Semi-supervised learning

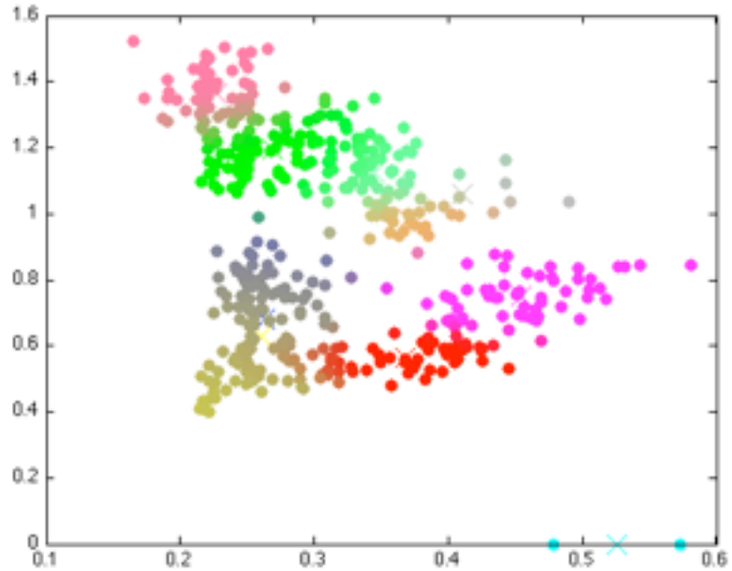
Computational Cognitive Science 2014

Dan Navarro

Where are we?

- Supervised learning
  - Similarities between cognitive science and machine learning
  - Prototype models linked with parametric classifiers
  - Exemplar models linked with non-parametric classifiers

- Unsupervised learning
  - k-means classifier
  - Mixture of Gaussians
  - Link to phoneme learning



- Next: semi supervised learning

People form theories about categories

- Our primary goal as learners is not actually classification
  - We want to make predictions about the broader world
  - We want causal knowledge about why things are how they are
  - We want theories

- "Theory theory" is complex
  - Learner needs to infer rich causal graphs, and schemas for new ones
  - Inference about counterfactual worlds, one-shot learning, etc
  - Way beyond the scope of these few lectures

- Let's set our sights a little lower for now…
  - Learning category structure

Volleyball

Volleyball

Volleyball

Football

Football

Football

Football

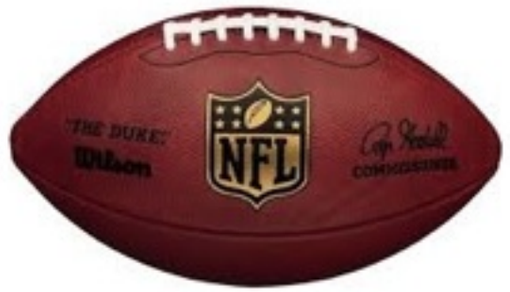
Football

Football


Wilson
avp
REPLICA GAME BALL


SHERRIN
®
KANGAROO BRAND
OFFICIAL BALL
AUSTRALIAN FOOTBALL LEAGUE


SHERRIN
MATCH
GENUINE LEATHER


"THE DUKE"
Wilson
NFL


Wilson
avp
REPLICA GAME BALL


SHERRIN
KANGAROO BRAND
OFFICIAL BALL
AUSTRALIAN FOOTBALL LEAGUE

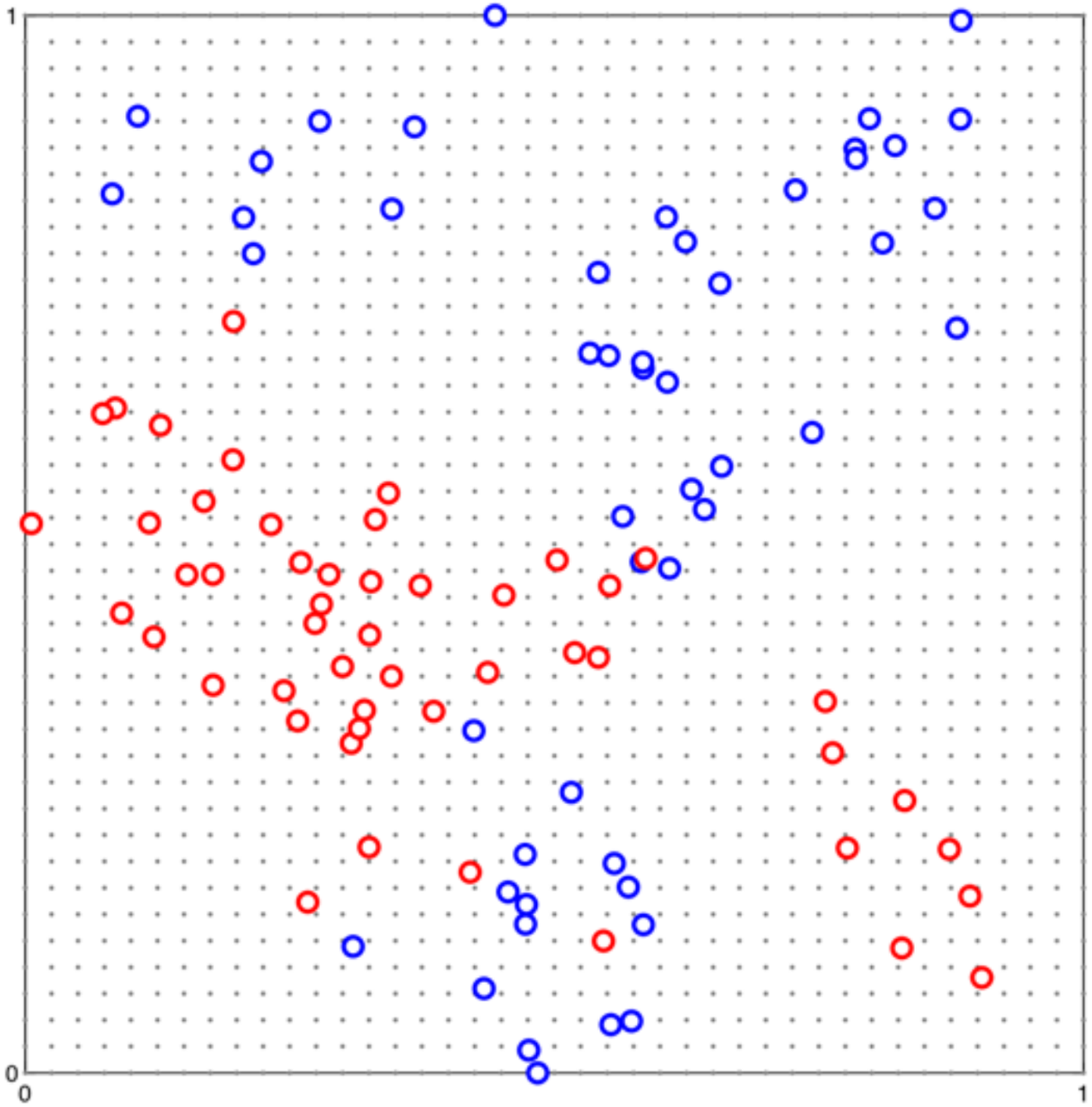

SHERRIN
MATCH
GENUINE LEATHER


"THE DUKE"
Wilson
NFL
COMMISSIONER

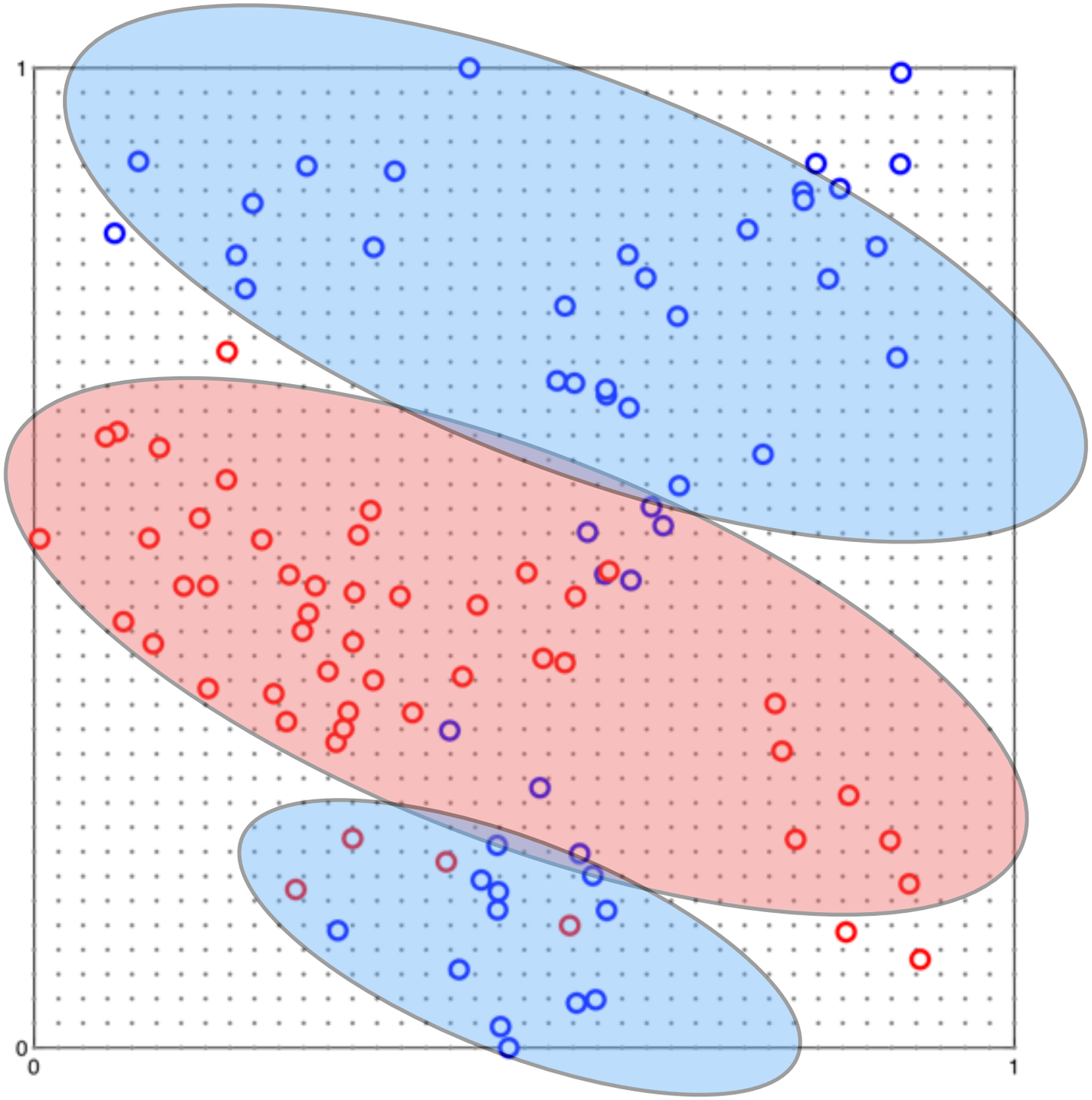
Categories have internal structure

- We often use the same label to cover a multitude of things
  - “football” is “U.S. football”, “soccer”, “rugby”, “Australian rules” etc
  - a “bank” is a moneylender, or the edge of a river
  - a “lecture” is an educational speech, or a disciplinary action
  - etc

- We want classifiers that discover this structure
  - Humans do it: you all saw this issue immediately
  - Simple Gaussian classifiers generally fail on these problems
  - k-NN or kernel classifiers can learn the messy classification boundaries that this structure produces, but they never explicitly identify this structure

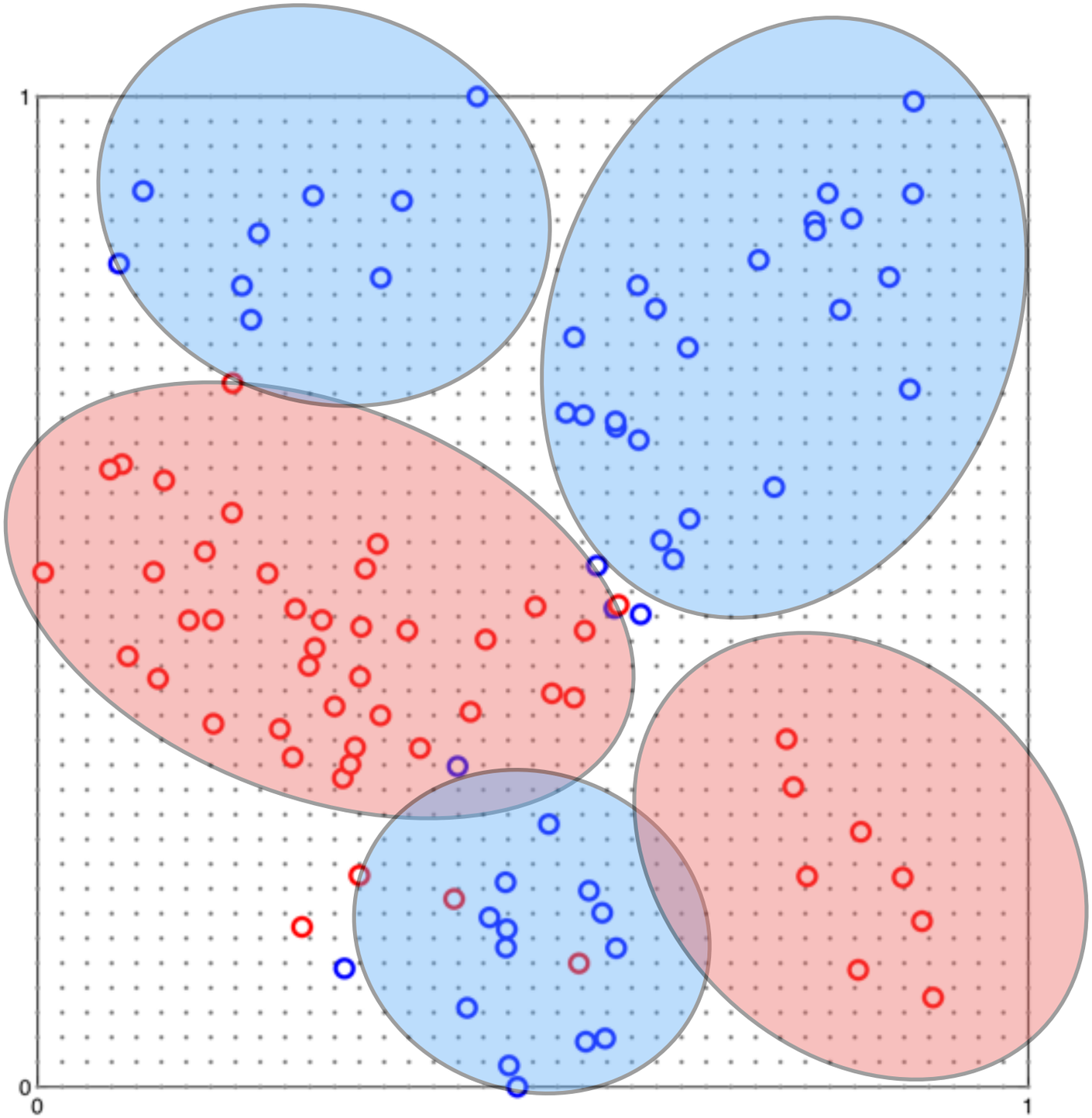


Our running example captures this issue nicely



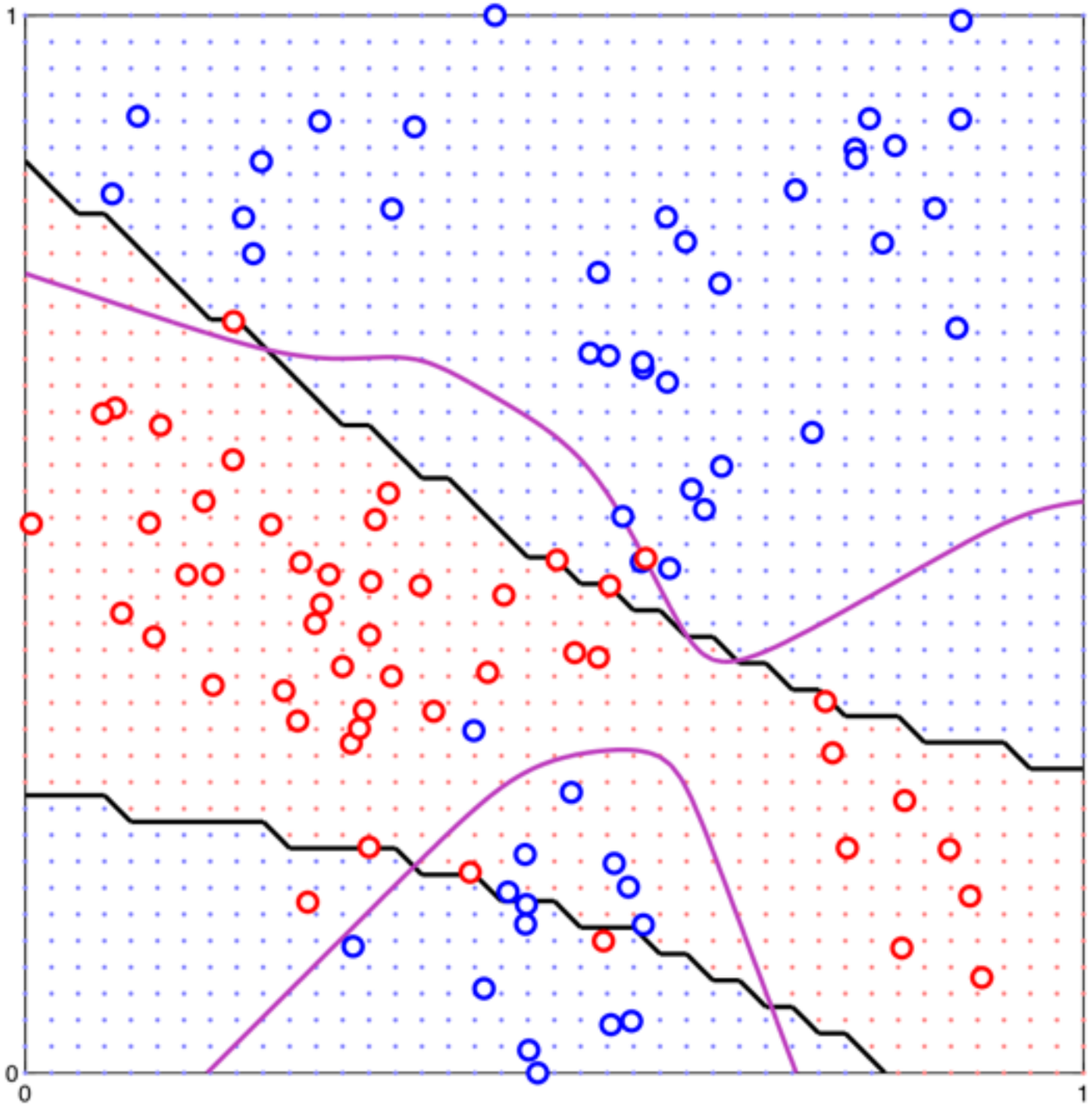
Our running example captures this issue nicely

We want a classifier that discovers sub-categories



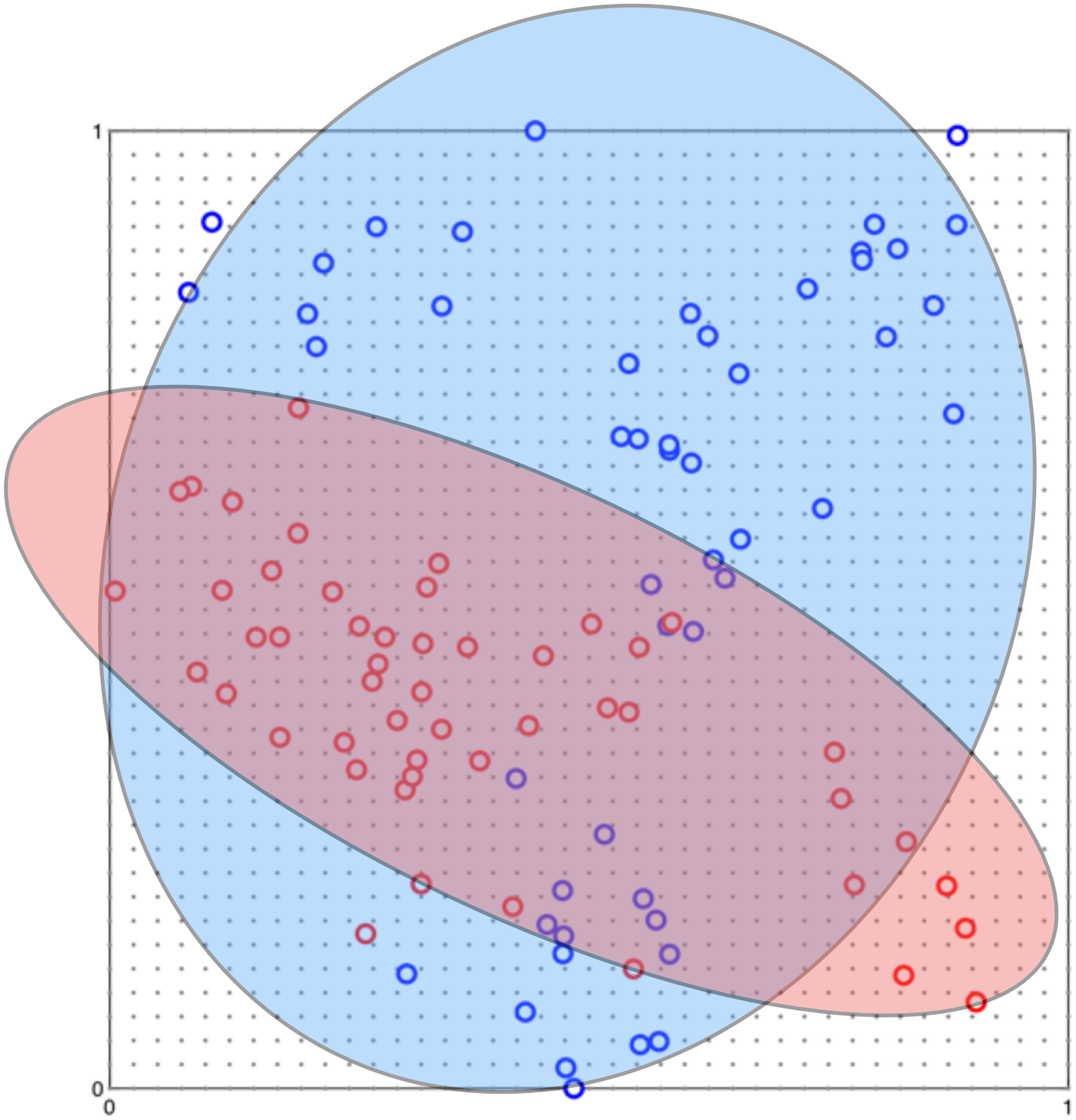
Our running example captures this issue nicely

We want a classifier that discovers sub-categories



This is the best a Gaussian classifier can do.

