Lecture 11 – Clustering, Introduction to Probability

DSC 40A, Fall 2021 @ UC San Diego

Suraj Rampure, with help from **many others**

Announcements

- Homework 4 due **tonight at 11:59pm**.
  - Remember to submit Survey 4 after finishing!

→ office hours today! Check calendar.

- Groupwork 5 due **Thursday 11/4 at 11:59pm**.
  - Discussion is back to being in-person, Wednesdays 6-6:50pm in Center Hall 113.

- Homework 5 due **Monday 11/8 at 11:59pm**.

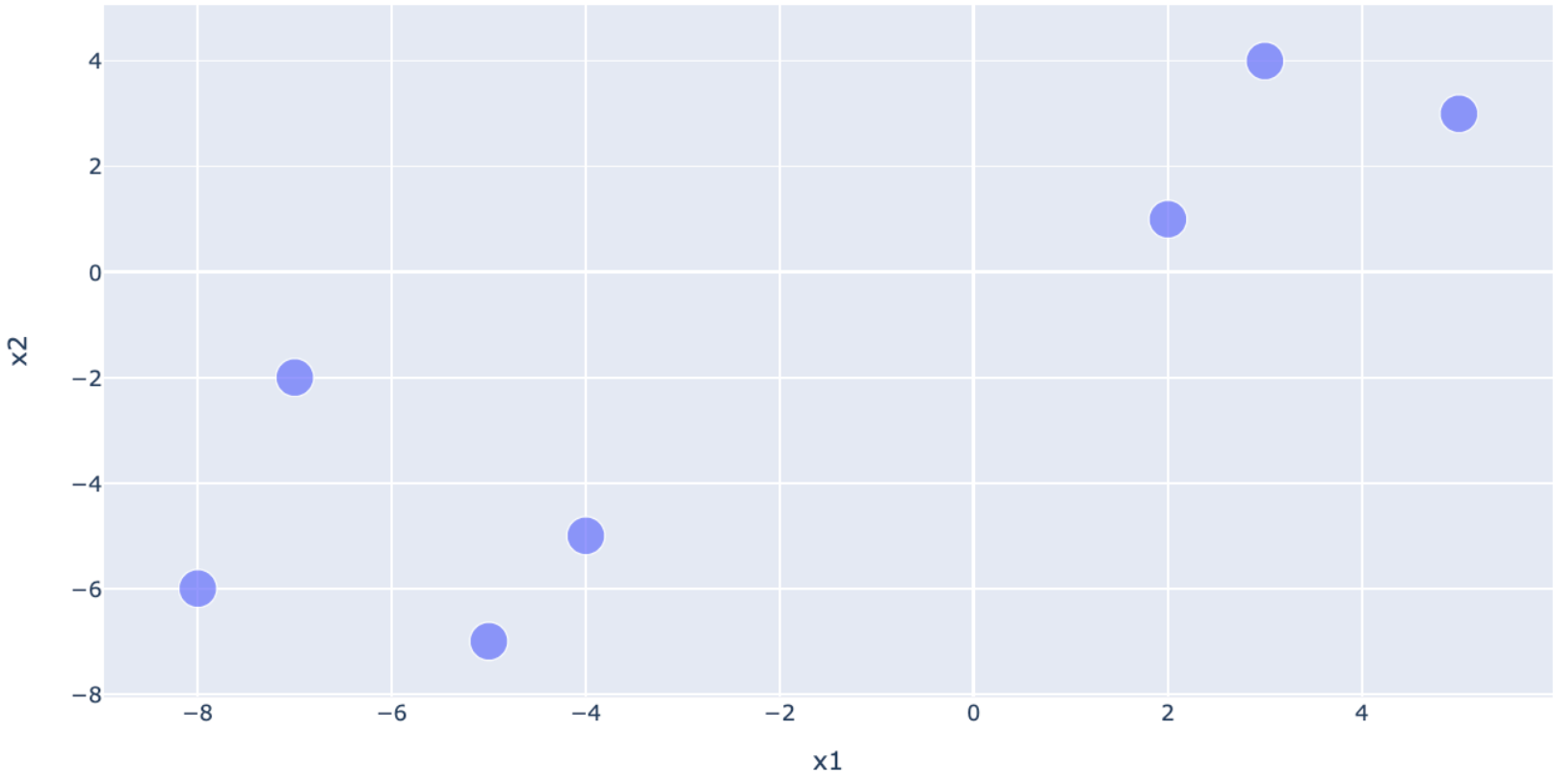
- Homework 3 grades are out.

