Expectation-Maximization (and related learning methods)

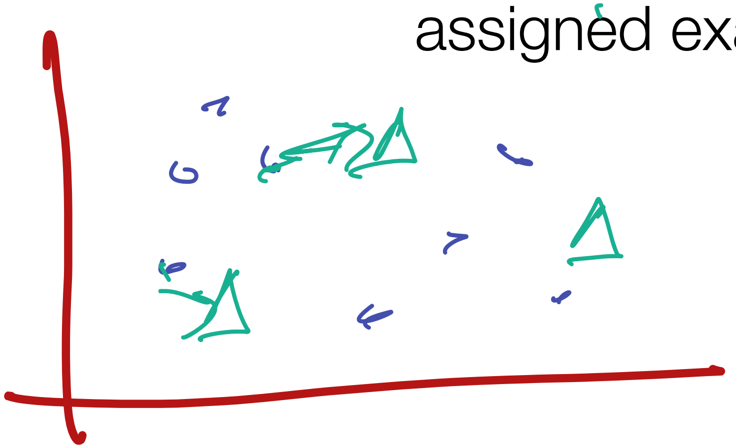
Lecture 18
CS 689, Spring 2023

Brendan O'Connor
College of Information and Computer Sciences
University of Massachusetts Amherst

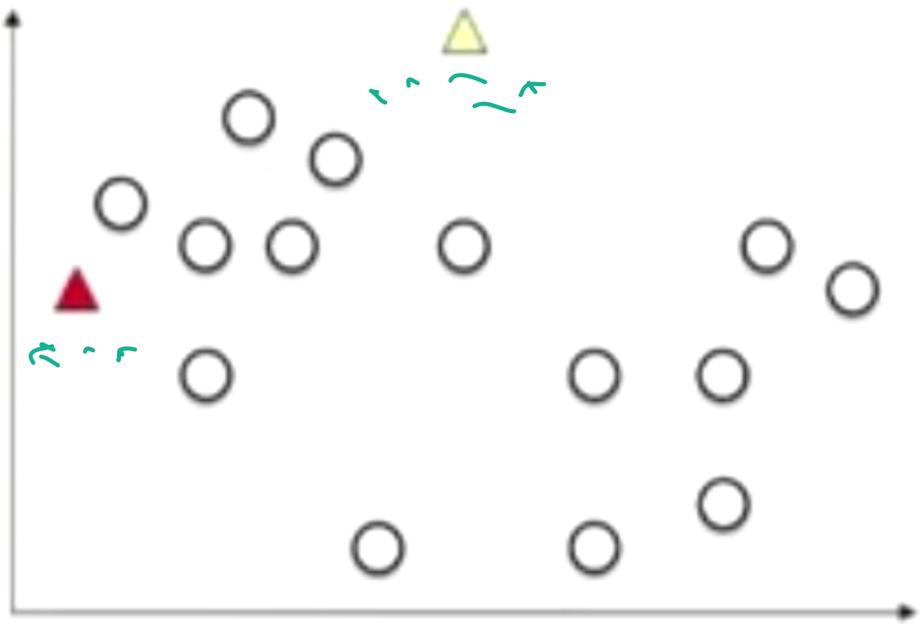
Clustering with (hard) EM

- K-Means is the most basic example of a (basically) probabilistic unsupervised learning algorithm
  - 1. Randomly initialize cluster centroids
  - 2. Alternate until convergence:
    - ("E"): Assign each example to closest centroid
    - ("M"): Update centroids to means of these newly assigned examples