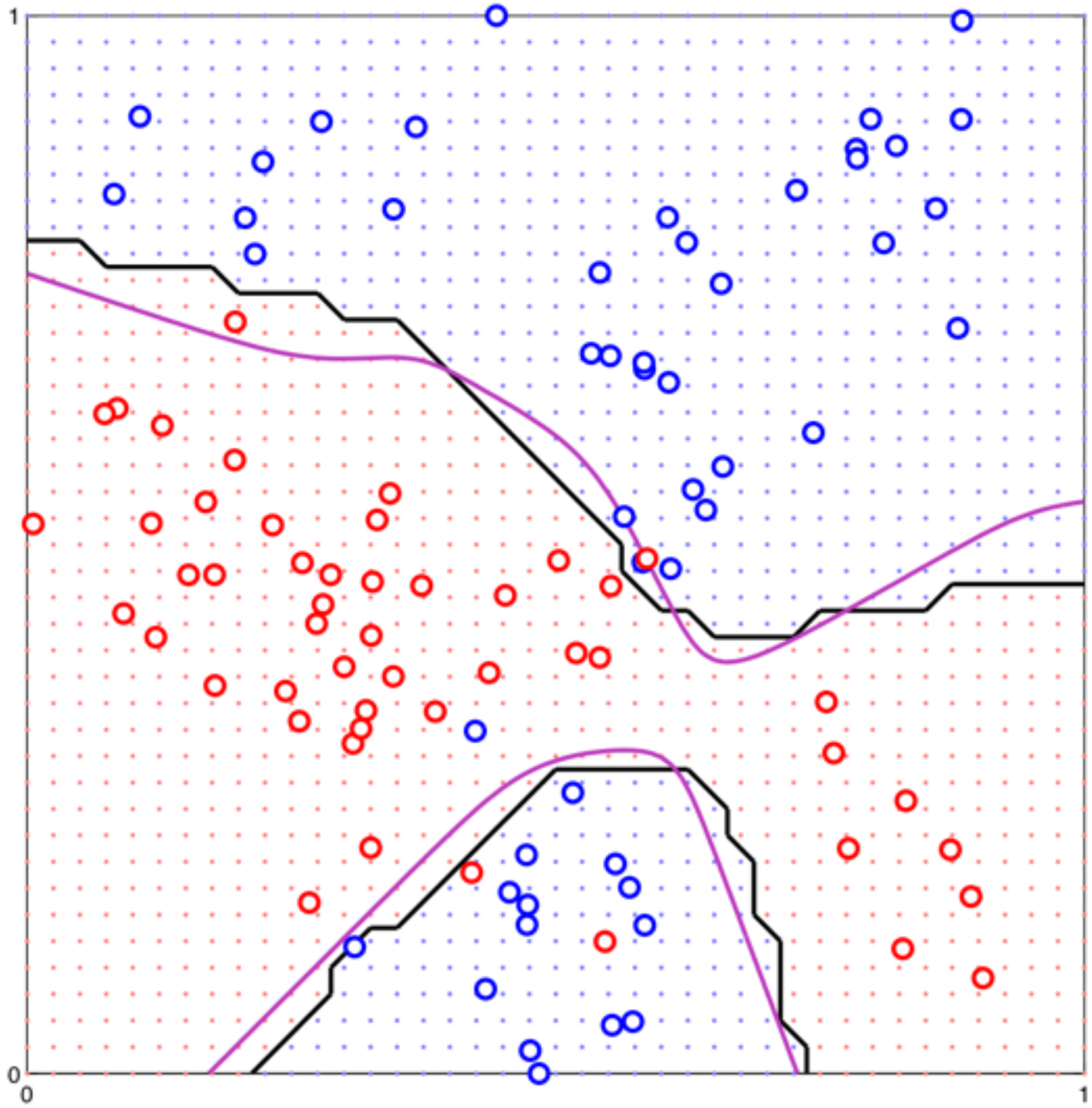
Performance is okay, but the two blue regions are not because it has learned two subcategories…



This is how it represents the categories.

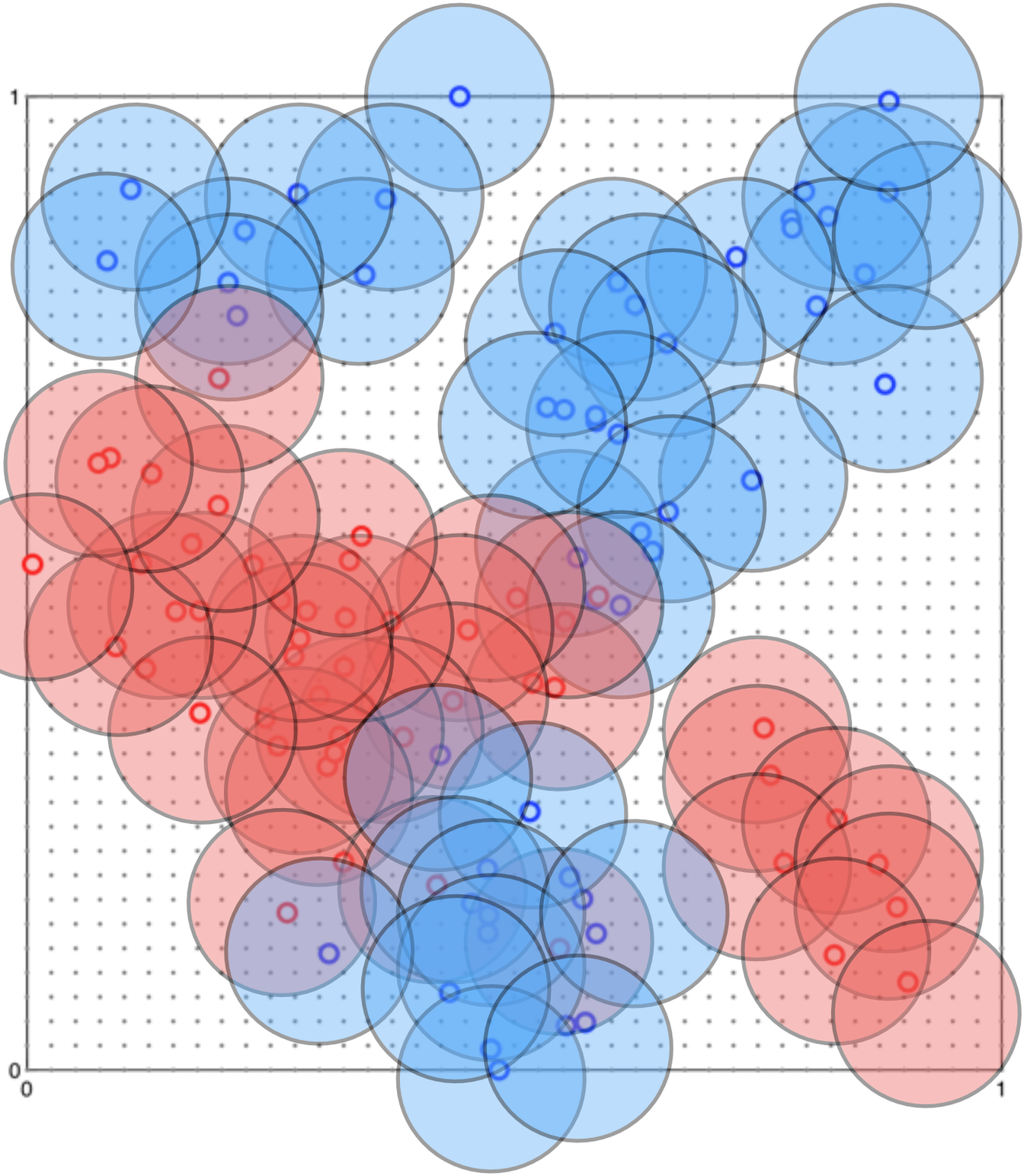
Obviously it's wrong.

It classifies to an acceptable standard, but it has no ability to say anything deeper about the category structure



Kernel classifiers aren't any better.

Performance is good, but the blue regions are not because it has learned about subcategories…



This is how it represents the categories.

That's even more absurd as a theory of category structure

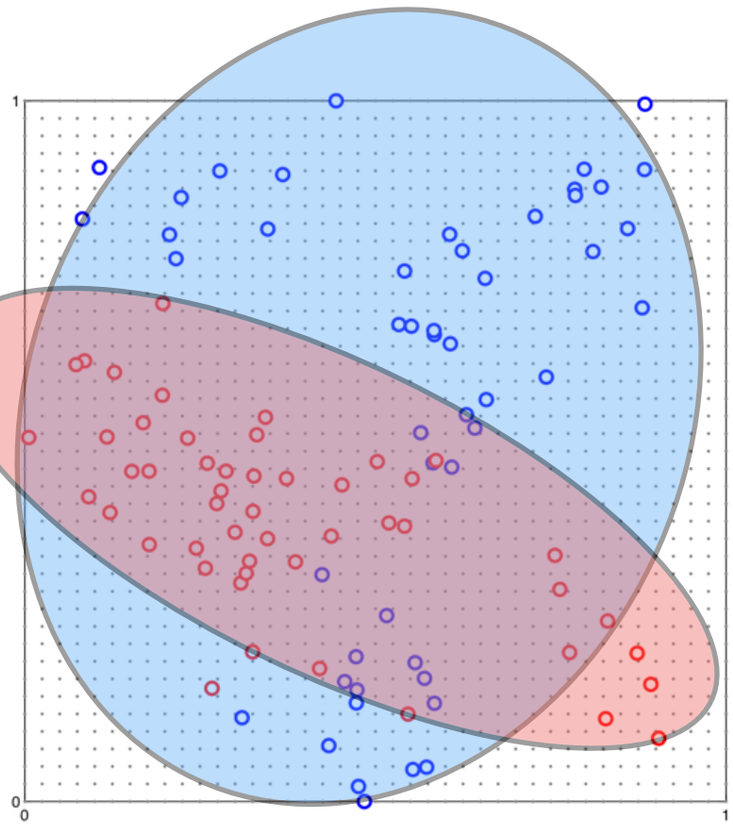
Cluster based category representation

The machine learning perspective

- Exemplar models (k-NN, kernel density estimators)
  - High storage requirement, no data reduction at all
  - Online computation is expensive unless you do something fancy (you really don't want to be doing N kernel function evaluations per classification decision… not with the kind of data the brain needs to process)

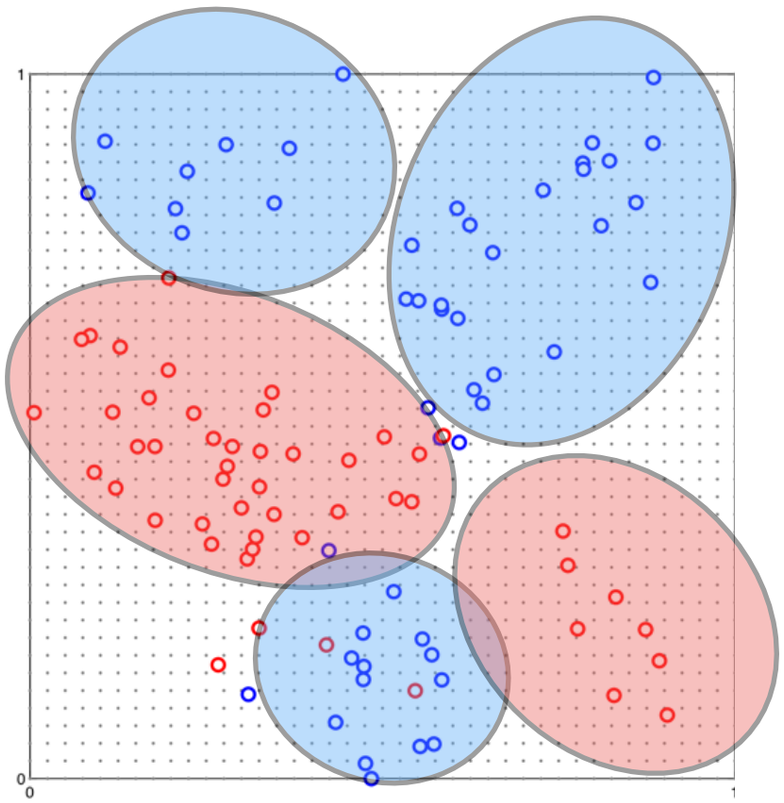
- Prototype models (the Gaussian classifier)
  - Low storage requirements: only a few numbers per category
  - Online computation is cheap: one Gaussian function evaluation per category per decision

Do not multiply entities beyond necessity



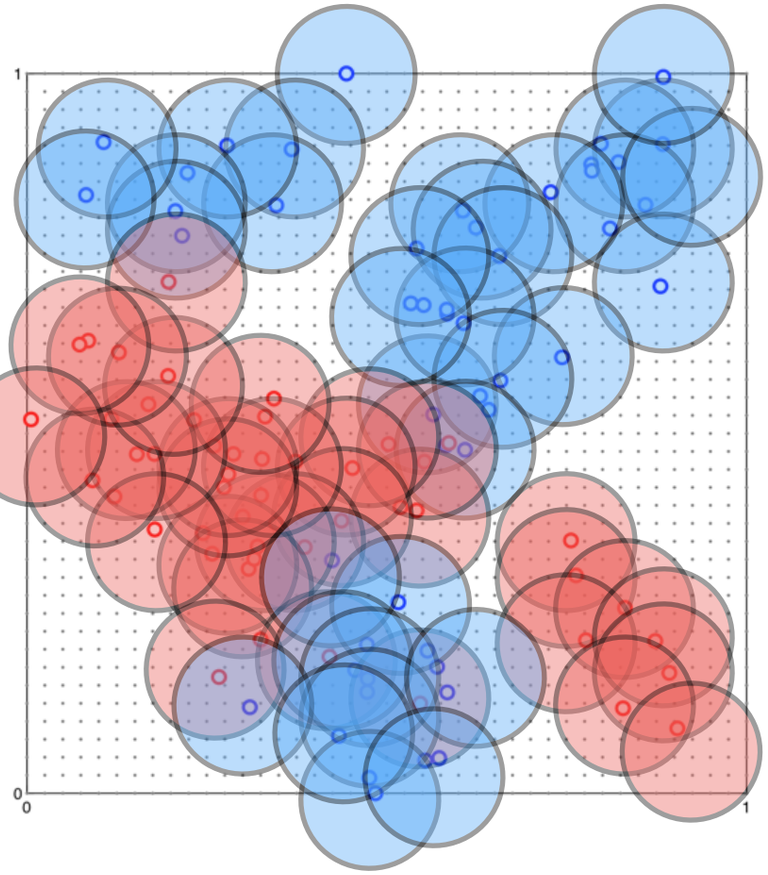
Prototype

Too few



Cluster

Just right?



Exemplar

Too many

A quick and dirty solution

- Apply k-means to each category
  - Start with N data points per category
  - Reduce it to k means per category

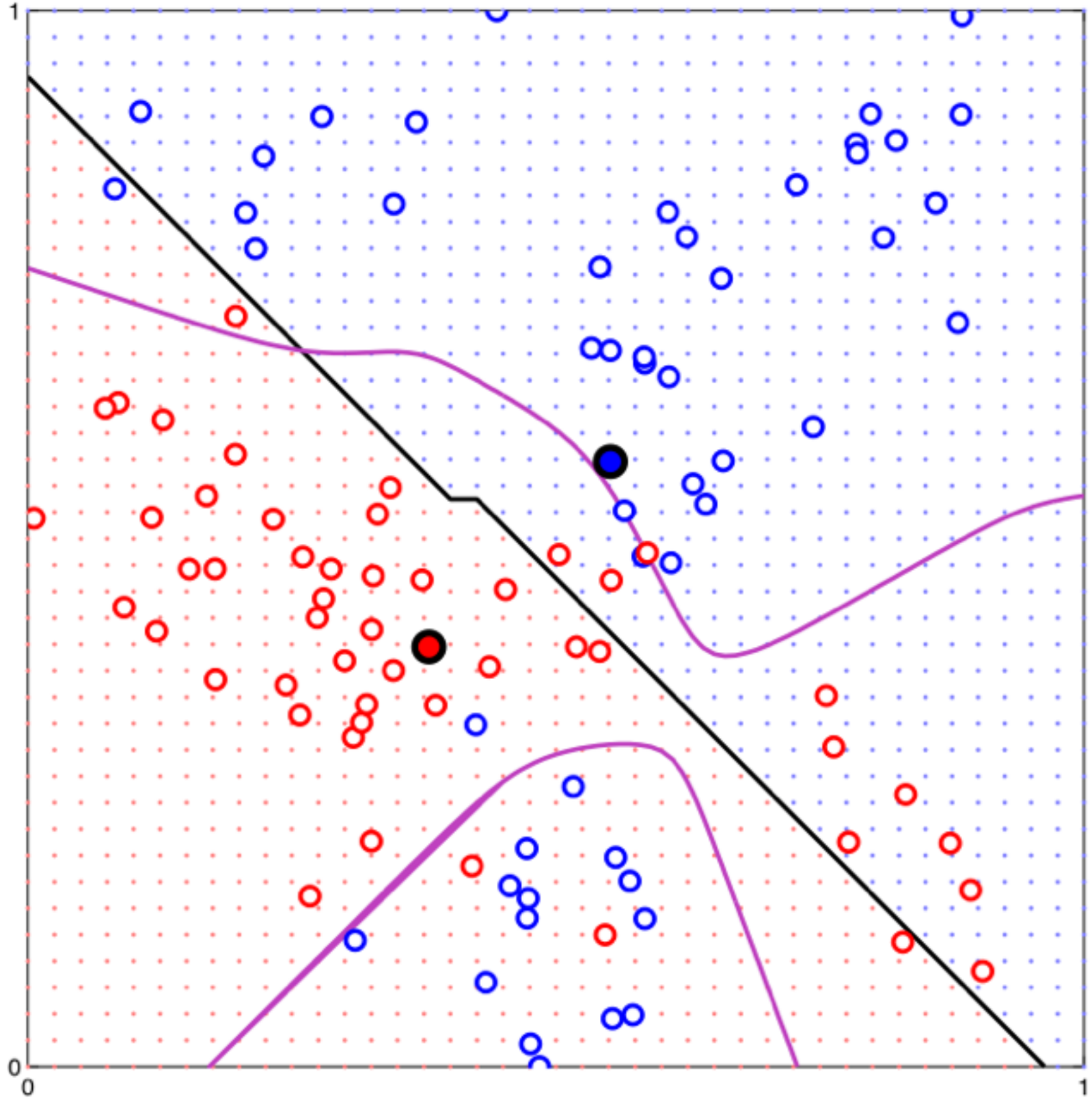
- Apply your classifier (e.g., k-NN) to the reduced data
  - I'll assume 1-NN in these examples

- It's not pretty
  - Lacks the elegance of a proper probabilistic model
  - But it works reasonably well
  - And we'll see the full probabilistic model soon enough…

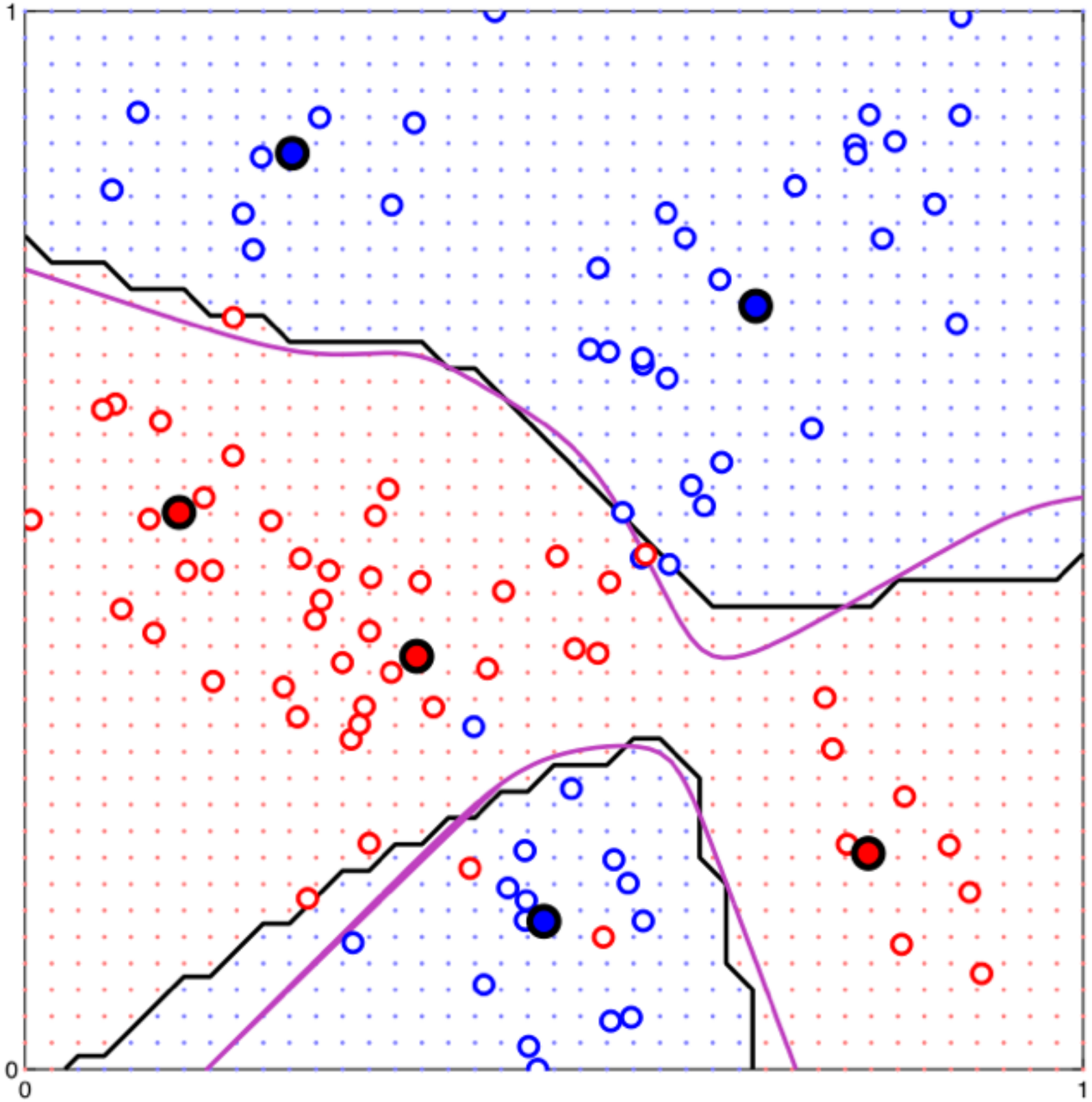
Demonstration code

(classifiers.R: kMeans and reducedNN)

