

# Unsupervised Learning, K Means

March 12, 2020

Data Science CSCI 1951A

Brown University

Instructor: Ellie Pavlick

HTAs: Josh Levin, Diane Mutako, Sol Zitter

# Announcements

- Here we go! Get cozy..PJs, coffee-in-hand, ready to talk ML :)
- Use the “raise hand” feature (under “participants”)
- I’ll scroll through periodically and see if here are any questions; if I call on you, unmute yourself

**How's everyone feeling? <3**

- (a) Super!**
- (b) Kinda freaked out but healthy**
- (c) A little sick**
- (d) Very sick, very scared**

# Today

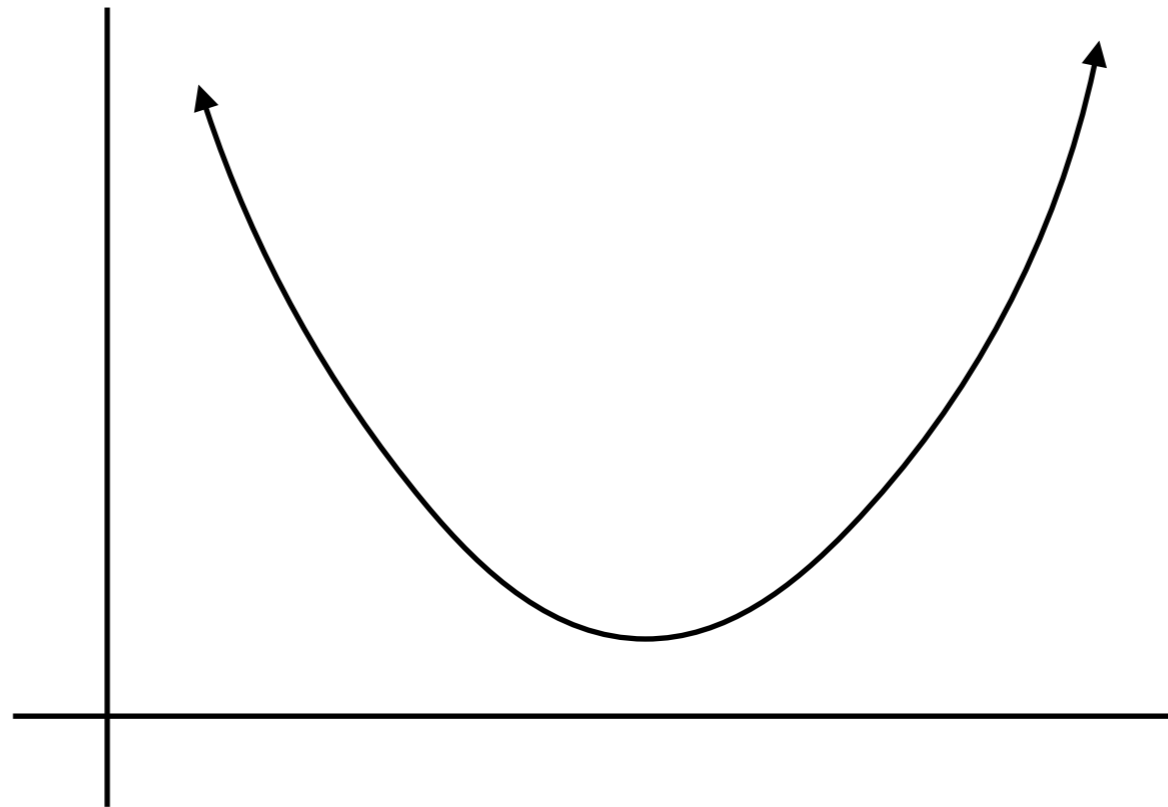
- Gradient Descent
- Supervised vs. Unsupervised Learning
- K-Means and EM