Agenda

- Recap: the k-Means Clustering algorithm.
- Why does k-Means work?
- Practical considerations.
- Introduction to probability.

k-Means Clustering

Question: how might we “cluster” these points into groups?



Problem statement: clustering

Goal: Given a list of n data points, stored as vectors in $\mathbb{R}^d$, $\vec{x}_1, \vec{x}_2, ..., \vec{x}_n$, and a positive integer k, **place the data points into k clusters of nearby points**.

- Clusters are defined by **centroids**, $\mu_1, \mu_2, ..., \mu_k$. Each data point "belongs" to the group corresponding to the nearest centroid.

- We want to find the centroids that minimize **inertia**:

$$C(\mu_1, \mu_2, ..., \mu_k) = \text{total squared distance of each data point } \vec{x}_i \text{ to its closest centroid } \mu_j$$

- k-Means Clustering is an algorithm that attempts to minimize inertia.

k-Means Clustering, i.e. Lloyd's Algorithm

choose from the data points

1. Pick a value of k and randomly initialize k centroids.

2. Keep the centroids fixed, and update the groups.
   - Assign each point to the nearest centroid.

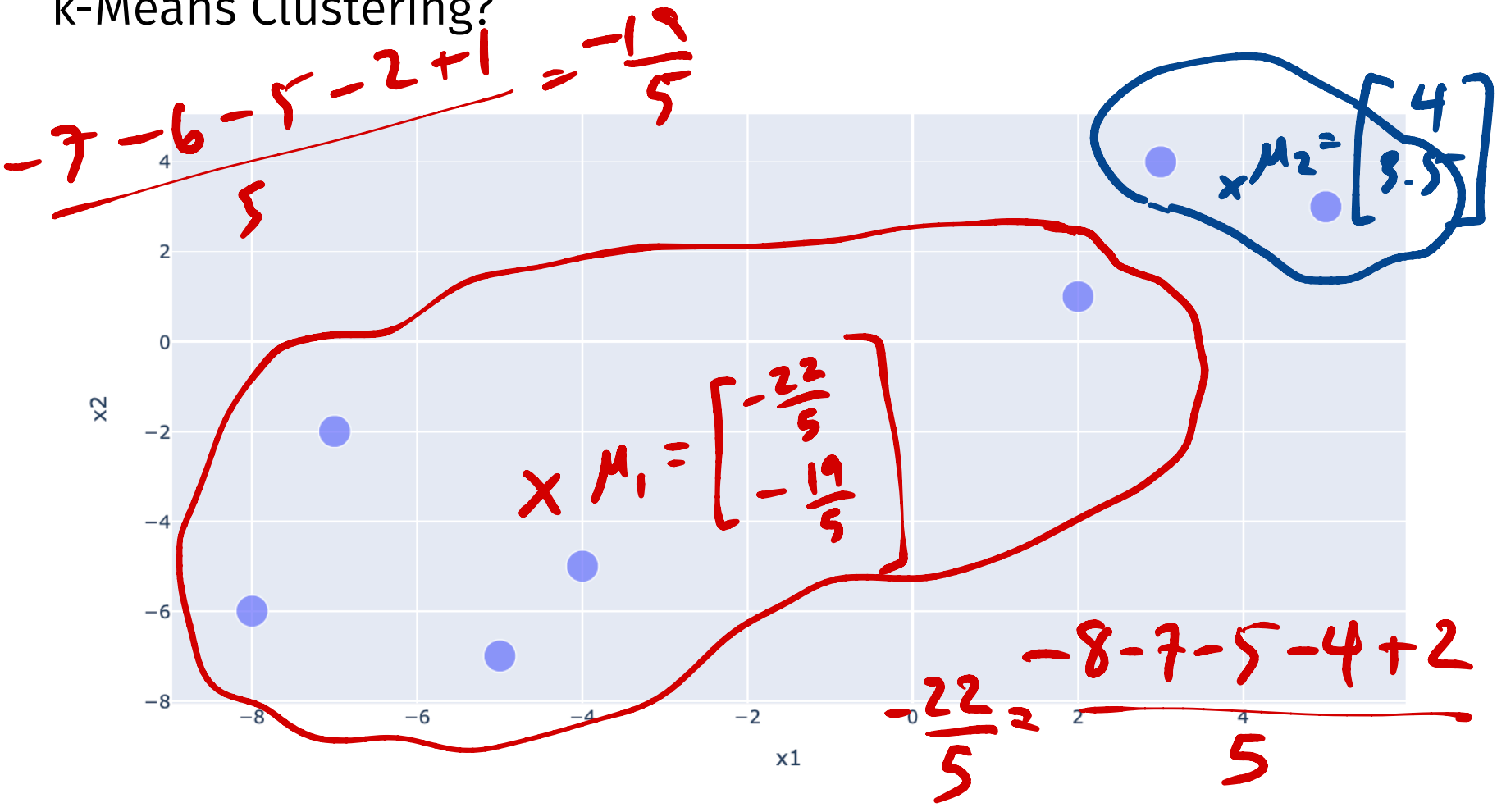
3. Keep the groups fixed, and update the centroids.
   - Move each centroid to the center of its group by averaging their coordinates.

