Carnegie Mellon University
Heinz College

# 94-775 Lecture 8: Topic Modeling, and a Preview of Predictive Data Analysis

George Chen

# Going from Similarities to Clusters

There's a whole zoo of clustering methods

Two major categories (there are others!):

**Generative models**

1. Pretend data generated by specific model with parameters
2. Learn the parameters ("fit model to data")
3. Use fitted model to determine cluster assignments

**Hierarchical clustering**

Top-down: Start with everything in 1 cluster and decide on how to recursively split

Bottom-up: Start with everything in its own cluster and decide on how to iteratively merge clusters

No time to go into detail =(

# What about if a data point is part of multiple clusters?

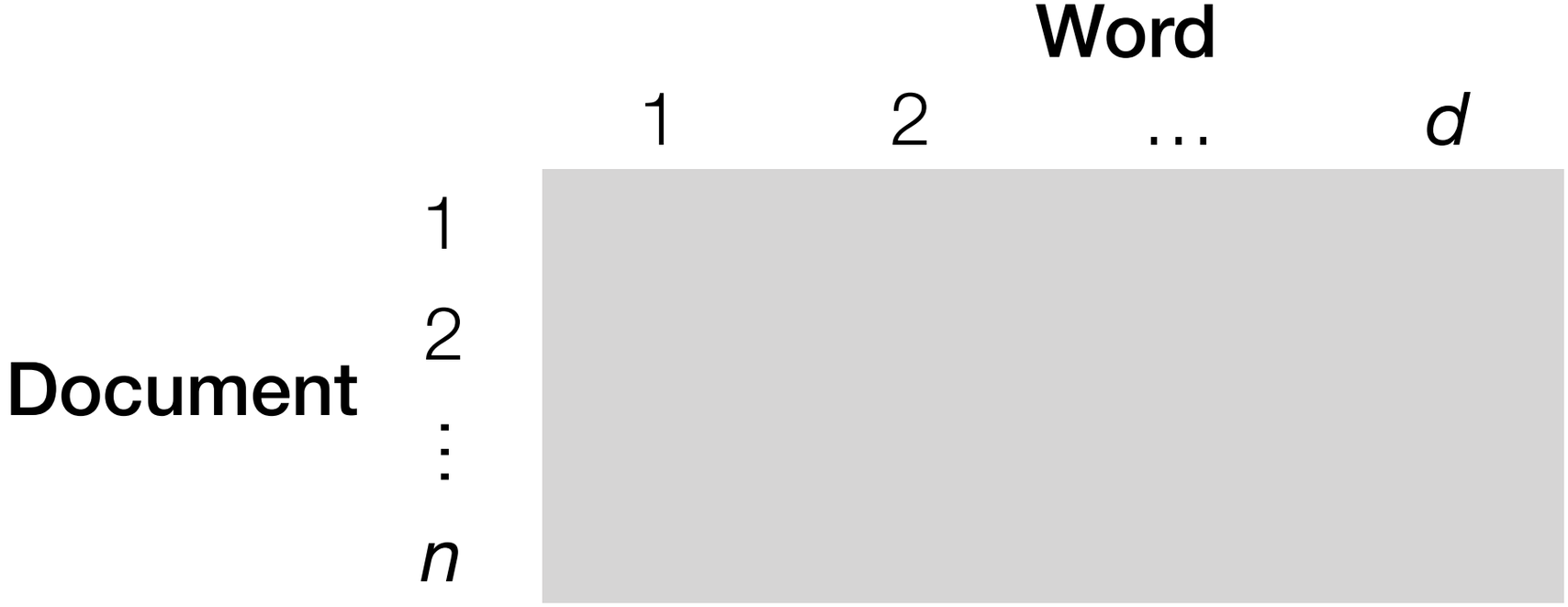
Example: a person is a member of multiple clubs

