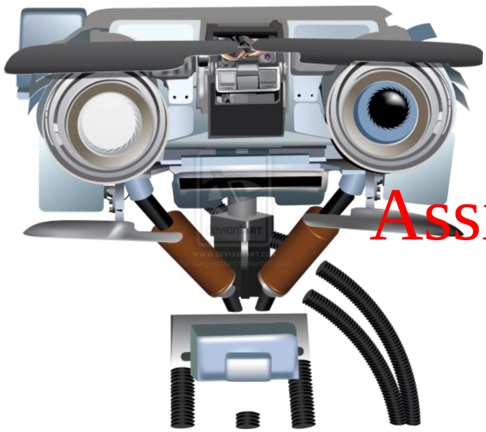
Announcements

Reminder: ps2 due Thursday at midnight (Boston)

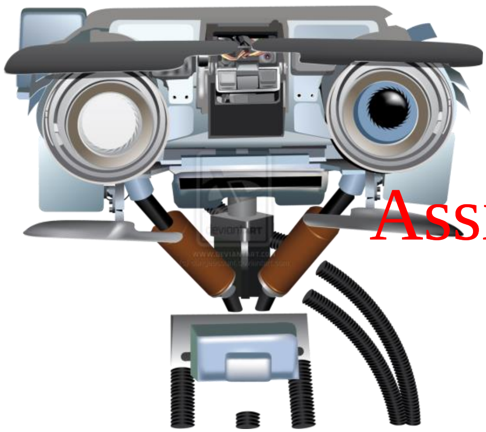
- Self-Grading form for ps1 out Friday 9/25 (1 week to turn in)
- Self-Grading form for ps2 out Monday 9/28 (1 week to turn in)
- Lab this week (no more rotations) – Linear/Logistic Regression, Anaconda



Unsupervised Learning I

Today

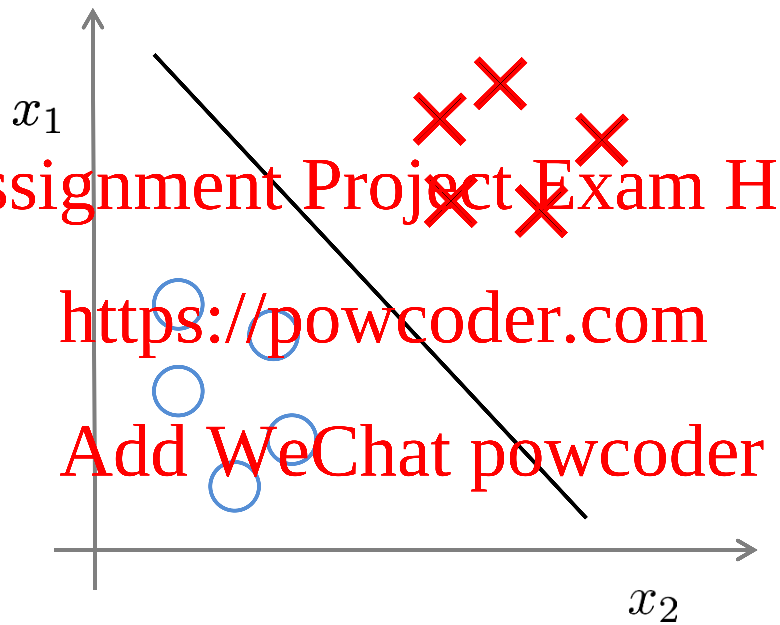
- Unsupervised learning
  - K-Means clustering
  - Gaussian Mixture clustering



Unsupervised Learning I

Clustering

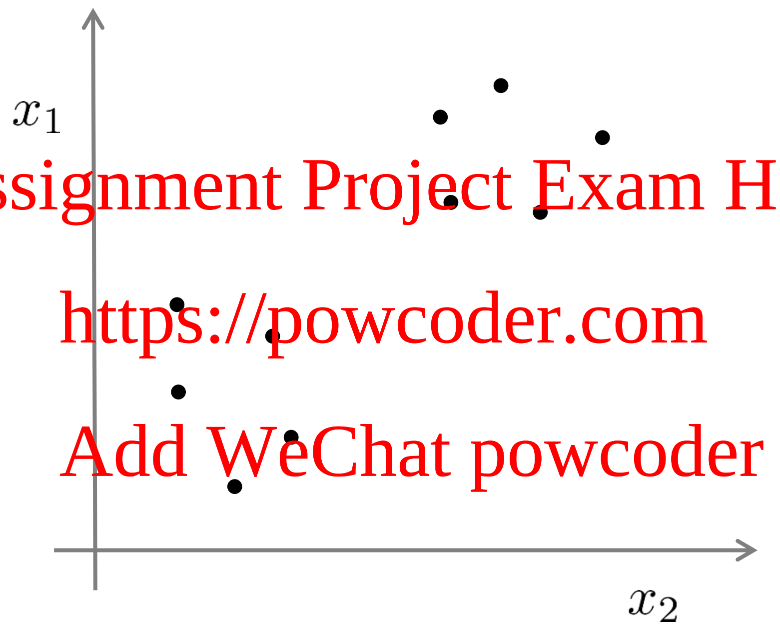
Supervised learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

slide credit: Andrew Ng

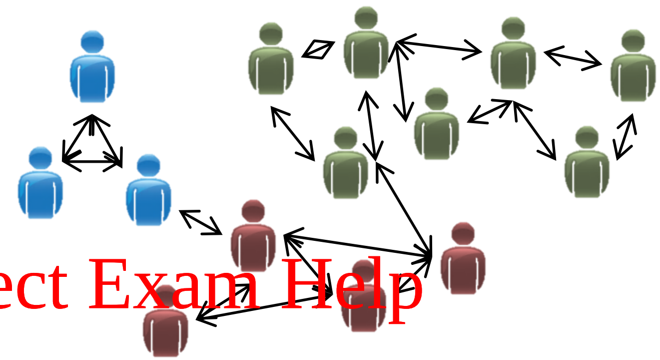
Unsupervised learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