# Training with Gradient Descent

$$\text{minimize } \sum_{i=1}^n (Y_i - \hat{Y})^2$$

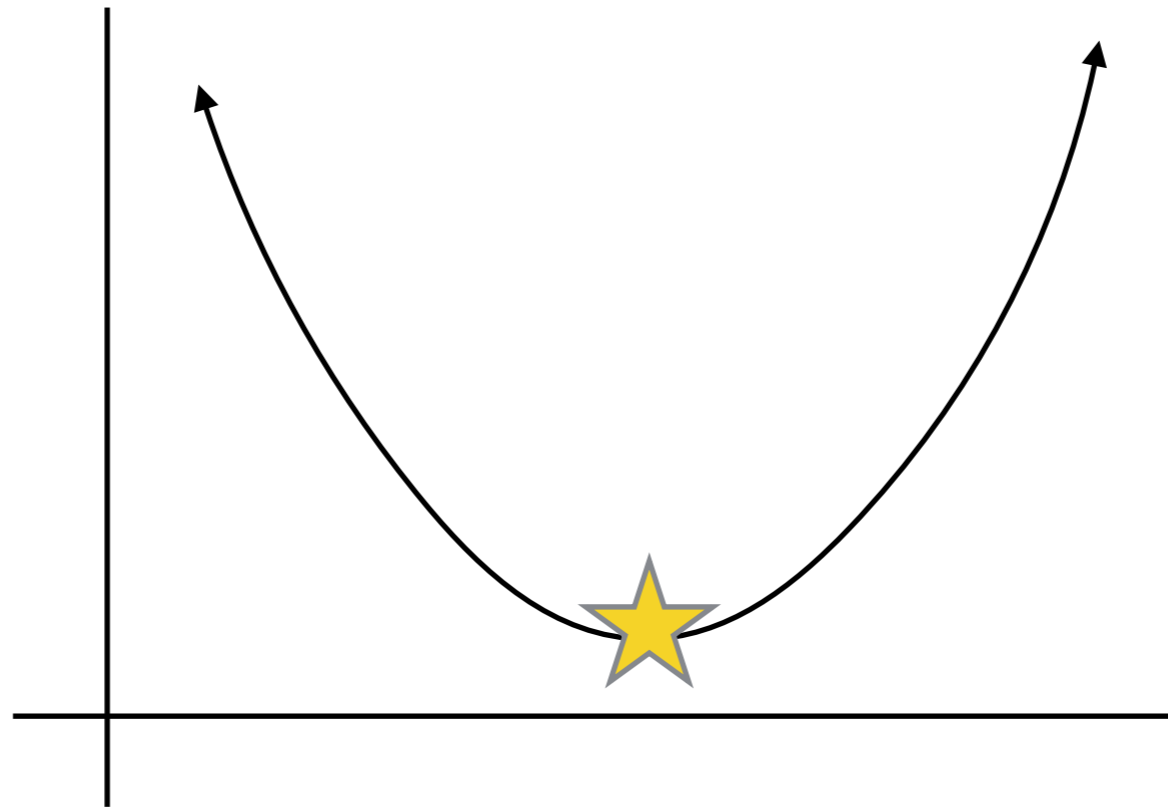
# Training with Gradient Descent

minimize  $\sum_{i=1}^n (Y_i - \hat{Y})^2$



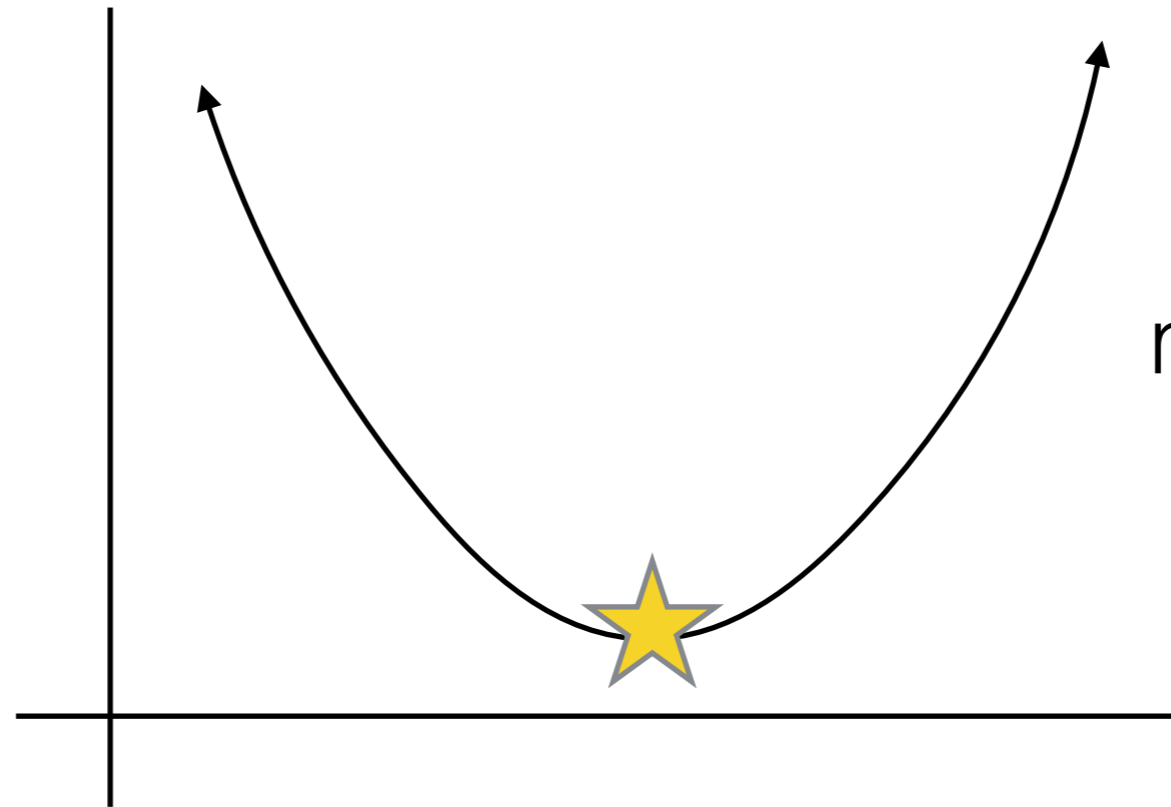
# Training with Gradient Descent

minimize  $\sum_{i=1}^n (Y_i - \hat{Y})^2$



# Training with Gradient Descent

minimize  $\sum_{i=1}^n (Y_i - \hat{Y})^2$



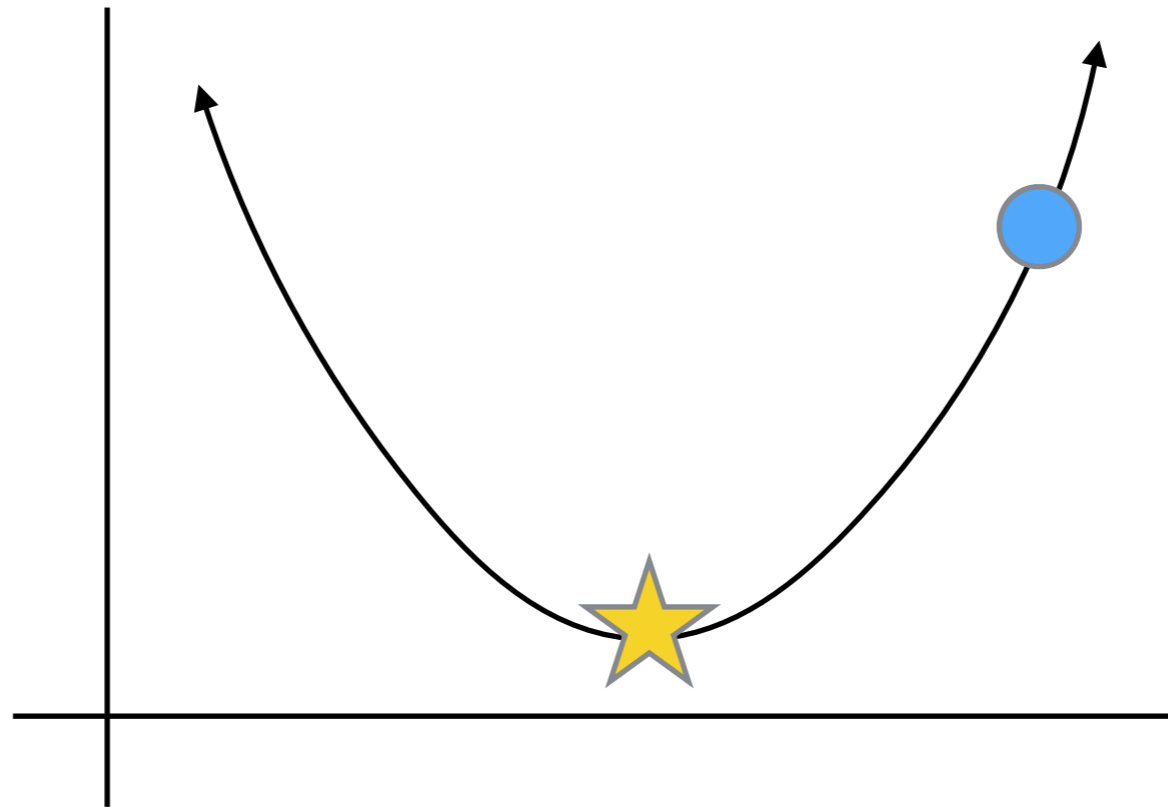
$$b = \bar{Y} - m\bar{X}$$

$$m = \frac{Cov(X, Y)}{Var(X)}$$



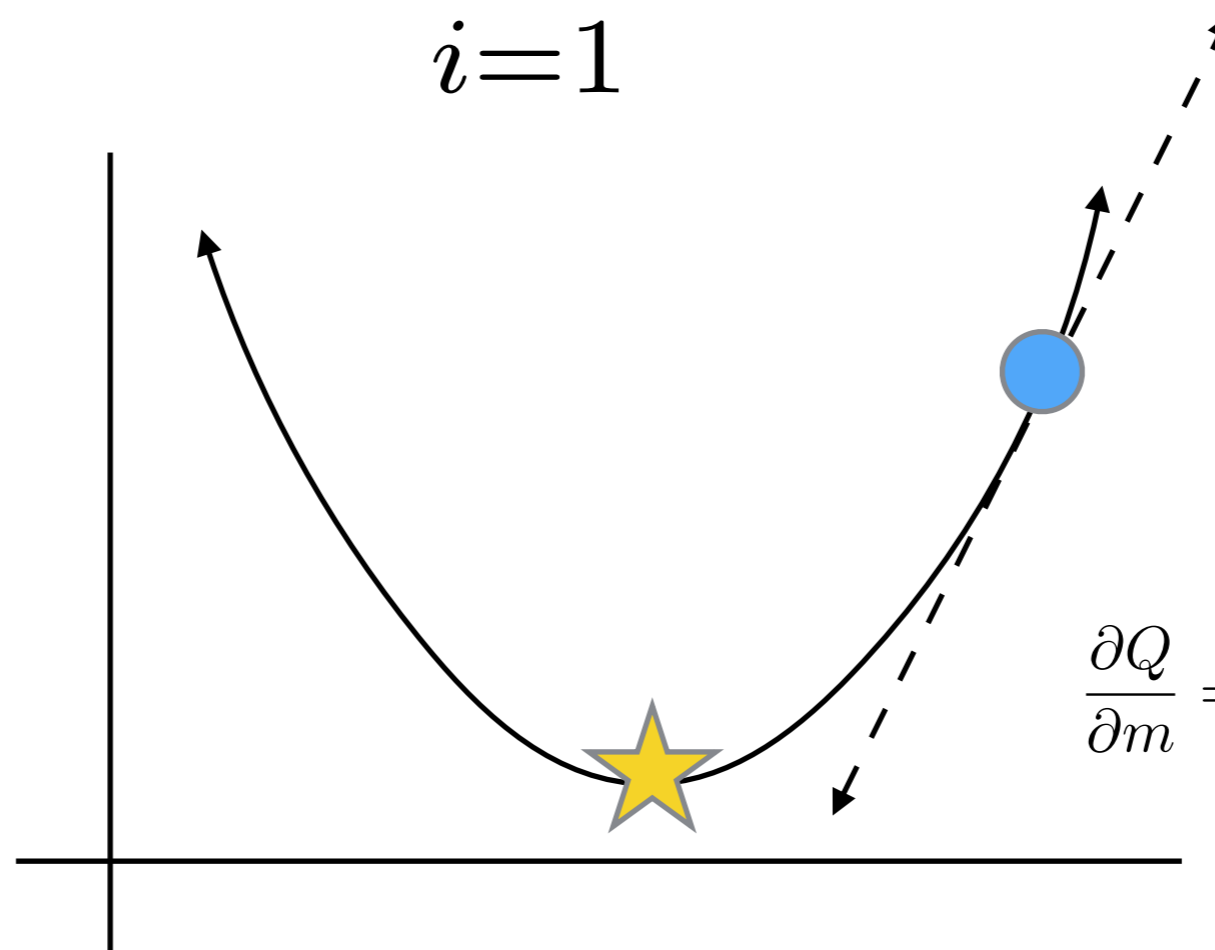
# Training with Gradient Descent

minimize  $\sum_{i=1}^n (Y_i - \hat{Y})^2$



# Training with Gradient Descent

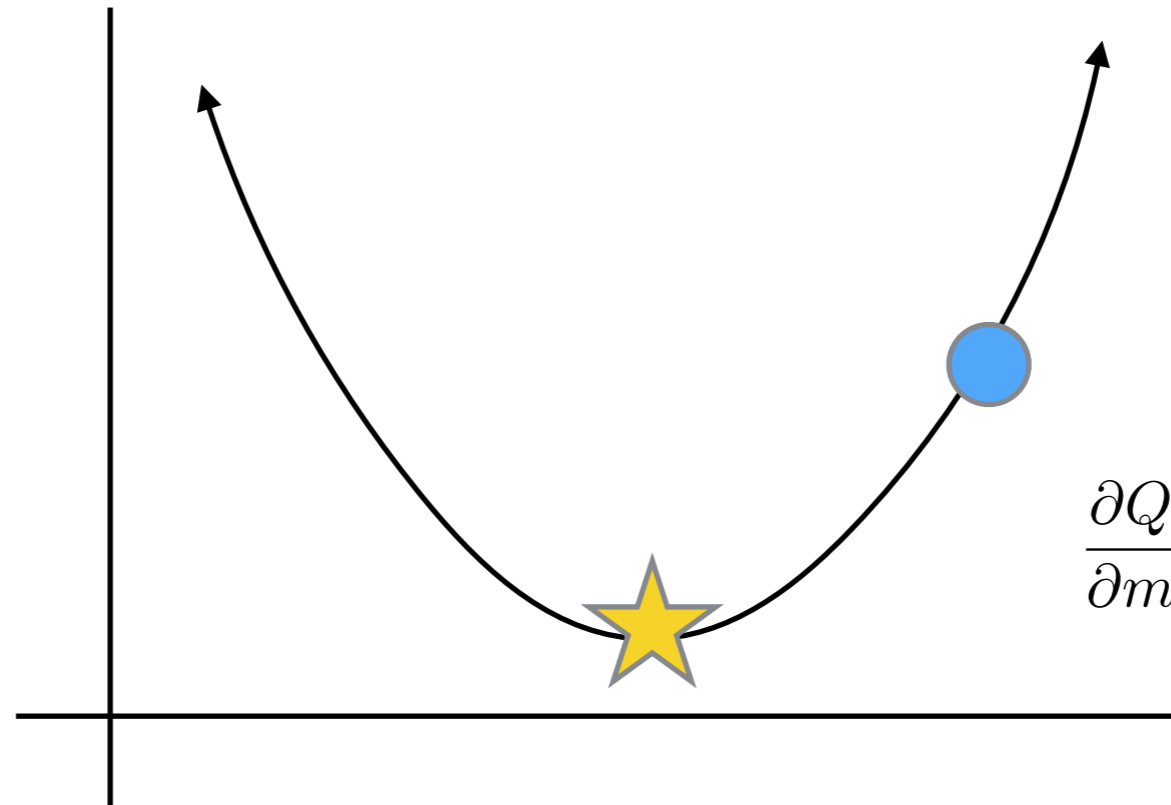
minimize  $\sum_{i=1}^n (Y_i - \hat{Y})^2$



$$\frac{\partial Q}{\partial m} = \sum_{i=1}^n -2X_i(Y_i - b - mX_i)$$

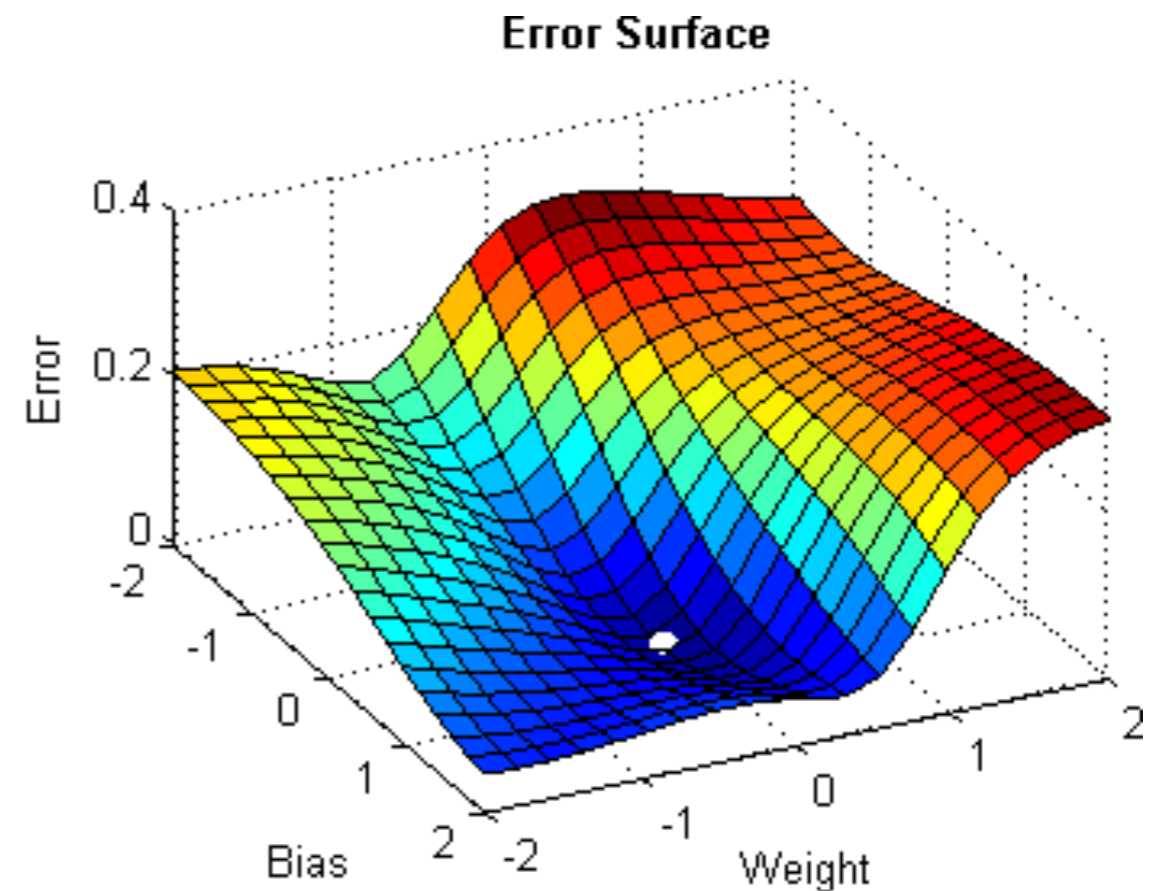
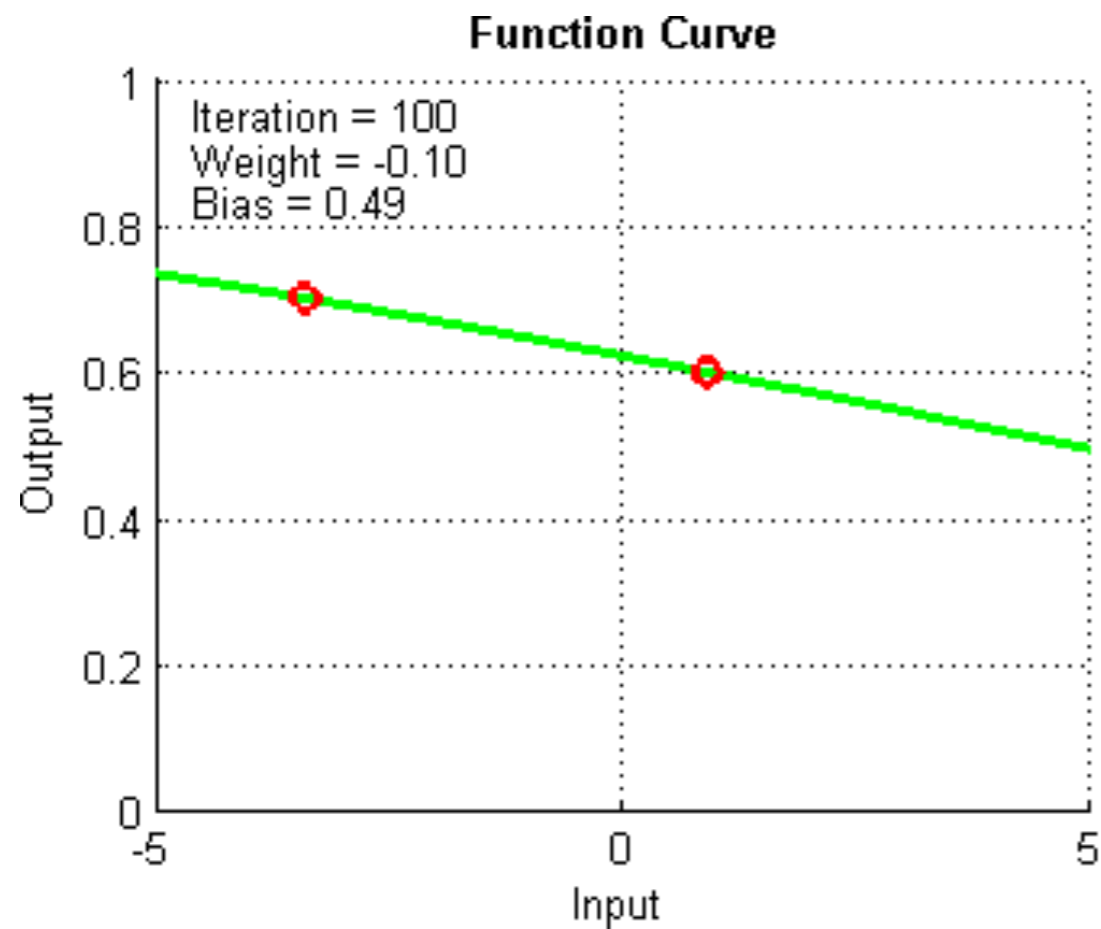
# Training with Gradient Descent

minimize  $\sum_{i=1}^n (Y_i - \hat{Y})^2$



$$\frac{\partial Q}{\partial m} = \sum_{i=1}^n -2X_i(Y_i - b - mX_i)$$

# Training with Gradient Descent



# Training with Gradient Descent

Helpful equations for following along in the jupyter notebook

$$Q = \sum_{i=1}^n (Y_i - (mX_i + b))^2$$

$$\frac{\partial Q}{\partial b} = \sum_{i=1}^n -2(Y_i - mX_i - b) = 0$$

$$\frac{\partial Q}{\partial m} = \sum_{i=1}^n -2X_i(Y_i - b - mX_i) = 0$$

$$m = \frac{Cov(X, Y)}{Var(X)} \quad b = \bar{Y} - m\bar{X}$$

# Supervised vs. Unsupervised Learning

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings



# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption
- Unsupervised: No explicit labels

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption
- Unsupervised: No explicit labels
  - Clustering—find groups similar customers

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption
- Unsupervised: No explicit labels
  - Clustering—find groups similar customers
  - Dimensionality Reduction—find features that differentiate individuals

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption
- Unsupervised: No explicit labels
  - Clustering—find groups similar customers
  - Dimensionality Reduction—find features that differentiate individuals

Today



# Supervised vs. Unsupervised Learning


- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption
- Unsupervised: No explicit labels
  - Clustering—find groups similar customers
  - Dimensionality Reduction—find features that differentiate individuals

Today  
↓

# Supervised vs. Unsupervised Learning

- Supervised: Explicit data labels
  - Sentiment analysis—review text -> star ratings
  - Image tagging—image -> caption
- Unsupervised: No explicit labels
  - Clustering—find groups similar customers
  - Dimensionality Reduction—find features that differentiate individuals

Tuesday



Oh you thought it was that  
simple? How cute...



# Oh you thought it was that simple? How cute...

- Semi Supervised—Combining large amounts of unlabelled with smaller amounts of labelled (pretraining)

# Oh you thought it was that simple? How cute...

- Semi Supervised—Combining large amounts of unlabelled with smaller amounts of labelled (pretraining)
- Weakly/Distantly Supervised—using noisy labels or partial labels (bootstrapping, automatically-labeled data)

# Oh you thought it was that simple? How cute...

- Semi Supervised—Combining large amounts of unlabelled with smaller amounts of labelled (pretraining)
- Weakly/Distantly Supervised—using noisy labels or partial labels (bootstrapping, automatically-labeled data)
- Reinforcement Learning—label on the result of a sequence of actions, but not on each action (games, robotics)

# Oh you thought it was that simple? How cute...

- Semi Supervised—Combining large amounts of unlabelled with smaller amounts of labelled (pretraining)
- Weakly/Distantly Supervised—using noisy labels or partial labels (bootstrapping, automatically-labeled data)
- Reinforcement Learning—label on the result of a sequence of actions, but not on each action (games, robotics)
- “Found” Data... (?)

# Unsupervised Learning

# Unsupervised Learning

- “Finding structure in data” (vs. predicting labels)

# Unsupervised Learning

- “Finding structure in data” (vs. predicting labels)
- In data science, this is typically for “exploratory analysis”. “What the \$@%! is this data even?! Enlighten me.”

# Unsupervised Learning

- “Finding structure in data” (vs. predicting labels)
- In data science, this is typically for “exploratory analysis”. “What the \$@%! is this data even?! Enlighten me.”
- Or for preprocessing/featurizing—e.g. so you can use article “topics” to predict clicks.



# Unsupervised Learning

- “Finding structure in data” (vs. predicting labels)
- In data science, this is typically for “exploratory analysis”. “What the \$@%! is this data even?! Enlighten me.”
- Or for preprocessing/featurizing—e.g. so you can use article “topics” to predict clicks.
- In ML, right now, used extensively for “pretraining” (e.g. autoencoding, dimensionality reduction, language modeling\*)

# Clicker Question!

# Clustering

**Discussion Question!**

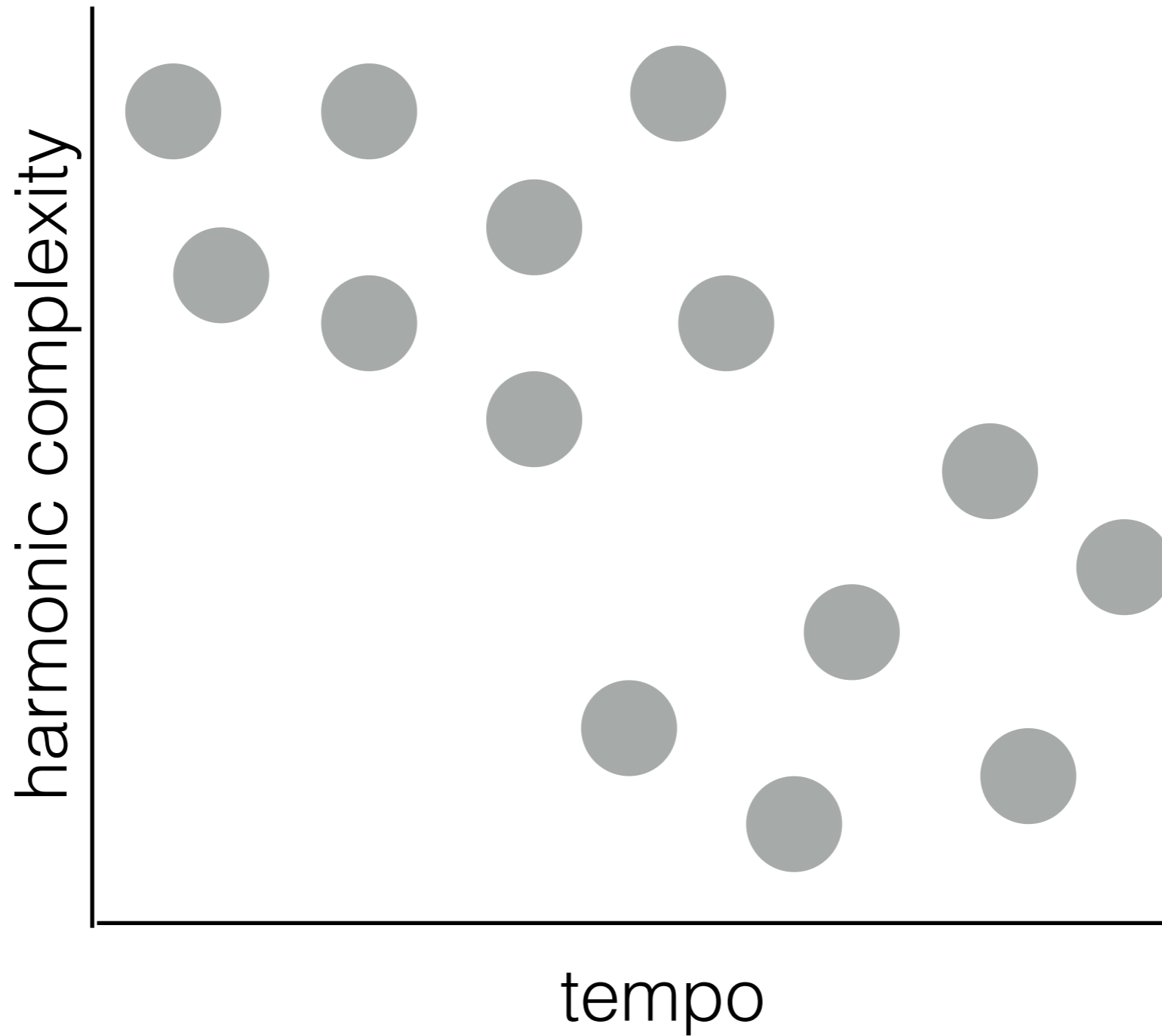
**What is it good for...?**

(...because those free-form answers were enlightening last time...)