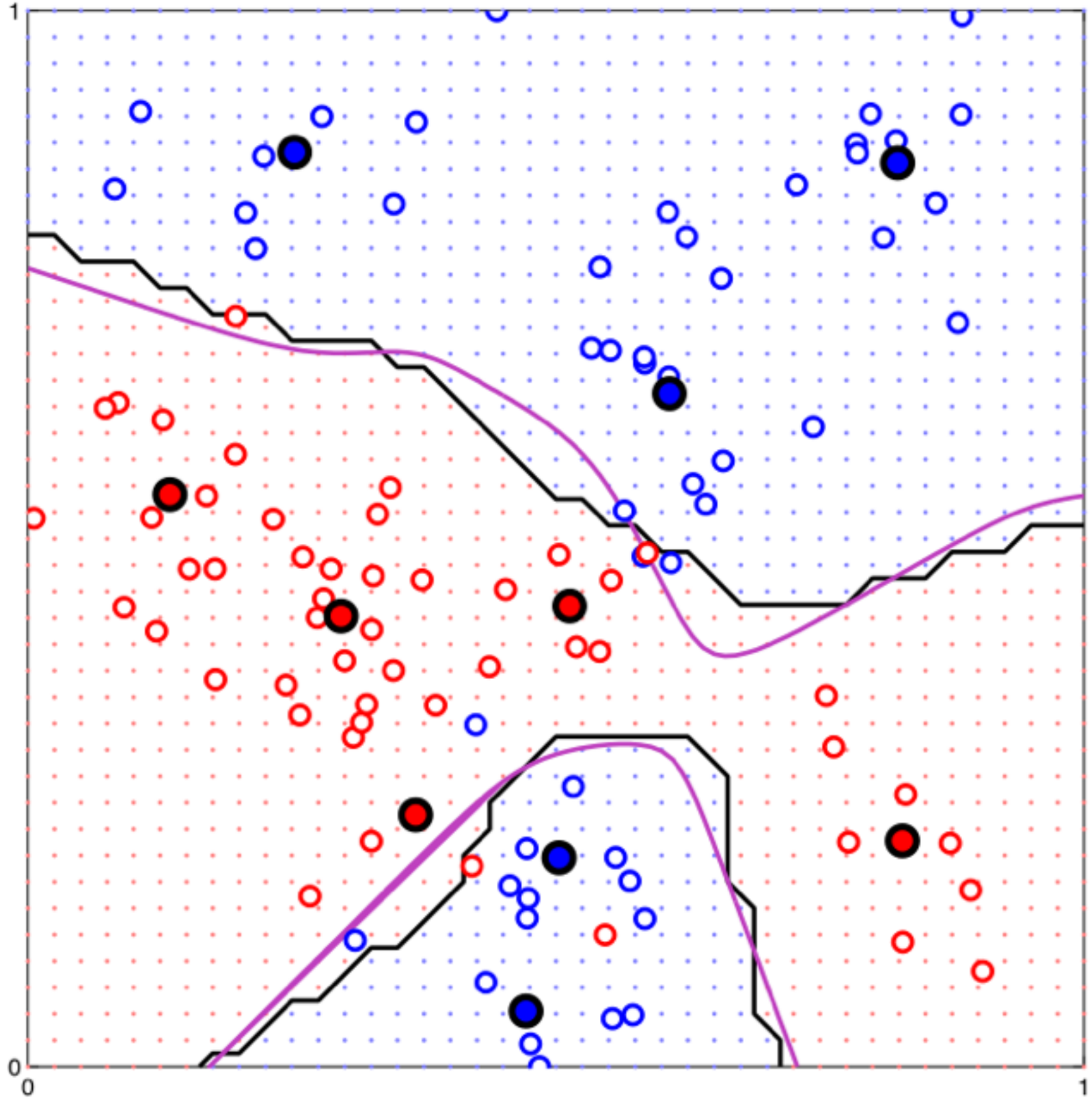
$k = 1$



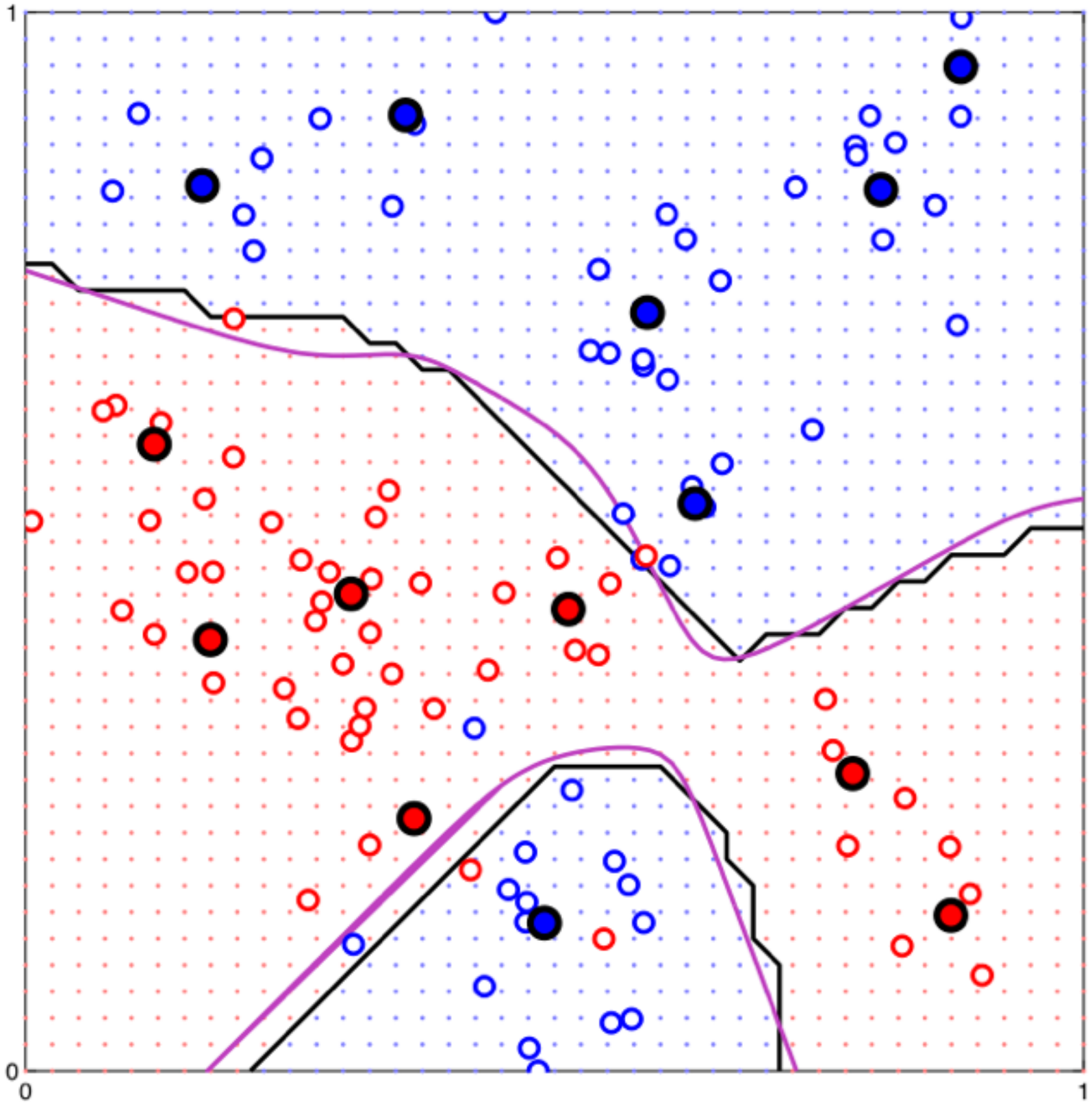
$k = 3$



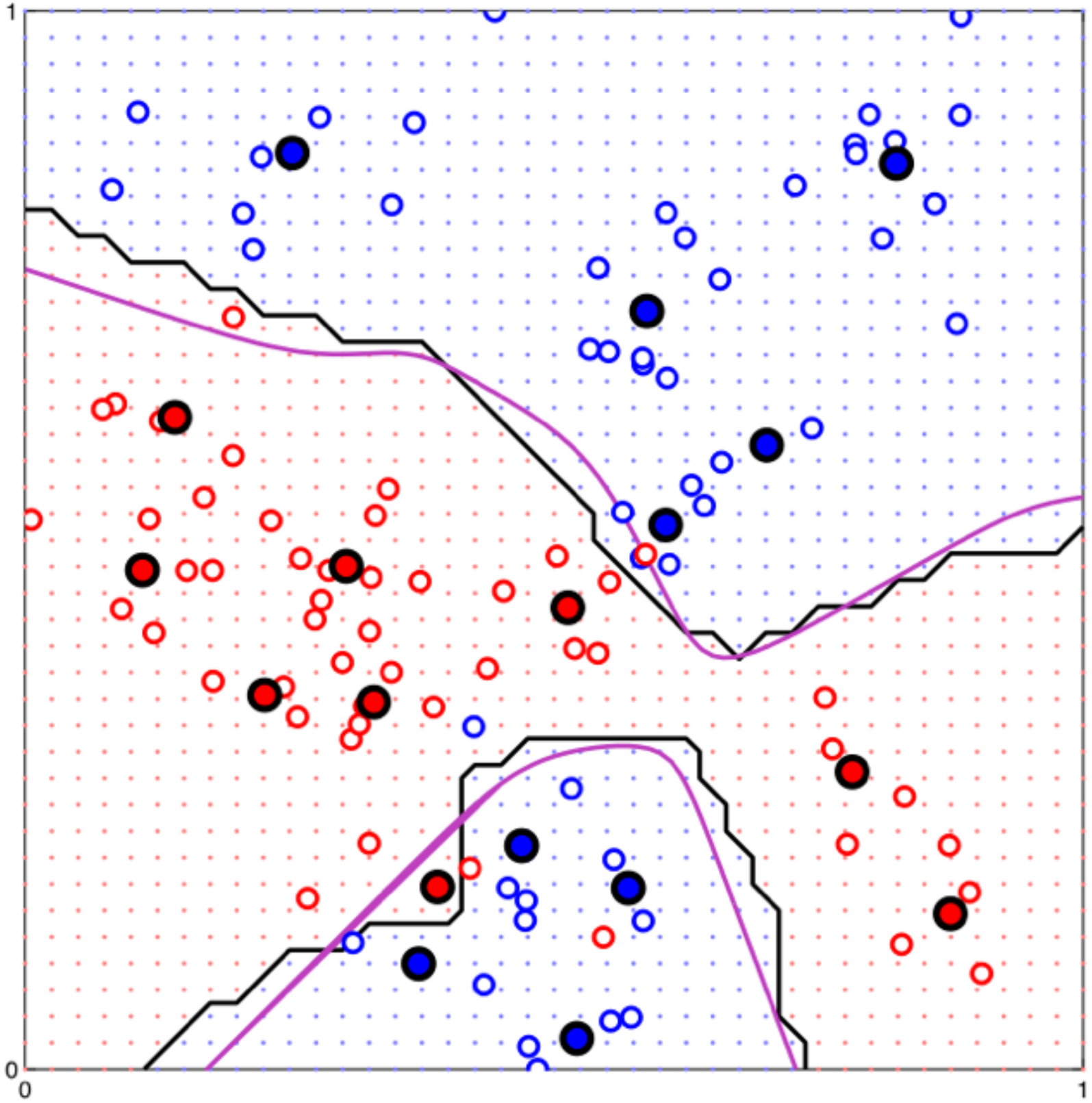
$k = 5$



$k = 7$



$k = 9$



An angry Bayes demands his due!

Where are your priors? What is your theory? I cry angry tears at your shameless short cuts

Fine. What*ever*

Okay, stop bugging me. Here's a fully Bayesian model that does supervised, unsupervised and semi-supervised learning using a cluster based representation. Good enough?

A tricky problem

- Any model that tries to extract a set of clusters…
  - k-means clustering for unsupervised classification
  - mixture of Gaussians for unsupervised classification
  - cluster-based supervised classification

- …has to choose the number of clusters somehow
  - the learner can't possible know the answer in advance
  - so you need to figure it out "on the fly"

- Bayesian solution:
  - specify a prior over the allocations of observations to clusters

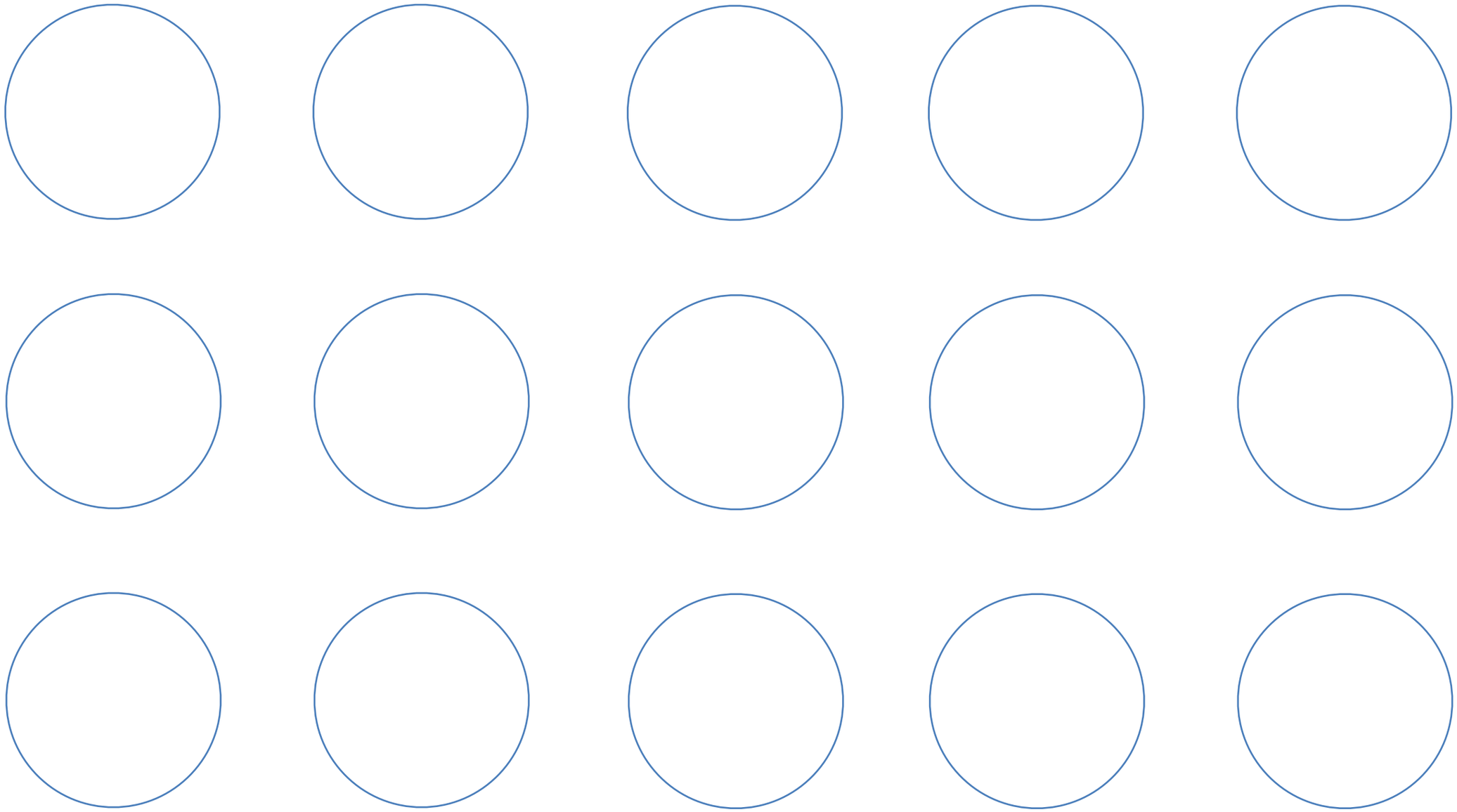
In the beginning there was an infinite Chinese restaurant…

“Chinese restaurant process” (CRP)

- Origin of the name comes from two observations:
  - Chinese restaurants in San Francisco don’t look very big, yet they appear to have infinite seating capacity
  - Statisticians sometimes get too cute with naming things

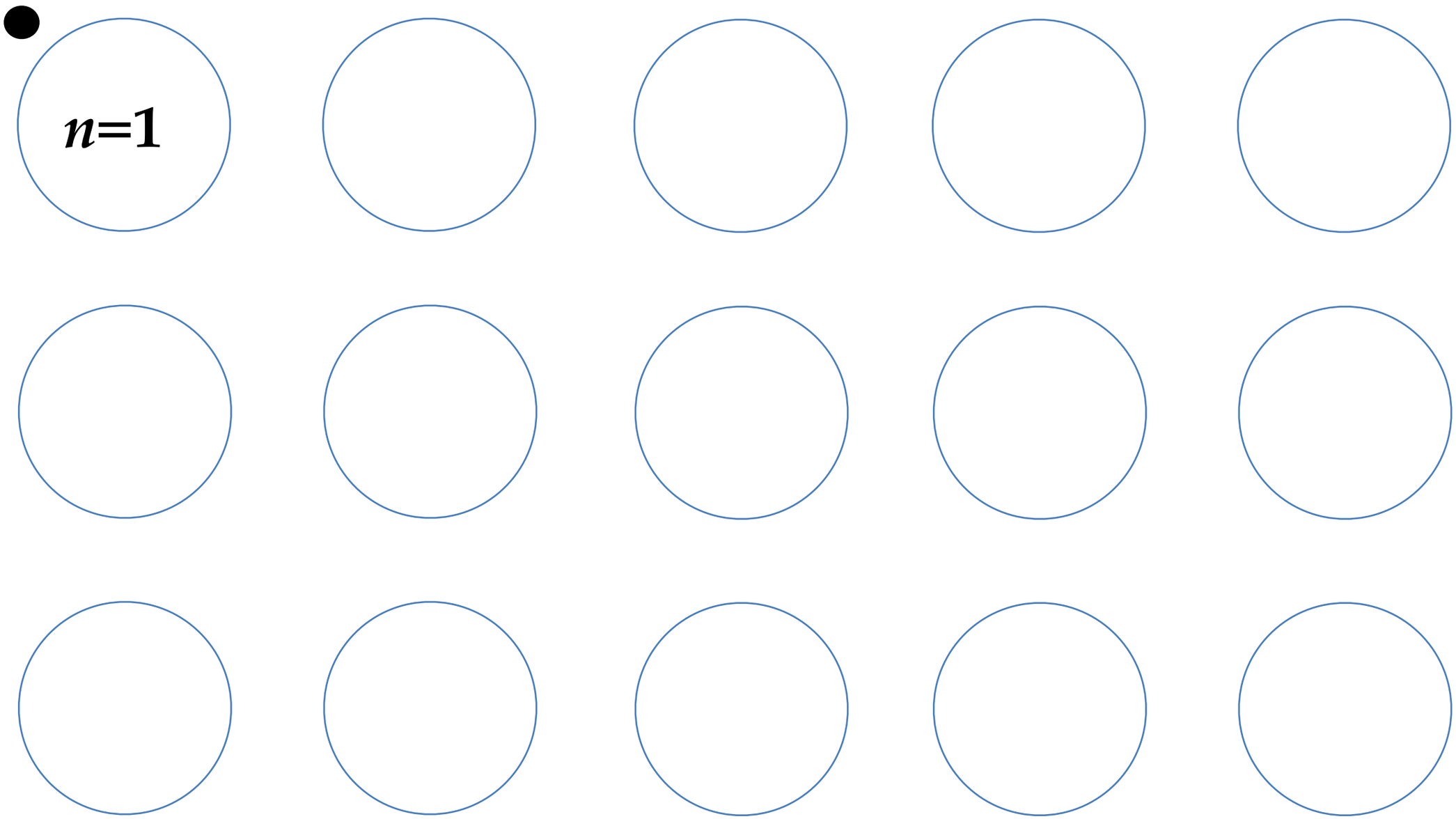
- Here’s the metaphor
  - Customers walk into a restaurant and sit down at tables
  - Customers = observations
  - Tables = clusters

In the beginning was an infinite restaurant, and all the tables looked the same

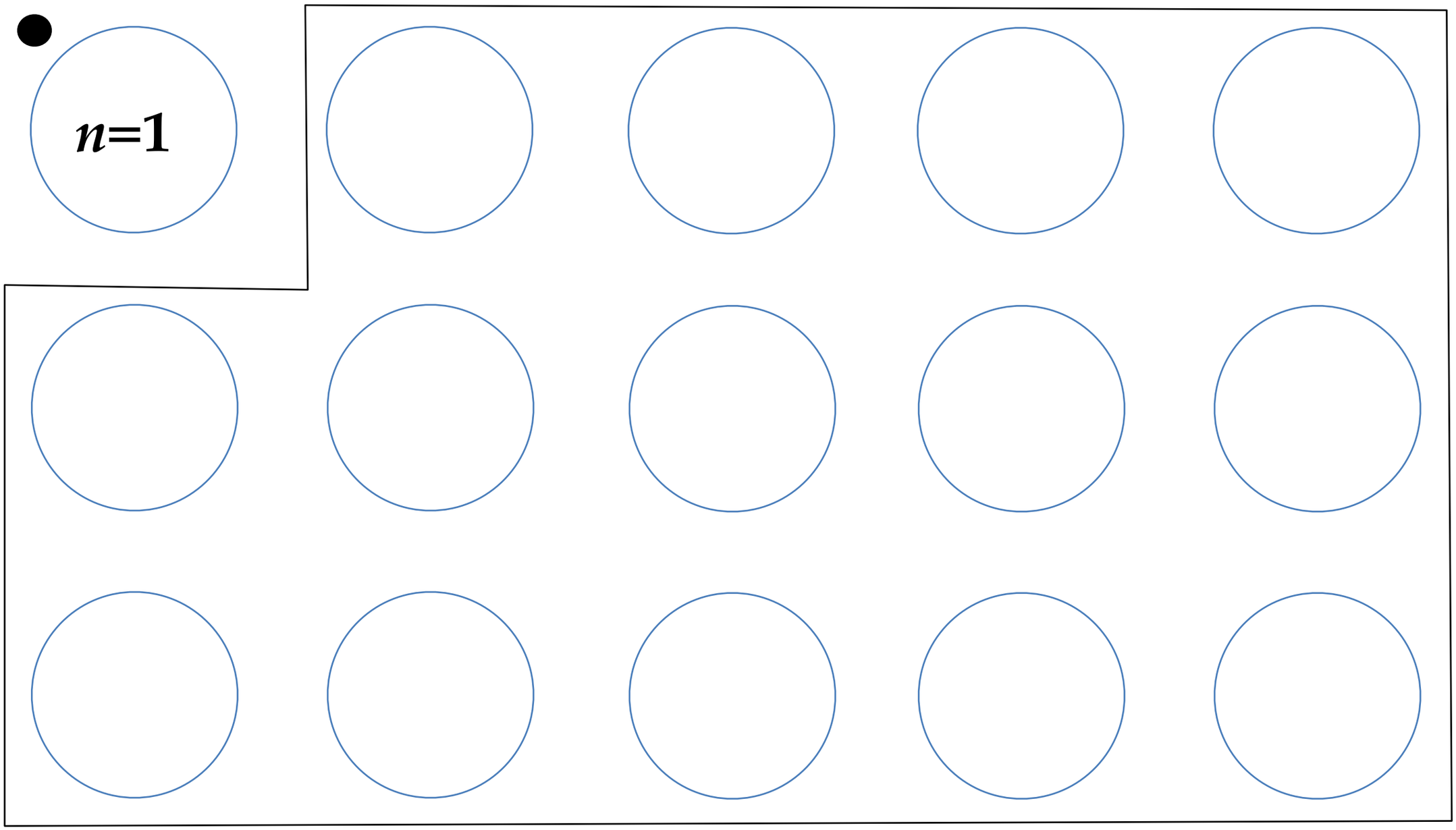


+ infinity more tables that I couldn’t fit onto the screen

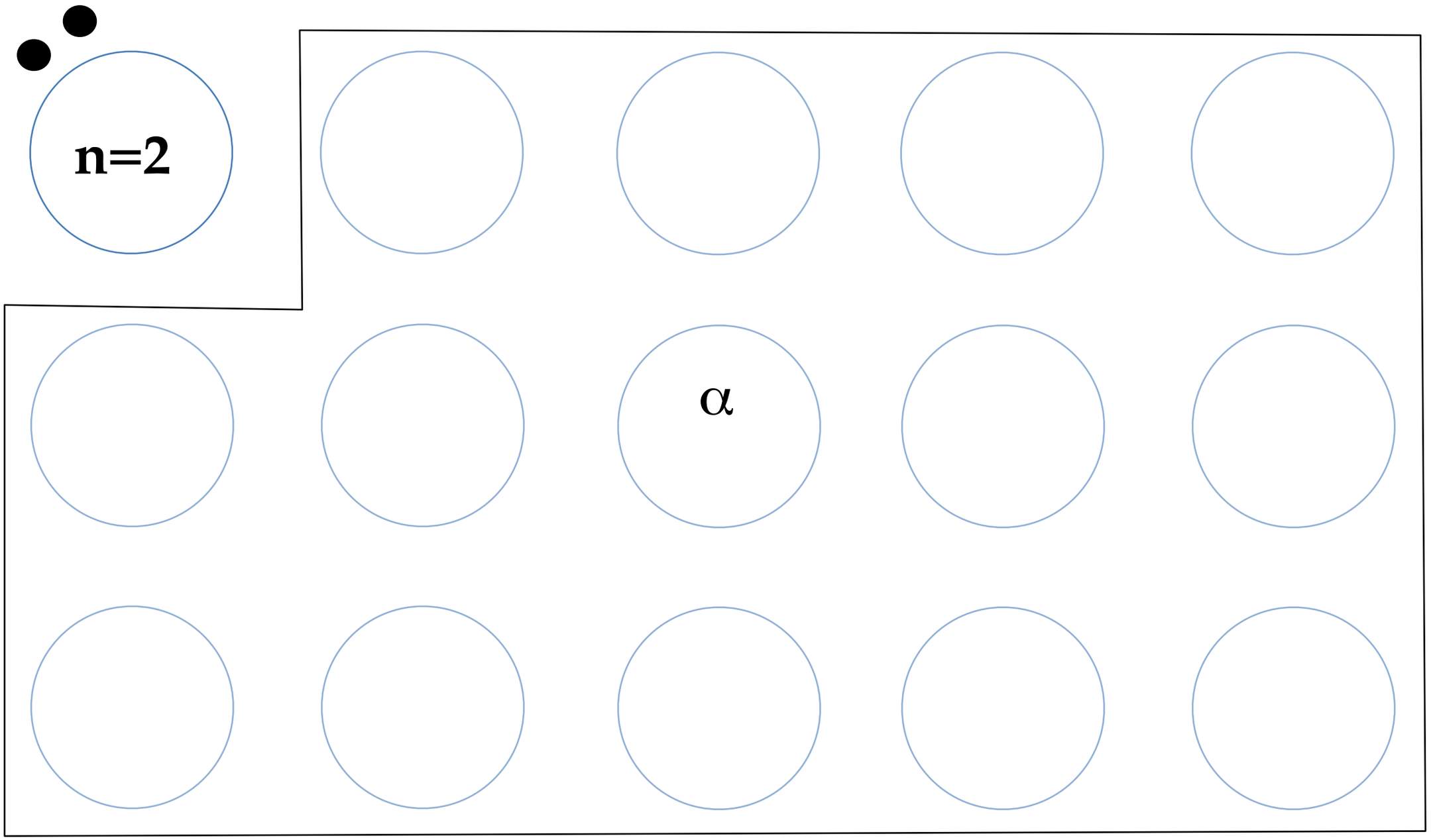
Until a customer sat at one of the tables, which makes it special…



