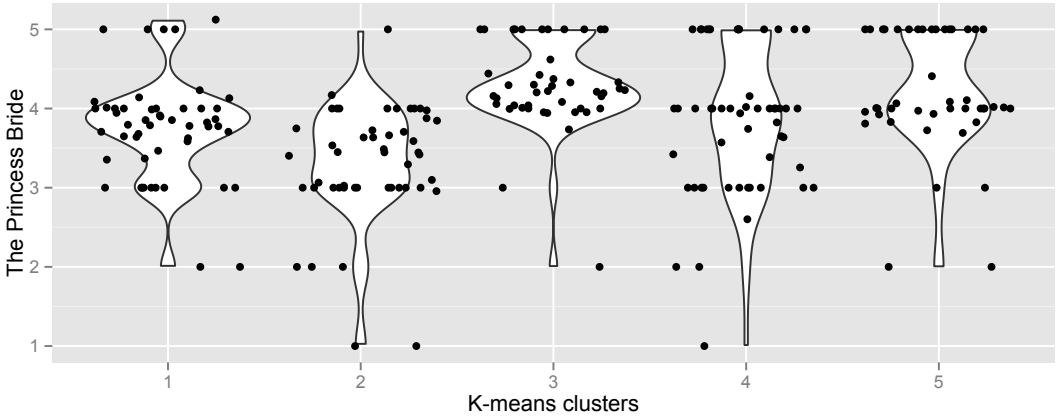
When $K = 3$, clusters of individuals based on movie ratings generally correspond to:

- Cluster 1: people who generally disliked mainstream movies, but liked *Pulp Fiction*
- Cluster 2: people who generally liked mainstream movies, but didn't care for *Pulp Fiction*;
- Cluster 3: people who loved Avatar, Love Actually, Frozen; disproportionally male.

Slight ability to distinguish the clusters from other individual-specific characteristics.

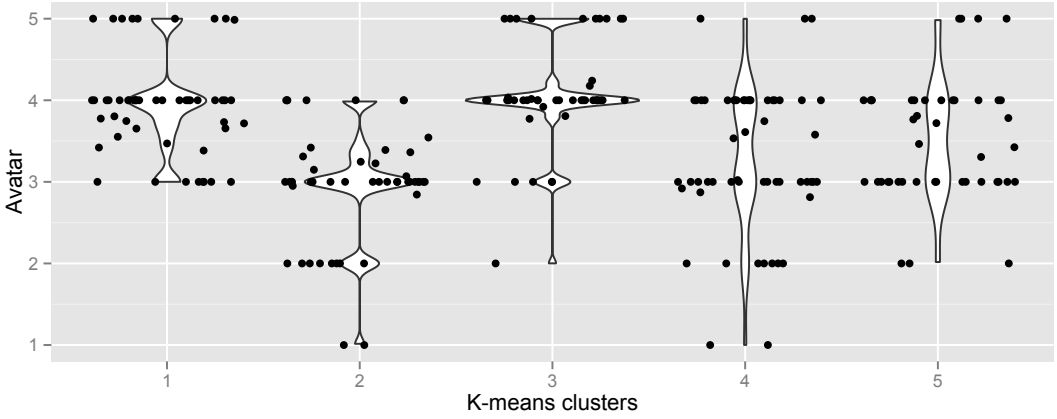
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, how does each cluster rate *The Princess Bride*?



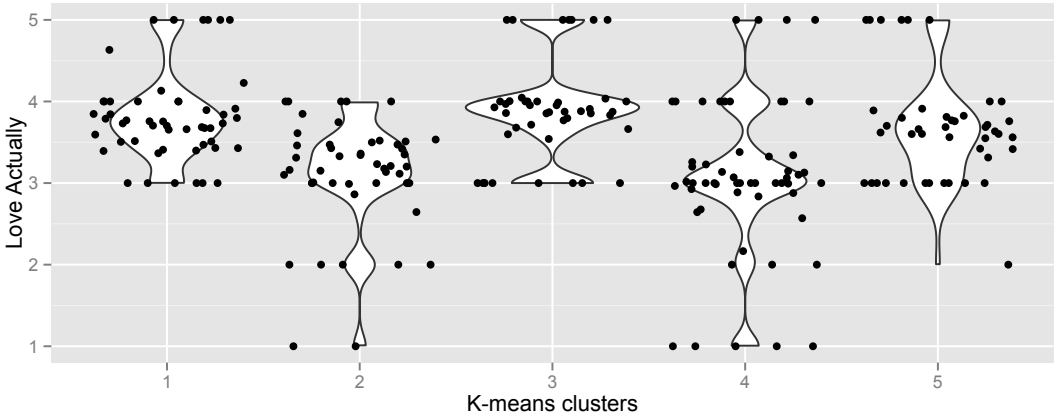
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, how does each cluster rate *Avatar*?