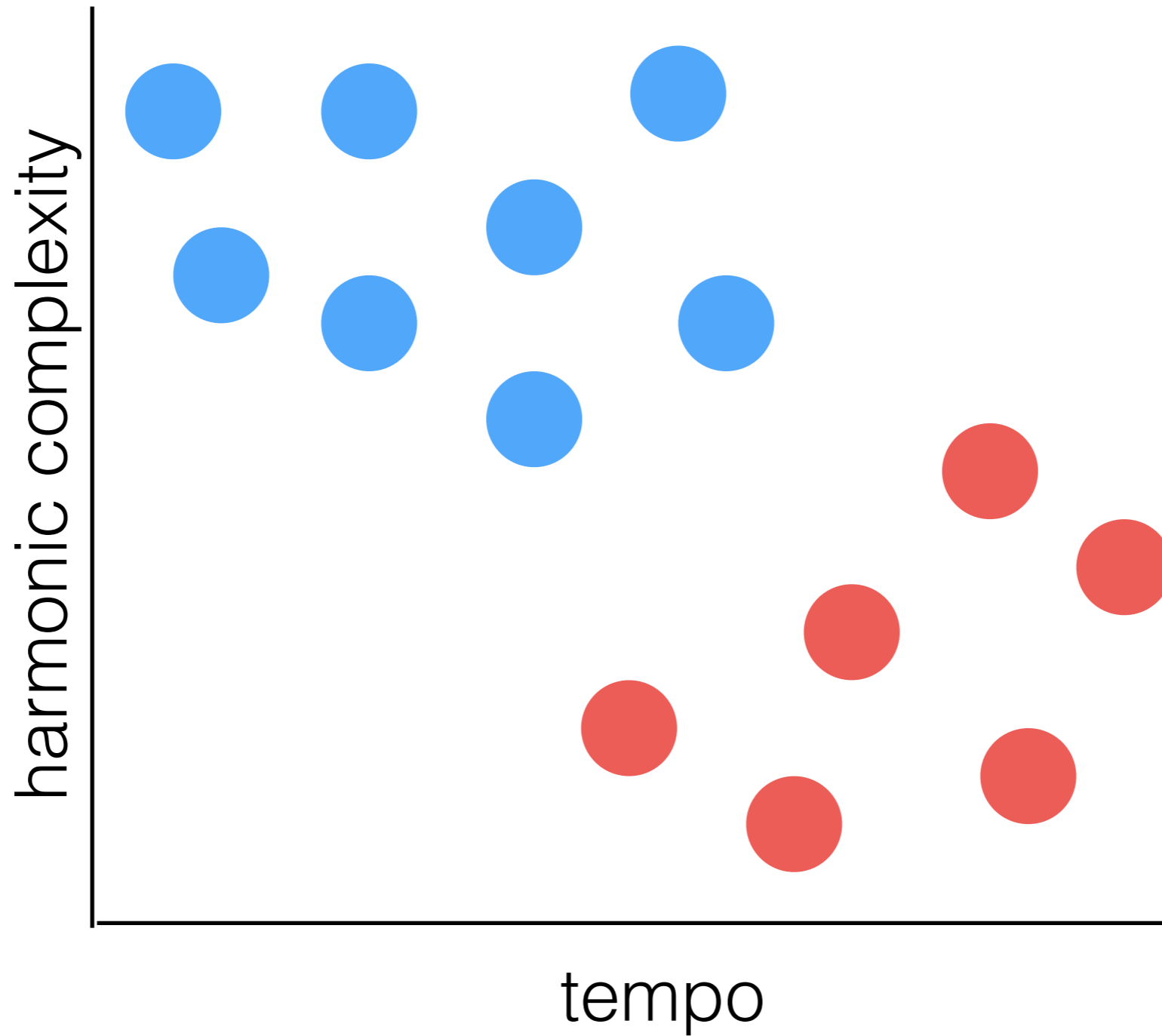
# Clustering

- Find groups of customers with similar tastes
- Find topics within a set of news articles
- Find genres within a library of music
- Extrapolating—make predictions about your new business based on behavior of similar old businesses

# Clustering



# Clustering



# K Means

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# K Means

“Hyperparameters” (i.e. not model parameters)

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```



# K Means

How many clusters we want to find

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# K Means

When to quit. Things aren't changing,  
or we have gotten bored.

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# K Means

Randomly guess what the means are  
(lots of ways to do this)