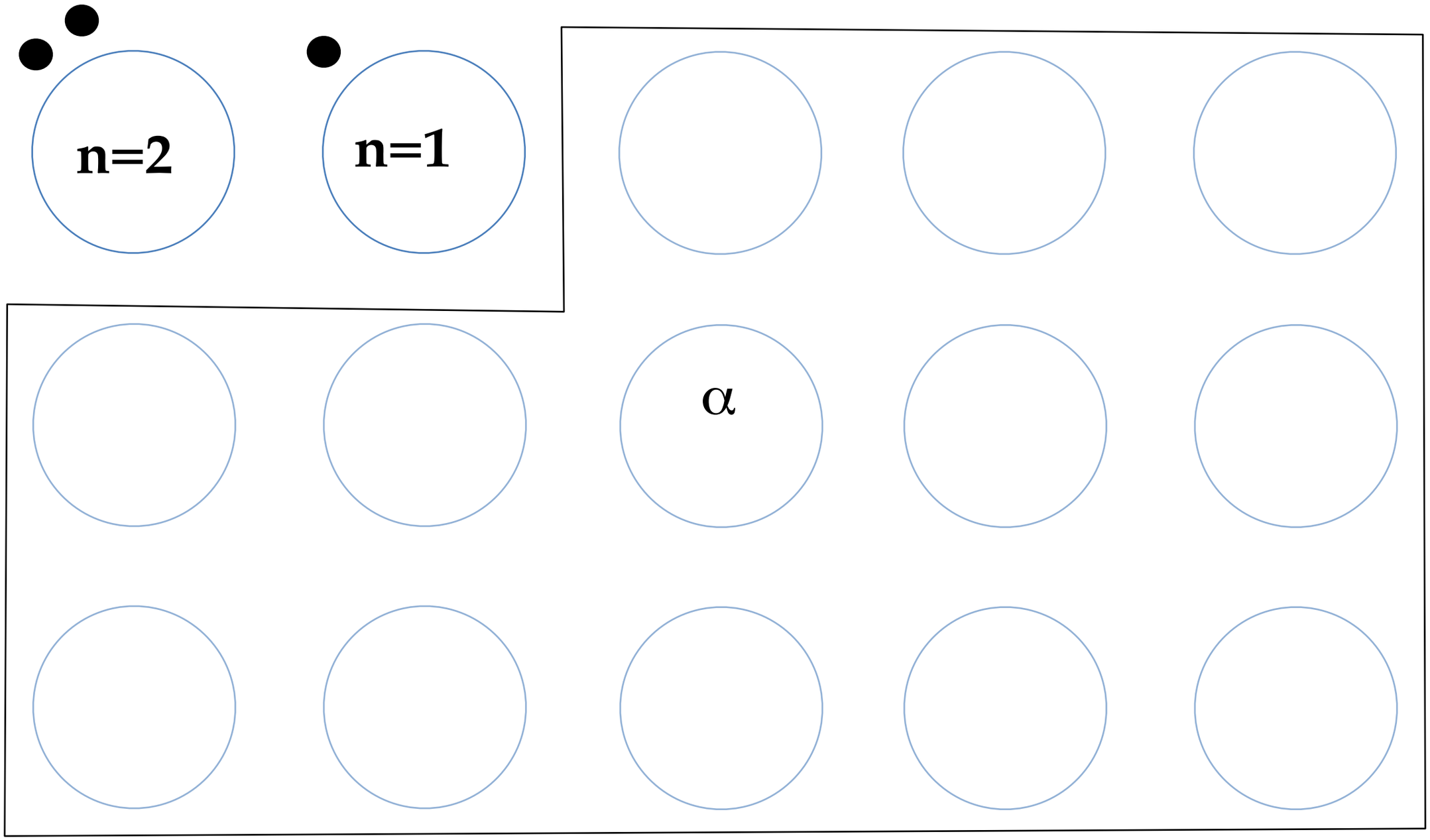
Collectively, an infinite collection of empty tables is as “attractive” as exactly α people



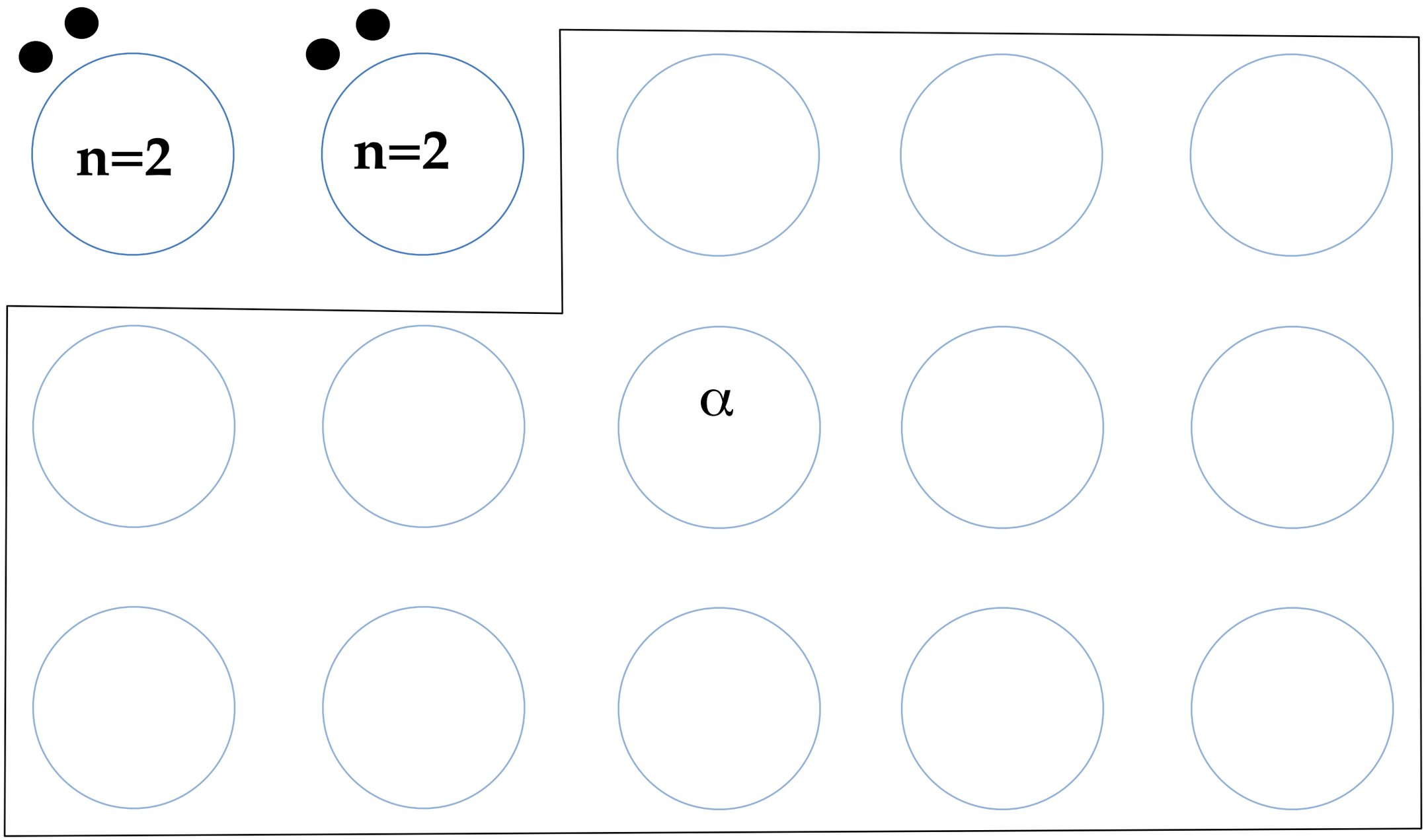
Customer 2



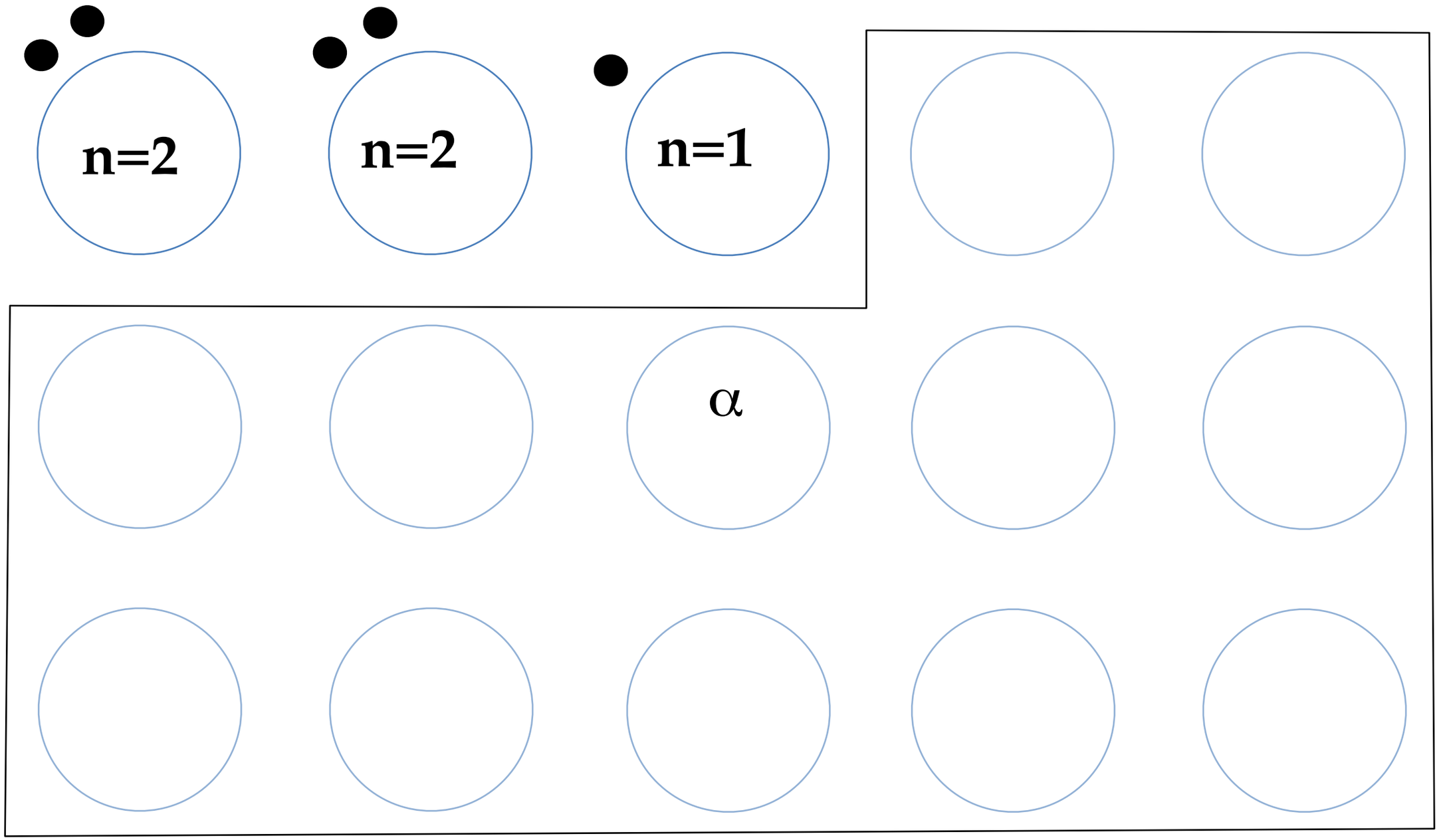
Customer 3



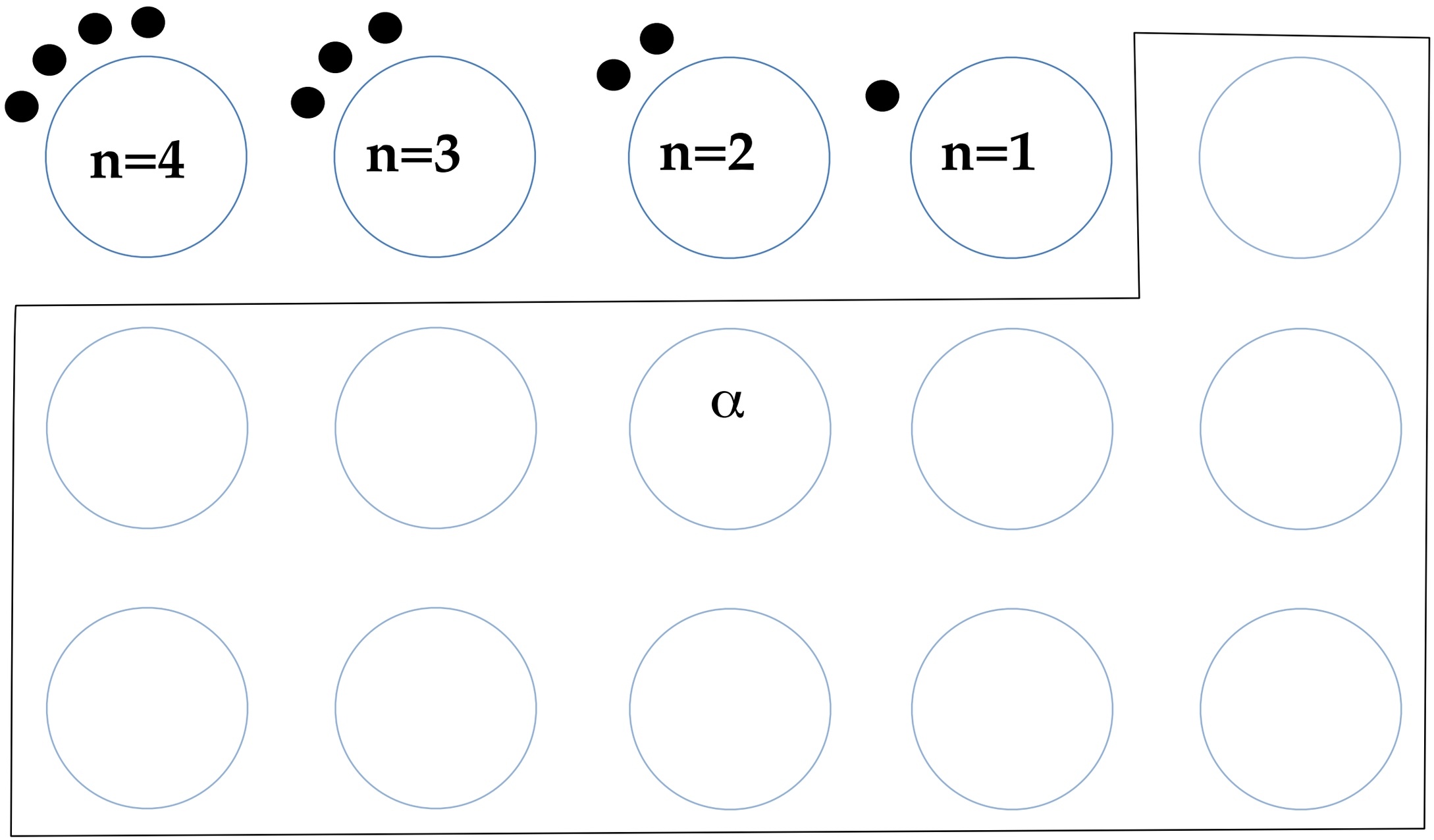
Customer 4



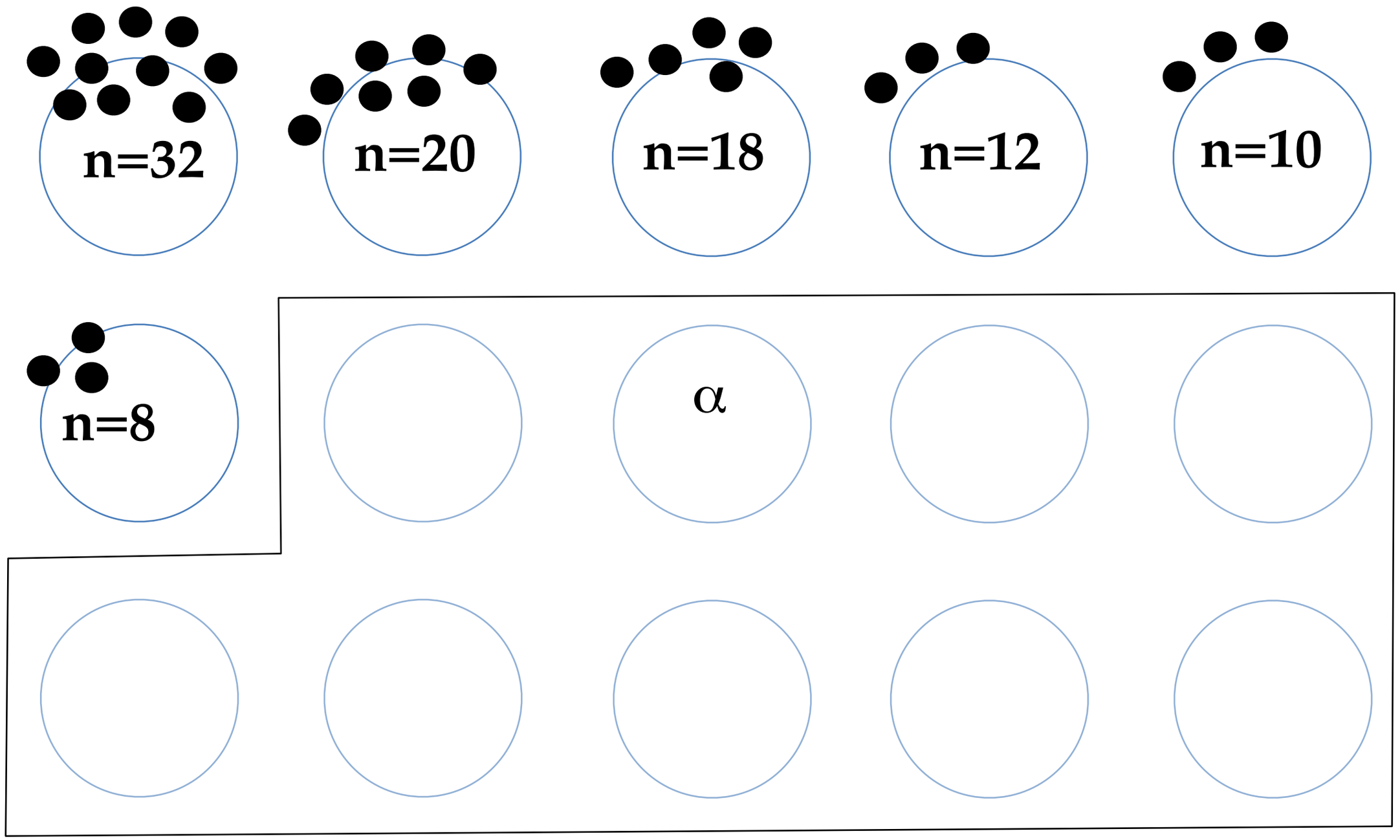
Customer 5

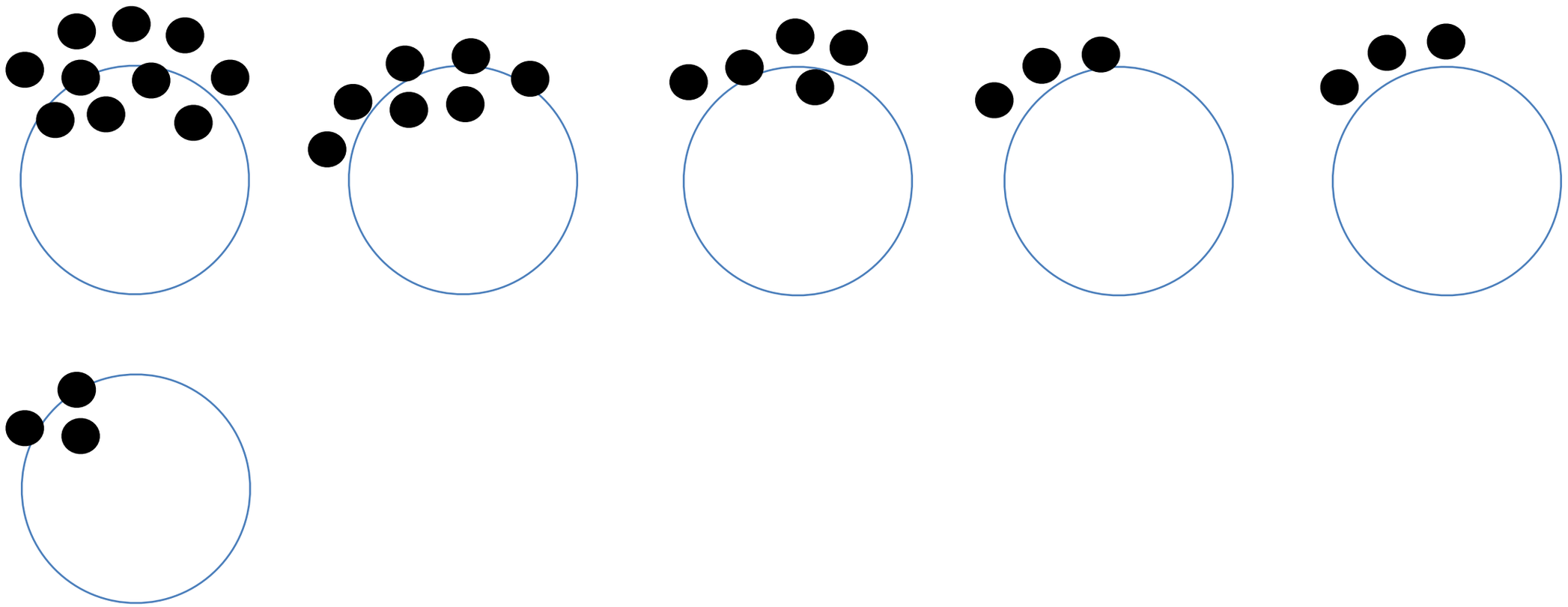


Customer 10



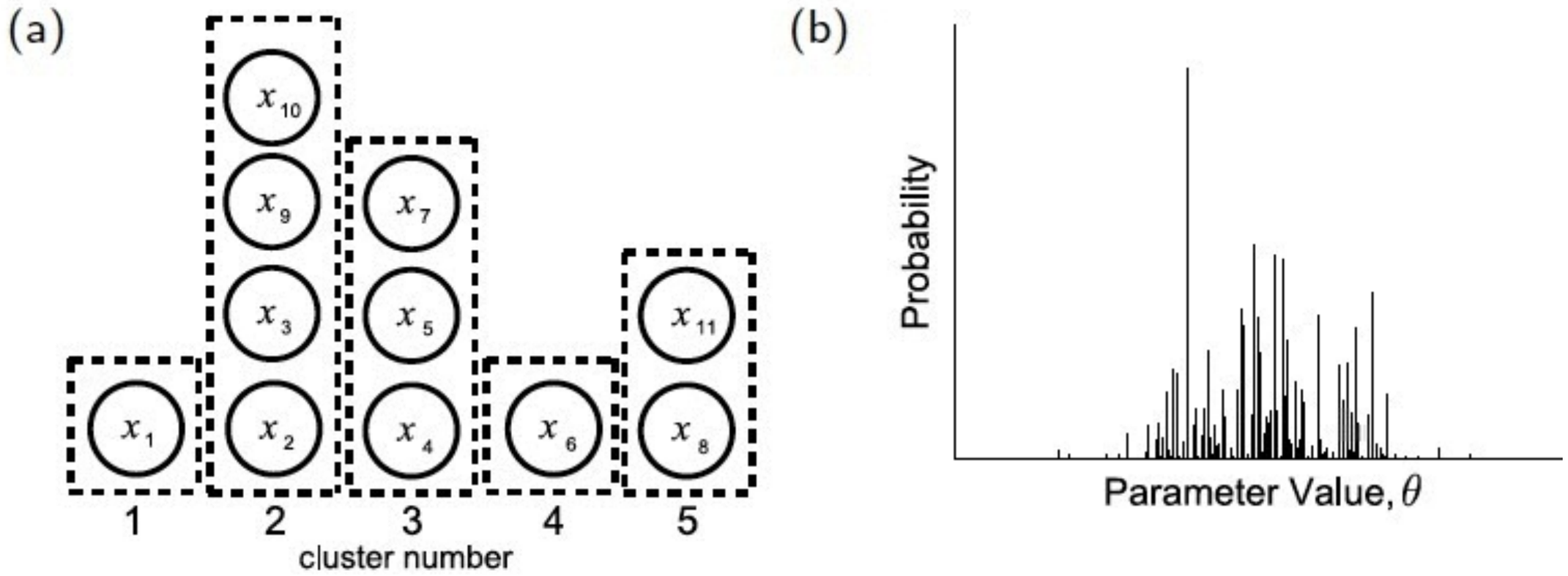
Customer 100





$$P(z_{n+1} = k | \mathbf{z}_n, \alpha) = \begin{cases} \frac{n_k}{n+\alpha} & \text{if old} \\ \frac{\alpha}{n+\alpha} & \text{if new} \end{cases}$$

From random clusterings to random probability distributions:



Beyond the scope of this class, but the CRP and similar tools are the foundations for “Bayesian nonparametrics”, which allow you to specify priors over the space of all probability distributions. (sort of)

For more details, see the tech note…

The Chinese restaurant process

COMPSCI 3016: Computational Cognitive Science
Dan Navarro & Amy Perfors
University of Adelaide

Abstract

The Chinese restaurant process (CRP) is an extremely simple and powerful tool. Unfortunately, it's also one of the most poorly described concepts in the statistical literature. This note tries to demystify the CRP. If anything the notes doesn't make sense please contact me (these notes were written by Dan: daniel.navarro@adelaide.edu.au) and I'll try to fix them!