Example: a text document is about multiple topics such as sports and finance

This question is answered via the problem of **topic modeling**

# Latent Dirichlet Allocation (LDA)

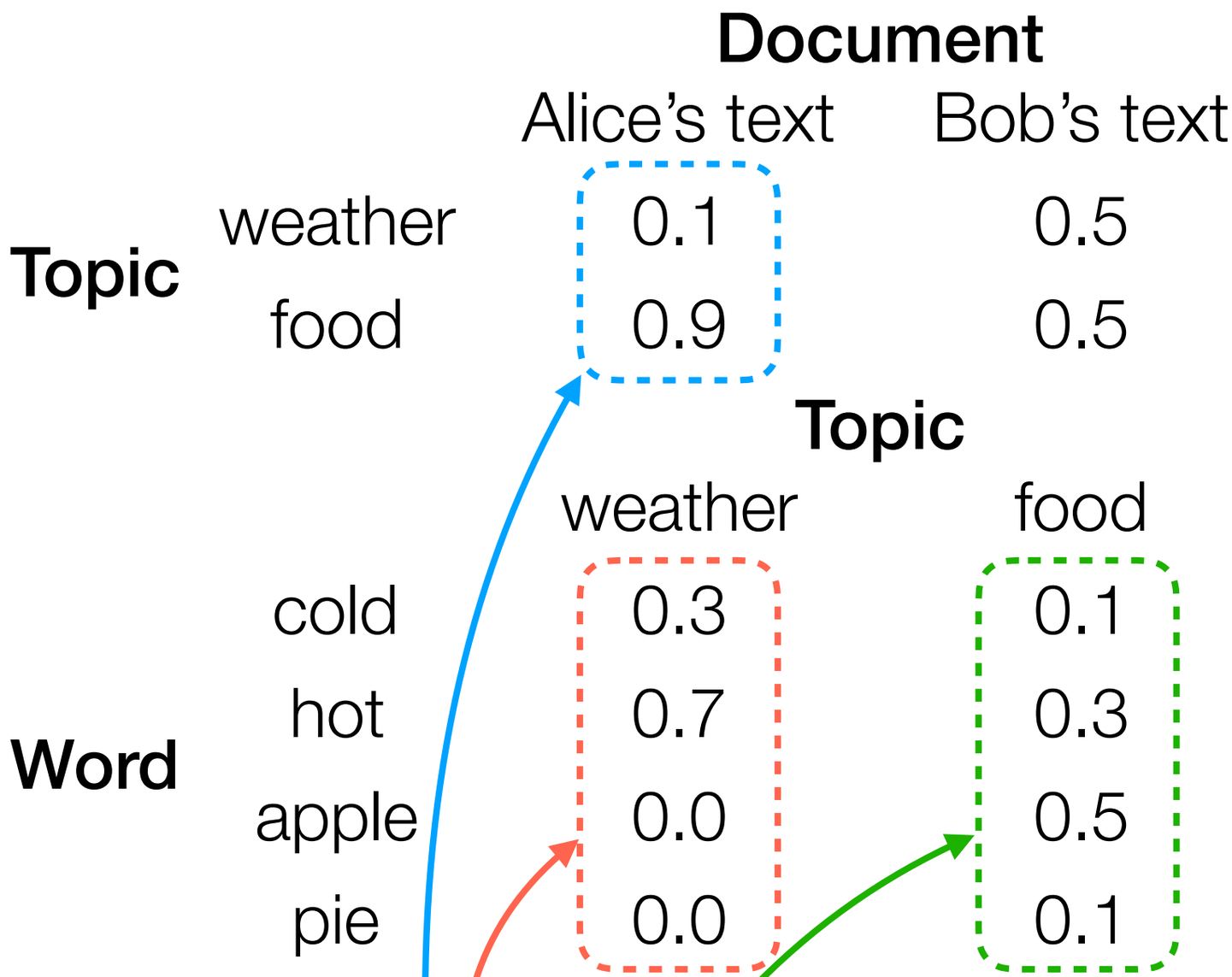
- Easy to describe in terms of text (but works for not just text)
- Input: “document-word” matrix, and pre-specified # topics *k*




*i*-th row, *j*-th column: # times word *j* appears in doc *i*

- Output: what the *k* topics are (details on this shortly)