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# K Means

Repeat until your hyperparameters say  
to stop

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# K Means

Assign each point to its closest mean

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

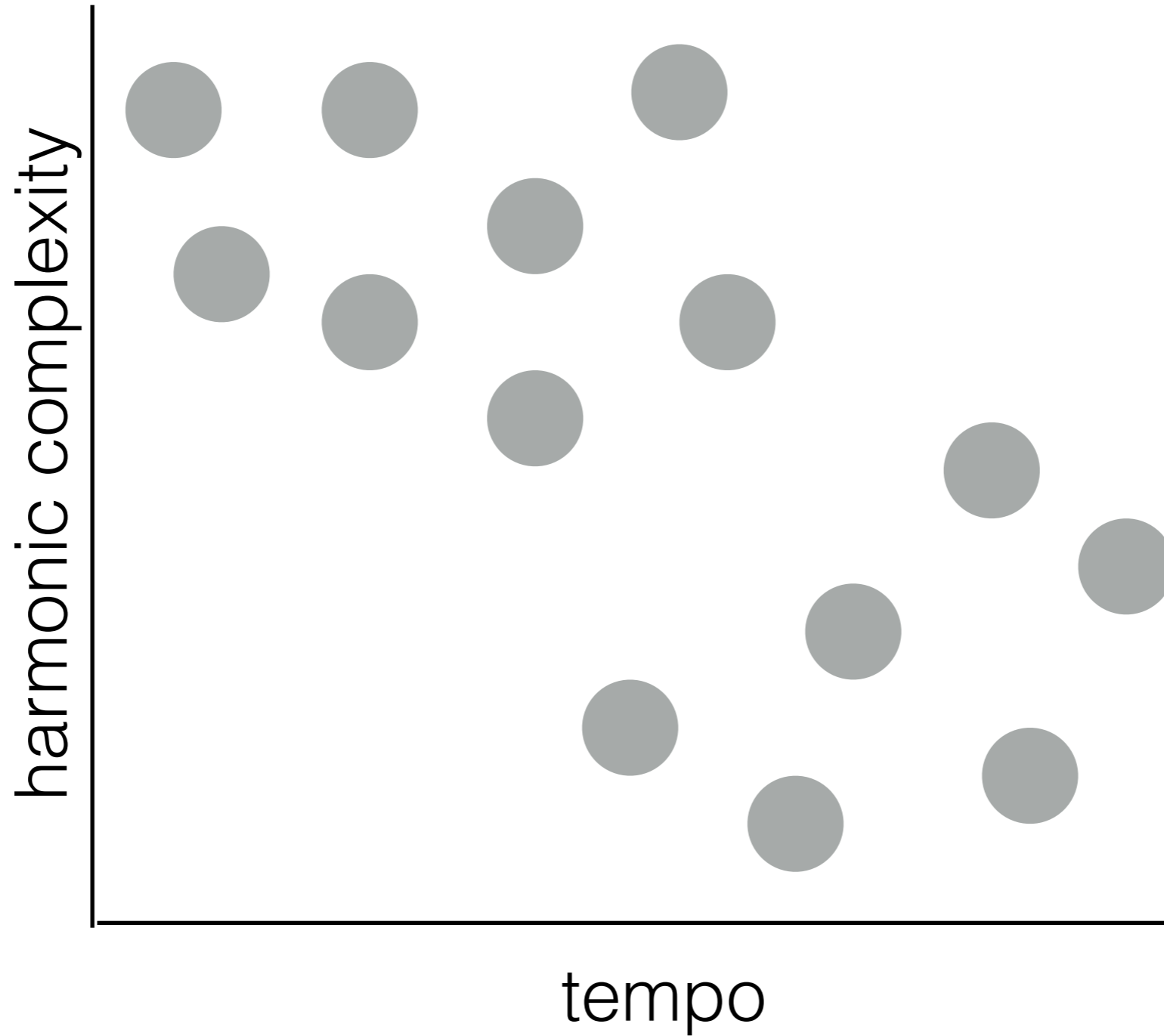
# K Means

Recompute the means to be the mean of the points assigned to each cluster

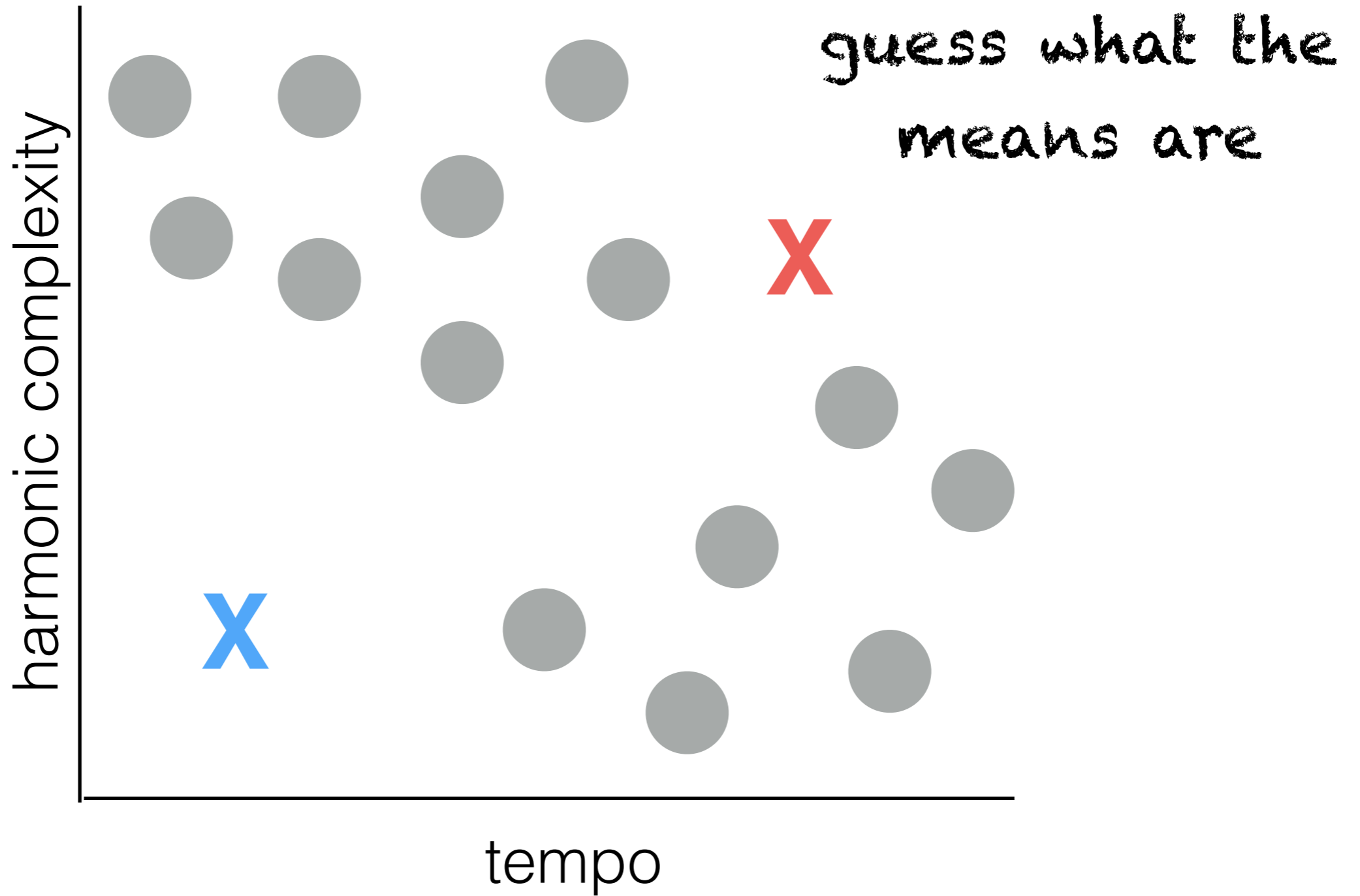
```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# K Means

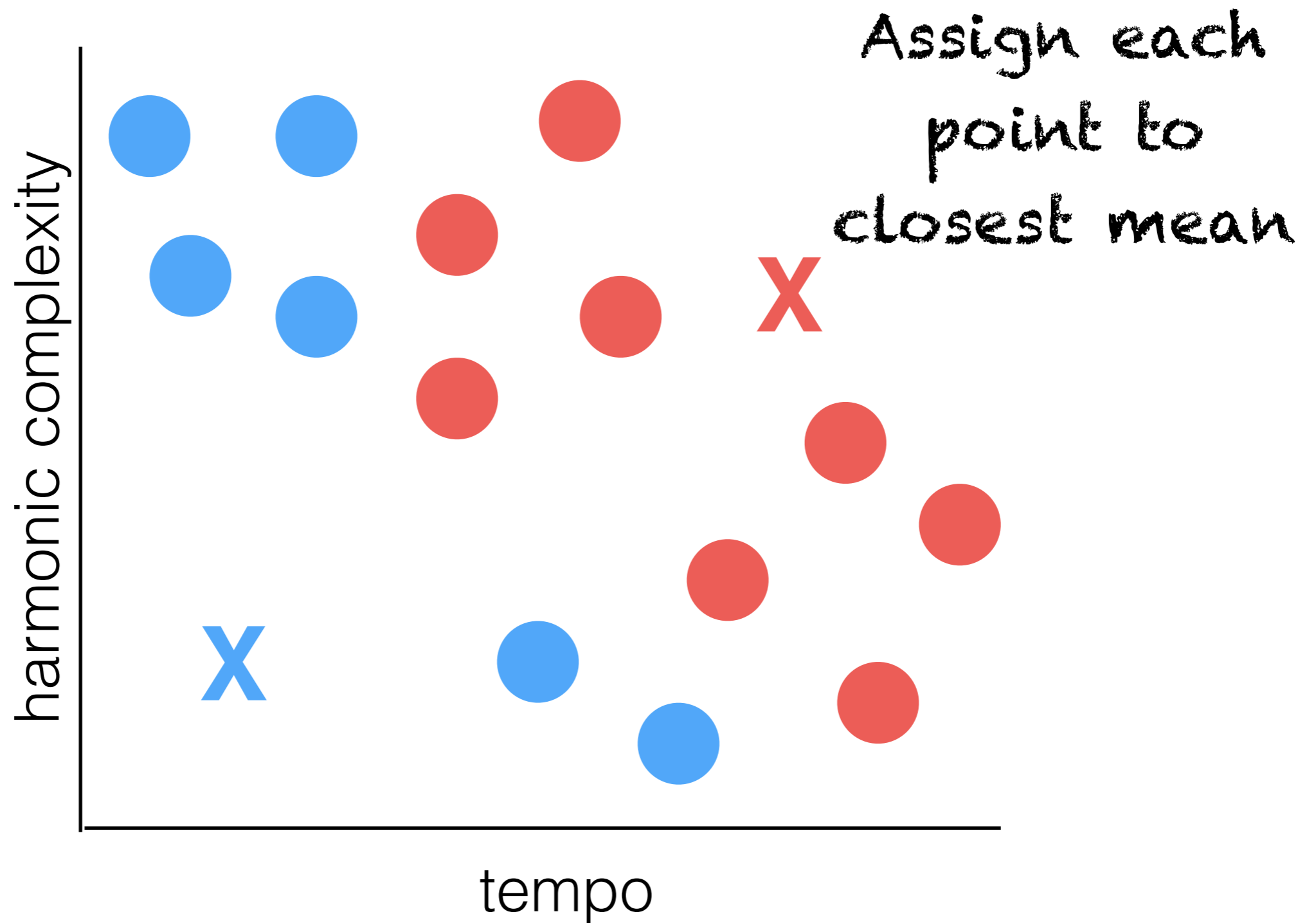


# K Means



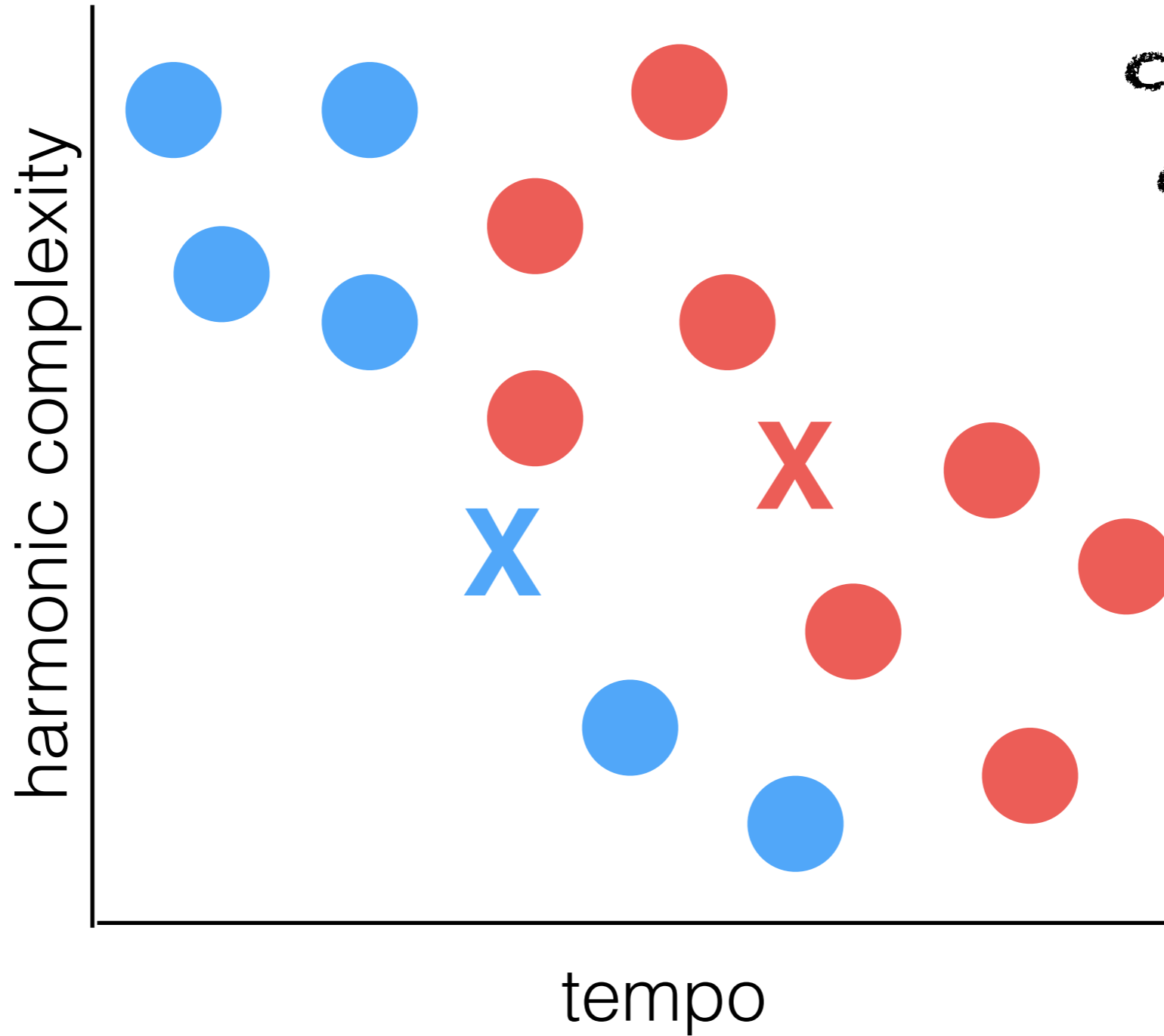


# K Means

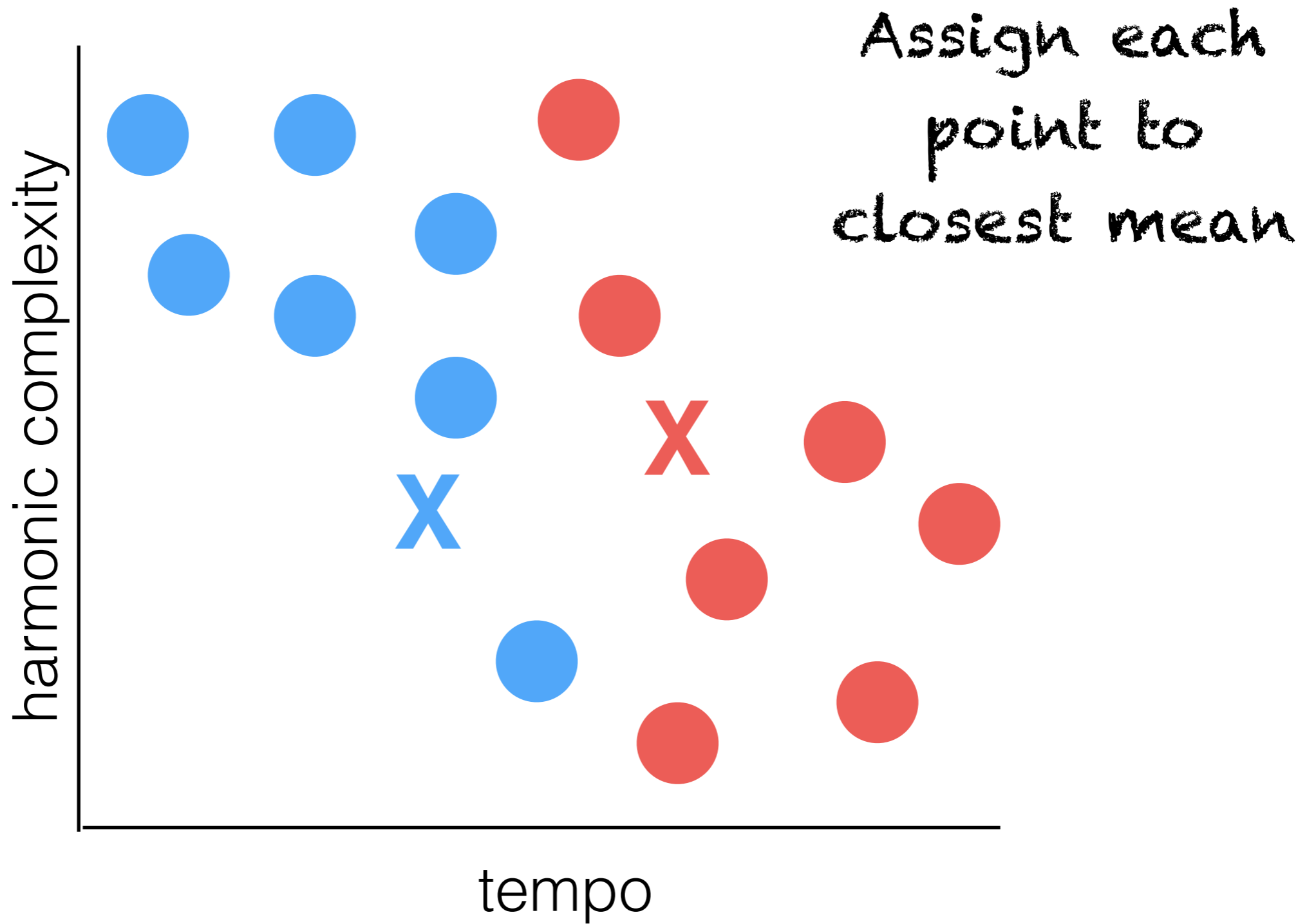


# K Means

re-compute  
means to be  
center of  
clusters

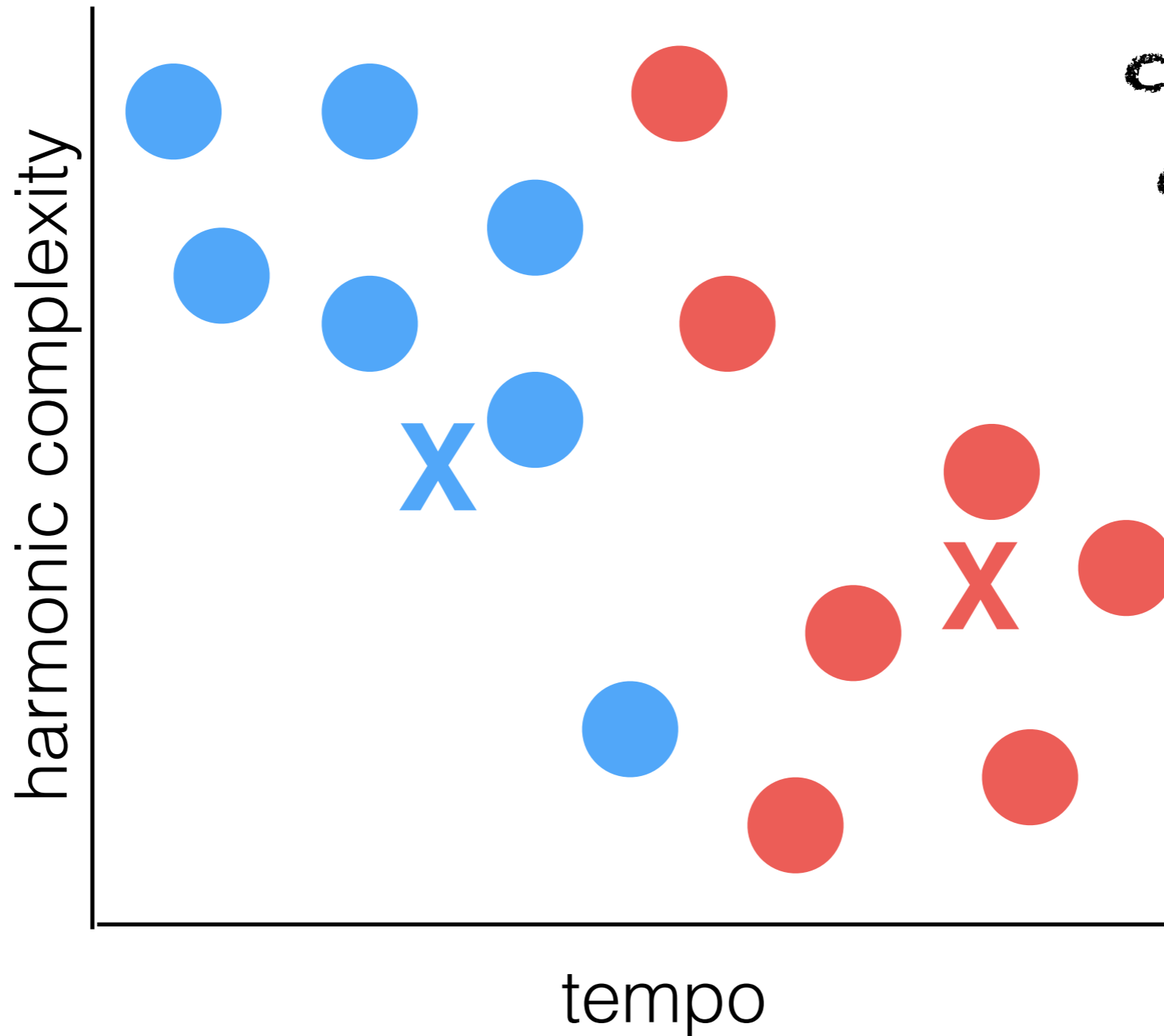


# K Means

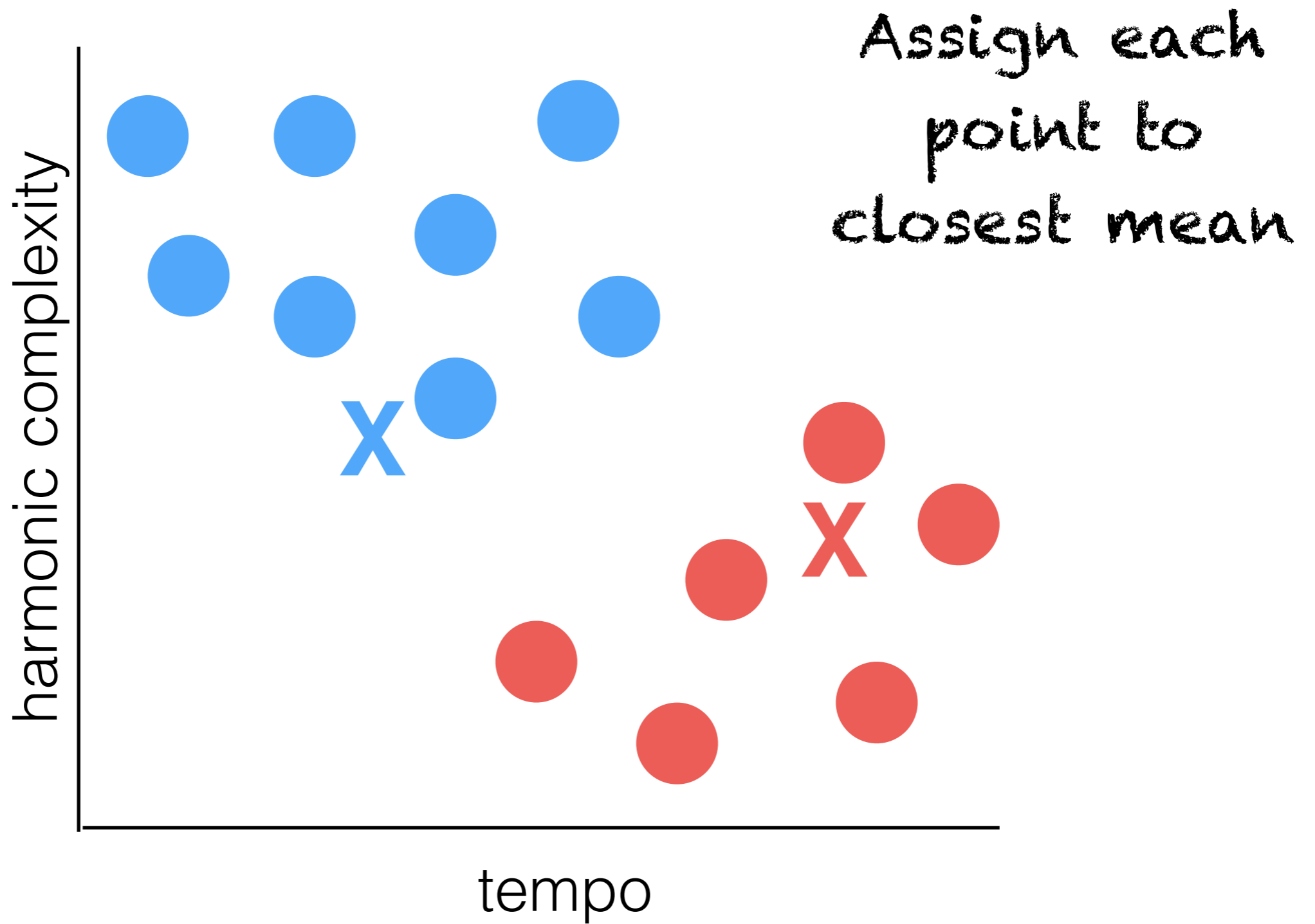


# K Means

re-compute  
means to be  
center of  
clusters

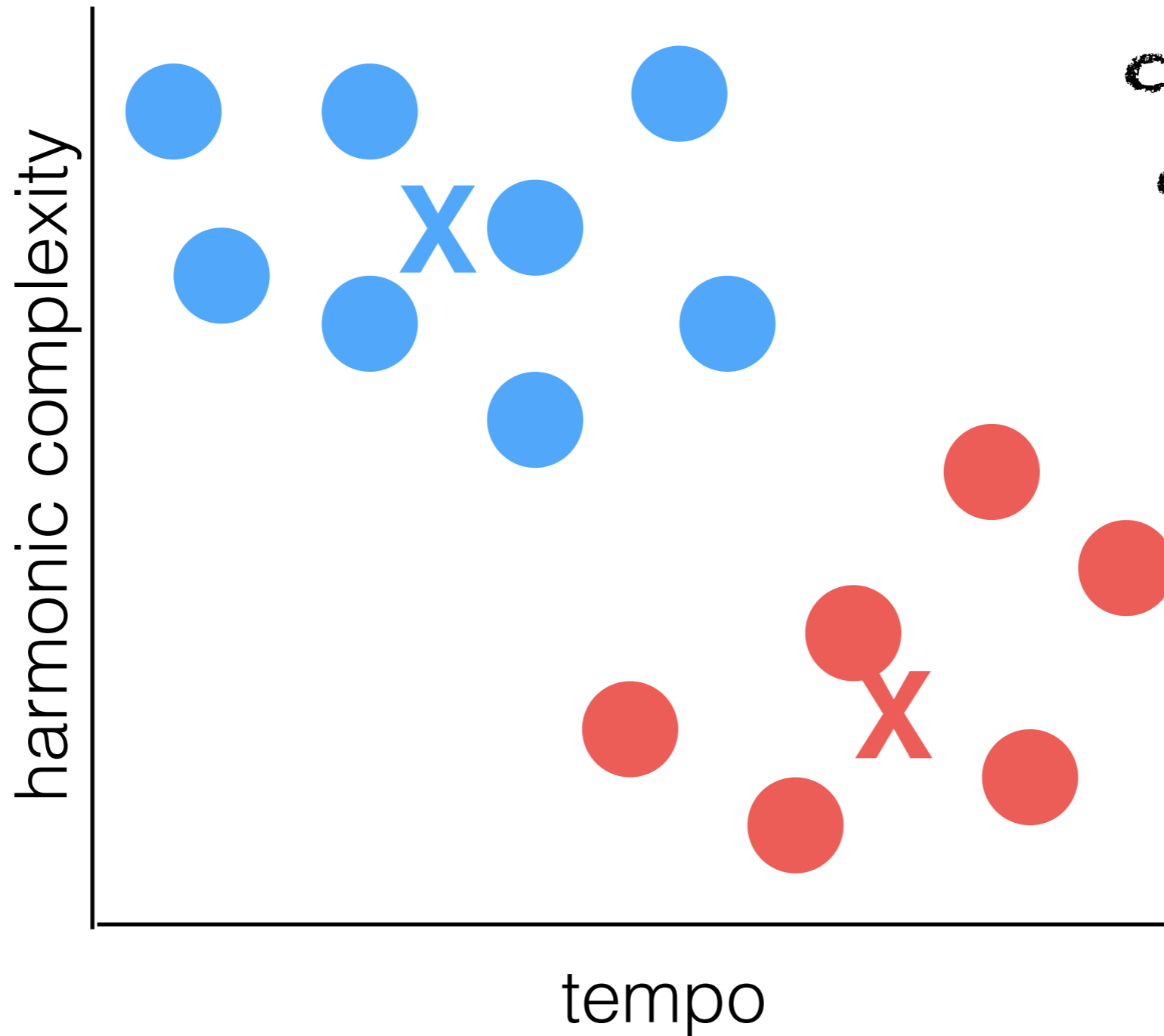


# K Means

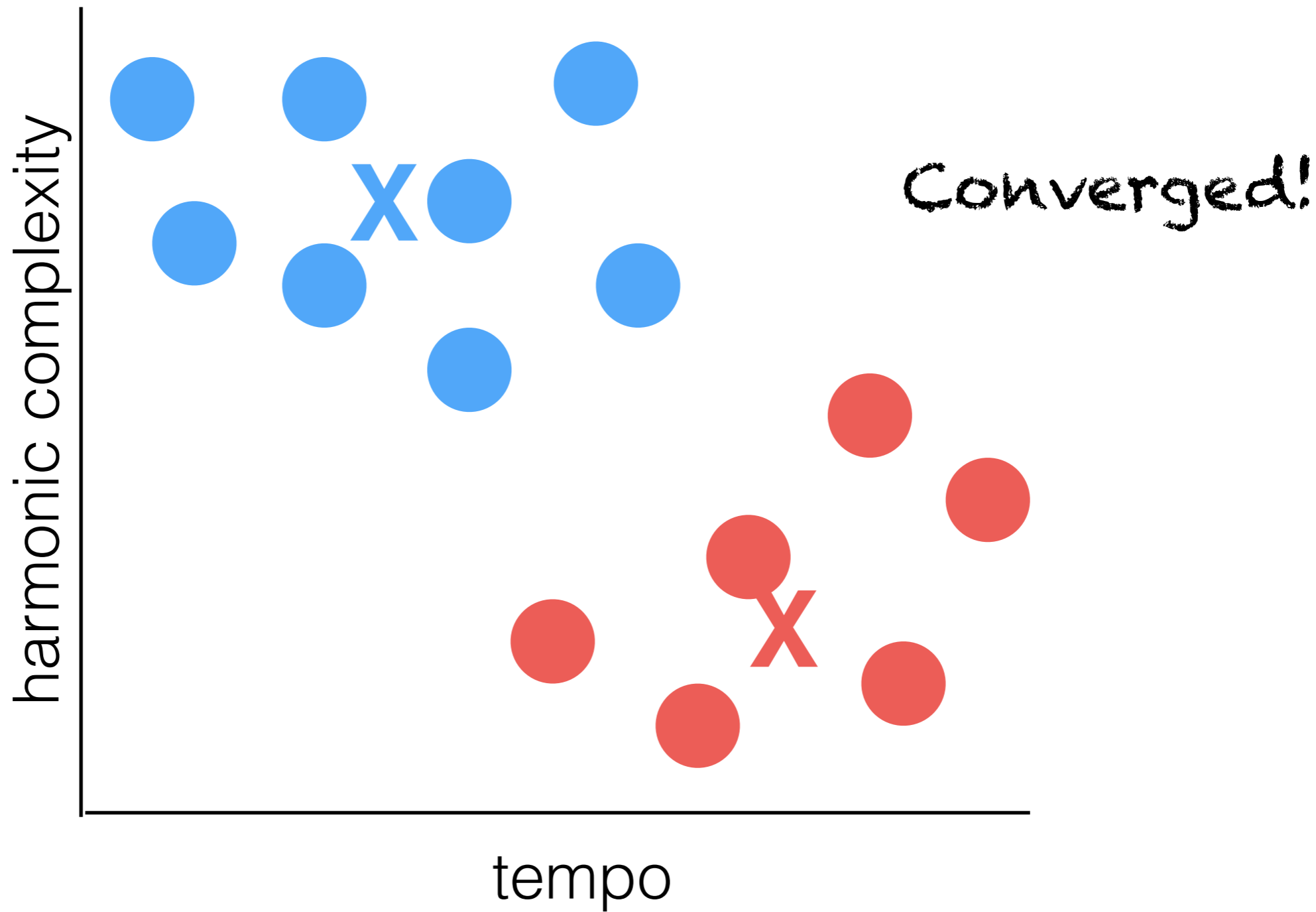


# K Means

re-compute  
means to be  
center of  
clusters