Coord descent on Obj

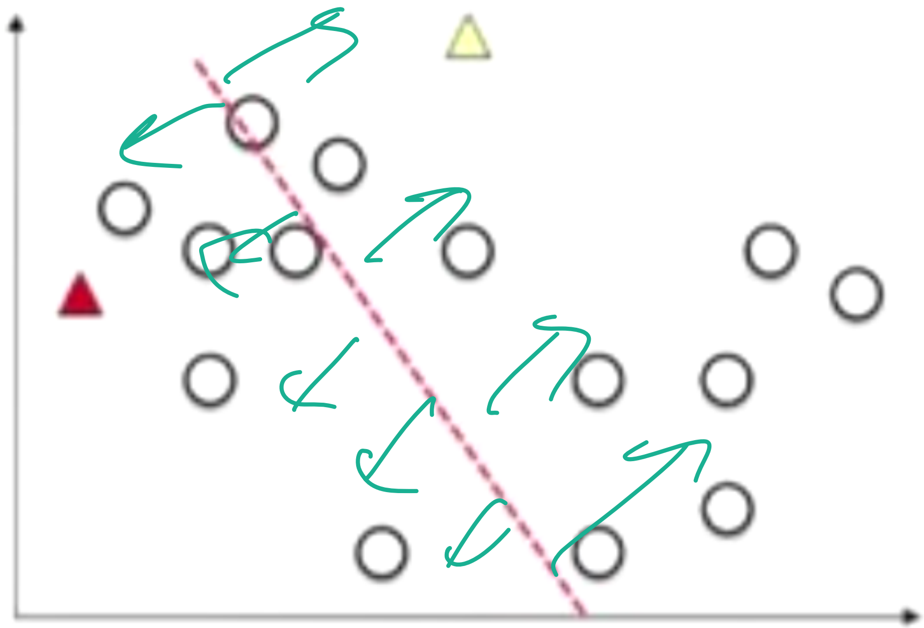


K-means clustering example



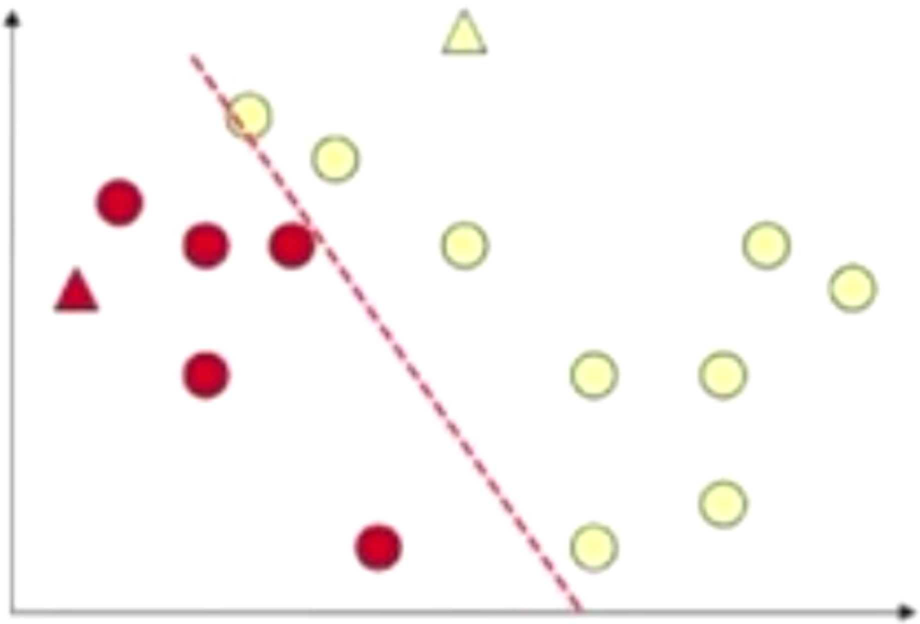
Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