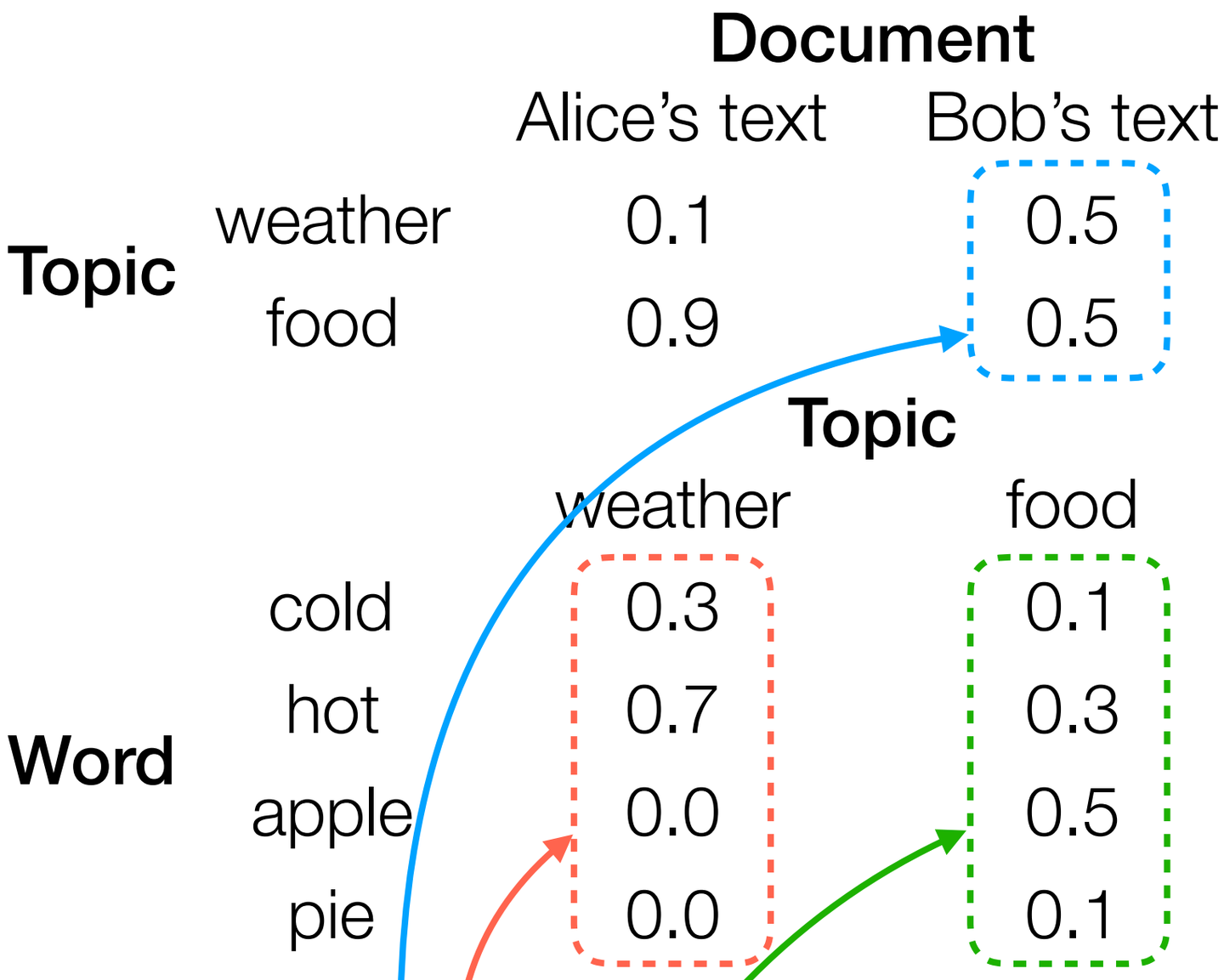
# LDA Example




Each word in Alice's text is generated by:

1. Flip 2-sided coin for Alice
2. If weather: flip 4-sided coin for weather
   If food: flip 4-sided coin for food

# LDA Example




Each word in Bob's text is generated by:

1. Flip 2-sided coin for Bob
2. If weather: flip 4-sided coin for weather
   If food: flip 4-sided coin for food

# LDA Example

**Document**

| **Topic** | Alice's text | Bob's text |
|---|---|---|
| weather | 0.1 | 0.5 |
| food | 0.9 | 0.5 |

**Topic**

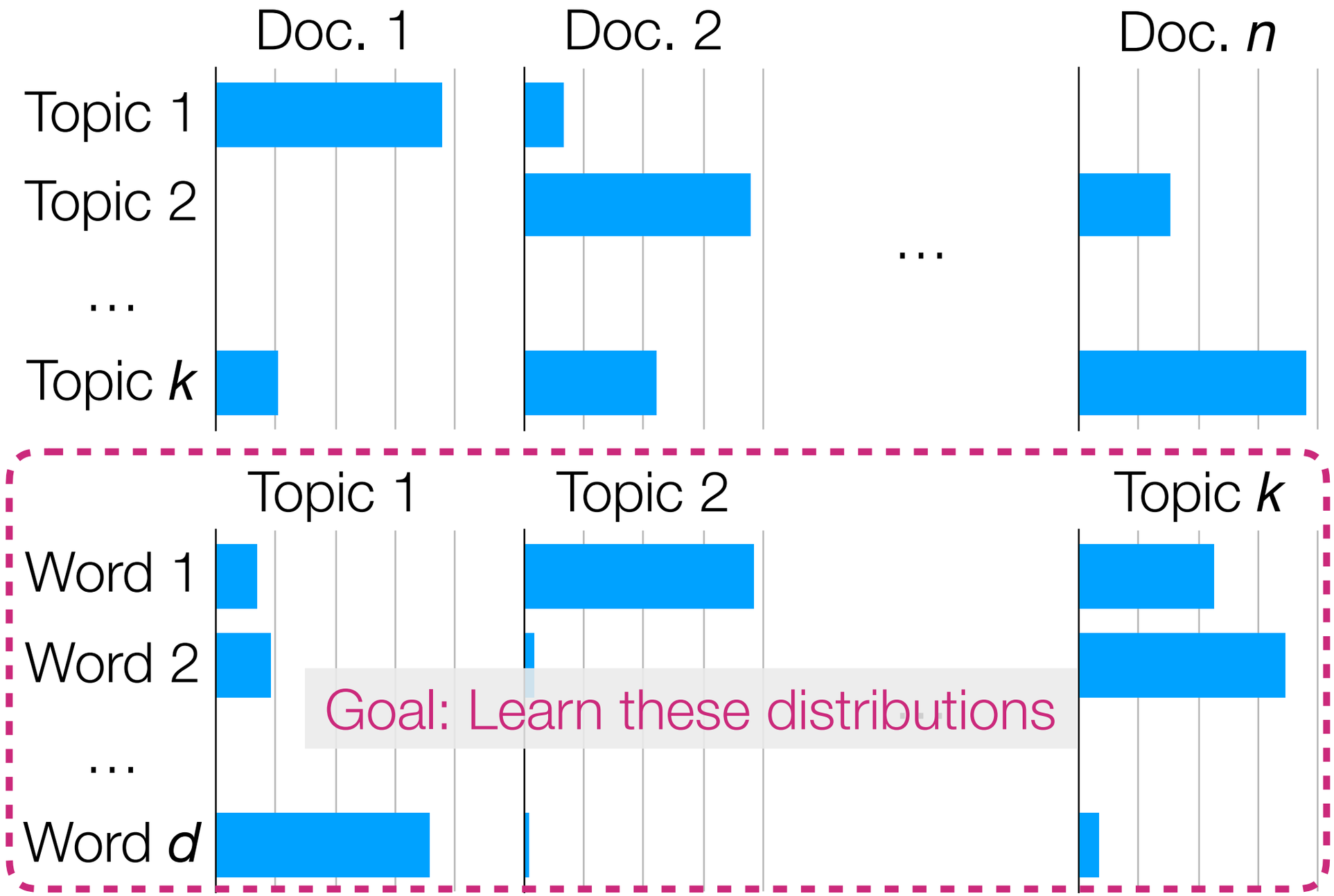
| **Word** | weather | food |
|---|---|---|
| cold | 0.3 | 0.1 |
| hot | 0.7 | 0.3 |
| apple | 0.0 | 0.5 |
| pie | 0.0 | 0.1 |