# K Means



# Clicker Question!



# Clicker Question!

What is the “loss” that we are trying to minimize here?

- (a) Number of clusters**
- (b) Distance of points to their respective clusters**
- (c) Distance between clusters**
- (d) Probability of observed data**

# Clicker Question!

What is the “loss” that we are trying to minimize here?

(a) Number of clusters

(b) Distance of points to their respective clusters

(c) Distance between clusters

(d) Probability of observed data

# Clicker Question!

What is the “loss” that we are trying to minimize here?

(a) Number of clusters

(b) Distance of points to their respective clusters

(c) Distance between clusters

(d) Probability of observed data

  
This in just a few slides!

# Clicker(/Discussion) Question!

Is this a good objective?

**(a) Yes**

**(b) No**

**(c) Sure, why not.**

# Clicker(/Discussion) Question!

Is this a good objective?

**(a) Yes**

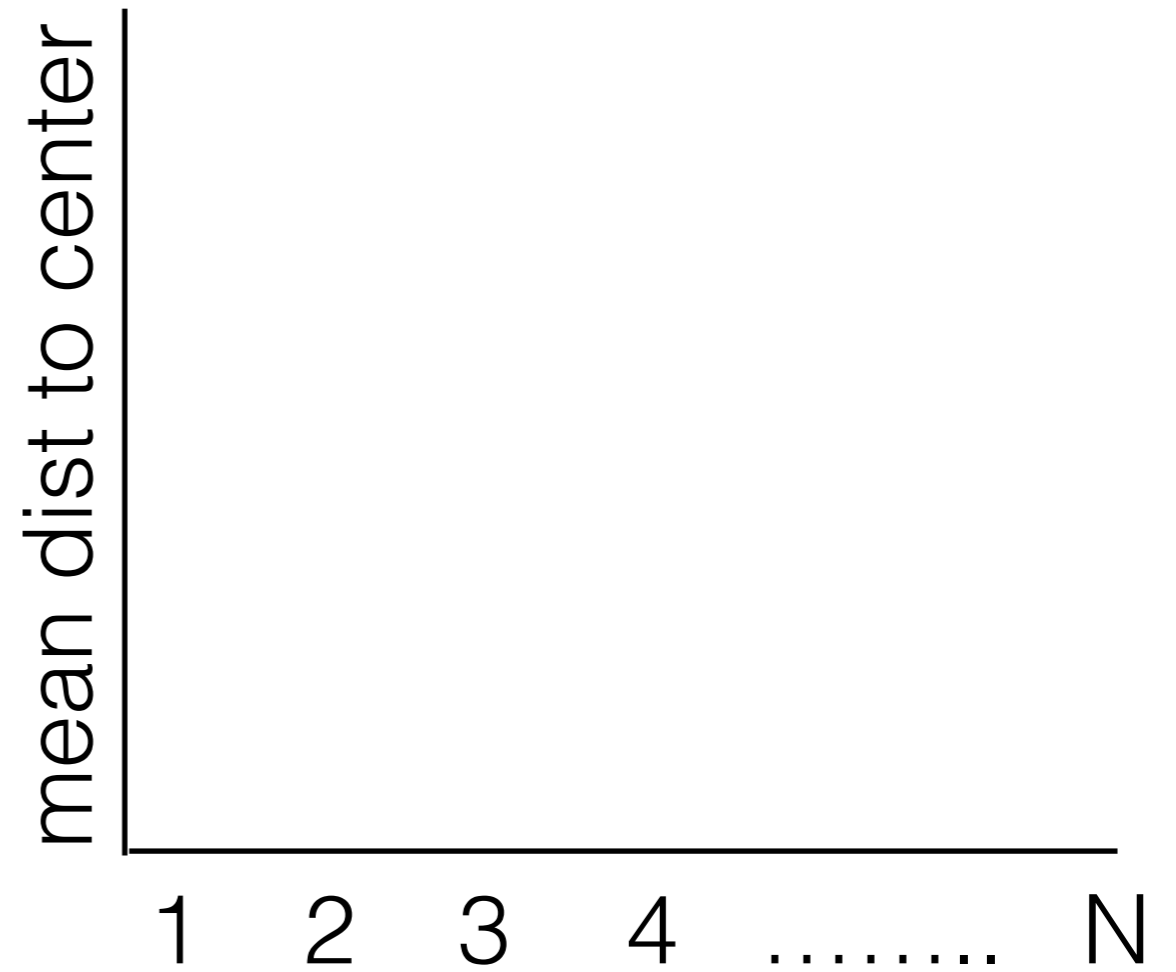
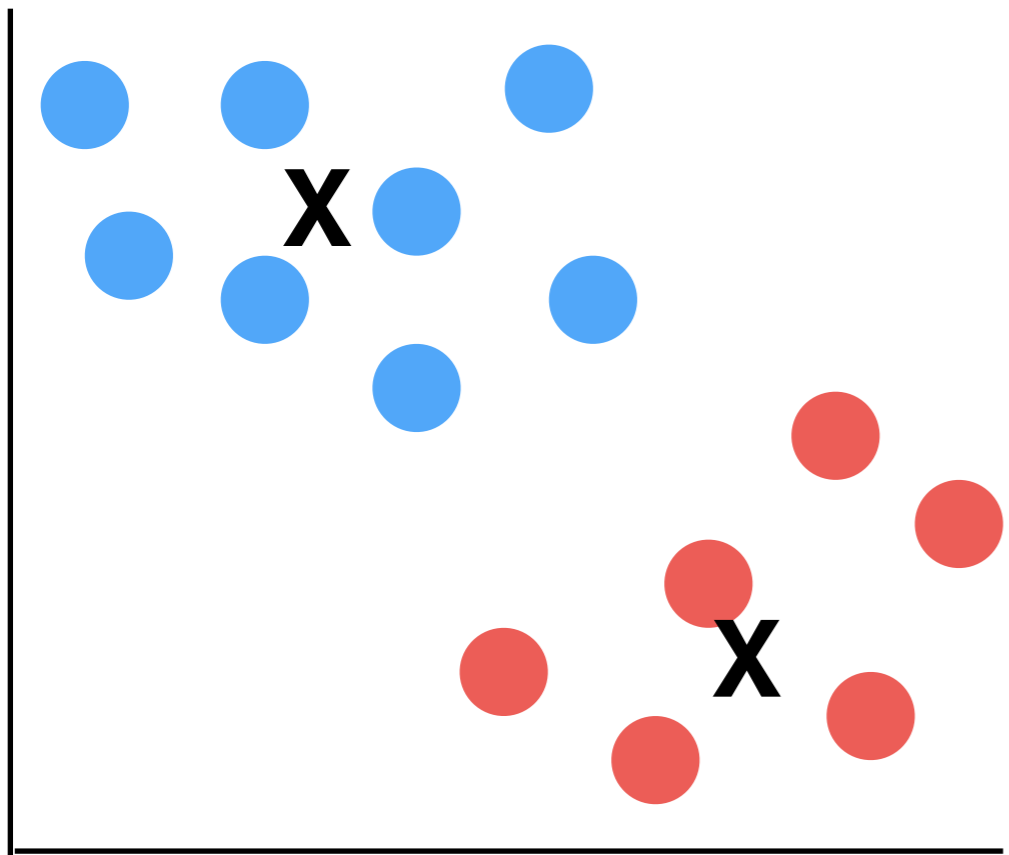
**(b) No**

**(c) Sure, why not.**

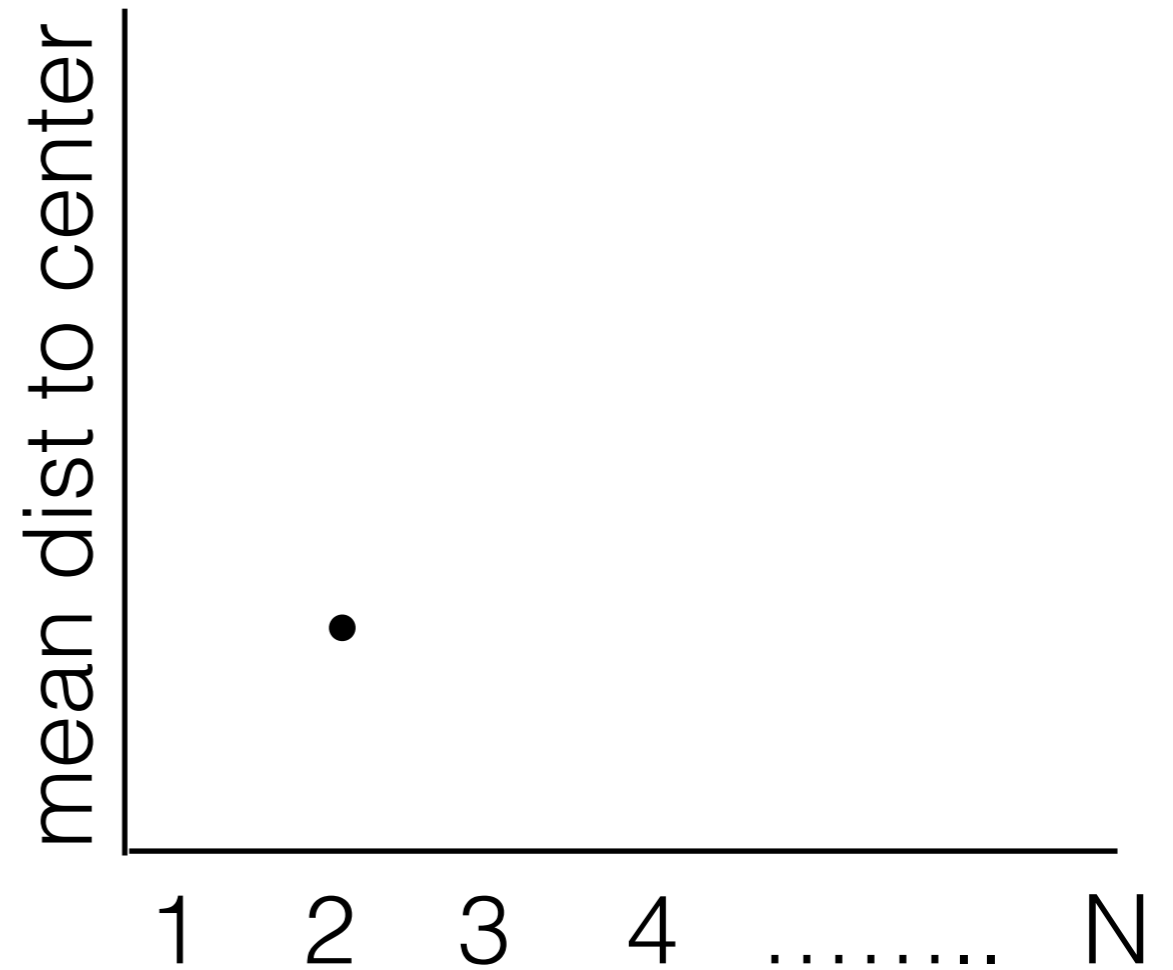