K-means clustering example



Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

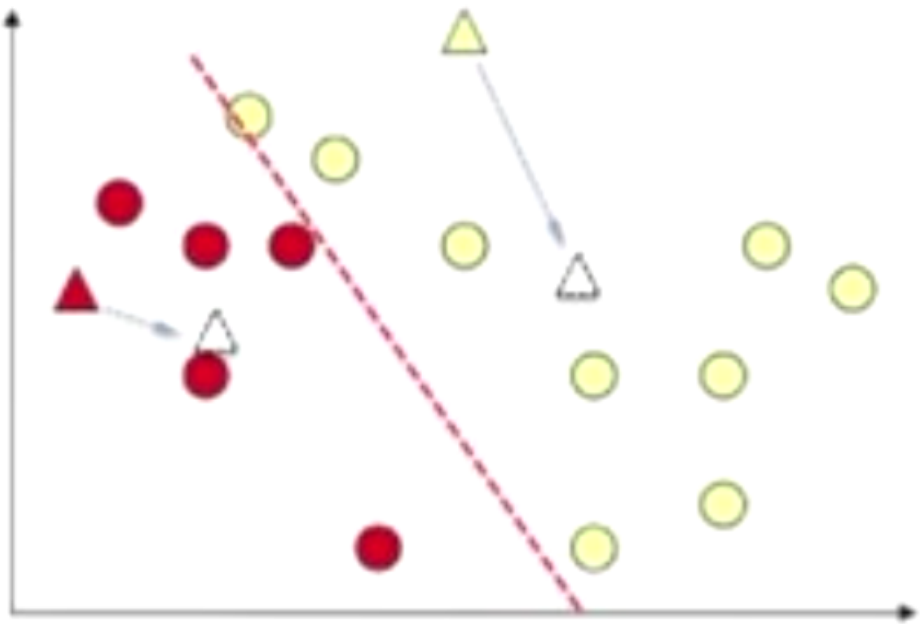
K-means clustering example



Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

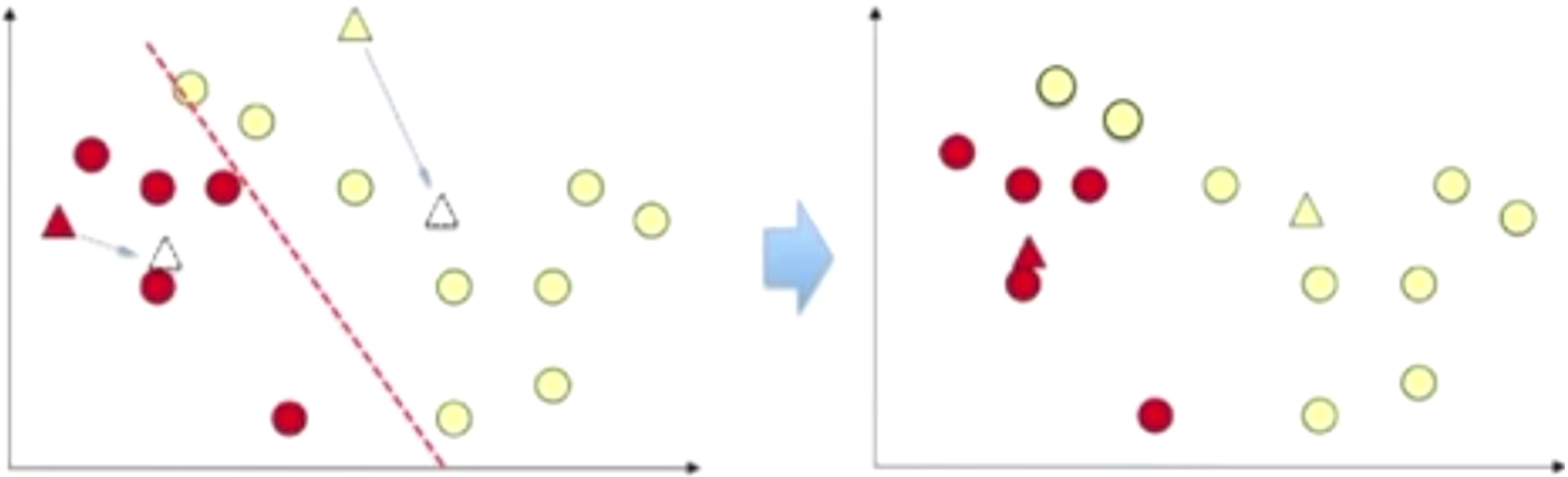
E-step: infer labels
M-step: update centroids

K-means clustering example



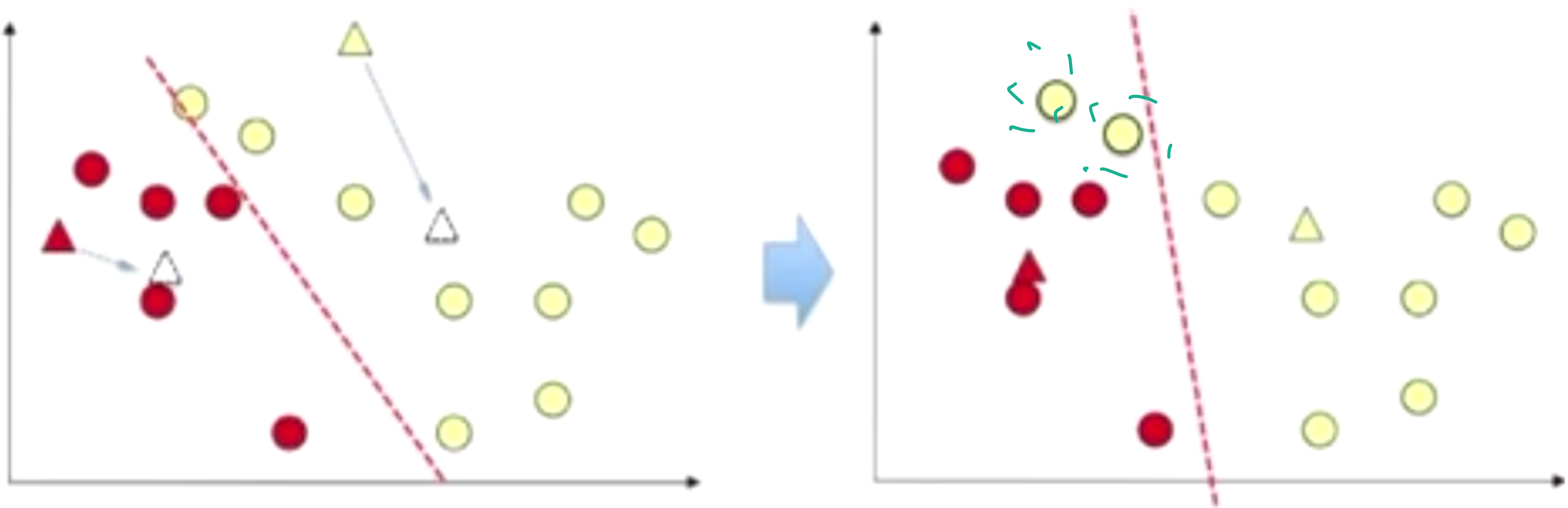
Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