Each word in doc. *i* is generated by:

1. Flip 2-sided coin for doc. *i*
2. If weather: flip 4-sided coin for weather
   If food: flip 4-sided coin for food

"Learning the topics" means figuring out these 4-sided coin probabilities

# LDA

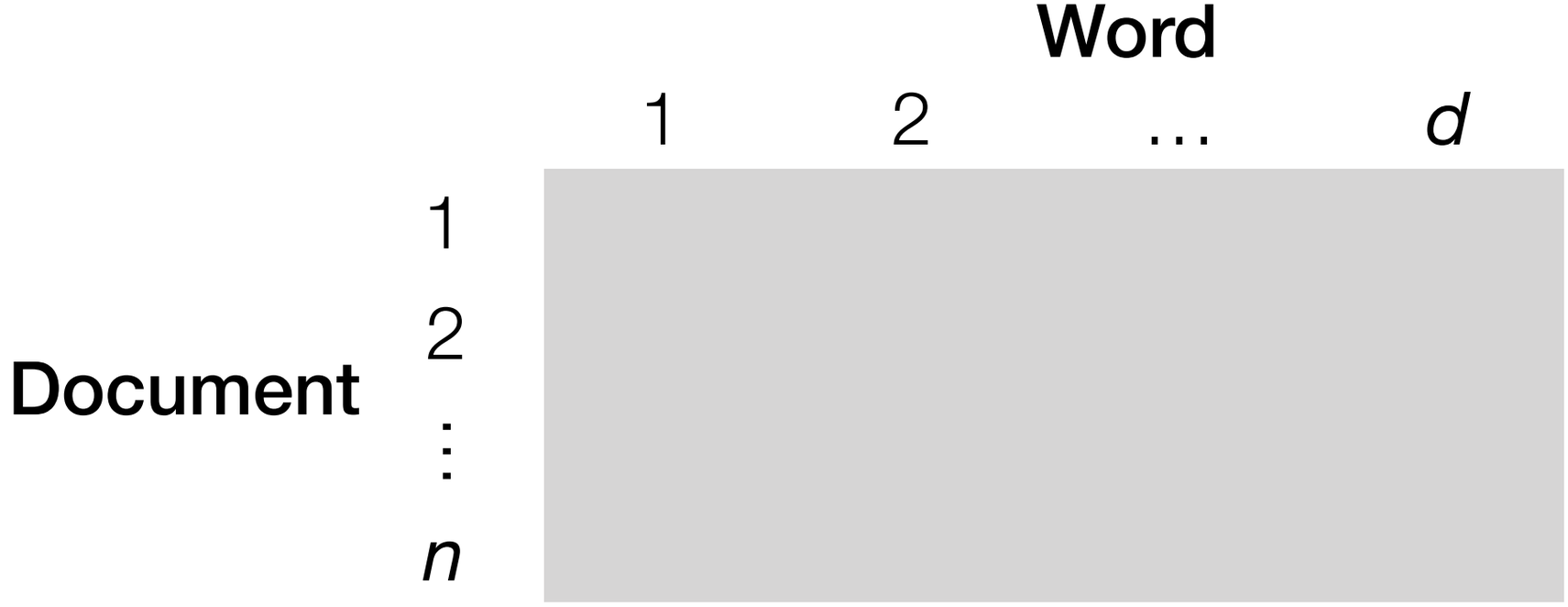



LDA models each word in document *i* to be generated as:

- Randomly choose a topic *Z* (use topic distribution for doc *i*)
- Randomly choose a word (use word distribution for topic *Z*)

# LDA

- Easy to describe in terms of text (but works for not just text)
- Input: “document-word” matrix, and pre-specified # topics ***k***




*i*-th row, *j*-th column: # times word *j* appears in doc *i*

- Output: the ***k*** topics’ distribution of words

# LDA

Demo

# How to Choose Number of Topics *k*?

Something like CH index is also possible!