4. Repeat steps 2 and 3 until the centroids stop changing.

An example by-hand

Suppose we choose the initial centroids $\mu_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\mu_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Where will the centroids move to after one iteration of k-Means Clustering?



Follow along with the demo by clicking the **code** link on the course website next to Lecture 11.

Why does k-Means work?

What is the goal of k-Means Clustering?

- Recall, our goal is to find the centroids $\mu_1, \mu_2, ..., \mu_k$ that minimize inertia:

$$C(\mu_1, \mu_2, ..., \mu_k) = \text{total squared distance of each data point } \vec{x}_i \text{ to its closest centroid } \mu_j$$

- Let's argue that each step of the k-Means Clustering algorithm reduces inertia.
  - After enough iterations, inertia will be small enough.

Why does k-Means work? (Step 1)

Let's look at each step one at a time.

Step 1: Pick a value of k and randomly initialize k centroids.

- After initializing our k centroids, we have an initial value of inertia. We are going to argue that this only decreases.

Why does k-Means work? (Step 2)

Step 2: Keep the centroids fixed, and update the groups by assigning each point to the nearest centroid.

- Assuming the centroids are fixed, for each $\vec{x}_i$ we have a choice — which group should it be a part of?
- Whichever group we choose, inertia will be calculated using the squared distance between $\vec{x}_i$ and that group's centroid.
- Thus, to minimize inertia, we assign each $\vec{x}_i$ to the group corresponding to the closest centroid.

Note that this analysis holds every time we're at Step 2, not just the first time.

Why does k-Means work? (Step 3)

Step 3: Keep the groups fixed, and update the centroids by moving each centroid to the center of its group (by averaging coordinates).

- Before we justify why this is optimal, let's re-visit inertia.

Aside: separating inertia

- Inertia:

$C(\mu_1, \mu_2, ..., \mu_k)$ = total squared distance of each data point $\vec{x}_i$ to its closest centroid μ_j

- Note that an equivalent way to write inertia is

$$C(\mu_1, \mu_2, ..., \mu_k) = C(\mu_1) + C(\mu_2) + ... + C(\mu_k) \text{ where}$$

$C(\mu_j)$ = total squared distance of each data point $\vec{x}_i$ in group j to centroid μ_j

- What's the point?