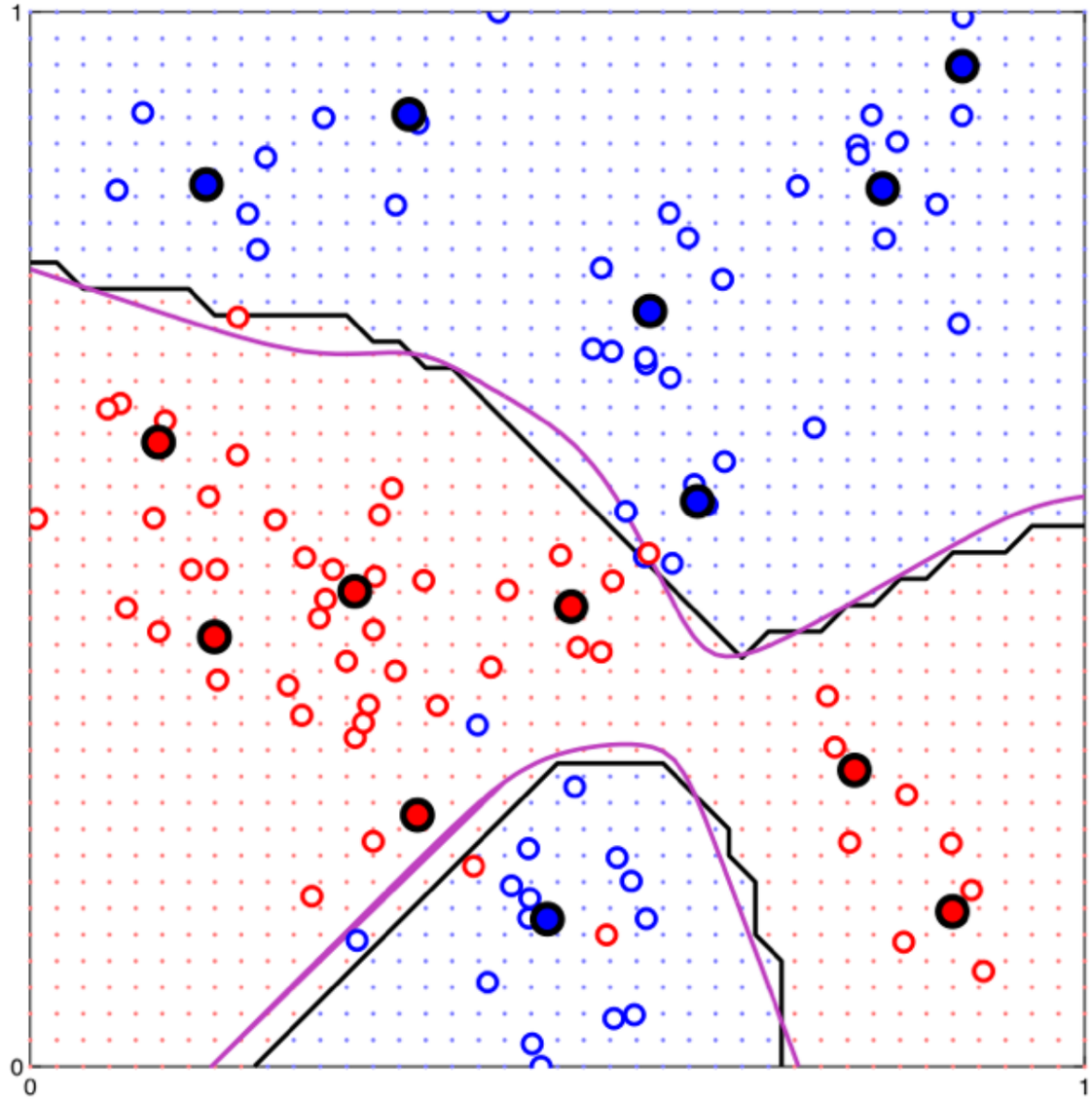
What is the Chinese restaurant process?

Reduced to the simplest possible description, the Chinese restaurant process (CRP) gives us a distribution over partitions. Suppose that we have a collection of observations, and we want to cluster/partition them into groups. We can do this using the CRP. The CRP gets its name from a metaphor based on Chinese restaurants in San Francisco that seem to have limitless seating capacity. In this metaphor, every possible group corresponds to a "table" in an infinitely large Chinese restaurant. Each observation corresponds to a "customer" entering the restaurant and sitting at a table. In this metaphor, the customers are assumed to prefer sitting at popular tables, but nevertheless there is always a non-zero probability that a new customer will sit at a currently unoccupied table. To see how this works, suppose

Anderson's "rational" model of categorisation

Mixture models

Previously, we introduced a simple heuristic for estimating a mixture model by combining k-means and k-NN



Let’s try to be more principled about it

- The model
  - Use the CRP to give us a prior over the assignment of observations to clusters
  - Use a mixture of Gaussians for the likelihood: each cluster specifies a multivariate Gaussian over feature values
  - Each cluster also specifies a probability distribution over category labels
- Numerical problems?
  - Simple approximations and good approximations

About this model

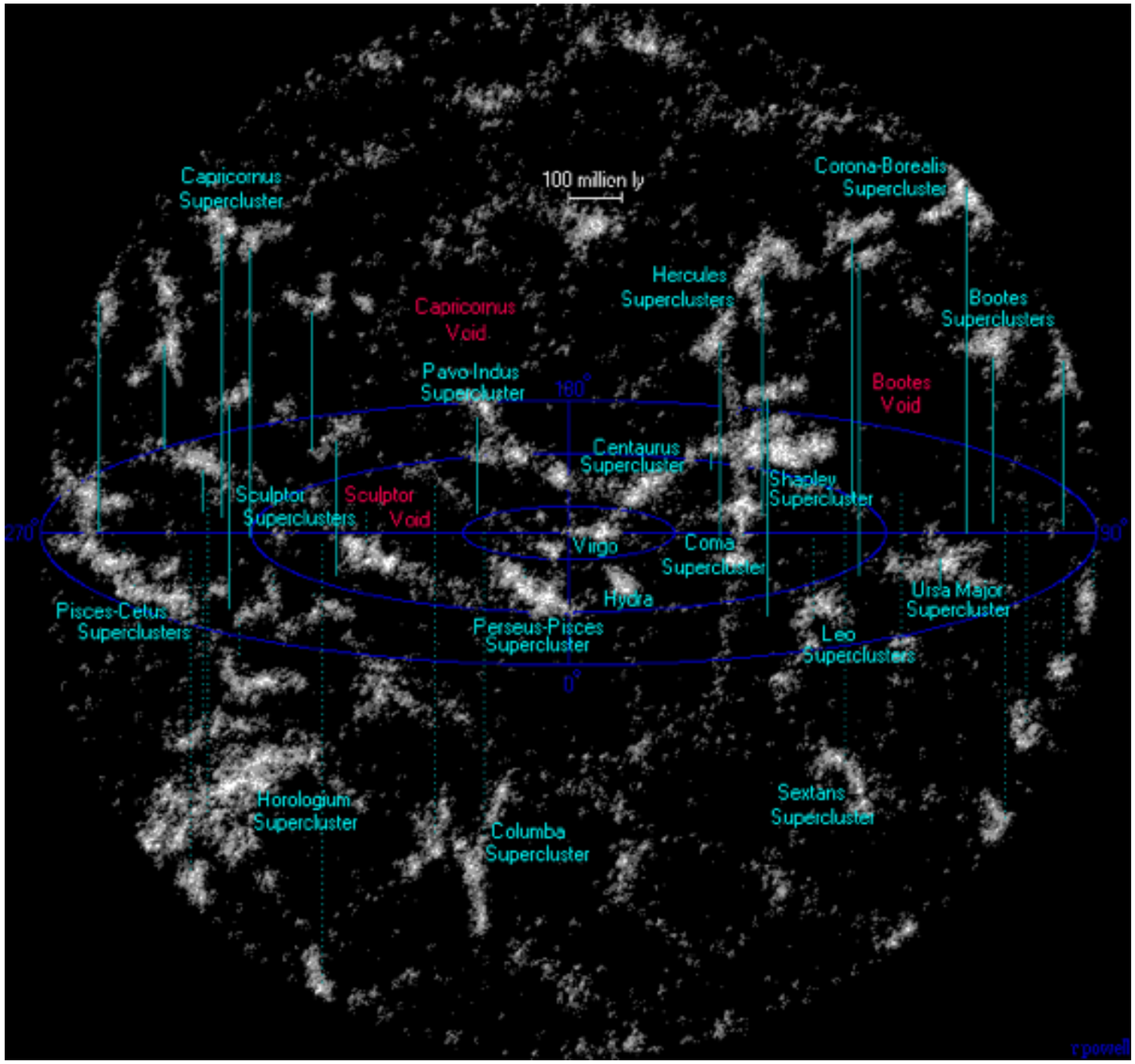
- This model has two different names:
  - Psychology: “rational model of categorisation” (RMC)
  - Machine learning: “Dirichlet process mixture model” (DPMM)
  - Developed independently in both fields

- Not too far off the state of the art: only about 20 years!
  - Developed in 1990s
  - Still used in both fields a lot
  - A lot of current work is an extension of the RMC/DPMM

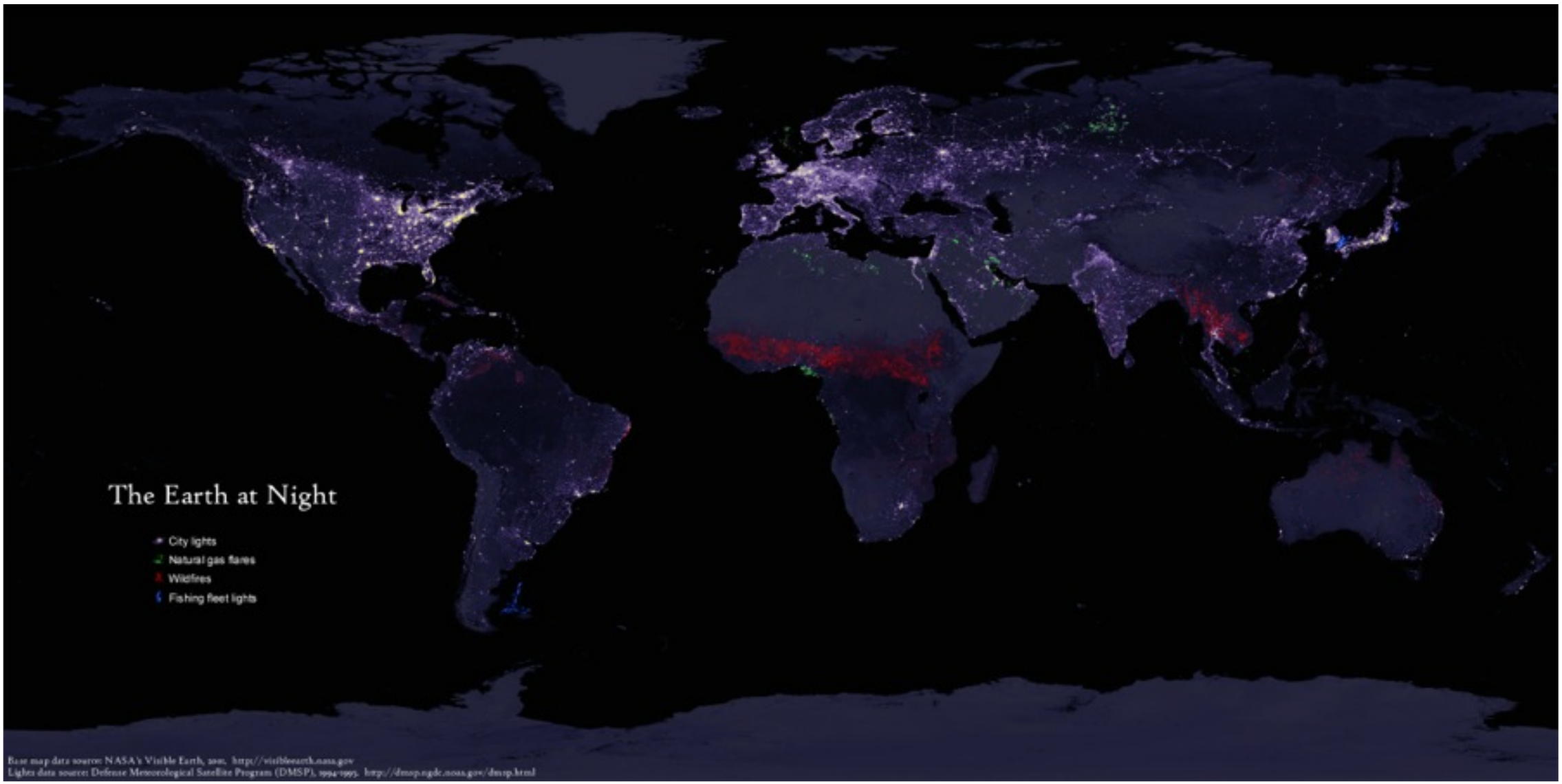
Theoretical question in psychology

- Why do people even *form* categories?
- What is their function?
  - Answer #1: **inductive power**. We assume things in the same category share similar properties. Categories are the tool we use to underpin our inductive inferences
  - Answer #2: **cognitive economy.** Instead of storing every single experience we've ever had, we use categories to simplify things. Categories are part of "psychological data compression".
  - Answer #3: **structure in the world**. The world is clumpy and categories are a natural way of describing this clumpiness.