For a specific topic, look at the *m* most probable words ("top words")

**Topic coherence (within cluster/topic variability):**

$$\sum_{\text{top words } v,w \text{ that are not the same}} \log \frac{\text{P( top word v appears | top word w appears )}}{\frac{\text{(\# documents with at least one appearance of } v \text{ and } w\text{)} + \varepsilon}{\text{\# documents with at least one appearance of } w}}$$

estimate with

choose something small like 0.01

**Inter-topic similarity (between cluster/topic variability):**

Count # top words that do not appear in any of the other topics' *m* top words

(number of "unique top words")

Can average each of these across the topics

# Topic Modeling

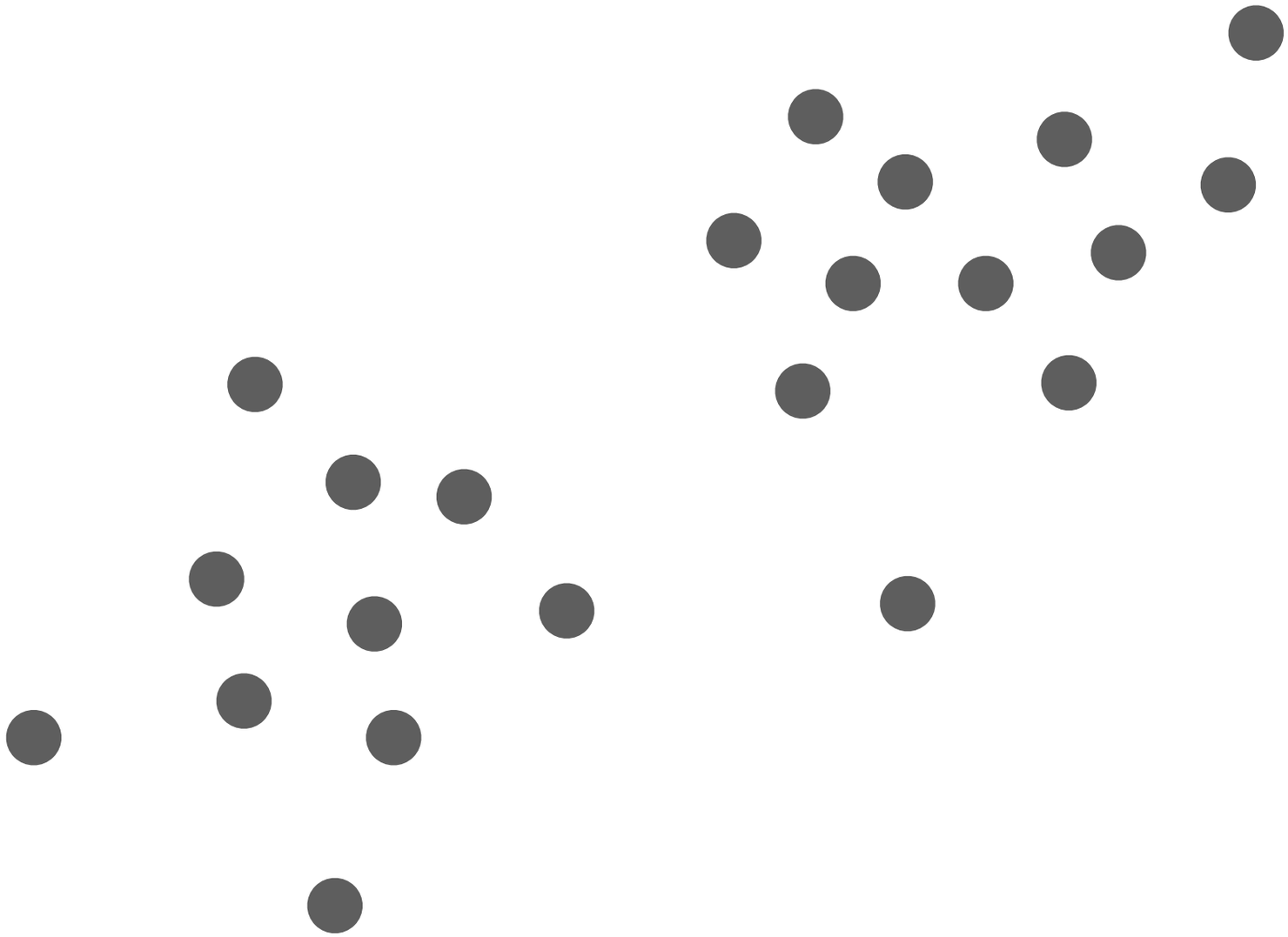
- There are actually *many* topic models, not just LDA
    - HDP, correlated topic models, Pachinko allocation, biterm topic models, anchor word topic models, …
- Dynamic topic models: tracks how topics change *over time*