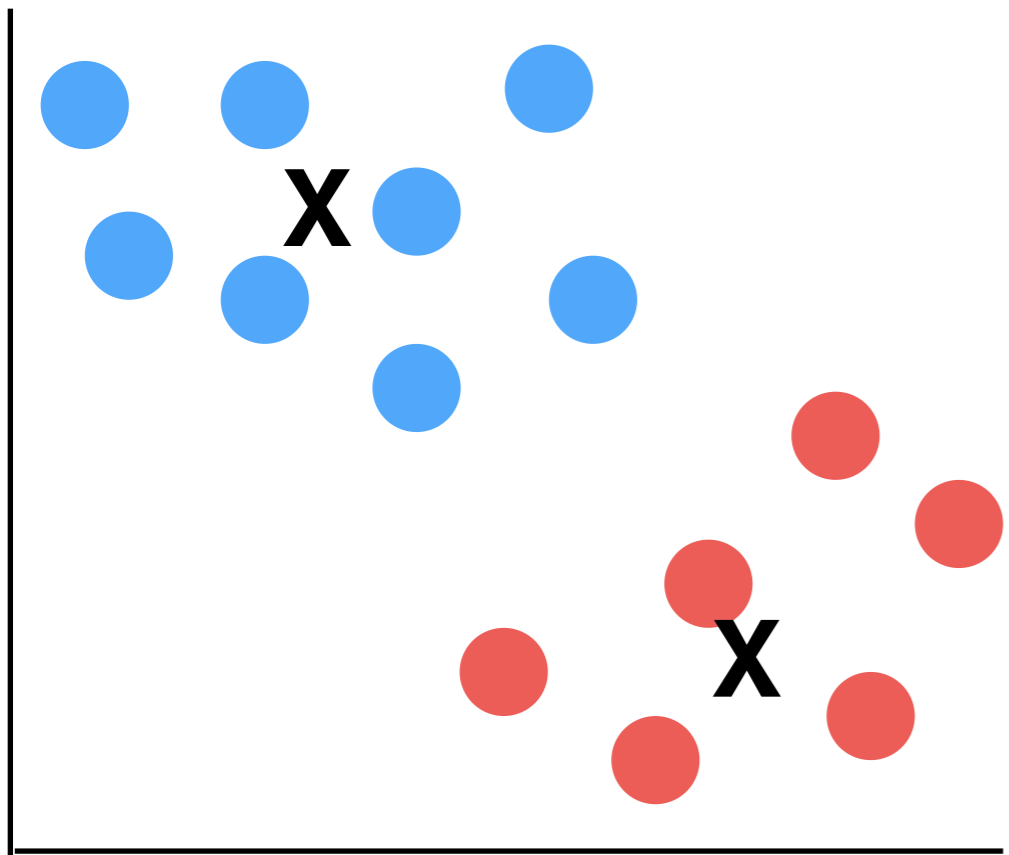
Potential problems?

(Hint: hyperparameters, generalization...)

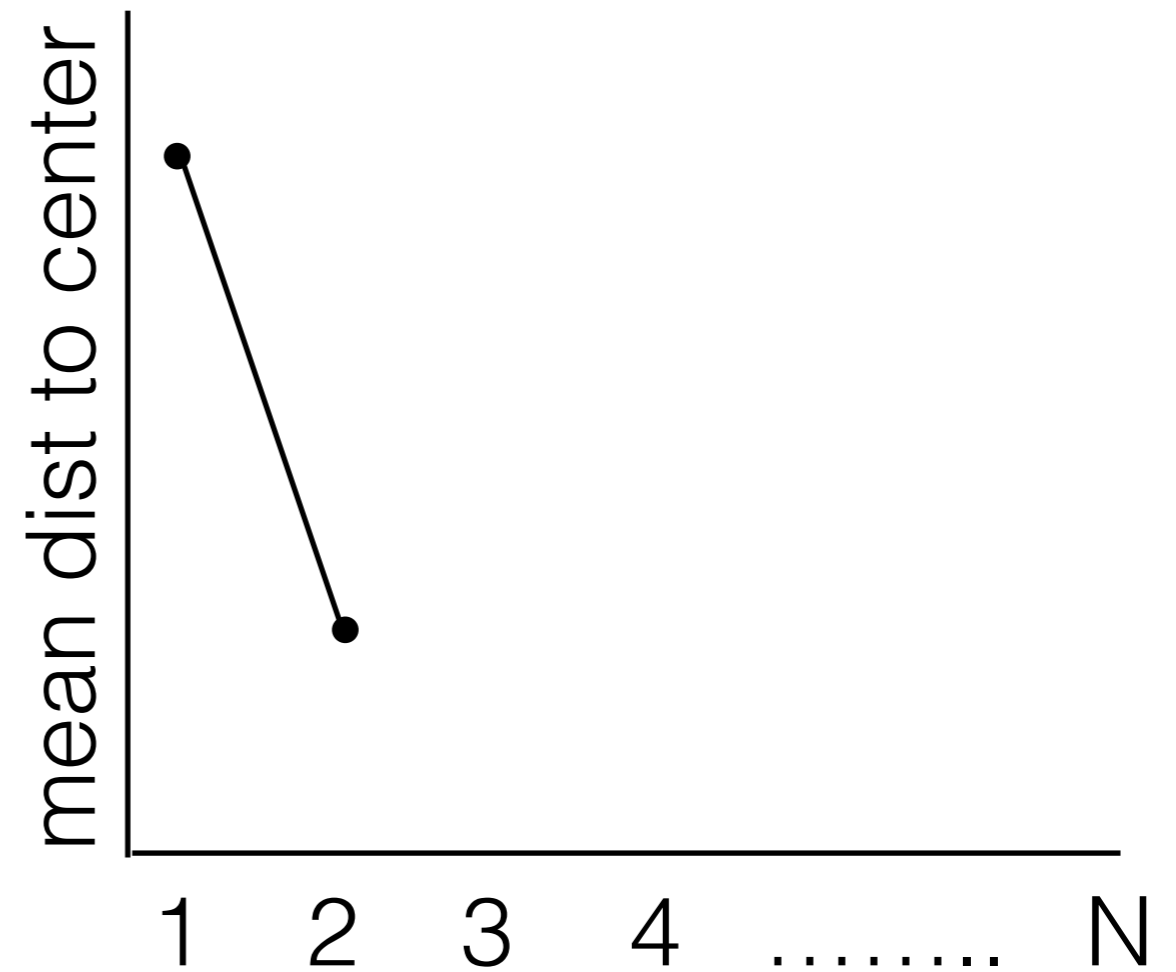
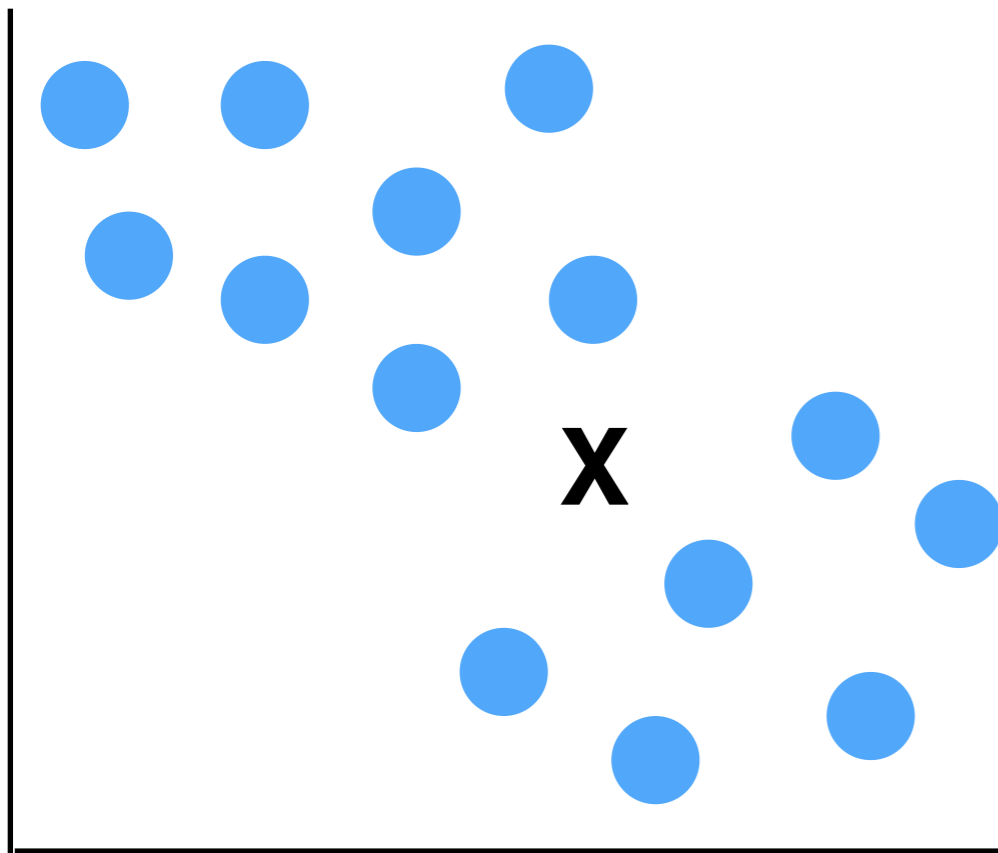
# How many clusters?



# How many clusters?

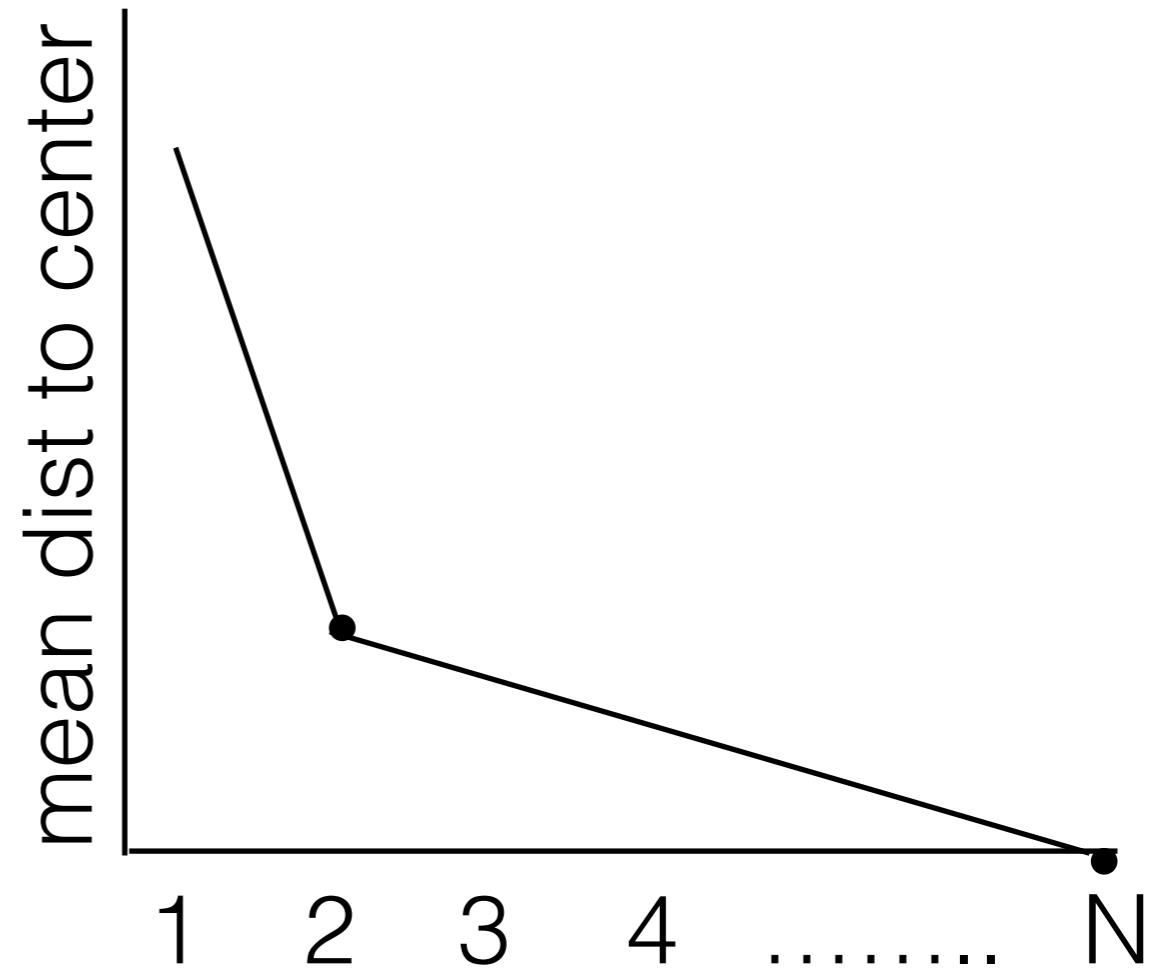
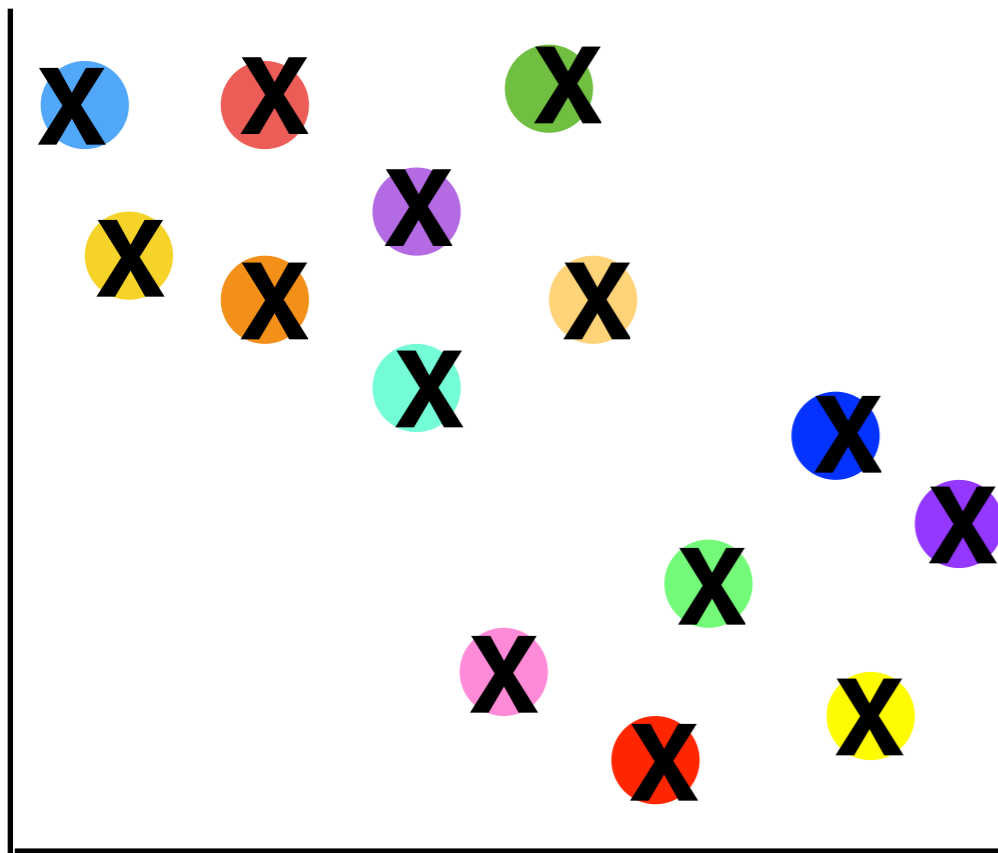


# How many clusters?

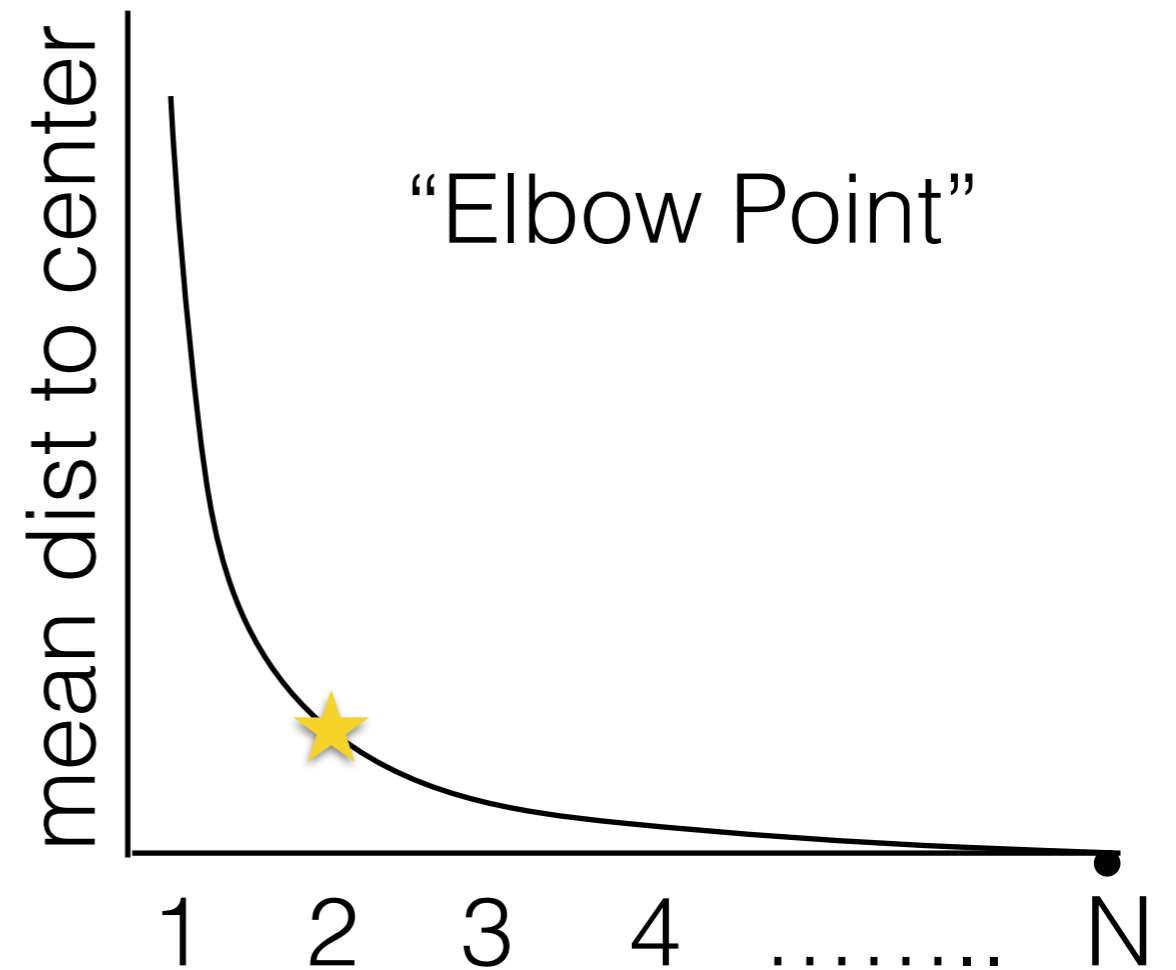
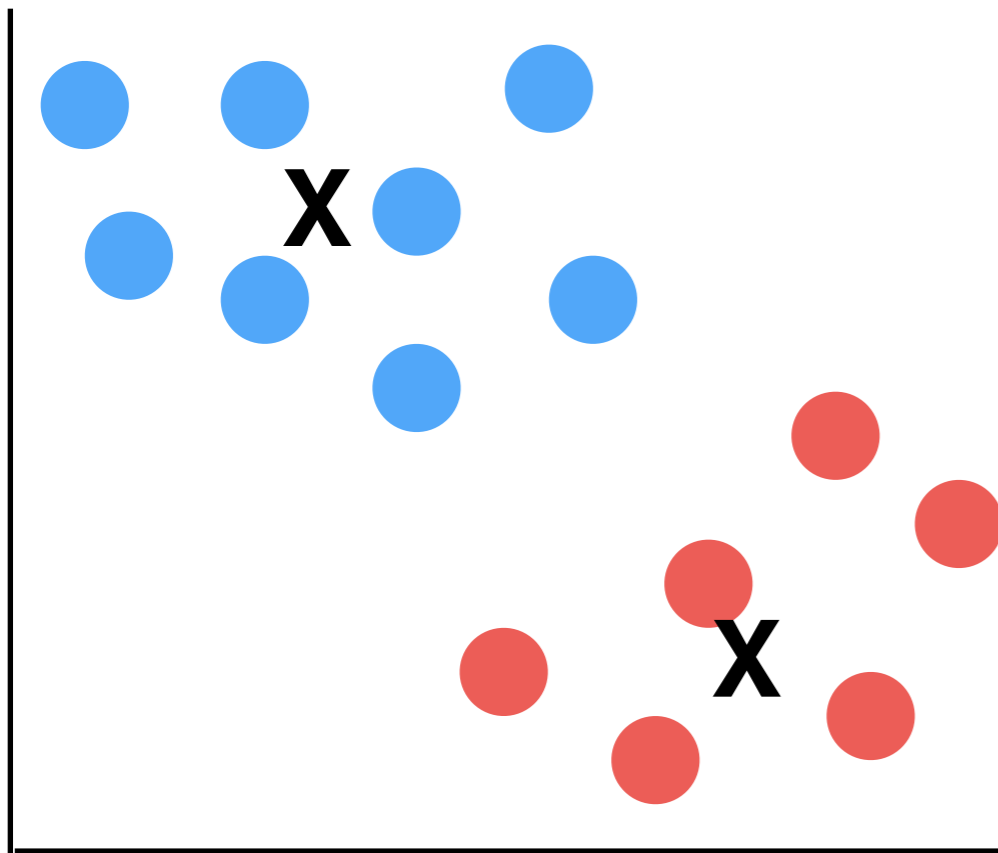




# How many clusters?



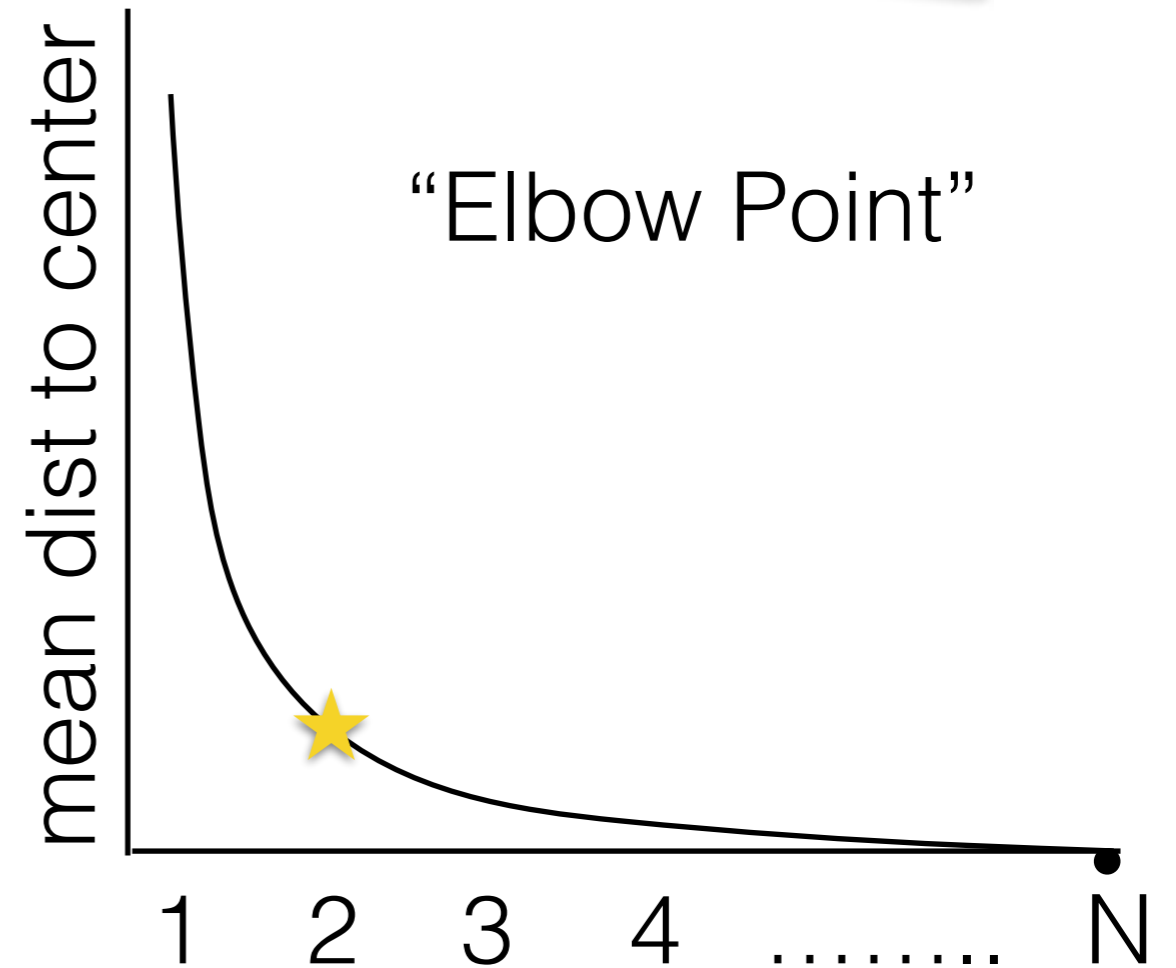
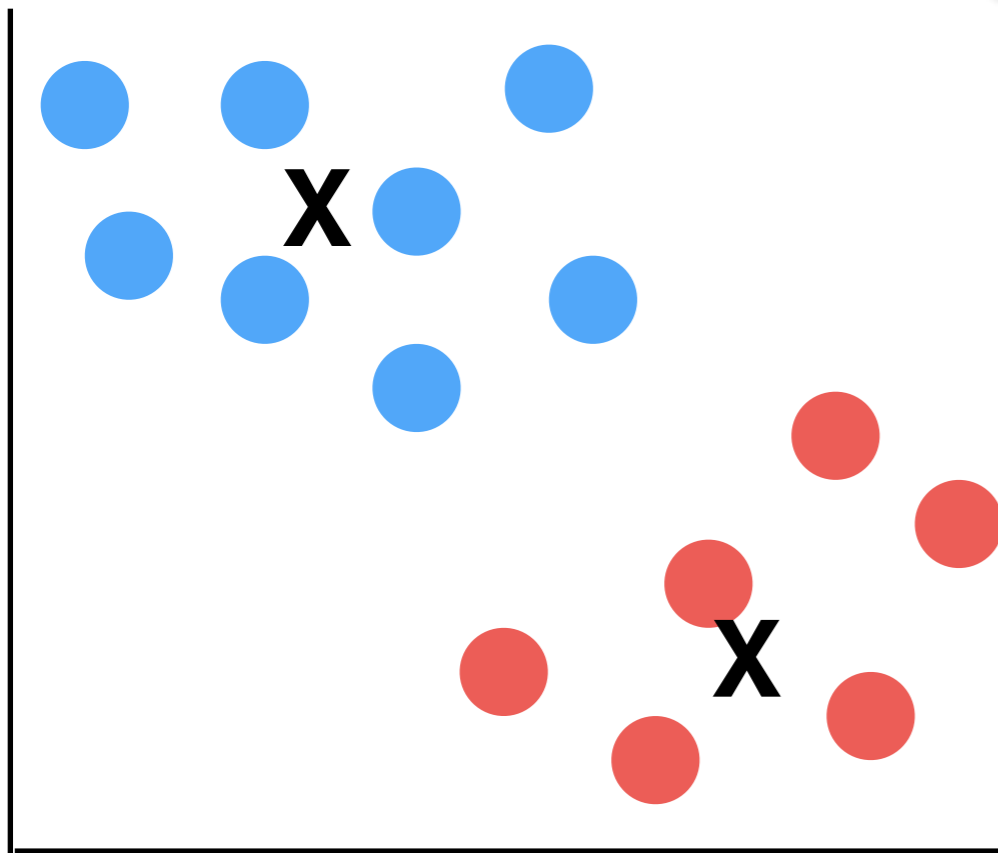
# How many clusters?



# How many clusters?

Other techniques:

- Silhouette
- Intuition/Divine Intervention
- LGTM



# How many

Other techniques:

- **Silhouette**
- Intuition/Divine Intervention
- LGTM

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

distance to own cluster

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

distance to next best cluster

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

Point”

N



# Expectation Maximization (EM)

# Expectation Maximization (EM)