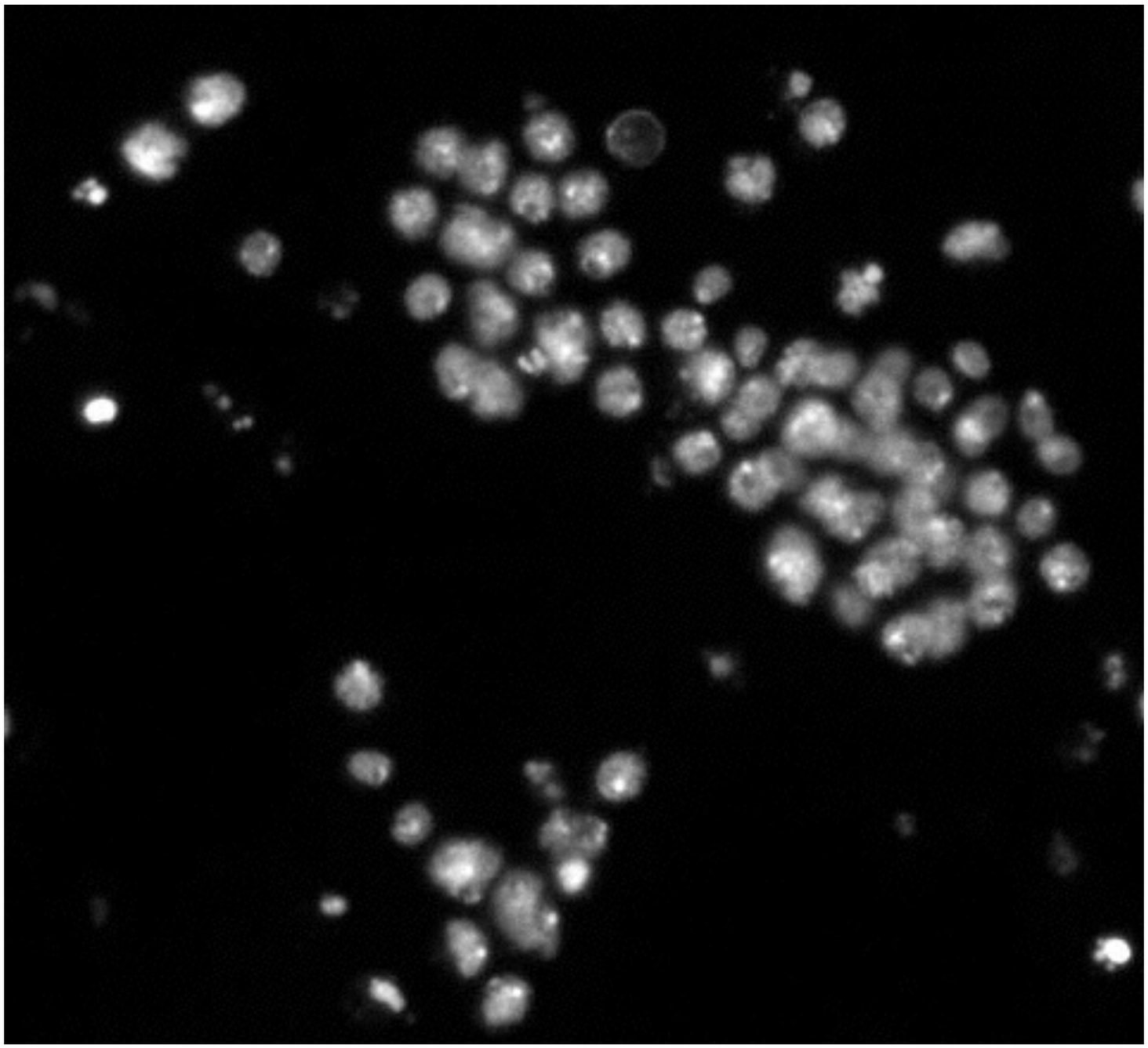
Stars cluster together



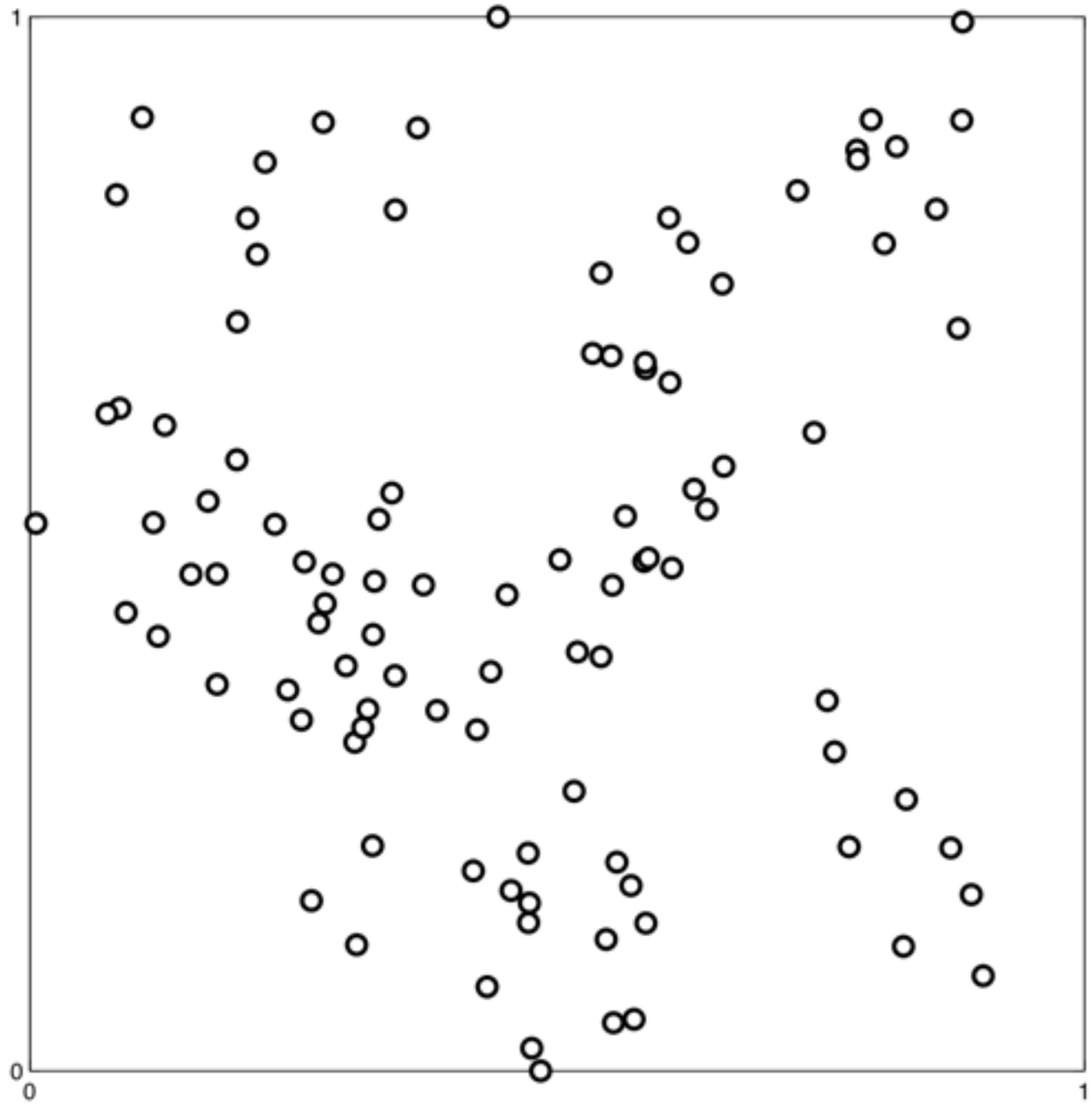
People cluster together



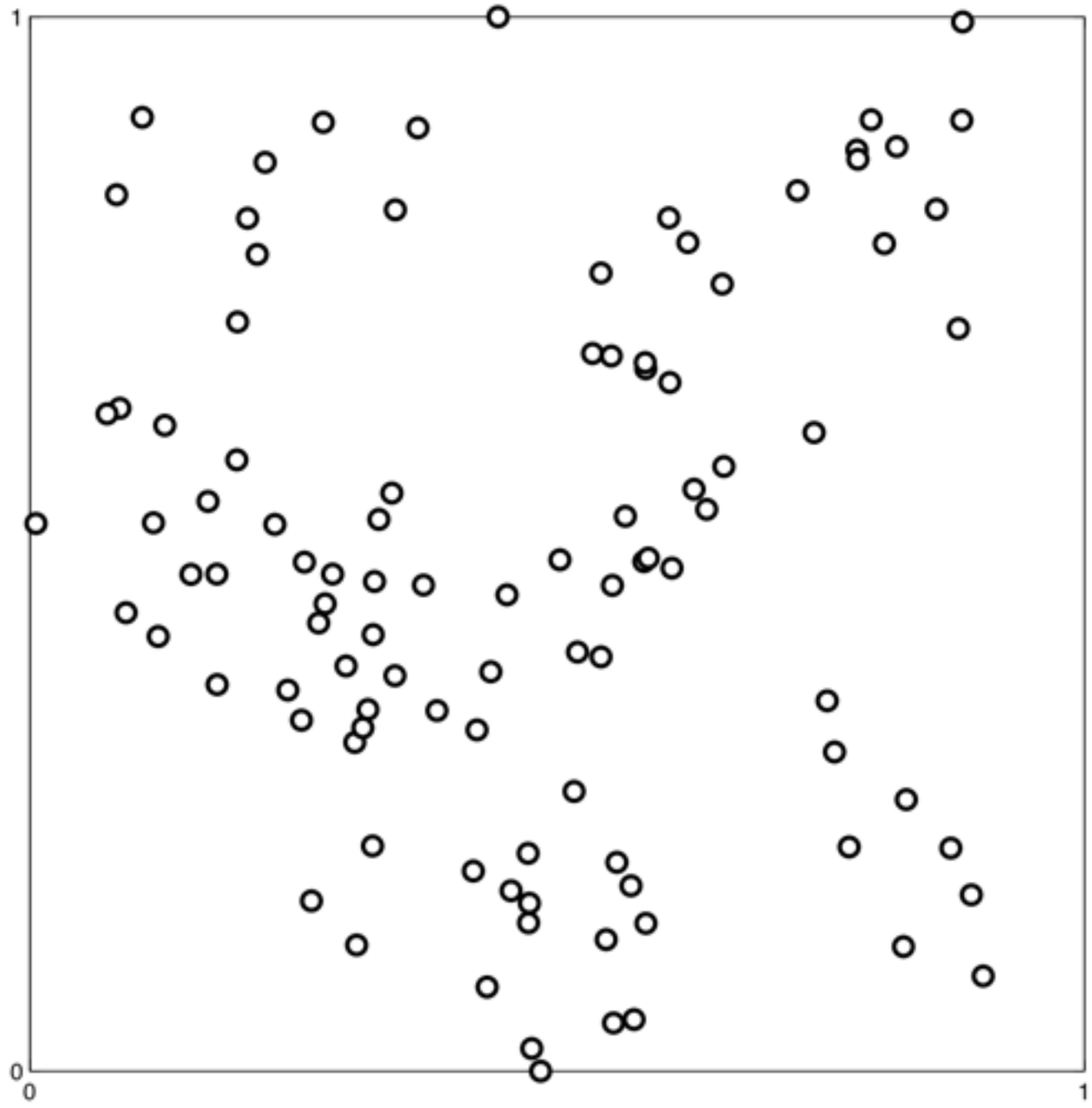
Cells cluster together



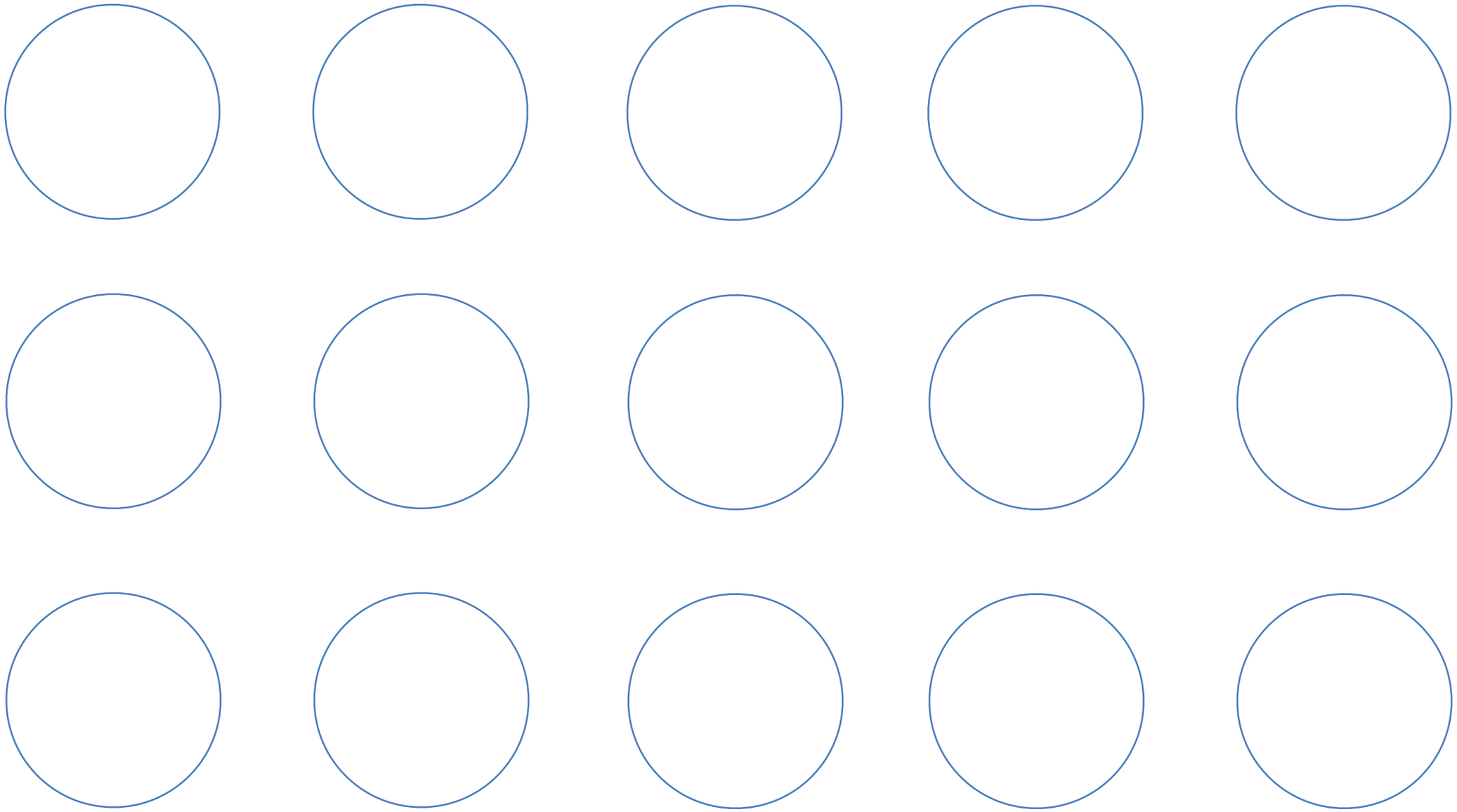
Data cluster together



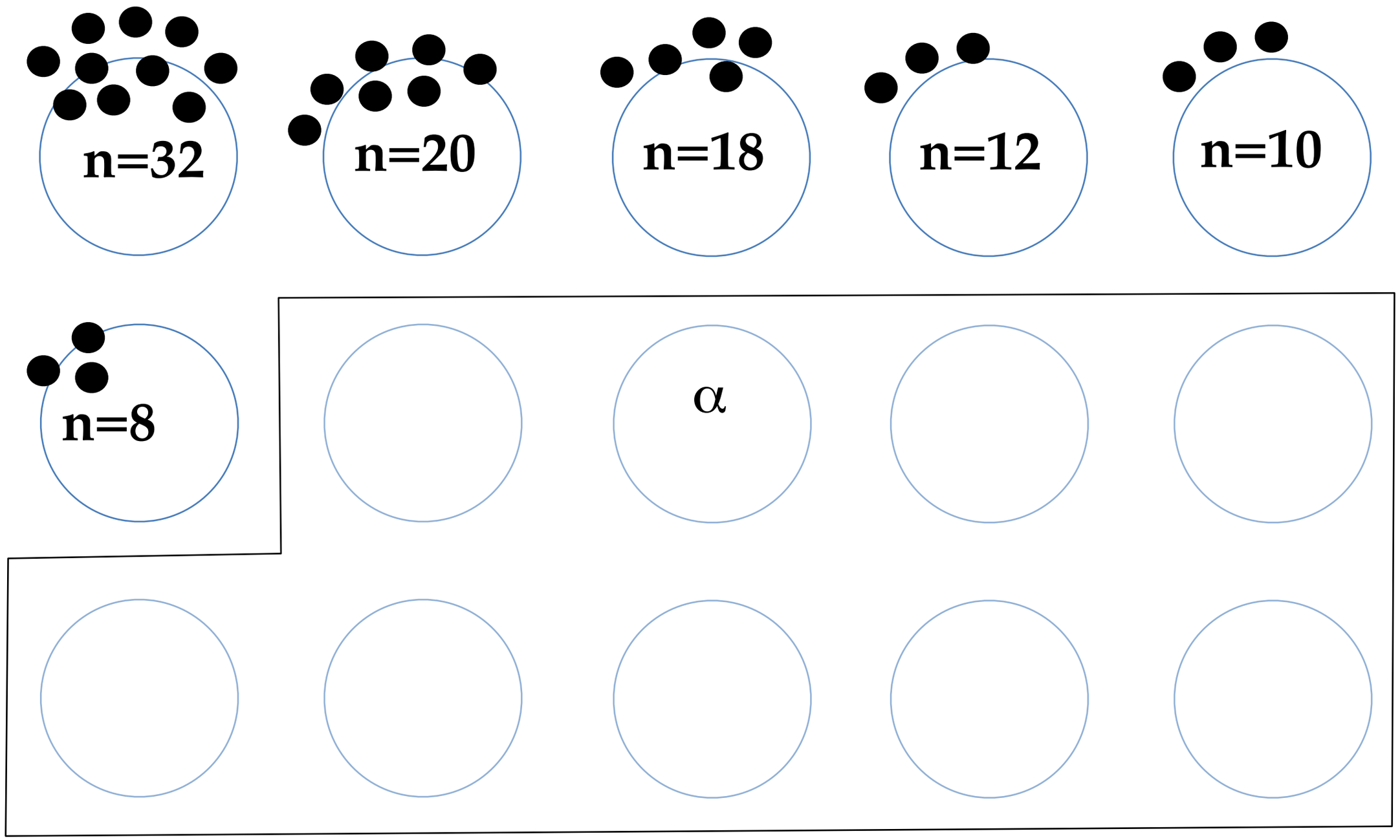
What we want is not just a “labelling system”, but rather a “story” about how our data came to look like this…

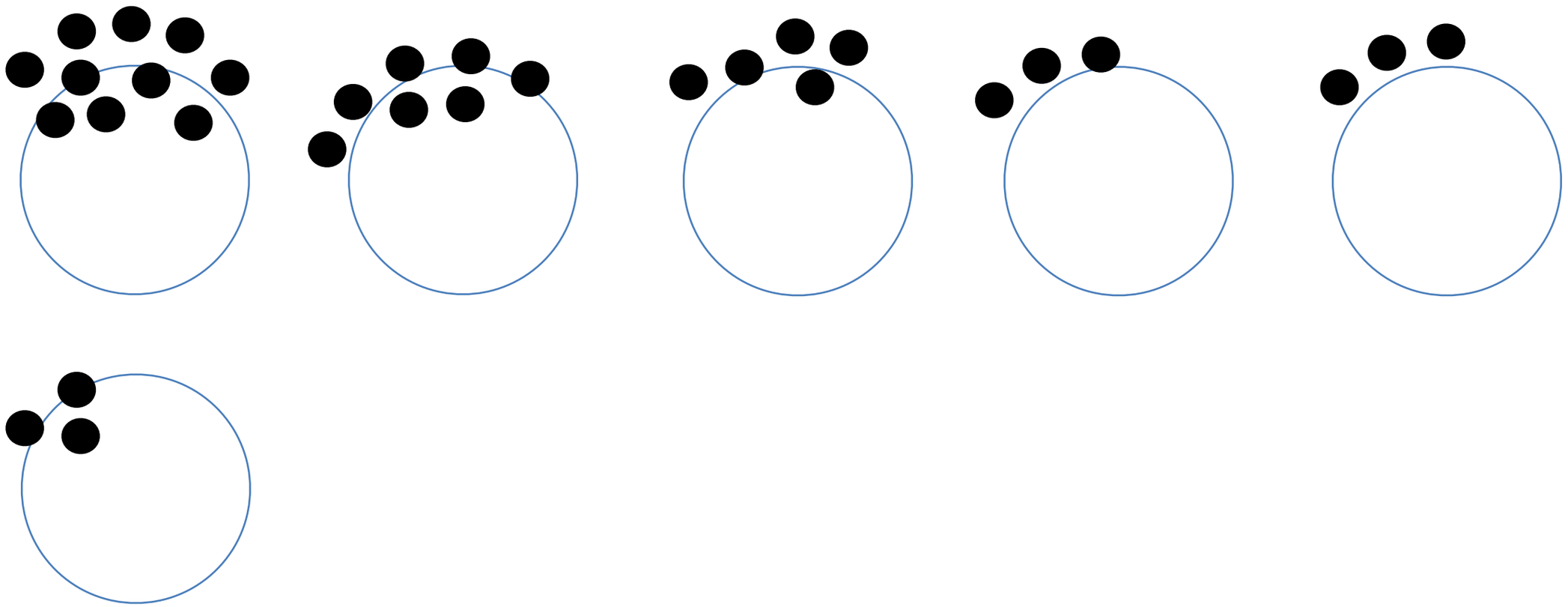


We start with an infinite blank slate

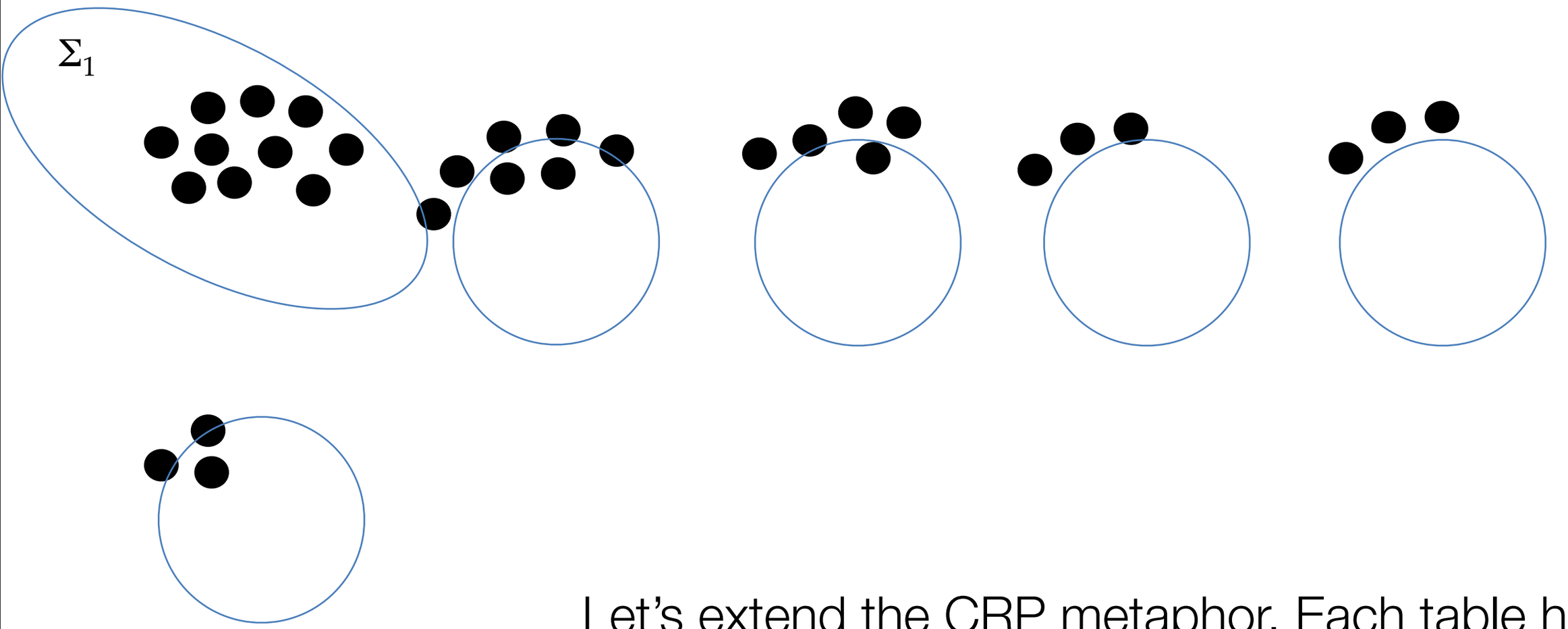


We imagine our data arriving via the CRP…

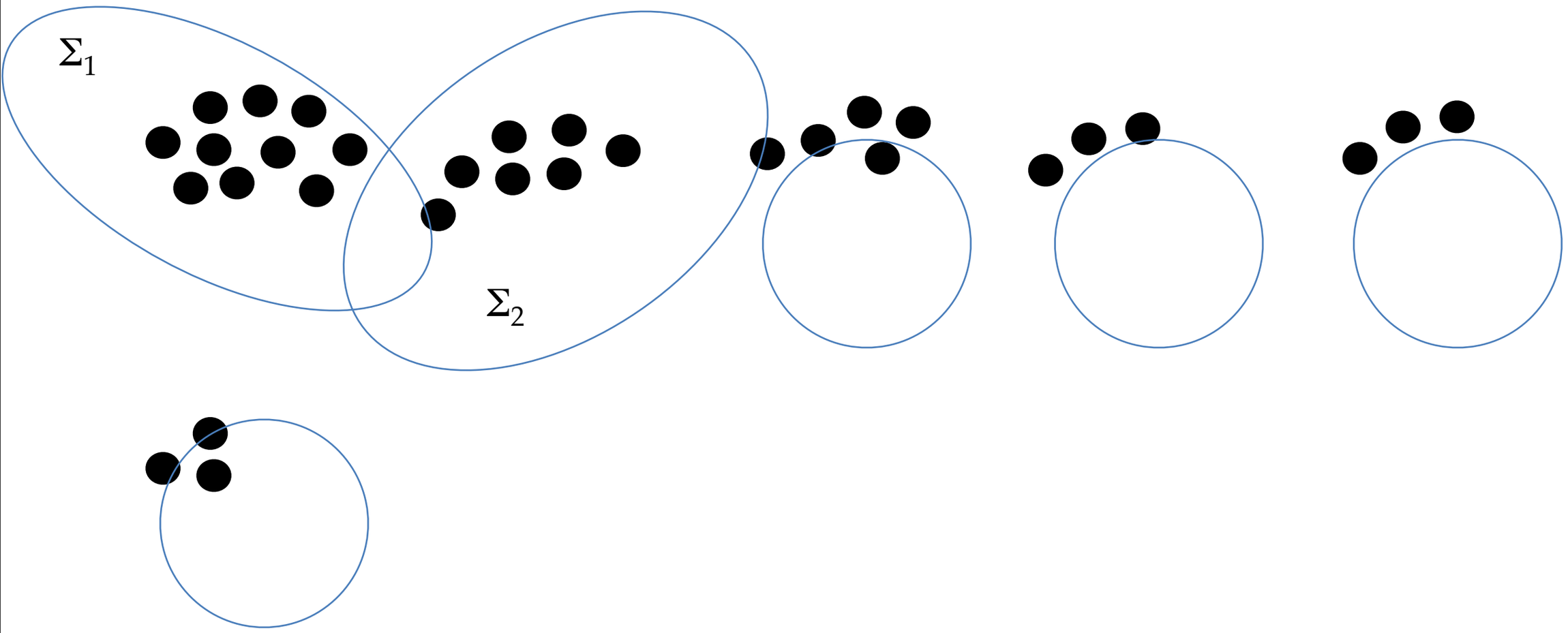


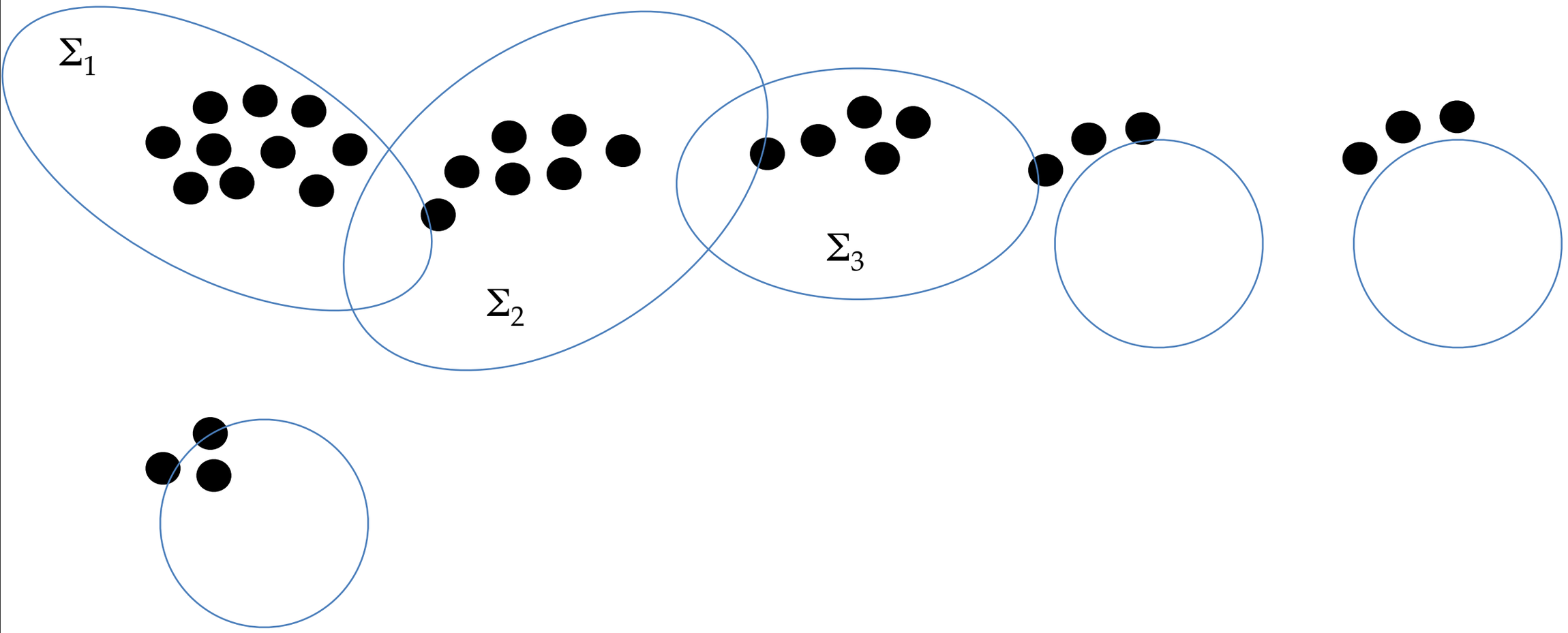


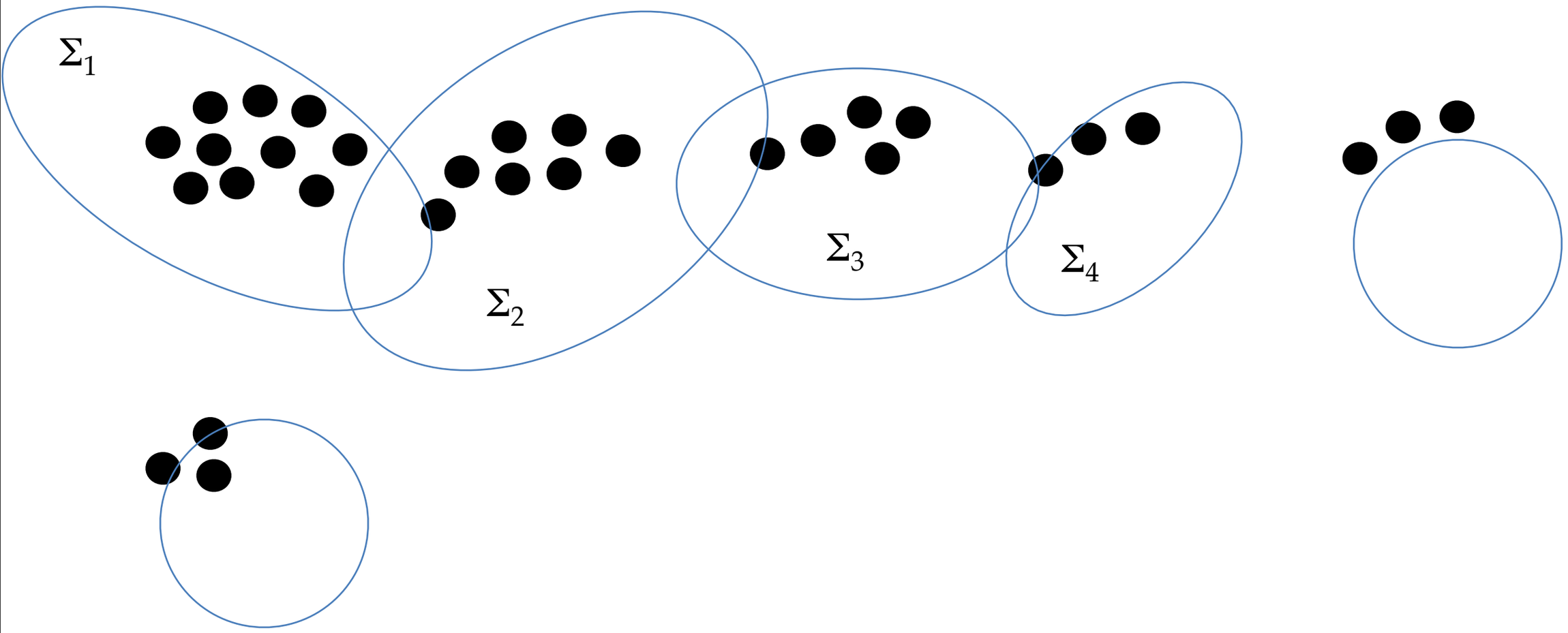
$$P(z_{n+1} = k | \mathbf{z}_n, \alpha) = \begin{cases} \frac{n_k}{n+\alpha} & \text{if old} \\ \frac{\alpha}{n+\alpha} & \text{if new} \end{cases}$$