# 94-775 Part III:

# Predictive Data Analysis

# What if we have labels?

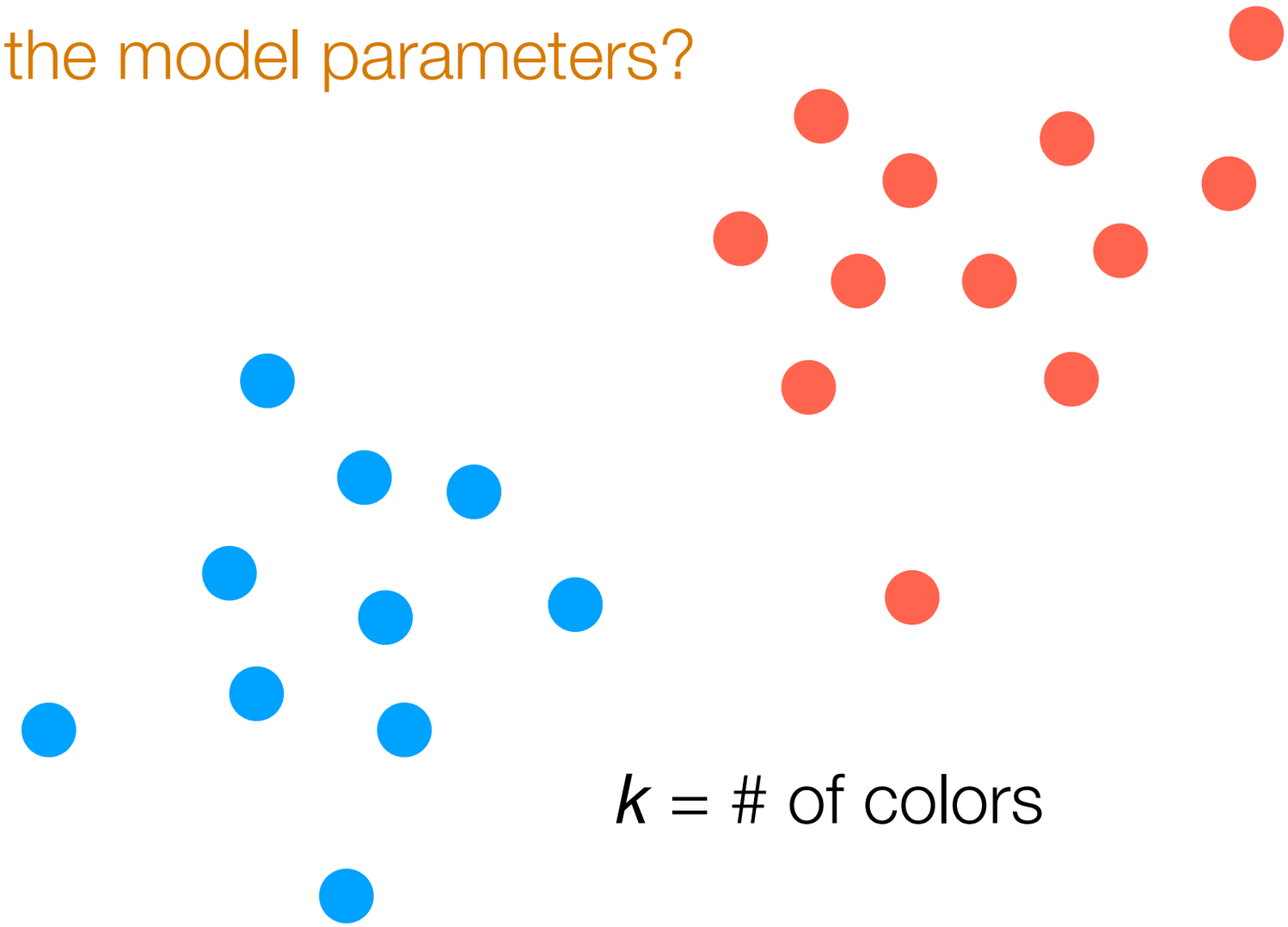
If the labels are known…



If the labels are known…

And we assume data generated by GMM…

What are the model parameters?



$k$ = # of colors

We can directly estimate cluster means, covariances

# Flashback: Learning a GMM

Don’t need this top part if we know the labels!