K-means clustering example



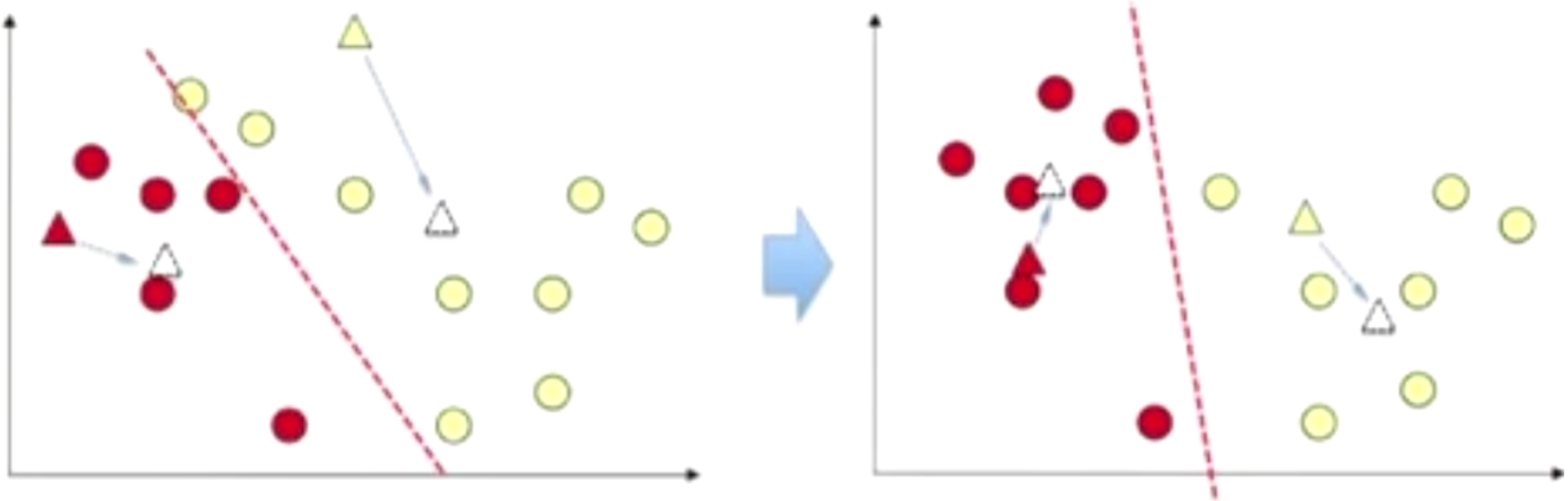
Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