```
define parameters: K, max_iter, min_diff

iter = 0
change = inf
means = [random() for _ in range(K)]
while iter < max_iter and change > min_diff:
    update_assignments()
    compute_new_means()
    change = max_i(dist(new_mean_i, old_mean_i))
    iter += 1
```

# Expectation Maximization (EM)

```
randomly initialize params
while not converged:
    data = estimate_likelihood(params)
    params = maximize_likelihood(data)
```



# Expectation Maximization (EM)

E Step: estimate the likelihood of data  
under current parameter setting

randomly initialize params

while not converged:

```
data = estimate_likelihood(params)
```

```
params = maximize_likelihood(data)
```

# Expectation Maximization (EM)

*M Step: adjust the the parameters so as to maximize the expectation of the data*

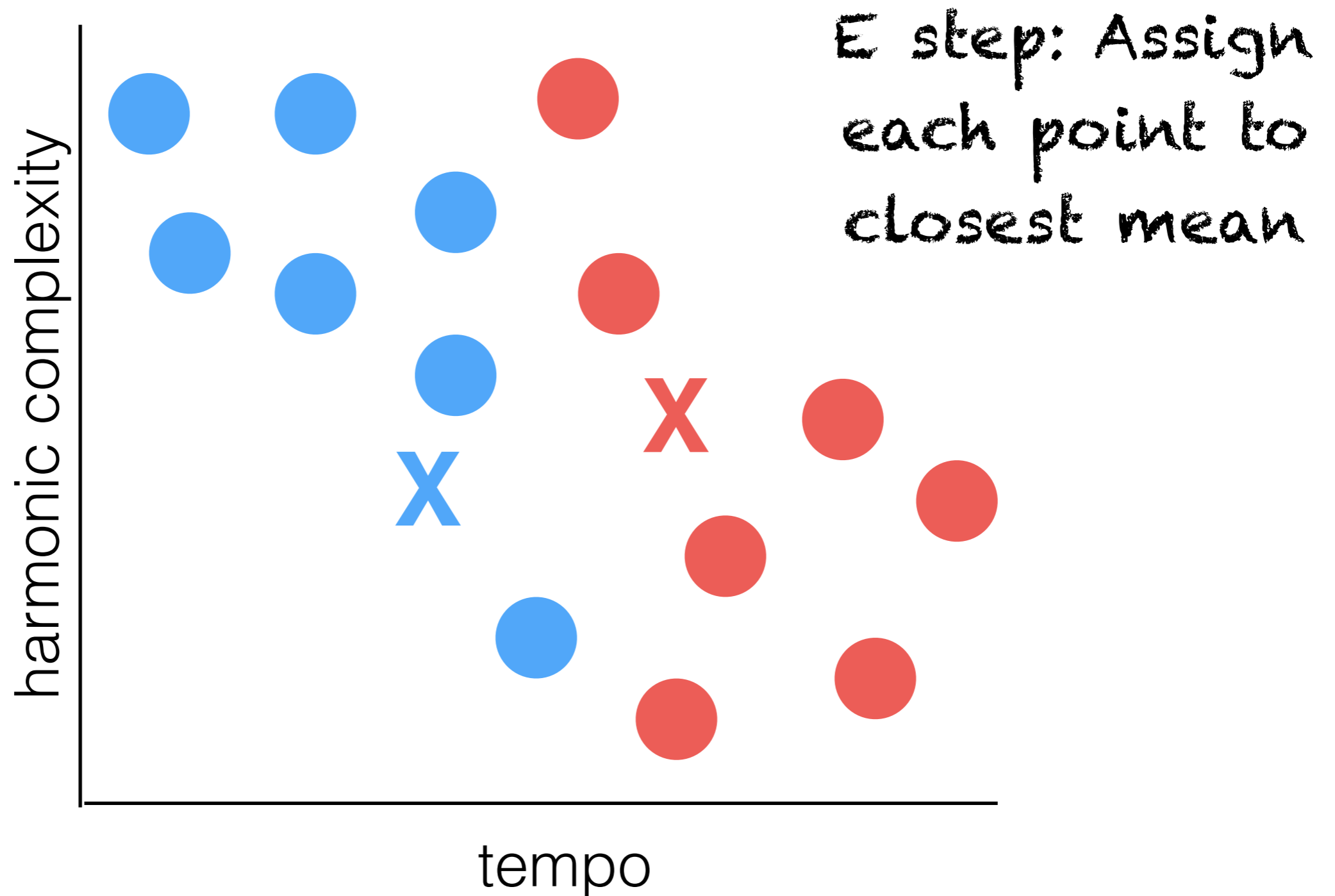
randomly initialize params

while not converged:

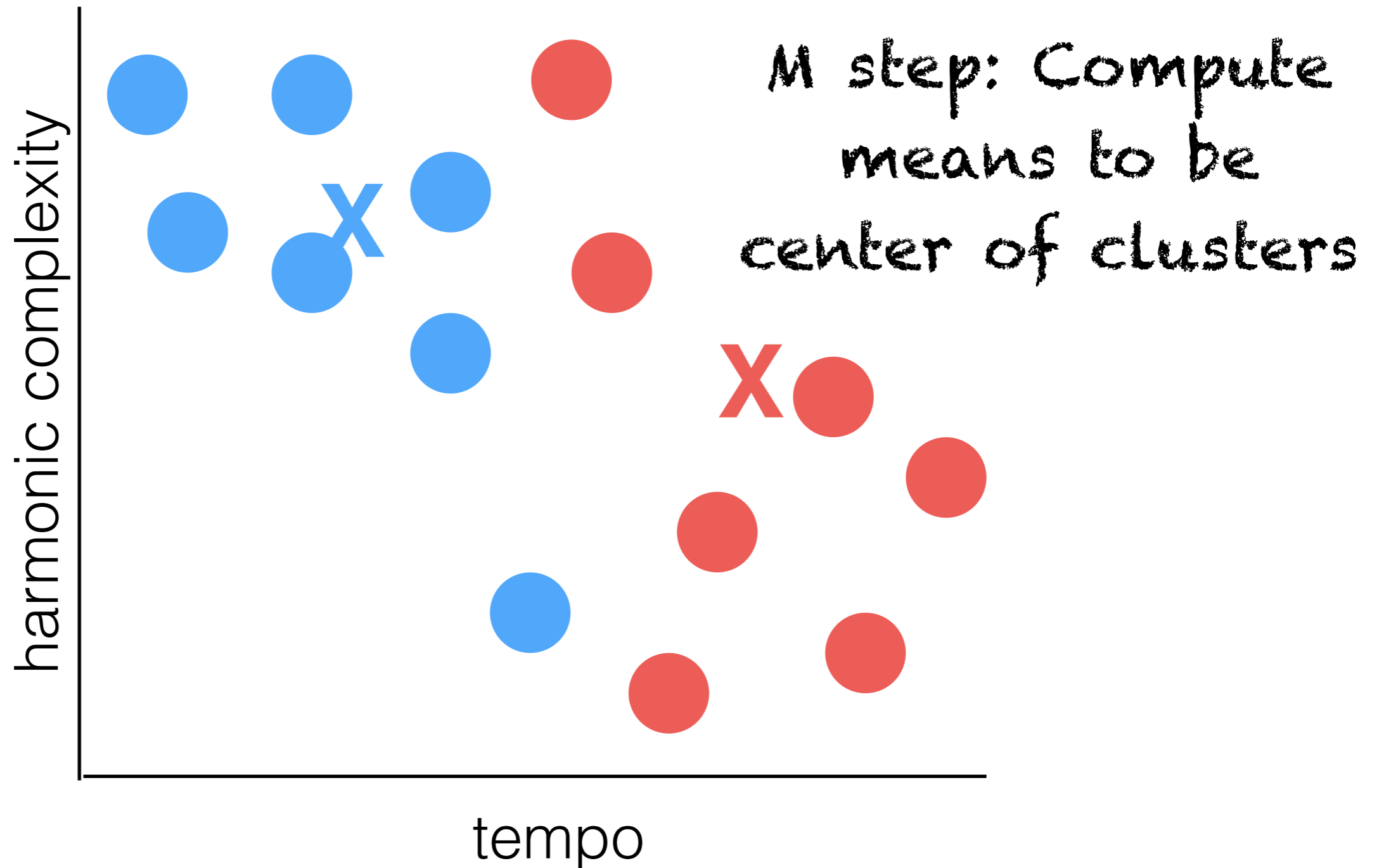
```
data = estimate_likelihood(params)
```