Step 0: Pick *k*

Step 1: Pick guesses for **cluster means and covariances**

**Repeat until convergence:**

Step 2: Compute probability of each point belonging to each of the *k* clusters

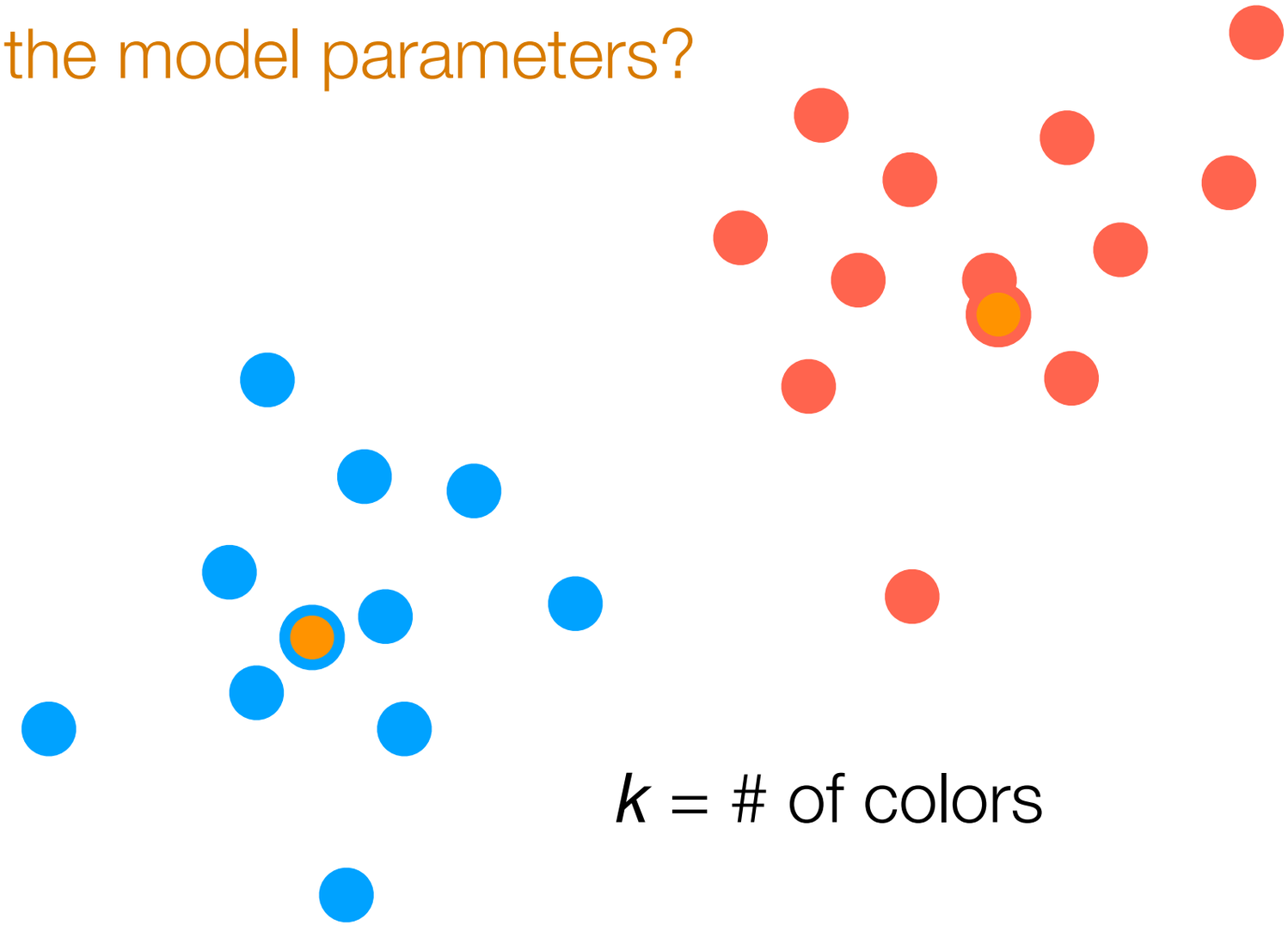
Step 3: Update **cluster means and covariances** carefully accounting for probabilities of each point belonging to each of the clusters