K-means clustering example



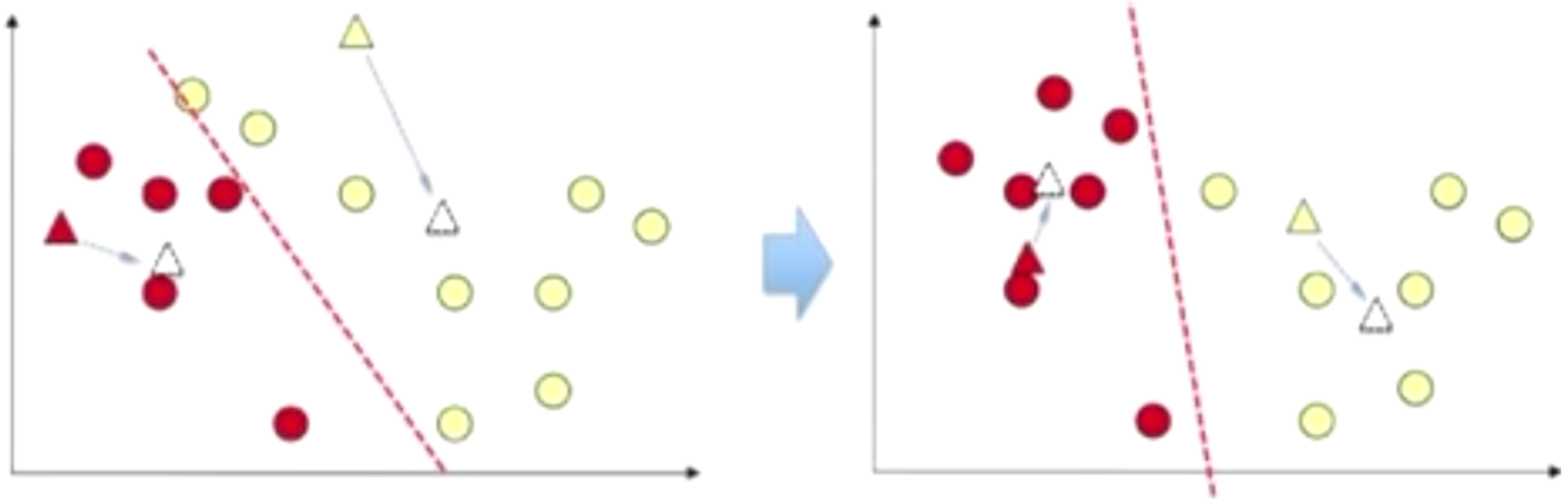
Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

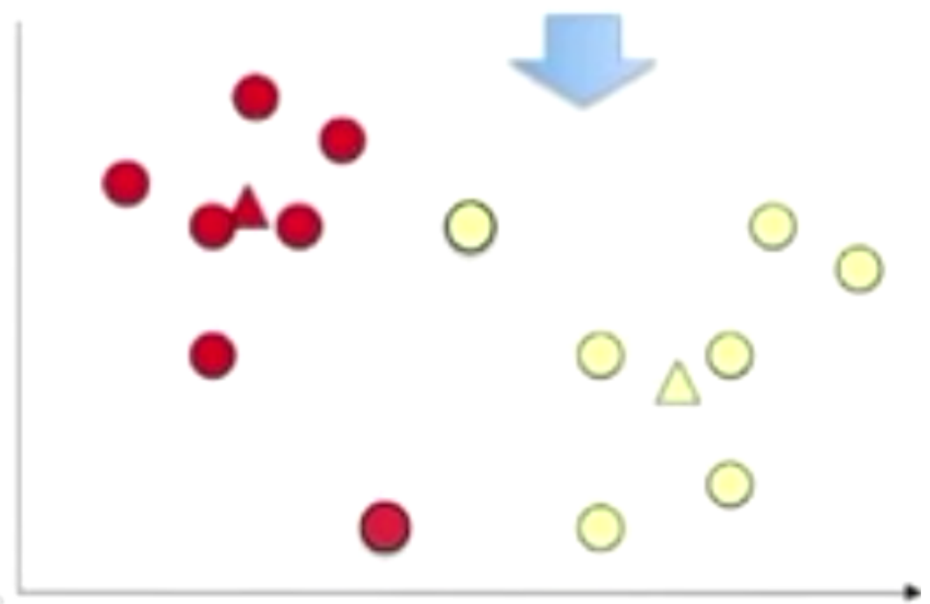
K-means clustering example



Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

K-means clustering example





Slides from UMass alum Victor Lavrenko, U. Edinburgh: https://www.youtube.com/watch?v=_aWzGGNrcic

K-Means as Gaussian Mixture

$x_i \in \mathbb{R}^d$

- Observed data $\boxed{\mathbf{x_1..x_n}}$
- Latent variables: cluster labels $\boxed{\mathbf{z_1..z_n}}$ discrete $z_i \in \{1...K\}$
- Parameters: Gaussian centroids $\underline{\boldsymbol{\mu_1..\mu_K}}$ $\mu_k \in \mathbb{R}^d$
- Assume Gaussian mixture model
  $p(x_i \mid z_i) \sim N(\mu_{z_i}, \text{var}(\sigma))$ constant μ_{z_i}
- Learning algorithm: alternate until convergence,
  - ("E"): Assign each example to closest centroid
    => z_i := argmax$_k$ $P(z_i=k \mid x, \mu_k)$
  - ("M"): Update centroids to averages
    => μ_k := argmax$_m$ $P(x \mid z, \mu=m)$
    = $(1/n_k) \sum_{i:z_i=k} x_i$