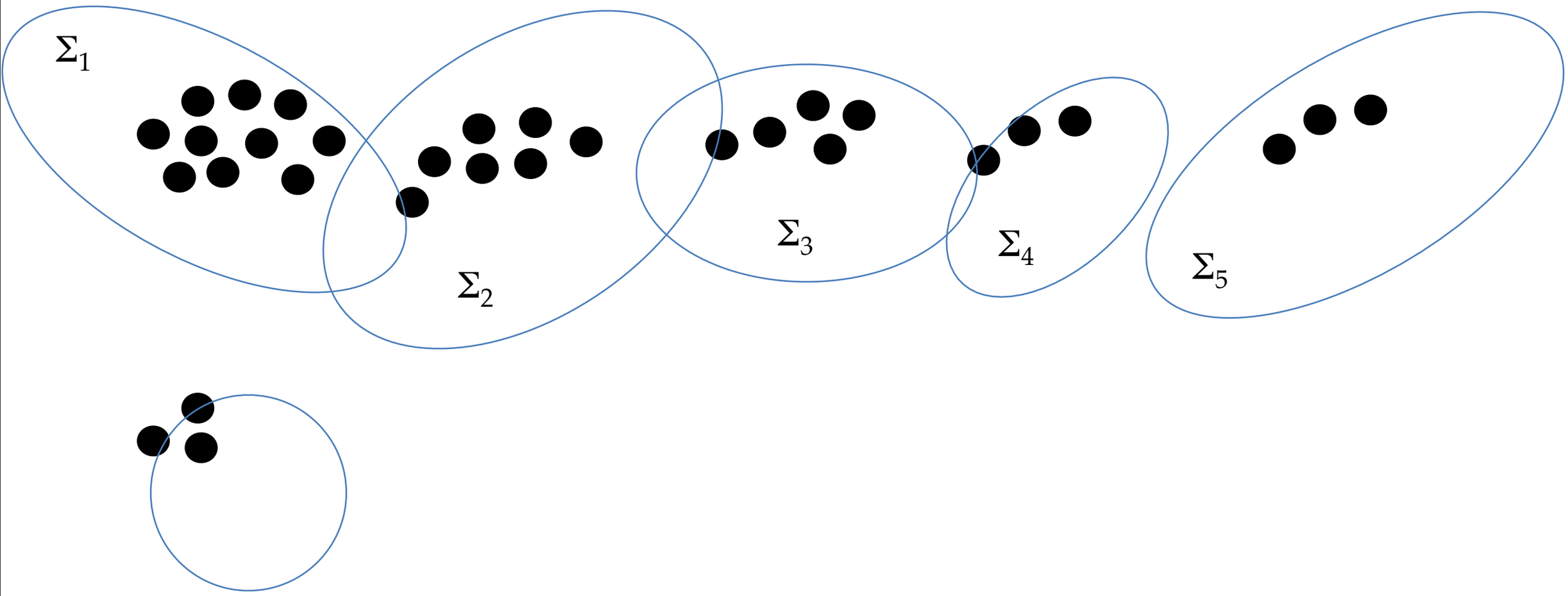


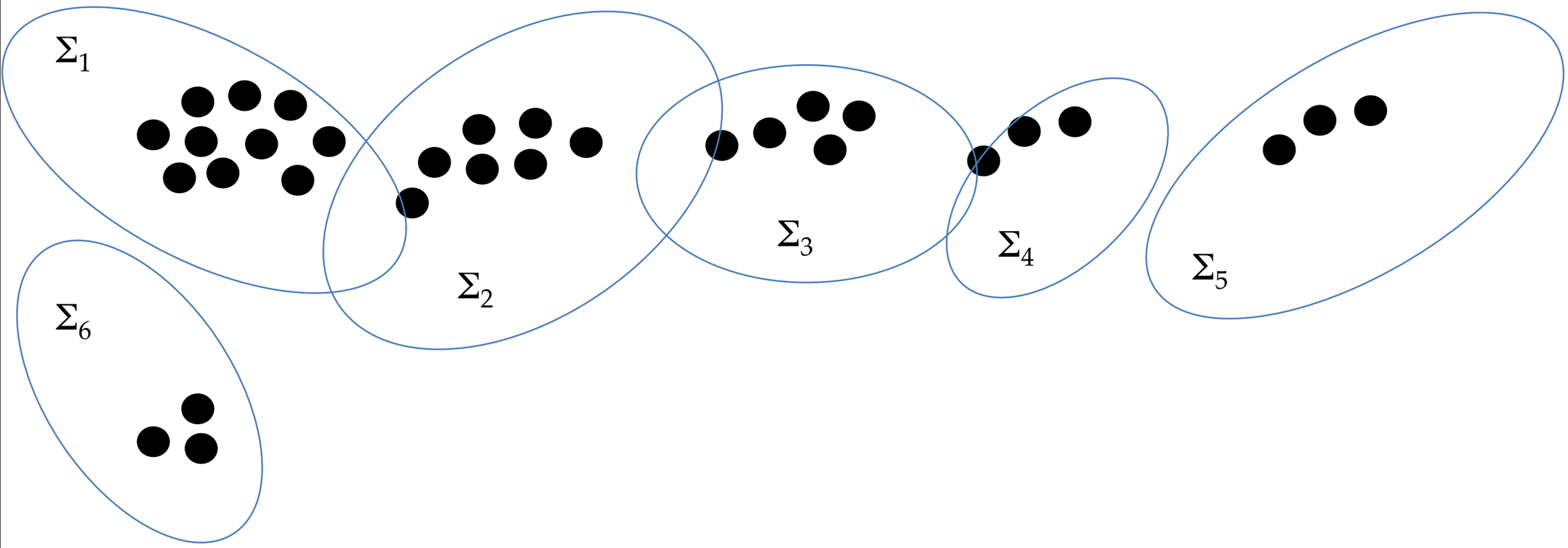
Let’s extend the CRP metaphor. Each table has a different elliptical **shape** (described by a covariance matrix Σ)