Why does k-Means work? (Step 3)

$$C(\mu_1, \mu_2, ..., \mu_k) = C(\mu_1) + C(\mu_2) + ... + C(\mu_k) \text{ where}$$

$C(\mu_j)$ = total squared distance of each data point $\vec{x}_i$ in group j to centroid μ_j

Step 3: Keep the groups fixed, and update the centroids by moving each centroid to the center of its group (by averaging coordinates).

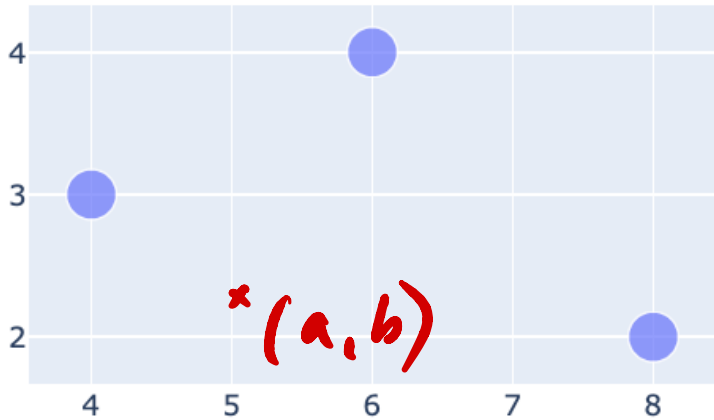
- Let's argue why this minimizes $C(\mu_j)$, for each group j.

Why does k-Means work? (Step 3)

$C(\mu_j)$ = total squared distance of each data point $\vec{x}_i$ in group j to centroid μ_j

Suppose group j contains the points $(4, 3)$, $(6, 4)$, and $(8, 2)$.

Where should we put $\mu_j = \begin{bmatrix} a \\ b \end{bmatrix}$ to minimize $C(\mu_j)$?



$$C(\mu_j) = C(a, b)$$
$$= (4-a)^2 + (3-b)^2$$
$$+ (6-a)^2 + (4-b)^2$$
$$+ (8-a)^2 + (2-b)^2$$

Why does k-Means work? (Step 3)

$$C(a,b) = (4-a)^2 + (3-b)^2 + (6-a)^2 + (4-b)^2 + (8-a)^2 + (2-b)^2$$

$$\frac{\partial C}{\partial a} = 2(4-a)(-1) + 2(6-a)(-1) + 2(8-a)(-1) = 0$$

$$4-a+6-a+8-a=0$$

$$3a = 4+6+8$$

$$a^* = \frac{4+6+8}{3}$$

$$\frac{\partial C}{\partial b} = -2(3-b) - 2(4-b) - 2(2-b) = 0$$

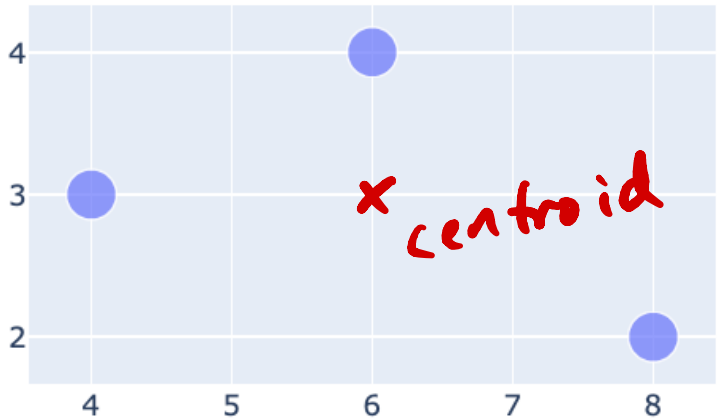
$$\Rightarrow b^* = \frac{3+4+2}{3}$$

Why does k-Means work? (Step 3)

$C(\mu_j)$ = total squared distance of each data point $\vec{x}_i$ in group j to centroid μ_j

Suppose group j contains the points $(4, 3)$, $(6, 4)$, and $(8, 2)$.