"Hard EM"

$P(z \mid x, \mu)$

$P(z) = \text{uniform}$

$P(z=1 \mid x, \mu) \propto p(x \mid z=1)\,p(z=1)$

$P(z=2 \mid x, \mu) \propto p(x \mid z=2)\,p(z=2)$

- ~~Is there a "Soft" EM (close variant) iteratively optimiz~~
- ~~This is "Hard EM". A very close variant~~
- ~~es P(x,z | μ)~~

Expectation-Maximization

GMM: cluster labels

Local

Global = component centroids & variances

- Observed **x**, latent **z**, parameters **θ**
- EM is a meta-algorithm for settings where MLE for **θ** is easy, if only you knew **z**

$$\max_{\theta} \; p(x|\theta)$$

$$\max_{\theta} \; p(x|z,\theta)\, p(z|\theta)$$

[for GMM]

Derivation of EM

- (new page)

Latent-variable generative models

Text

$$P(w, z \mid \theta)$$

Parameters

(Sometimes) latent quantity to help explain the language you see

- Document category
- World context
- Grammatical category
- Semantic structure
- Real-valued embedding

Easy stuff

- Supervised learning: $\mathbf{argmax}_\theta\ \mathbf{P(w^{train}, z^{train} \mid \theta)}$
- Prediction (via posterior inference): $\mathbf{P(z \mid w^{input}, \theta)}$

Unsupervised stuff with *marginal inference*

- Latent (unsupervised) learning: $\mathbf{argmax}_\theta\ \mathbf{P(w^{train} \mid \theta)}$
- Language modeling (via marginal inference): $\mathbf{P(w^{input} \mid \theta)}$

EM performance

- Guaranteed to find a locally maximum likelihood solution. Guaranteed to converge.
  - But can take a while
- Dependent on initialization

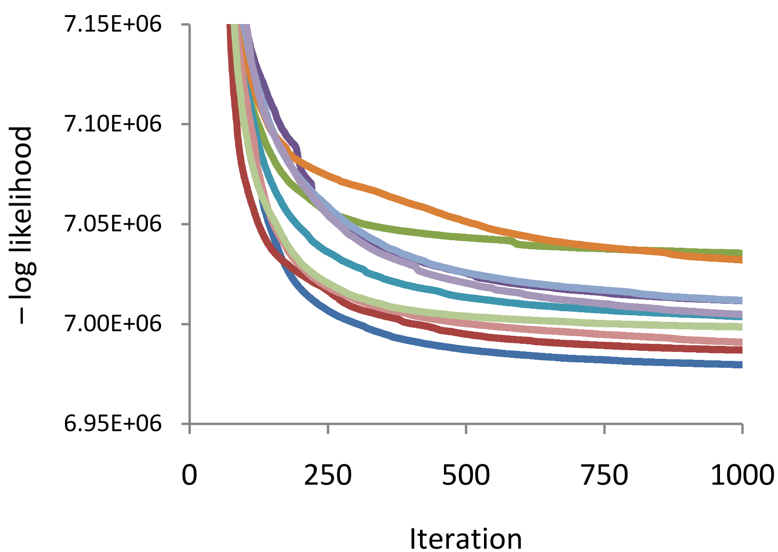


Figure 1: Variation in negative log likelihood with increasing iterations for 10 EM runs from different random starting points.

Johnson 2007, "Why doesn't EM find good HMM POS-taggers?"

EM pros/cons

- Works best for a simple model with rapid E/M-step inference
- Requires probabilistic modeling assumptions
- Dependent on initialization
  - Many alternative methods (e.g. MCMC), but can similar issues with local optima
- EM originally invented for Hidden Markov Models in speech recognition
  - E-step infers structured posteriors
- General issue: Closed form M-steps only available for pretty simple models (Gaussians, count-based multinomials...)

EM versus direct gradients

- What if the M-step requires gradient ascent?
  - Running LBFGS or many iterations of GD inside the M-step can be slow

- Partial/incremental EM variants (Neal and Hinton, 1998): Why not just 1 gradient step? Gradient step on only a few examples?
- Or... consider the direct gradient. We can interpret it as an EM-like method itself.