```
params = maximize_likelihood(data)
```

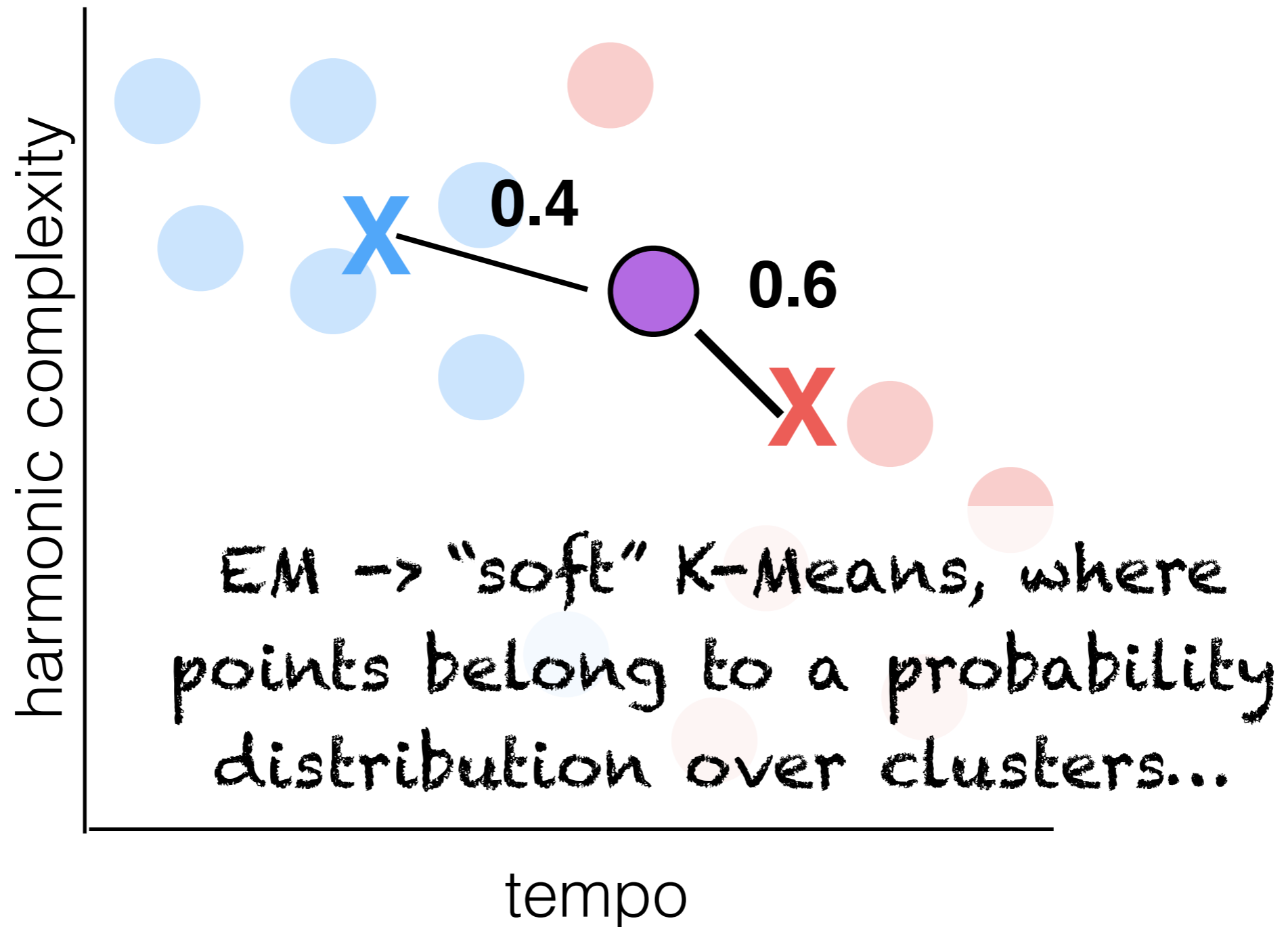
# Expectation Maximization (EM)



# Expectation Maximization (EM)



# Expectation Maximization (EM)

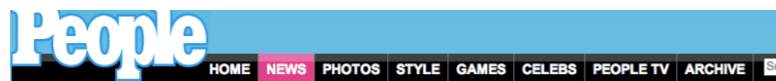


#tbot

Slide from crowdsourcing lecture



# Quality Control



Was arrested actress Heather Locklear because of the driving under the effect of an unknown medicine



Heather Locklear  
Photo by: Santa Barbara County Sheriff's Department

SPONSORED LINKS

The actress Heather Locklear that is known to the Amanda through the role from the series "Melrose Place" was arrested at this weekend in Santa Barbara (California) because of the driving under the effect of an unknown medicine. A female witness observed she attempted in quite strange way how to go from their parking space in Montecito, speaker of the traffic police of California told the warehouse "People". The female witness told in detail, that Locklear pressed "after 16:30 clock accelerator and a lot of noise did when she attempted to move their car towards behind or forward from the parking space, and when it went backwards, she pulled itself together unites Male at their sunglasses". A little later the female witness that did probably

- Why was Heather Locklear arrested?
- Why did the bystander call emergency services?
- Where did the witness see her acting abnormally?

Second-Pass HIT

MLB WORLD SERIES SURVEY (< 1 min survey) **Eligible for \$5 bonus, US only**

Requester: [Danielle Limberg](#)

HIT E  
Time

Incentive Pay

$$L(\theta; \mathbf{X}) = p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

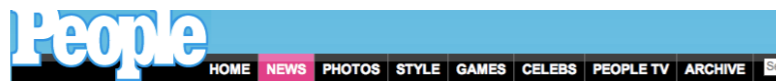
Statistical Models

#tbot

Slide from crowdsourcing lecture



# Quality Control



Was arrested actress Heather Locklear because of the driving under the effect of an unknown medicine



Heather Locklear  
Photo by: Santa Barbara County Sheriff's Department

SPONSORED LINKS

The actress Heather Locklear that is known to the Amanda through the role from the series "Melrose Place" was arrested at this weekend in Santa Barbara (California) because of the driving under the effect of an unknown medicine. A female witness observed she attempted in quite strange way how to go from their parking space in Montecito, speaker of the traffic police of California told the warehouse "People". The female witness told in detail, that Locklear pressed "after 16:30 clock accelerator and a lot of noise did when she attempted to move their car towards behind or forward from the parking space, and when it went backwards, she pulled itself together unites Male at their sunglasses". A little later the female witness that did probably

- Why was Heather Locklear arrested?
- Why did the bystander call emergency services?
- Where did the witness see her acting abnormally?

Second-Pass HIT

MLB WORLD SERIES SURVEY (< 1 min survey) **Eligible for \$5 bonus, \$5 only**

Requester: [Danielle Limberg](#)

HIT E  
Time

Incentive Pay

$$L(\theta; \mathbf{X}) = p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

Statistical Models

(That I don't think we actually covered, but its cool its fine...)

# Goal: Find “true” labels despite noisy annotations from workers...

worker1 worker2 worker3 worker4 worker5

email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam



# Goal: Find “true” labels despite noisy annotations from workers...

worker1 worker2 worker3 worker4 worker5

Easy! If you tell me how much to trust each worker, I can trivially compute labels

			not	not	spam
		spam	spam	spam	spam
			not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

# Goal: Find “true” labels despite noisy annotations from workers...

worker1 worker2 worker3 worker4 worker5

Easy! If you tell me how much to trust each worker, I can trivially compute labels

Sure, just tell me the labels and I can easily figure out which workers to trust.

email1	spam	spam	not	not	spam
email2	spam	spam	spam	spam	spam
email3	spam	spam	spam	spam	spam
email4	spam	spam	spam	spam	spam
email5	spam	not	not	not	spam

# Goal: Find “true noisy annotation **EM EVERYTHING!!!!**

worker1 worker2 \



Easy! If you tell me how much to trust each worker, I can trivially compute labels

spam spam spam

Sure, just tell me the labels and I can easily figure out which workers to trust.

email3 spam

email4 spam spam

email5 spam not not not spam

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	?	?
email2	?	?
email3	?	?
email4	?	?
email5	?	?

84

w1	spam	not
spam	?	?
not	?	?

w2	spam	not
spam	?	?
not	?	?

w3	spam	not
spam	?	?
not	?	?

w4	spam	not
spam	?	?
not	?	?

w5	spam	not
spam	?	?
not	?	?

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

$P(\text{email1 is spam})$

	spam	not
email1	?	?
email2	?	?
email3	?	?
email4	?	?
email5	?	?

w1	spam	not
spam	?	?
not	?	?

w2	spam	not
spam	?	?
not	?	?

w3	spam	not
spam	?	?
not	?	?

w4	spam	not
spam	?	?
not	?	?

w5	spam	not
spam	?	?
not	?	?

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam

$P(w1 \text{ says spam} \mid \text{not spam})$

email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	?	?
email2	?	?
email3	?	?
email4	?	?
email5	?	?

w1	spam	not
spam	?	?
not	?	?

w2	spam	not
spam	?	?
not	?	?

w3	spam	not
spam	?	?
not	?	?

w4	spam	not
spam	?	?
not	?	?

w5	spam	not
spam	?	?
not	?	?

Assume  
all  
workers  
are  
perfect

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	?	?
email2	?	?
email3	?	?
email4	?	?
email5	?	?

	w1	spam	not
spam	1	0	
not	0	1	

	w2	spam	not
spam	1	0	
not	0	1	

	w3	spam	not
spam	1	0	
not	0	1	

	w4	spam	not
spam	1	0	
not	0	1	

	w5	spam	not
spam	1	0	
not	0	1	

Compute labels using majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	?	?
email2	?	?
email3	?	?
email4	?	?
email5	?	?

w1	spam	not
spam	1	0
not	0	1

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1



# Clicker Question!

Compute labels using majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	?	?

## Clicker Question!

- (a) 0.4, 0.6
- (b) 0.6, 0.4
- (c) 0.8, 0.2
- (d) 1.0, 0.0

w1	spam	not
spam	1	0
not	0	1

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Compute labels using majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	?	?

# Clicker Question!

- (a) 0.4, 0.6
- (b) 0.6, 0.4
- (c) 0.8, 0.2
- (d) 1.0, 0.0

w1	spam	not
spam	1	0
not	0	1

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Compute labels using majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	<b>0.4</b>	?
email2	?	?
email3	?	?
email4	?	?
email5	?	?

w1	spam	not
spam	1	0
not	0	1

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Compute labels using majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	<b>0.6</b>
email2	?	?
email3	?	?
email4	?	?
email5	?	?

	w1	spam	not
spam	1	0	
not	0	1	

	w2	spam	not
spam	1	0	
not	0	1	

	w3	spam	not
spam	1	0	
not	0	1	

	w4	spam	not
spam	1	0	
not	0	1	

	w5	spam	not
spam	1	0	
not	0	1	

Compute labels using majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	1	0
email3	0.4	0.6
email4	0.8	0.2
email5	0.4	0.6

	w1	spam	not
spam	1	0	
not	0	1	

	w2	spam	not
spam	1	0	
not	0	1	

	w3	spam	not
spam	1	0	
not	0	1	

	w4	spam	not
spam	1	0	
not	0	1	

	w5	spam	not
spam	1	0	
not	0	1	

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	1	0
email3	0.4	0.6
email4	0.8	0.2
email5	0.4	0.6

w1	spam	not
spam	1	0
not	0	1

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	1	0
email3	0.4	0.6
email4	0.8	0.2
email5		0.6

w1	spam	not
spam		
not		

w2	spam	not
spam		
not		

w3	spam	not
spam		
not		

w4	spam	not
spam		
not		

w5	spam	not
spam		
not		



# Clicker Question!

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	1	0
email3	0.4	0.6
email4	0.8	0.2
email5		0.6

w1	spam	not
spam	?	?
not		

w2	spam	not
spam		
not		

# Clicker Question!

- (a) 0.4, 0.6
- (b) 0.6, 0.4
- (c) 0.8, 0.2
- (d) 1.0, 0.0

spam	
not	

w5	spam	not
spam		
not		

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	1	0
email3	0.4	0.6
email4	0.8	0.2
email5		0.6

w1	spam	not
spam	?	?
not		

w2	spam	not
spam		
not		

spam	
not	

w5	spam	not
spam		
not		

# Clicker Question!

- (a) 0.4, 0.6
- (b) 0.6, 0.4
- (c) 0.8, 0.2
- (d) 1.0, 0.0**

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	<b>1</b>	0
email3	0.4	0.6
email4	<b>0.8</b>	0.2
email5		0.6

100

w1	spam	not
spam	1	
not		

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	<b>1</b>	0
email3	0.4	0.6
email4	<b>0.8</b>	0.2
email5		0.6

w1	spam	not
spam	1	0
not		

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	<b>0.6</b>
email2	1	0
email3	0.4	<b>0.6</b>
email4	0.8	0.2
email5		<b>0.6</b>

w1	spam	not
spam	1	0
not	0.67	

w2	spam	not
spam	1	0
not	0	1

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	1	0
not	0	1

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	<b>0.6</b>
email2	1	0
email3	0.4	<b>0.6</b>
email4	0.8	0.2
email5		<b>0.6</b>

	w1	<b>spam</b>	not
spam		1	0
<b>not</b>		0.67	0.33

	w2	spam	not
spam		1	0
not		0	1

	w3	spam	not
spam		1	0
not		0	1

	w4	spam	not
spam		1	0
not		0	1

	w5	spam	not
spam		1	0
not		0	1

Assume these labels, and recompute confusion matrices

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	0.4	0.6
email2	1	0
email3	0.4	0.6
email4	0.8	0.2
email5		0.6

	w1	spam	not
spam	1	0	
not	0.67	0.33	

	w2	spam	not
spam	1	0	
not	0.33	0.67	

	w3	spam	not
spam	1	0	
not	0	1	

	w4	spam	not
spam	1	0	
not	0	1	

	w5	spam	not
spam	0.5	0.5	
not	1	0	



Recompute labels using (weighted) majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	1.5	
email2		
email3		
email4		
email5		

105

	w1	spam	not
spam	<b>1</b>	0	
not	0.67	0.33	

	w2	spam	not
spam	1	<b>0</b>	
not	0.33	0.67	

	w3	spam	not
spam	1	<b>0</b>	
not	0	1	

	w4	spam	not
spam	1	<b>0</b>	
not	0	1	

	w5	spam	not
spam	<b>0.5</b>	0.5	
not	1	0	

Recompute labels using (weighted) majority vote

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

	spam	not
email1	1.5	4.34
email2		
email3		
email4		
email5		

106

	w1	spam	not
spam	1	0	
not	<b>0.67</b>	0.33	

	w2	spam	not
spam	1	0	
not	0.33	<b>0.67</b>	

	w3	spam	not
spam	1	0	
not	0	<b>1</b>	

	w4	spam	not
spam	1	0	
not	0	<b>1</b>	

	w5	spam	not
spam	0.5	0.5	
not	<b>1</b>	0	

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

Renormalize

	spam	not
email1	0.26	0.74
email2	0.69	0.31
email3	0.29	0.71
email4	0.82	0.18
email5	0.26	0.74

w1	spam	not
spam	1	0
not	0.67	0.33

w2	spam	not
spam	1	0
not	0.33	0.67

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	0.5	0.5
not	1	0

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

Iterate until convergence!

	spam	not
email1	0.26	<b>0.74</b>
email2	<b>0.69</b>	0.31
email3	0.29	<b>0.71</b>
email4	<b>0.82</b>	0.18
email5	0.26	<b>0.74</b>

w1	spam	not
spam	1	
not		

w2	spam	not
spam	1	0
not	0.33	0.67

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	0.5	0.5
not	1	0

	w1	w2	w3	w4	w5
email1	spam	not	not	not	spam
email2	spam	spam	spam	spam	spam
email3	not	spam	not	not	spam
email4	spam	spam	spam	spam	not
email5	spam	not	not	not	spam

(This example converges after 1 iteration)

	spam	not
email1	0.26	0.74
email2	0.69	0.31
email3	0.29	0.71
email4	0.82	0.18
email5	0.26	0.74

w1	spam	not
spam	1	0
not	0.67	0.33

w2	spam	not
spam	1	0
not	0.33	0.67

w3	spam	not
spam	1	0
not	0	1

w4	spam	not
spam	1	0
not	0	1

w5	spam	not
spam	0.5	0.5
not	1	0

```
iter == max_iter or  
change == min_diff
```