Where should we put $\mu_j = \begin{bmatrix} a \\ b \end{bmatrix}$ to minimize $C(\mu_j)$?



$a^* = \frac{4+6+8}{3} = 6$

$b^* = \frac{3+4+2}{3} = 3$

Cost and empirical risk

- On the previous slide, we saw a function of the form

$$C(\mu_j) = C(a, b) = (4 - a)^2 + (3 - b)^2 + (6 - a)^2 + (4 - b)^2 + (8 - a)^2 + (2 - b)^2$$

$f(a)$ $g(b)$

- $C(a, b)$ can be thought of as the sum of two separate functions, $f(a)$ and $g(b)$.
  - $f(a) = (4 - a)^2 + (6 - a)^2 + (8 - a)^2$ computes the total squared distance of each x_1 coordinate to a.
  - From earlier in the course, we know that $a^* = \frac{4+6+8}{3} = 6$ minimizes $f(a)$.

Practical considerations

Initialization

- Depending on our initial centroids, k-Means may "converge" to a clustering that doesn't actually have the lowest possible inertia.
  - In other words, like gradient descent, k-Means can get caught in a **local minimum**.

- Some solutions:
  - Run k-Means several times, each with different randomly chosen initial centroids. Keep track of the inertia of the final result in each attempt. Choose the attempt with the lowest inertia.

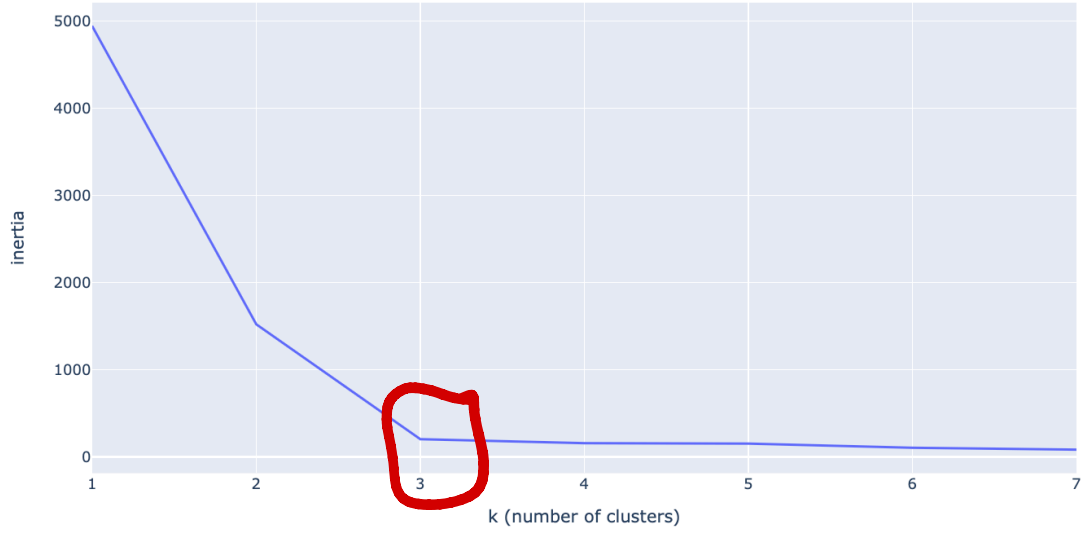
  - **k-Means++**: choose one initial centroid at random, and choose the remaining initial centroids by maximizing distance from all other centroids.

Choosing k

- Note that as k increases, inertia decreases.
  - Intuitively, as we add more centroids, the distance between each point and its closest centroid will drop.
- But the goal of clustering is to put data points into groups, and having a large number of groups may not be meaningful.
- This suggests a tradeoff between k and inertia.