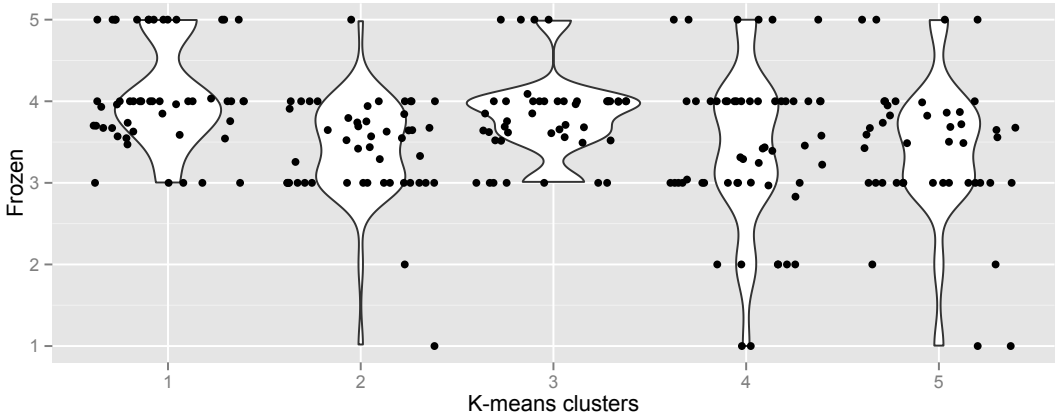
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, how does each cluster rate *Love Actually*?



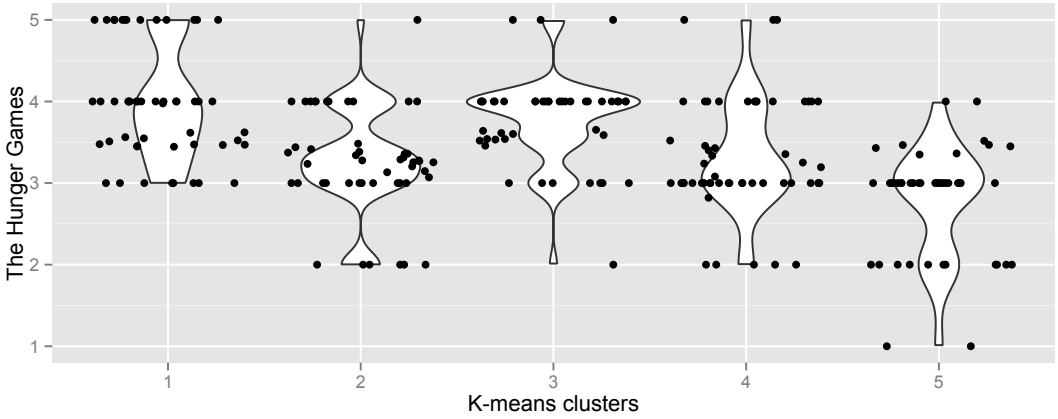
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, how does each cluster rate *Frozen*?



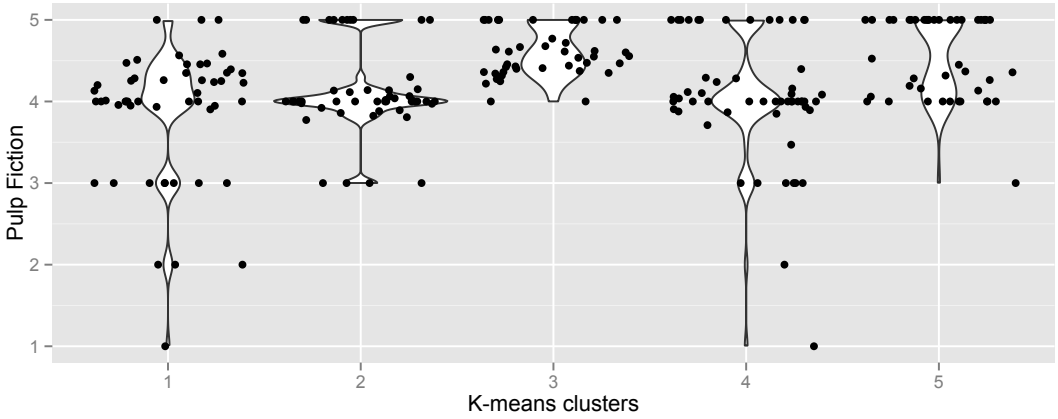
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, how does each cluster rate *Hunger Games*?