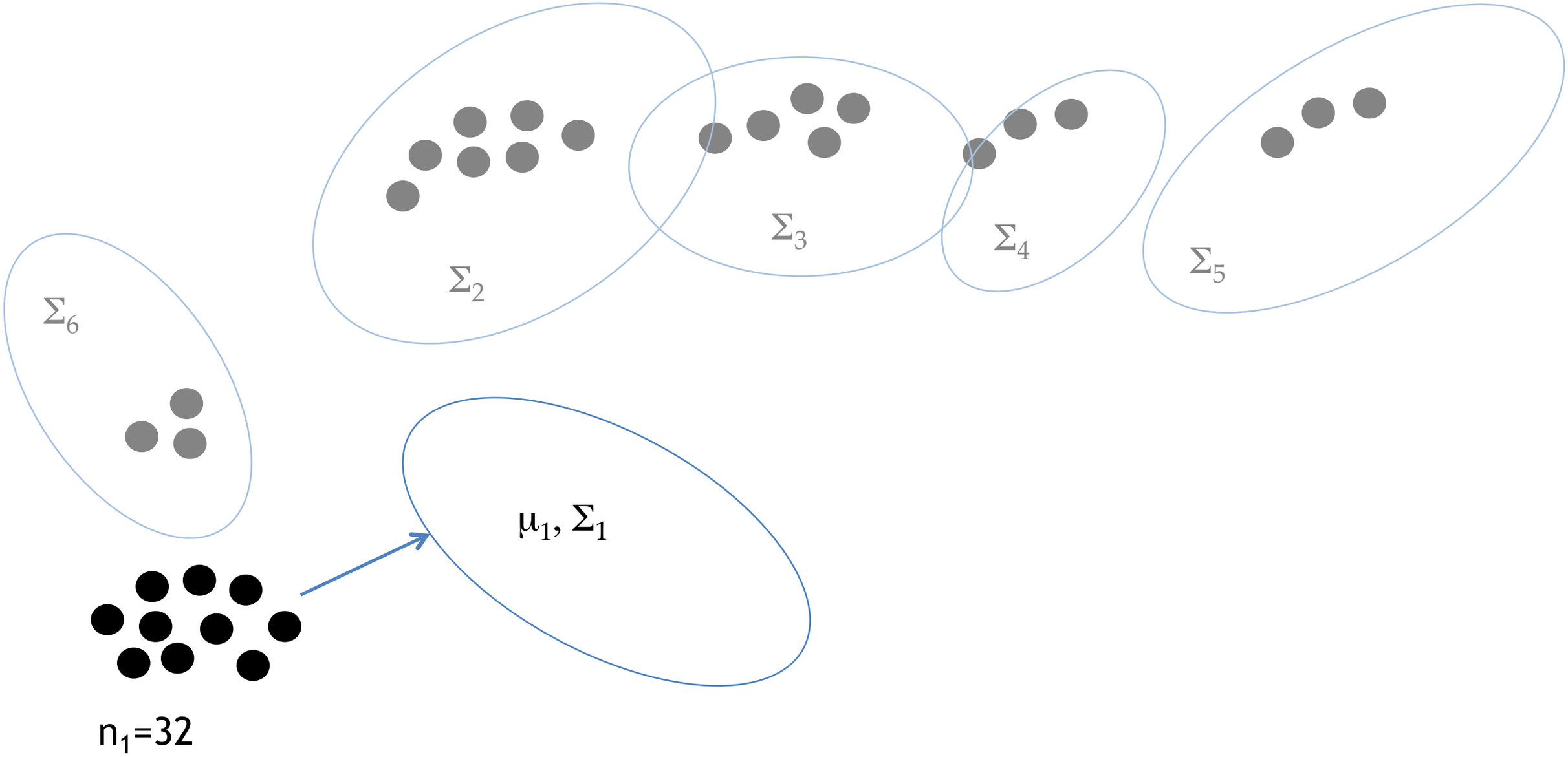

Σ_1
Σ_2


Σ_1
Σ_2
Σ_3

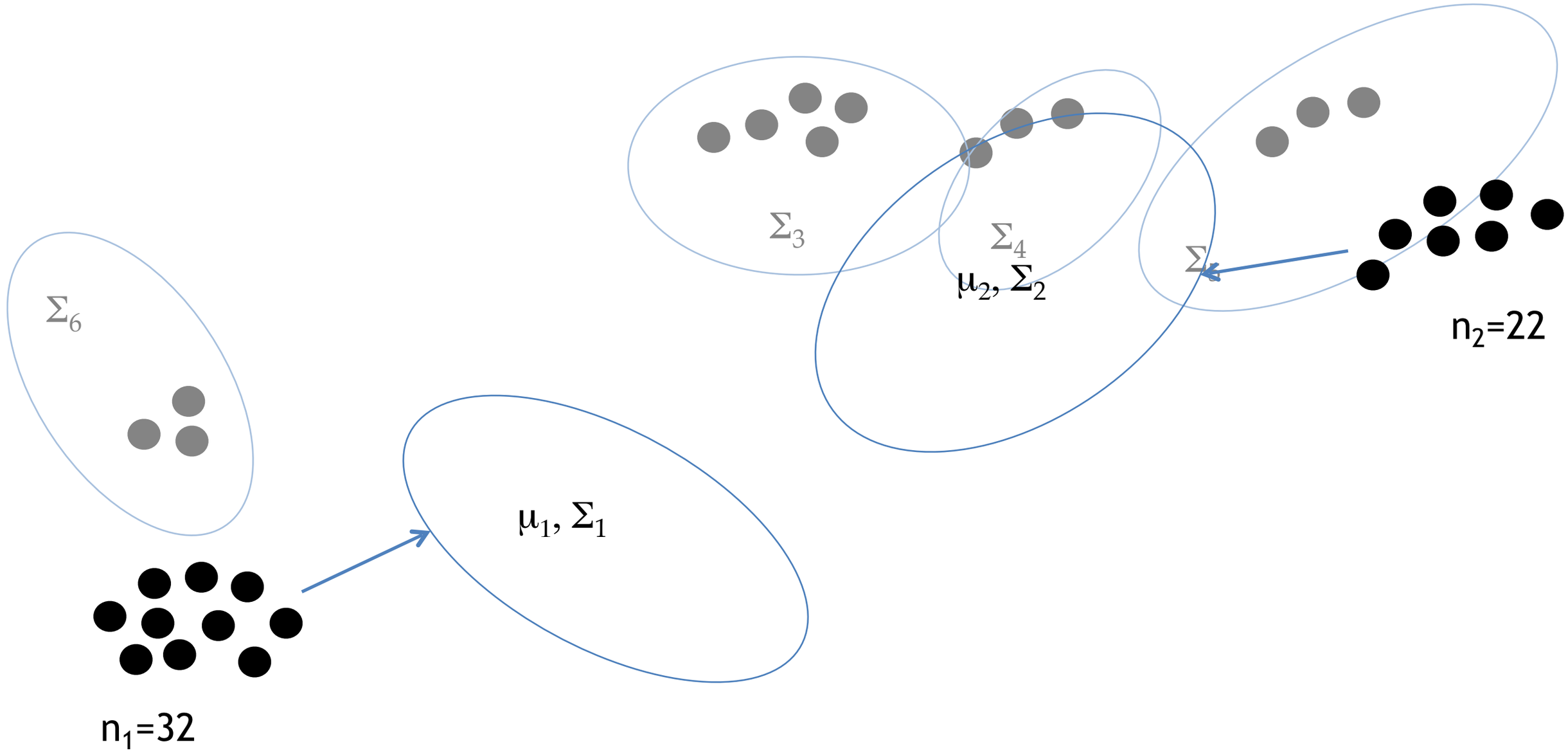

Σ_1
Σ_2
Σ_3
Σ_4

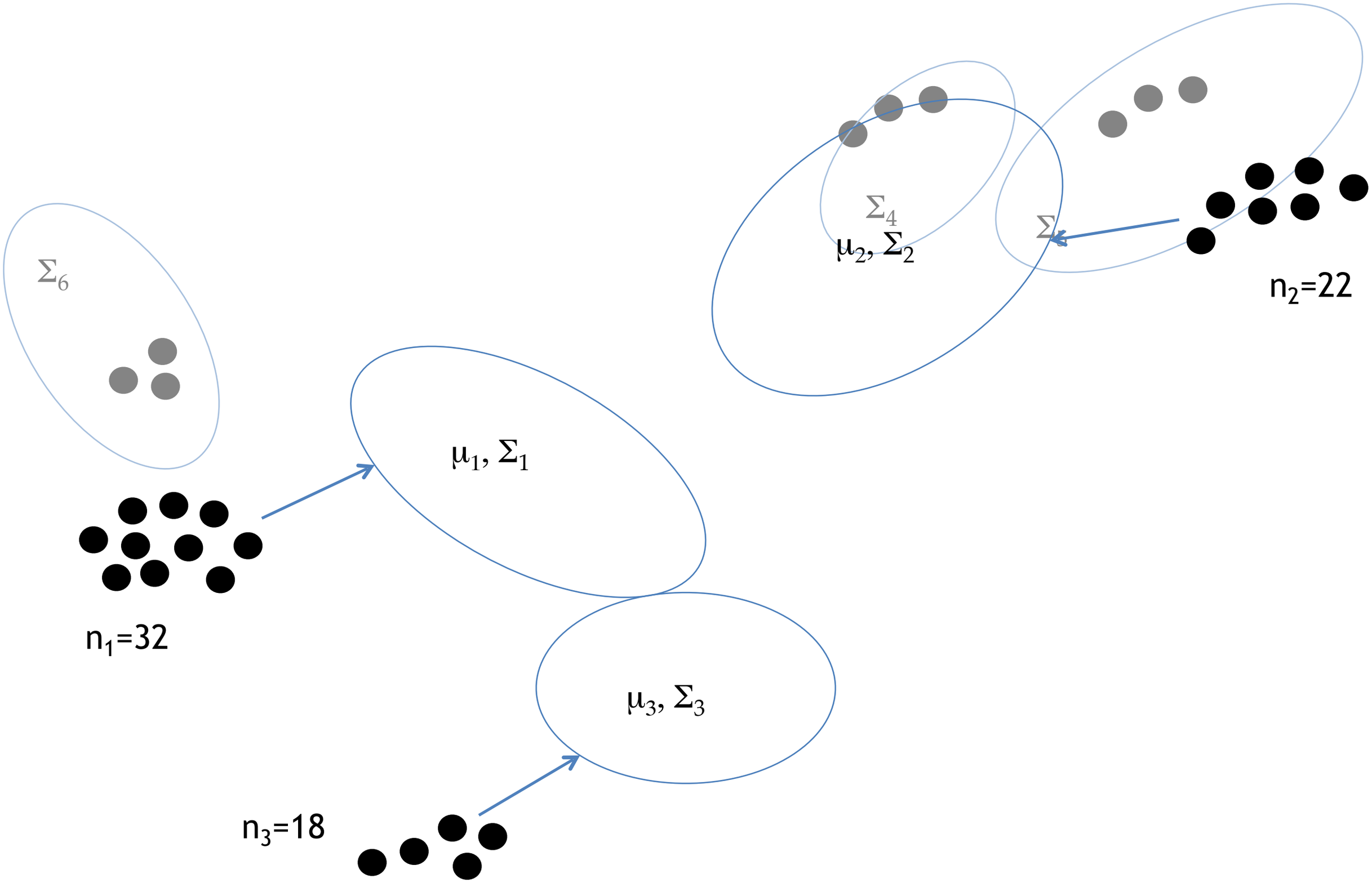

Σ_1
Σ_2
Σ_3
Σ_4
Σ_5

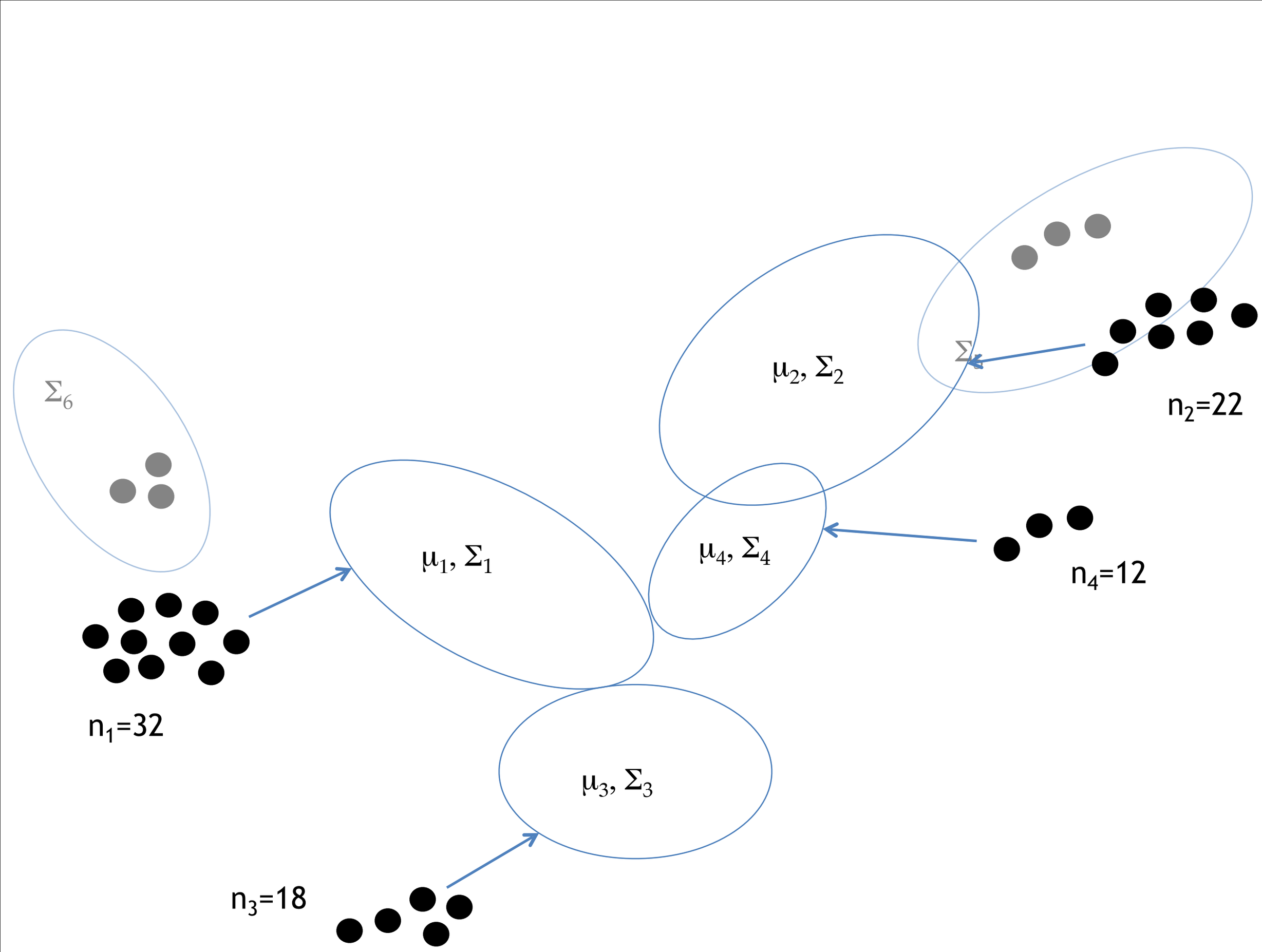

Σ_1
Σ_2
Σ_3
Σ_4
Σ_5
Σ_6

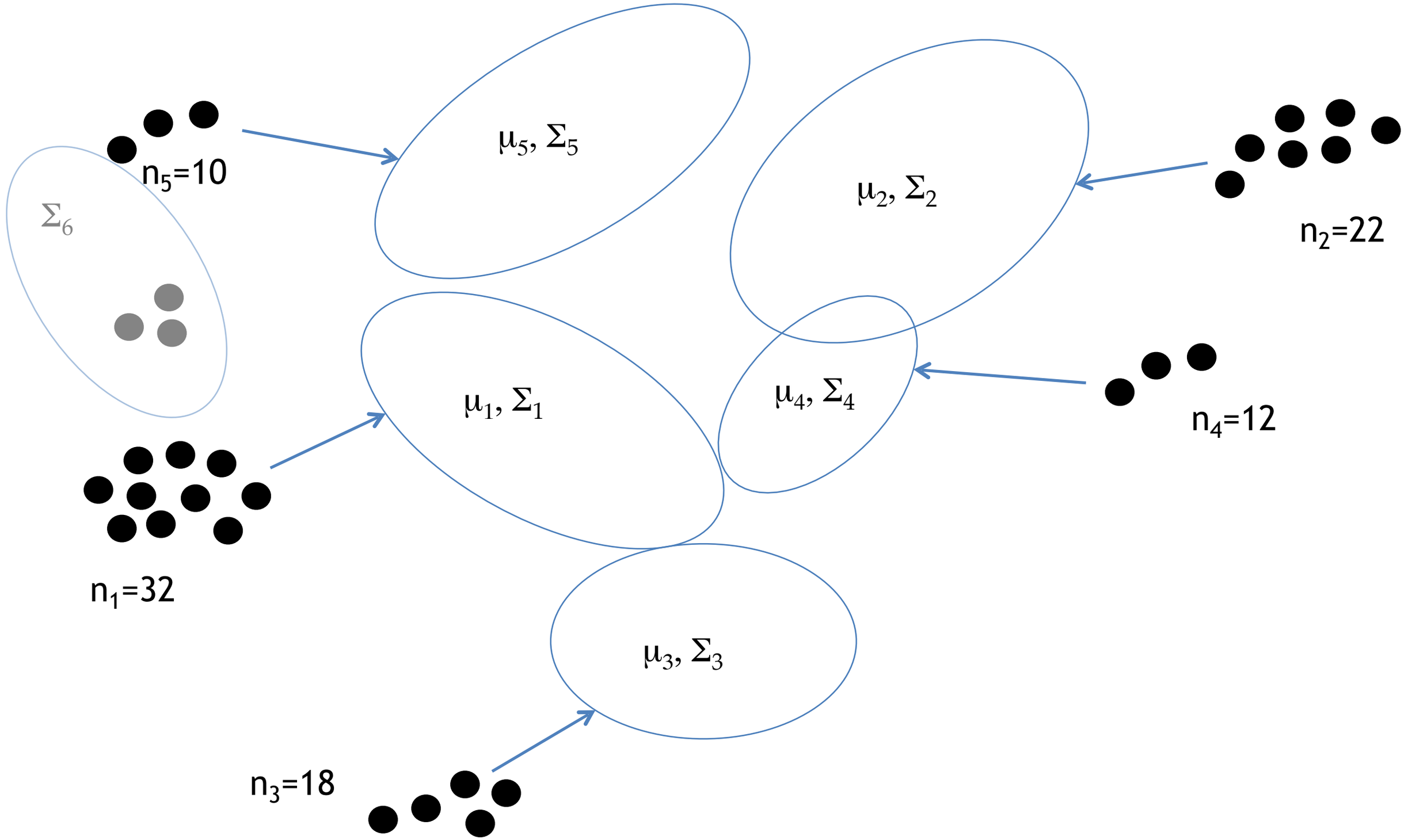


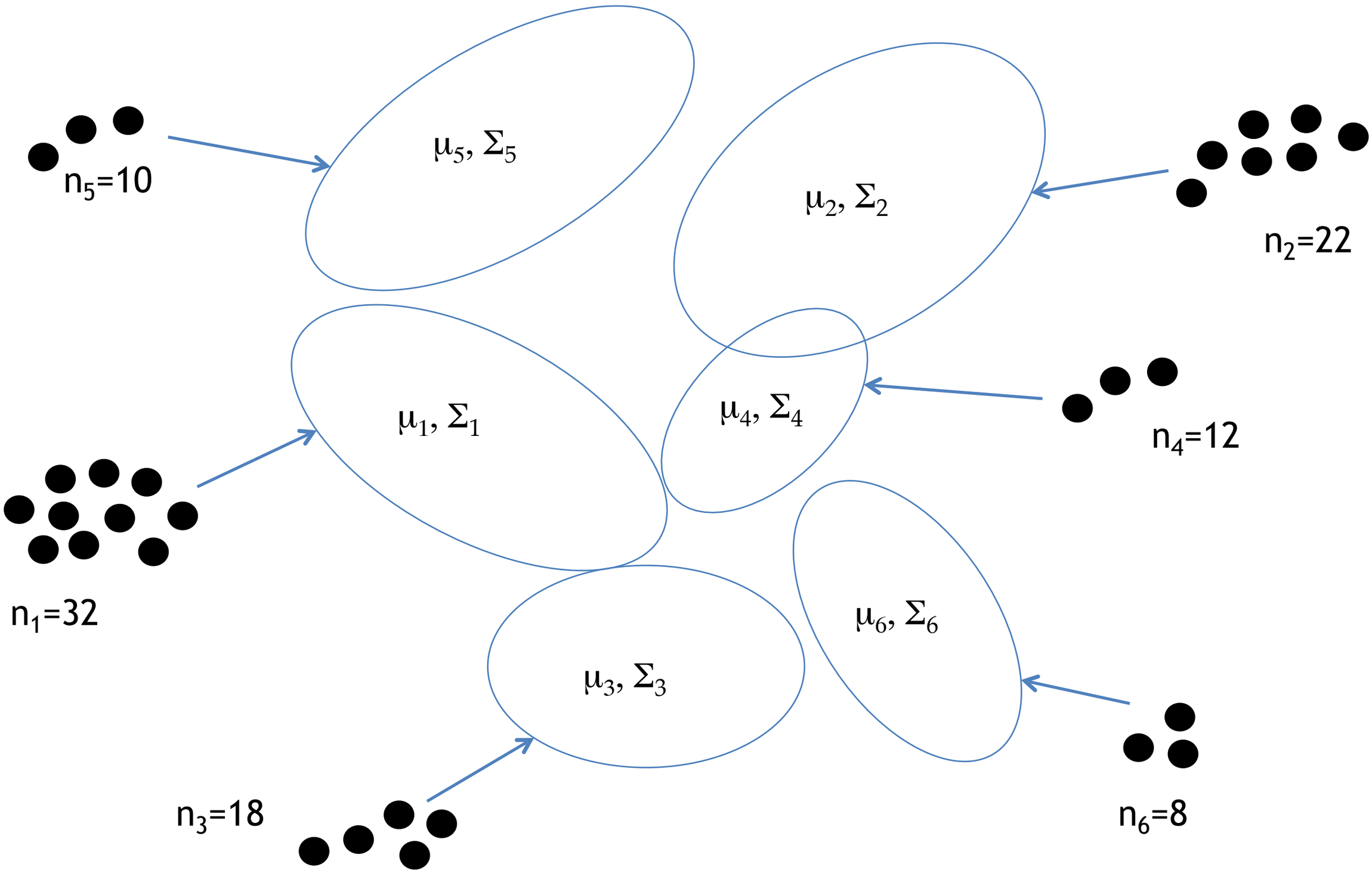
Tables don't just have shapes. Each table has a **location** in the restaurant (described by mean μ)


Σ_3
Σ_4
μ_2, Σ_2
Σ_6
n_2=22
μ_1, Σ_1
n_1=32

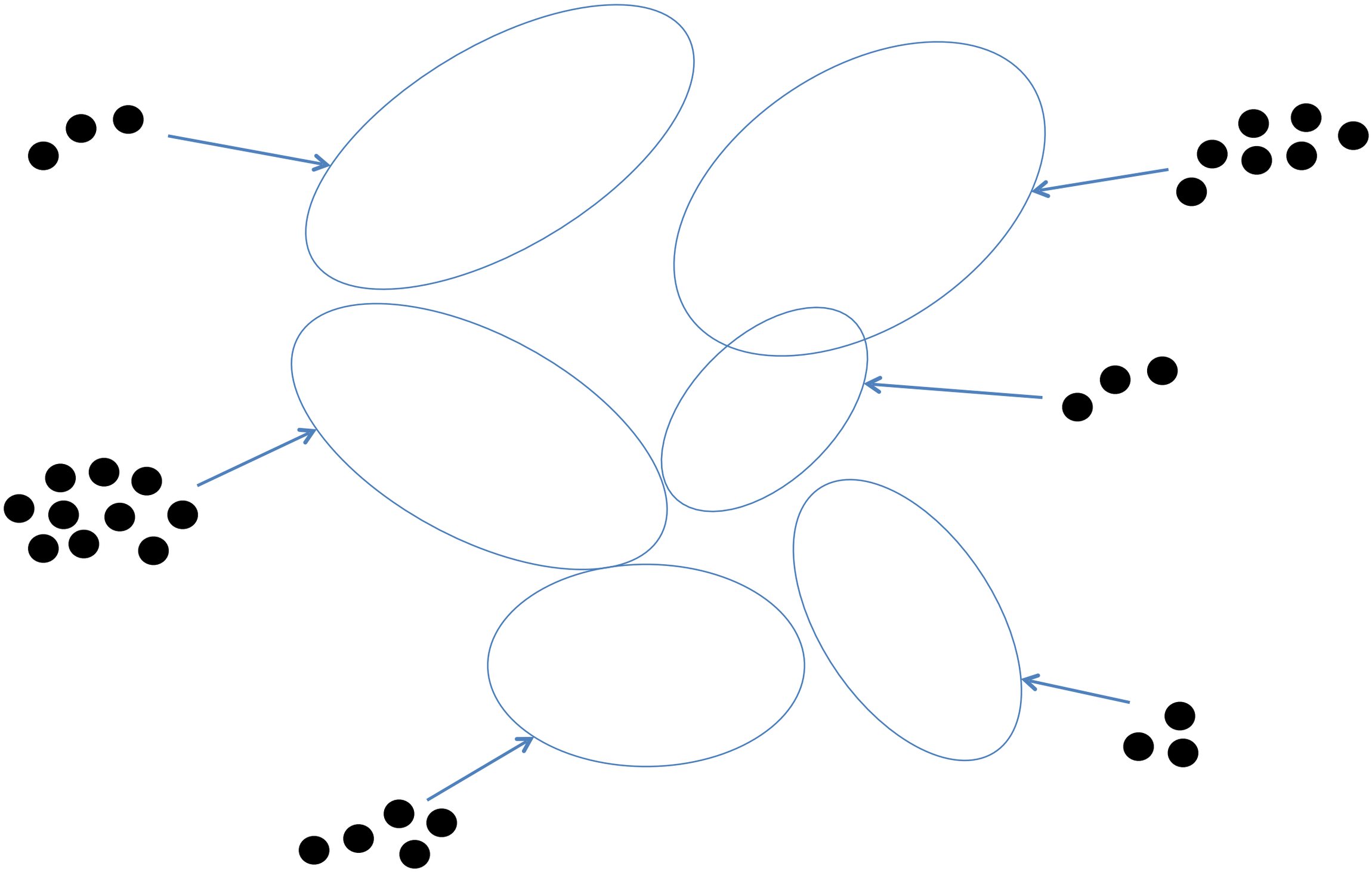

Σ_6
Σ_4
μ_2, Σ_2
Σ_2
n_2=22
μ_1, Σ_1
n_1=32
μ_3, Σ_3
n_3=18


Σ_6
μ_2, Σ_2
Σ_2
n_2=22
μ_1, Σ_1
μ_4, Σ_4
n_4=12
n_1=32
μ_3, Σ_3
n_3=18

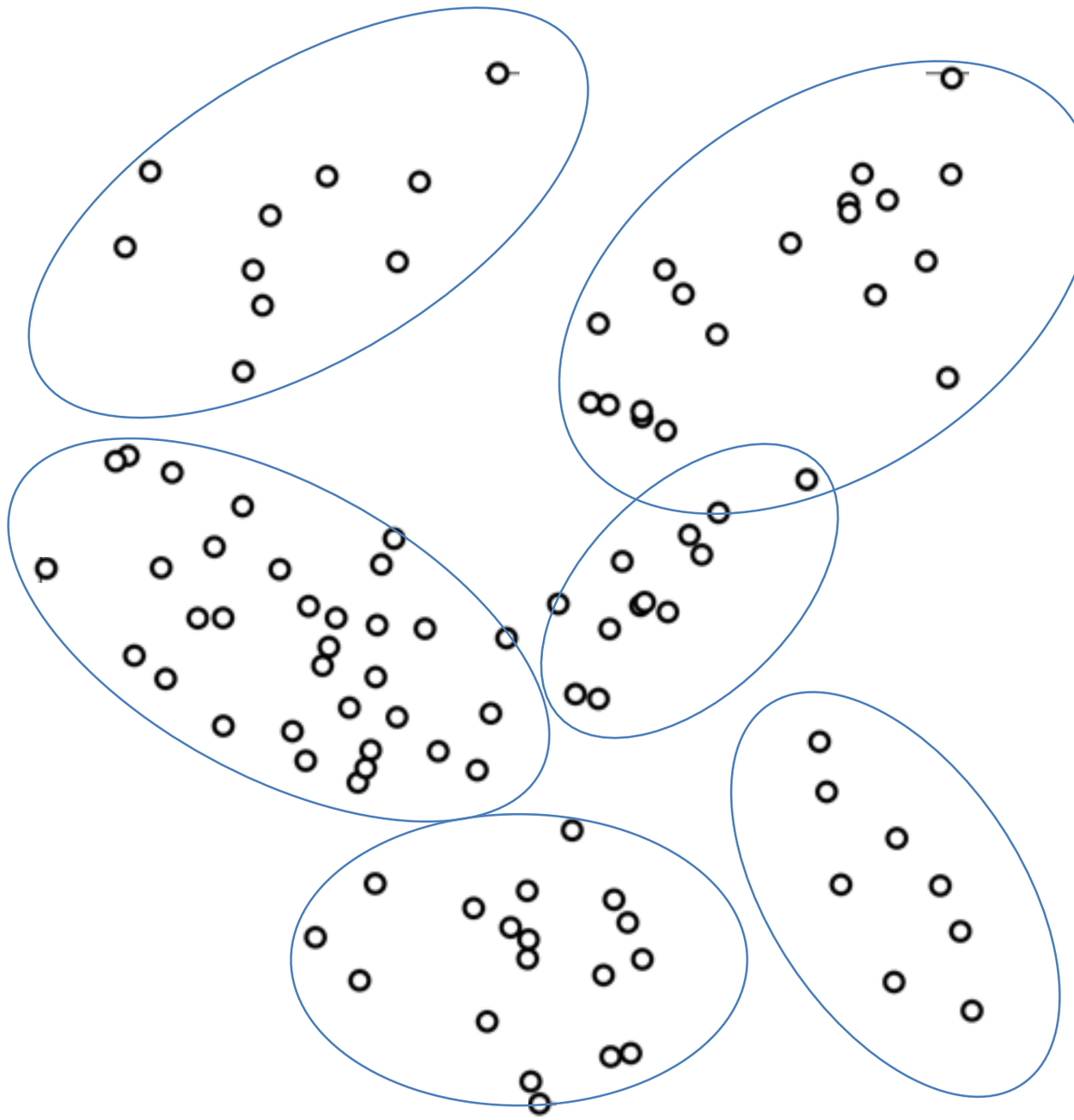

μ_5, Σ_5
n_5=10
Σ_6
μ_2, Σ_2
n_2=22
μ_1, Σ_1
μ_4, Σ_4
n_4=12
n_1=32
μ_3, Σ_3
n_3=18


μ_5, Σ_5
n_5=10
μ_2, Σ_2
n_2=22
μ_1, Σ_1
μ_4, Σ_4
n_4=12
n_1=32
μ_3, Σ_3
μ_6, Σ_6
n_3=18
n_6=8

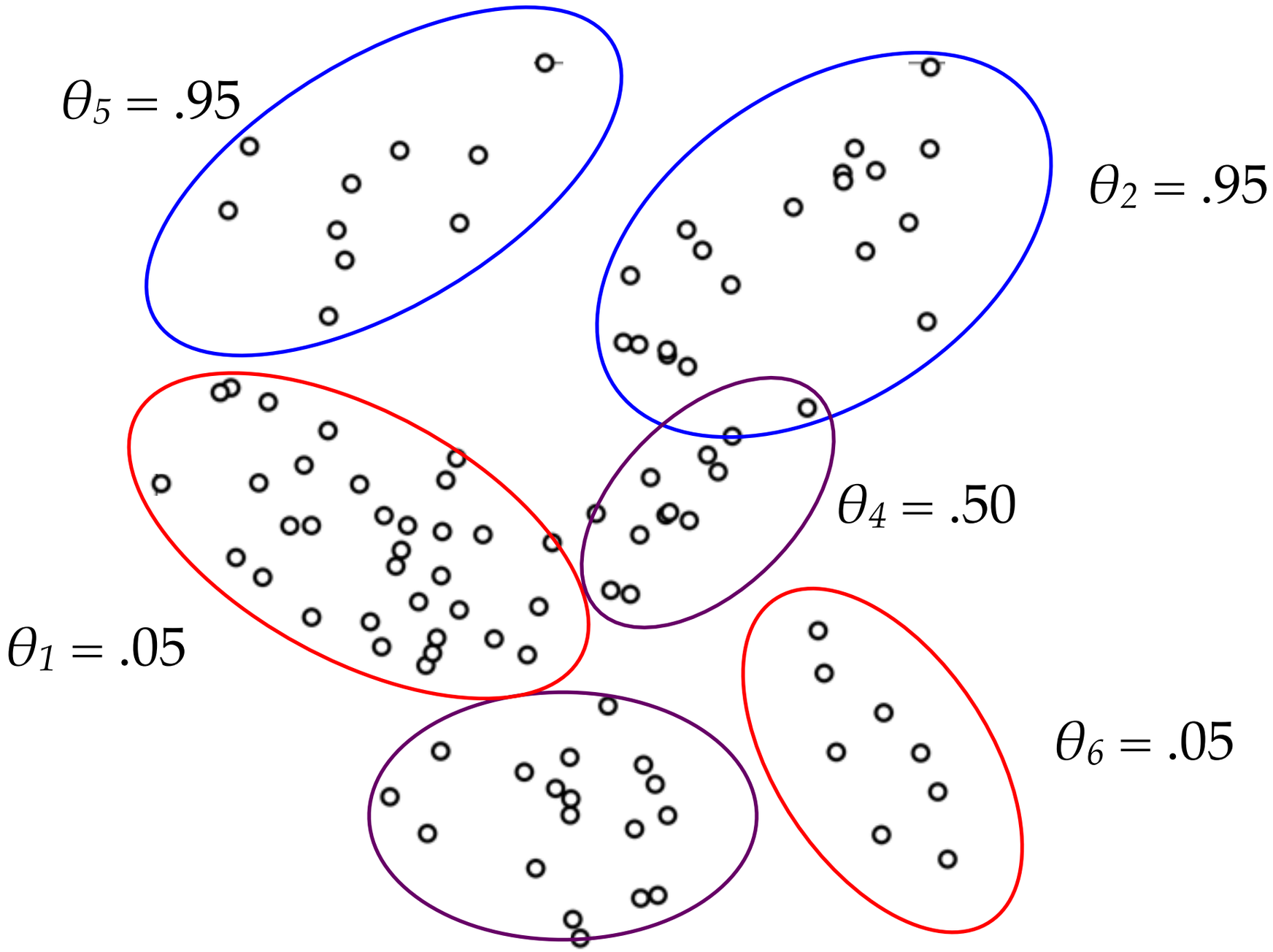
Obviously, each of these six “tables” is a **cluster** that describes a multivariate normal distribution over possible feature values. So…



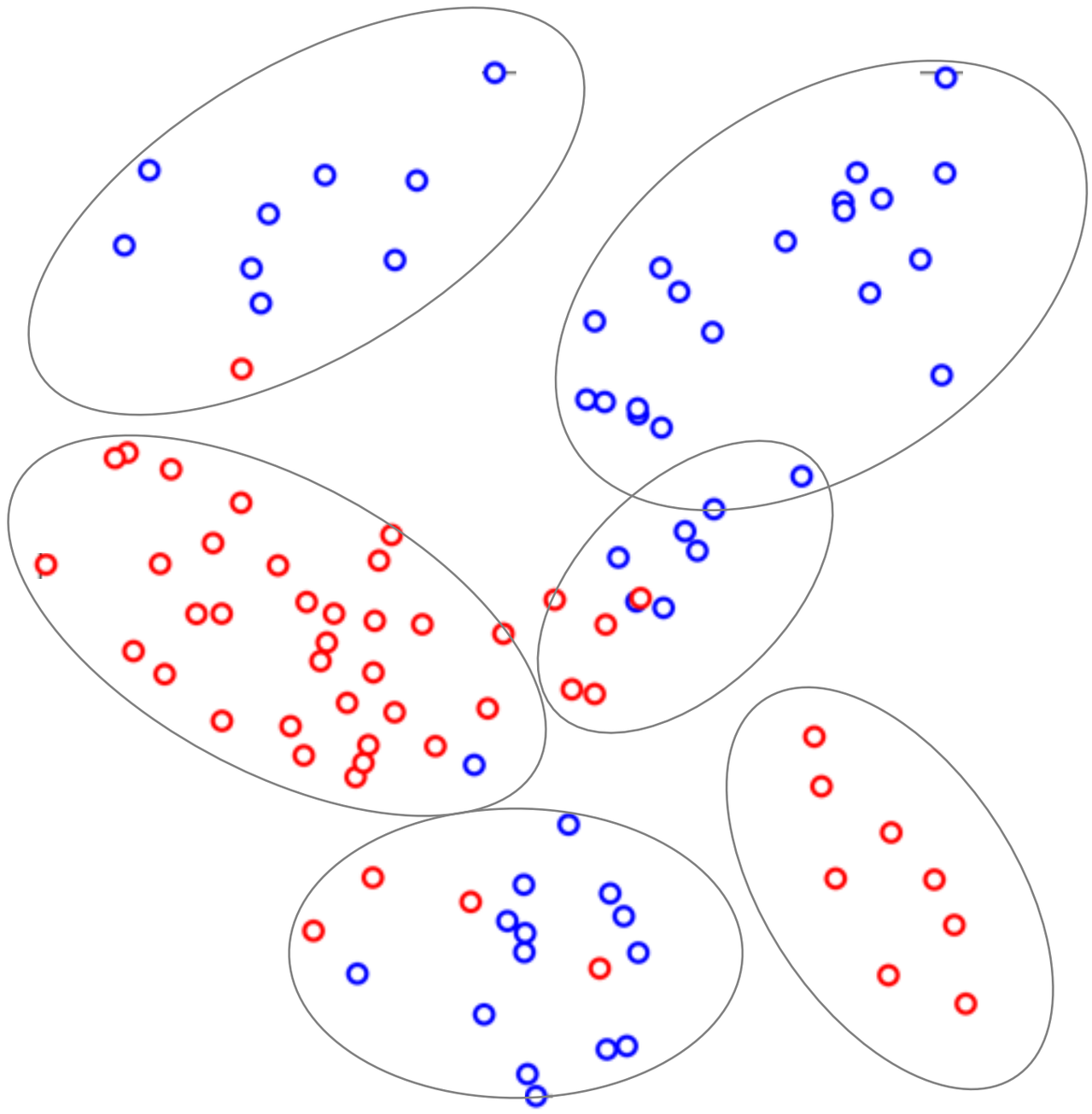
… we'll generate the actual locations of the raw data by sampling from the corresponding probability distribution!



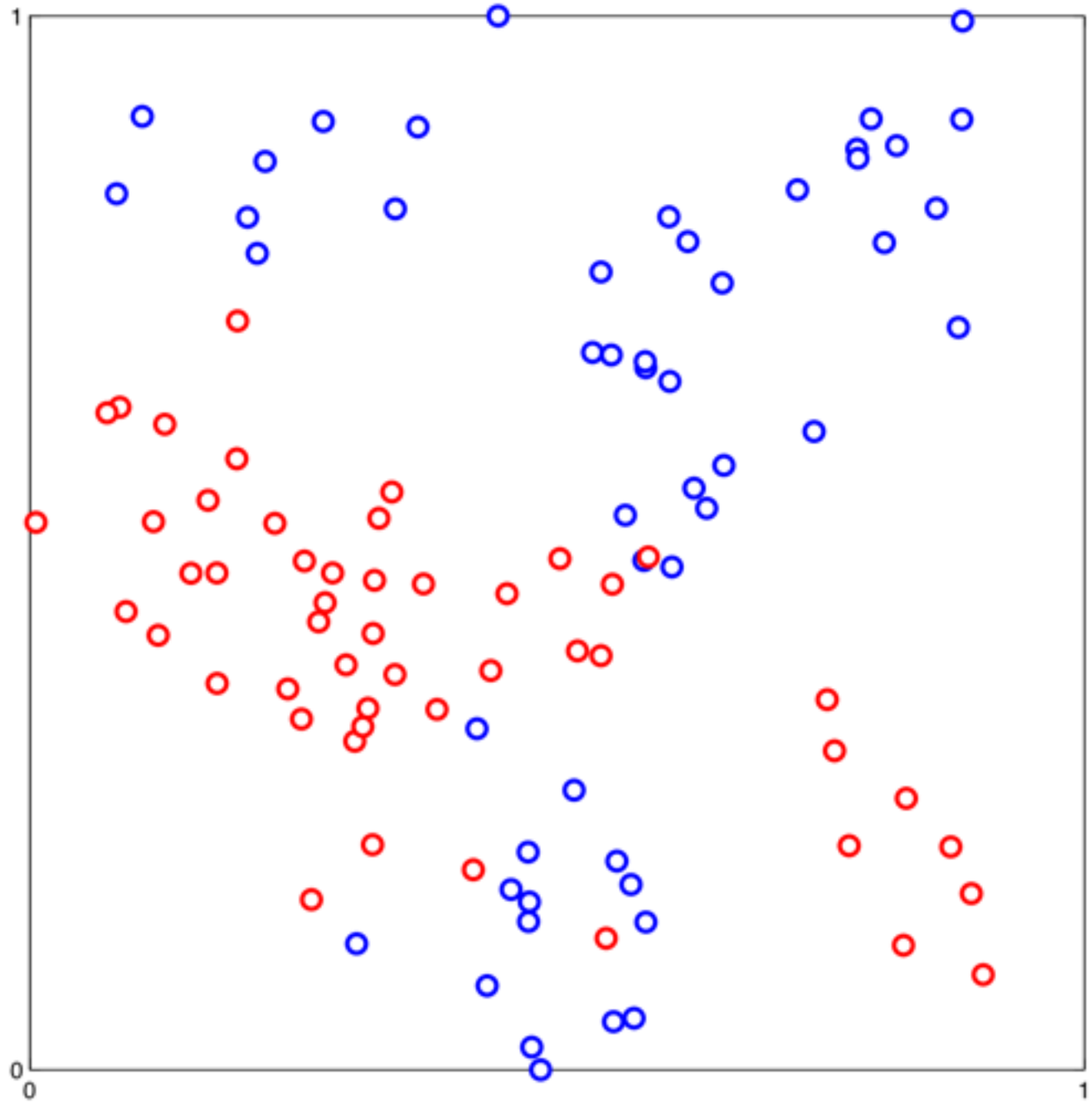
Let's also assume that each cluster describes a probability distribution over **category labels**. That is, some probability θ that the category label will be "blue"



Now we can generate the actual category labels…



And that's the story of how our data came to look like this.



Next lecture…

- Formalising this as a classifier that can do
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
- Two algorithms for doing inference with this model
- Application to our running example
- Application to a novel problem in cognitive science