The “elbow” method

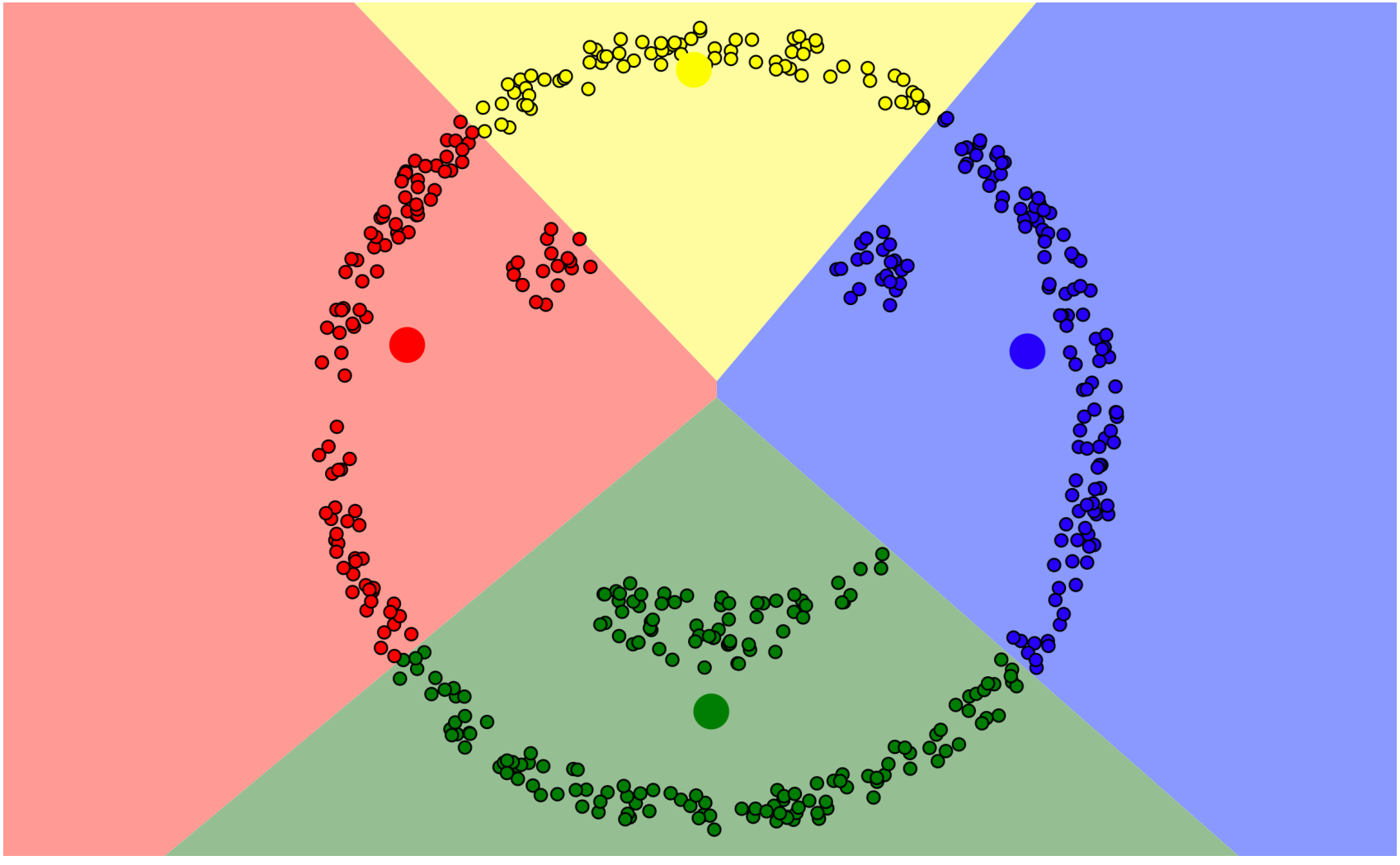
- Strategy: run k-Means Clustering for many choices of k (e.g. $k = 1, 2, 3, ..., 8$).
- Compute the value of inertia for each resulting set of centroids.
- Plot a graph of inertia vs k.
- Choose the value of k that appears at an “elbow”.



See the notebook for a demo.

Low inertia isn't everything!