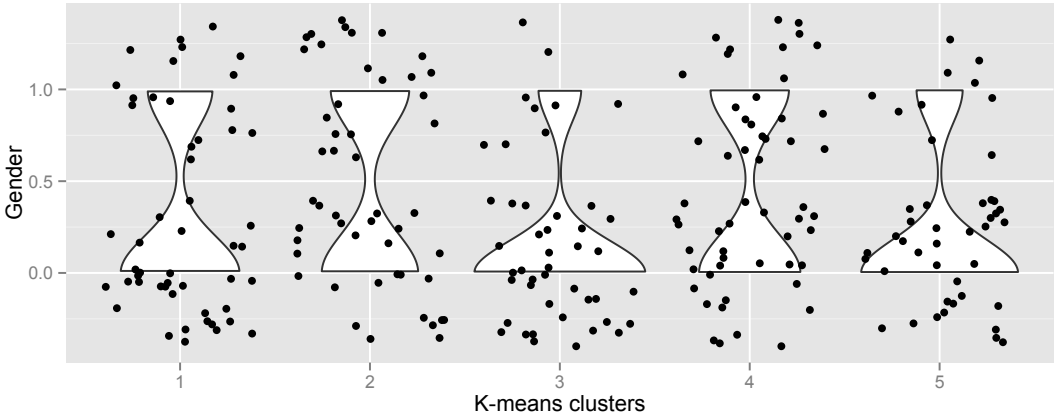
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, how does each cluster rate *Pulp Fiction*?



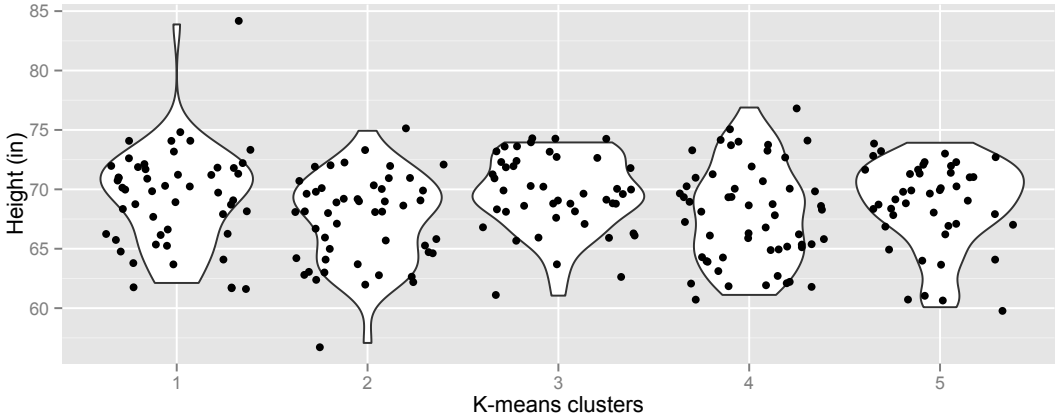
Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, do we see clusters corresponding to males or females?



Cluster individuals by their movie ratings, $K = 5$

With $K = 5$, do we see clusters corresponding to height?



Cluster individuals by their movie ratings, $K = 5$

What did we learn from this analysis?

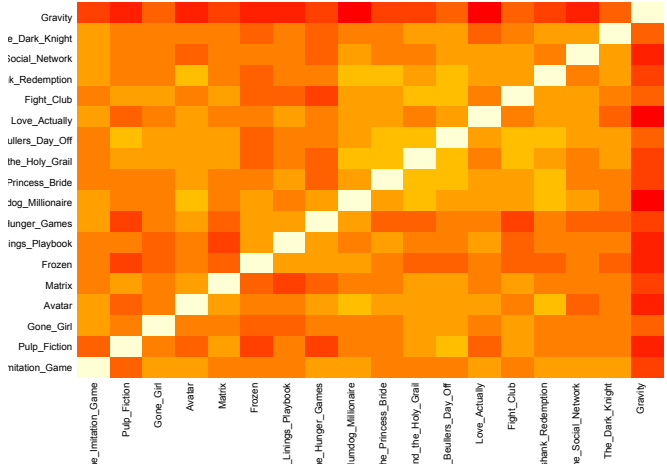
When $K = 5$, clusters of individuals from movie ratings correspond to:

- Cluster 1: people who loved Avatar, Frozen, and Hunger Games, but disliked *Pulp Fiction*;
- Cluster 2: people who were indifferent to most mainstream movies, hated Love Actually; disproportionally female
- Cluster 3: people who really liked Love Actually, loved *Pulp Fiction*; disproportionally male.
- Cluster 4: people who disliked mainstream movies, including *Pulp Fiction*, but liked *Princess Bride*; disproportionally female
- Cluster 5: people disliked mainstream movies, but loved *Pulp Fiction*; disproportionally male

Difficult to distinguish clusters from individual-specific characteristics.