We don’t need to repeat until convergence

If the labels are known…

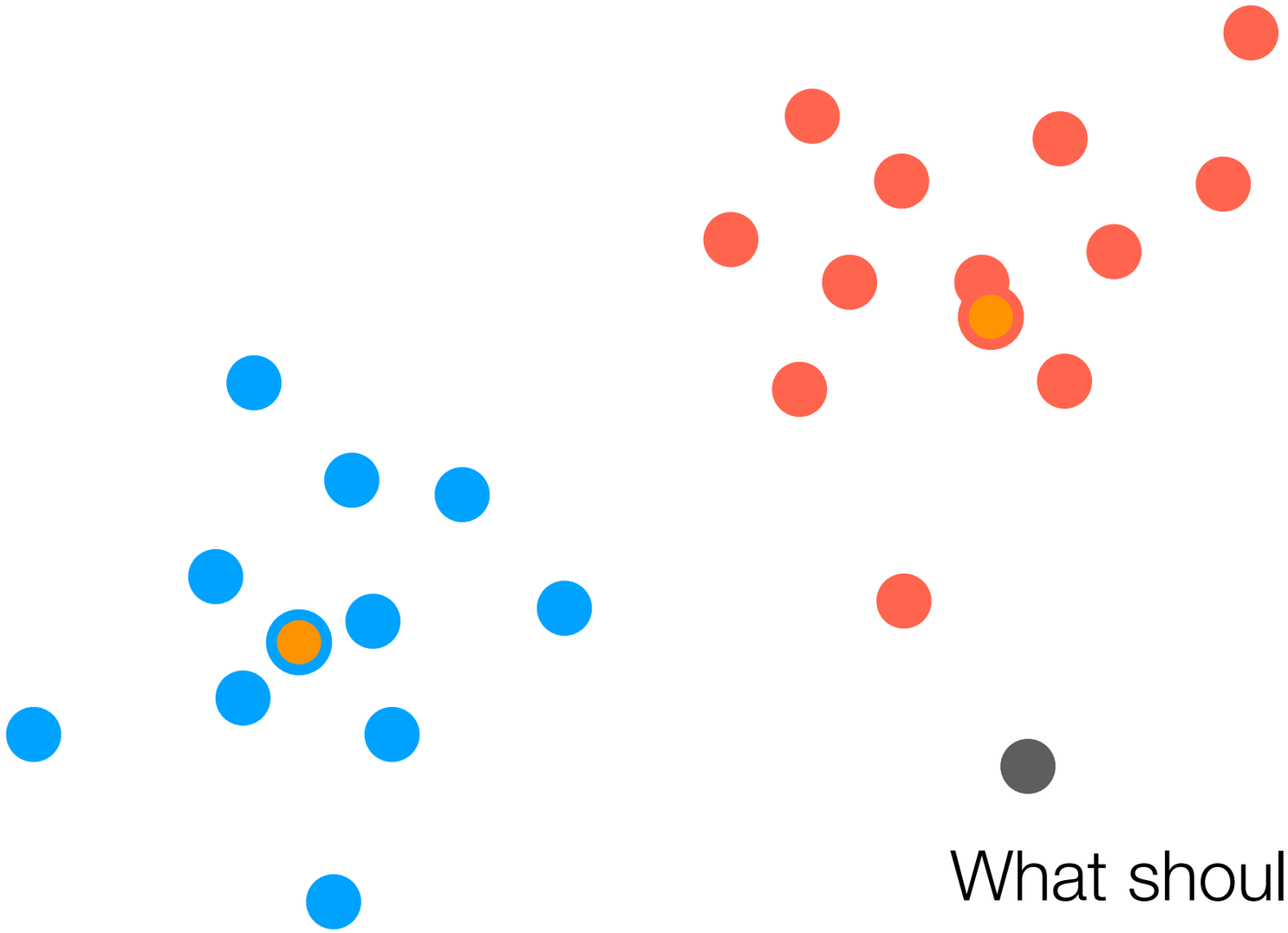
And we assume data generated by GMM…

What are the model parameters?



$k$ = # of colors

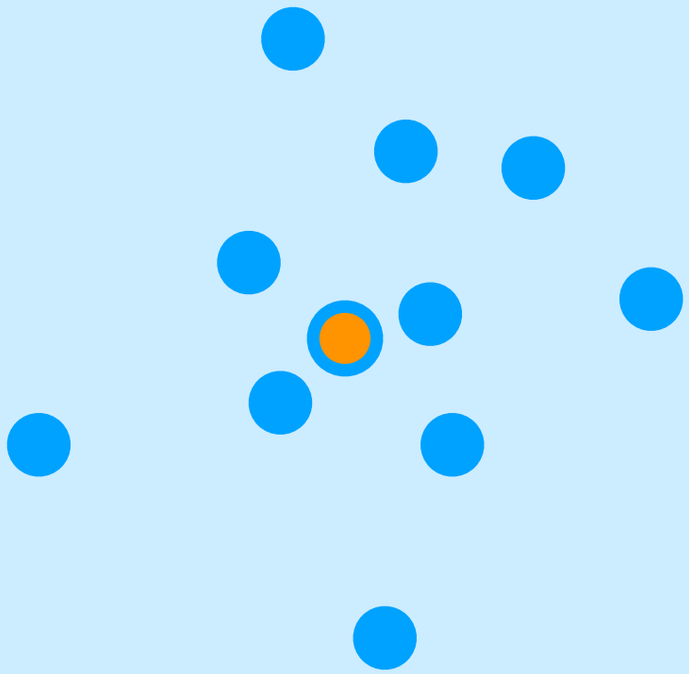
We can directly estimate cluster means, covariances



What should the label of this new point be?

Whichever cluster has higher probability!

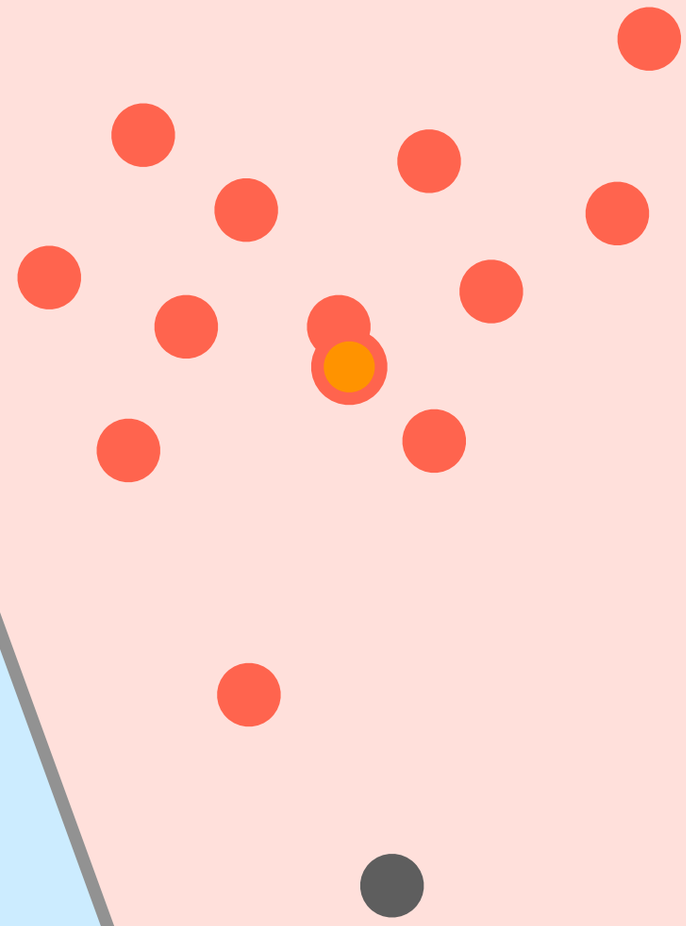
Decision boundary



This classifier we've created assumes a *generative model*

We just created a **classifier**

(a procedure that given a new data point tells us what "class" it belongs to)



What should the label of this new point be?

Whichever cluster has higher probability!