- Even if k-Means works as intended and finds the choice of centroids that minimize inertia, the resulting clustering may not look "right" to us humans.
  - Recall, inertia measures the total squared distance to centroids.
  - This metric doesn't always match our intuition.
- Let's look at some examples at https://tinyurl.com/40akmeans.
  - Go to "I'll Choose" and "Smiley Face". Good luck!



Other clustering techniques

- k-Means Clustering is just one way to cluster data.
- There are many others, each of which work differently and produce different kinds of results.
- Another common technique: **agglomerative clustering**.
  - High level: start out with each point being in its own cluster. Repeatedly combine clusters until only k are left.
- Check out **this chart**.

Introduction to probability

From Lecture 1: course overview

Part 1: Learning from Data (Lectures 1-11)

- Summary statistics and loss functions; mean absolute error and mean squared error.
- Linear regression (incl. linear algebra).
- Clustering.

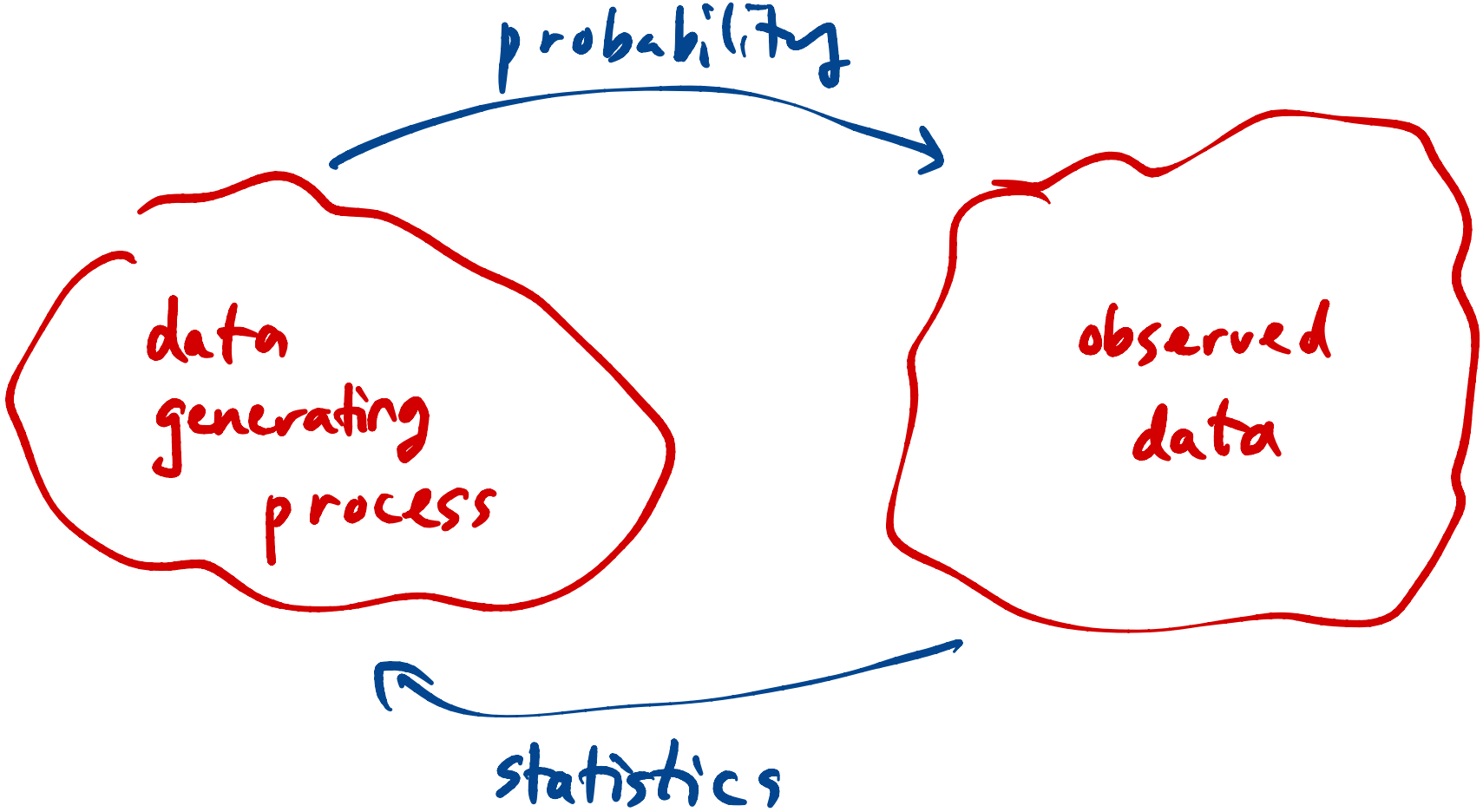
Part 2: Probability (Lectures 12-18)