Cluster movies by their ratings

Next, let's cluster movies based on their (imputed) movie ratings.



Starting with $K = 2$, how can we understand the clusters?

Cluster individuals by their movie ratings, $K = 3$

When $K = 3$, clusters of movies based on movie ratings:

$K = 3$ movie clusters

- Cluster 1: Gravity
- Cluster 2: The Imitation Game, Pulp Fiction, Gone Girl, Matrix, The Princess Bride, Monty Python and the Holy Grail, Ferris Bueller's Day Off, Fight Club, Shawshank Redemption, The Dark Knight
- Cluster 3: Avatar, Frozen, Silver, Linings Playbook, The Hunger Games, Slumdog Millionaire, Love Actually, The Social Network

Cluster individuals by their movie ratings, $K = 5$

When $K = 5$, clusters of movies based on movie ratings:

$K = 5$ movie clusters

- Cluster 1: The Dark Knight
- Cluster 2: Avatar, Frozen, Silver Linings Playbook, The Hunger Games, Love Actually
- Cluster 3: The Imitation Game, Pulp Fiction, Matrix, Monty Python and the Holy Grail, Fight Club, Shawshank Redemption
- Cluster 4: Gravity
- Cluster 5: Gone Girl, Slumdog Millionaire, The Princess Bride, Ferris Bueller's Day Off, The Social Network

No guarantees that subsequent K will split existing clusters evenly.

K-means clustering: issues raised in movie rating analysis

- Problem: selecting K
- Problem: label switching
- Problem: analyzing quality of clusters
- Problem: labeling clusters
- Problem: disjoint clusters

K-means clustering: Selecting K

How do we select the most appropriate K?

- Choose a number of K and run them all; look at results.
- Use a hierarchical clustering approach
- Use a non-parametric approach (infinite mixture models)
- Consider tree-like representations (when possible) of subsequent K partitions of data

K-means clustering: Label switching

How can we compare clusters across runs or different K?

- Compare the distances of centroids across clusterings
- Align each cluster in one clustering with its closest cluster in another

K-means clustering: Analyzing quality of clusters

How do we determine what patterns were identified in the cluster assignments?

- Use "held out" information to find enrichment in clusters
- Use included features to find enrichment in clusters
- Try different approaches (clustering, starting points, distance functions) and compare
- Metric: frequency that each pair of samples is clustered together across all clusterings

K-means clustering: labeling clusters

How can we describe the "type" of sample in each cluster?

- Use "held out" information to find enrichment in clusters
- Use included features to find enrichment in clusters
- This is a very manual process!

K-means clustering: disjoint clusters

What if samples are not one "type," but a combination of many types?

- K-means does not allow partial cluster membership, or samples in more than one cluster.
- We will learn in the next class about soft cluster membership assignment
- We will learn in a later lecture about topic models, or *admixture models*, that allow proportional membership in many clusters.

K-means clustering: extensions

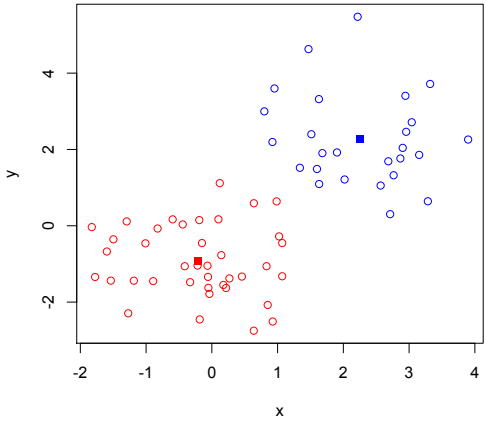
Variation on the k-means method:

- make the cluster assignments *soft*, or probabilistic.
- make the clusters multivariate Gaussian distributions (or an arbitrary distribution).
- weight a cluster assignment by the number of samples assigned to that cluster (recall naive Bayes).

In the next class, we will look at a generative model of this type.

K-means clustering: summary

- Simplest way to cluster data into K groups;
- Choose an initial set of centroids, K, and a distance function;
- Iterate between assigning each point to a cluster and updating the cluster definitions;
- Conditional on cluster assignment, K-means has the look of a supervised classification problem.



Additional Resources

- MLAPA: Chapter 11.4
- *Pattern Recognition and Machine Learning*: Chapter 9.1

- Metacademy: *K-means*
- Visualizing K-means clustering (Naftali Harris)
- StackOverflow: *Kmeans without knowing the number of clusters?*