- Set theory and combinatorics; probability fundamentals.
- Conditional probability and independence.
- Naïve Bayes (mix of both parts of the class).

Why do we need probability?

- So far in this class, we have made predictions based on a dataset.
- This dataset can be thought of as a **sample** of some population.
- For a prediction rule to be useful in the future, the sample that was used to create the prediction rule needs to look similar to samples that we'll see in the future.

Probability and statistics



Statistical inference

Given observed data, we want to know how it was generated or where it came from, for the purposes of

- predicting outcomes for other data generated from the same source.
- know how different our sample could have been.
- draw conclusions about our entire population and not just our observed sample (i.e. generalize).

Probability

Given a certain model for data generation, what kind of data do you expect the model to produce? How similar is it to the data you have? Probability is the tool to answer these questions.

- expected value vs. sample mean.
- variance vs. sample variance.
- likelihood of producing exact observed data.

Terminology

- An **experiment** is some process whose outcome is random (e.g. flipping a coin, rolling a die).
- A **sample space**, S, is the set of all possible outcomes of an experiment.
  - Could be finite or infinite!
- An **event** is a subset of the sample space. = sets of outcomes.

Example: Rolling a 6-sided die.

$S = \{1, 2, 3, 4, 5, 6\}$

3 alone is an outcome

$\{2, 4, 6\}$: event I roll an even #

$\{5, 6\}$: event my roll is ≥ 5

Probability distributions

- A probability distribution, p, describes the **probability** of each outcome s in a sample space S.
  - The probability of each outcome must be between 0 and 1: $0 \leq p(s) \leq 1$.
  - The sum of the probabilities of each outcome must be exactly 1: $\sum_{s \in S} p(s) = 1$.

↑ for all outcomes in sample space

- The probability of an **event** is the sum of the probabilities of the outcomes in the event.
  - $P(E) = \sum_{s \in E} p(s)$.

rolling a fair die

$E = \{2, 4, 6\}$

$P(E) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$

$A: \{2\}$

$P(A) = \frac{1}{6}$

Equally-likely outcomes

- If S is a sample space with n possible outcomes, and all outcomes are equally-likely, then the probability of any one outcome occurring is $\frac{1}{n}$.
- The probability of an event A, then, is

$$P(A) = \frac{1}{n} + \frac{1}{n} + \ldots + \frac{1}{n} = \frac{\text{\# of outcomes in A}}{\text{\# of outcomes in S}} = \frac{|A|}{|S|}$$

- **Example:** Flipping a coin three times.

A = some set of outcomes

$S = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$

$P(\text{first coin is heads}) = \frac{4}{8} = \frac{1}{2}$

Summary, next time

Summary

- k-Means Clustering attempts to minimize inertia.
  - We showed that it minimizes inertia on each step, but it's possible that it converges to a local minimum.
  - Different initial centroids can lead to different clusterings.
- To choose k, the number of clusters, we can use the elbow method.
- Other clustering techniques may work better than k-Means Clustering in certain cases.
- Outcomes, sample spaces, and events are the "building blocks" of probability.

Next time

- A deep-dive on the fundamentals rules of probability.
- **Important:** We've posted **many** probability resources on the resources tab of the course website. These will no doubt come in handy.
  - No more DSC 40A-specific readings.