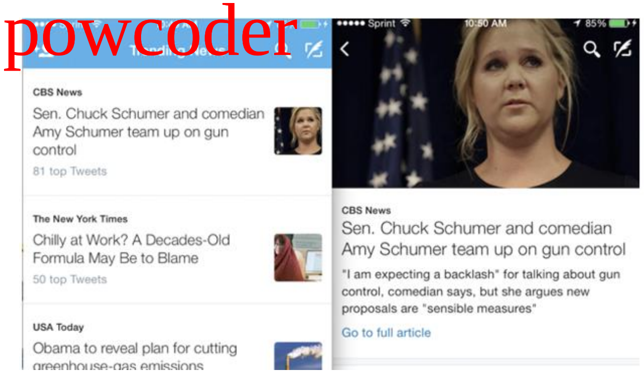
Clustering

Gene analysis

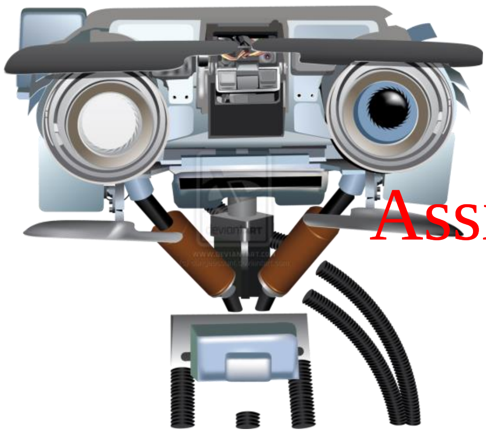


Social network analysis

Types of voters

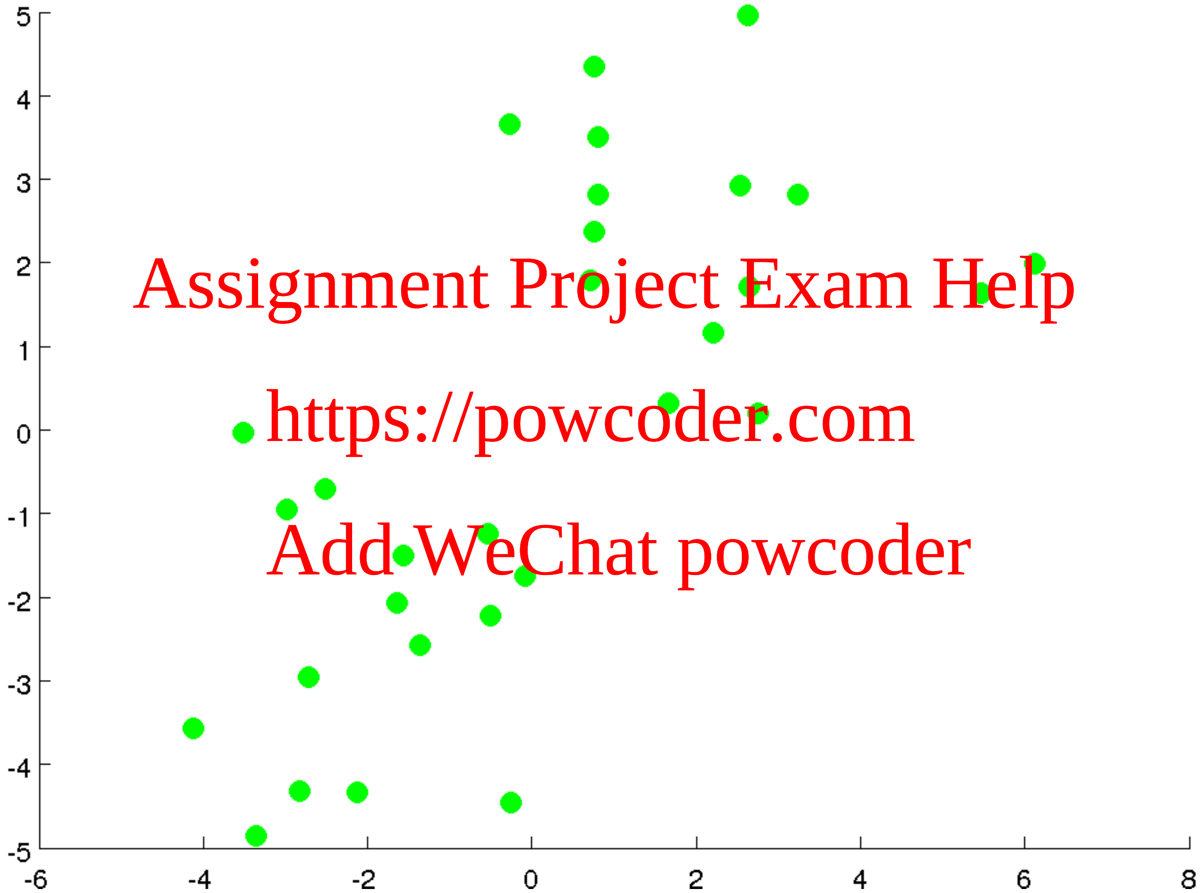


Trending news

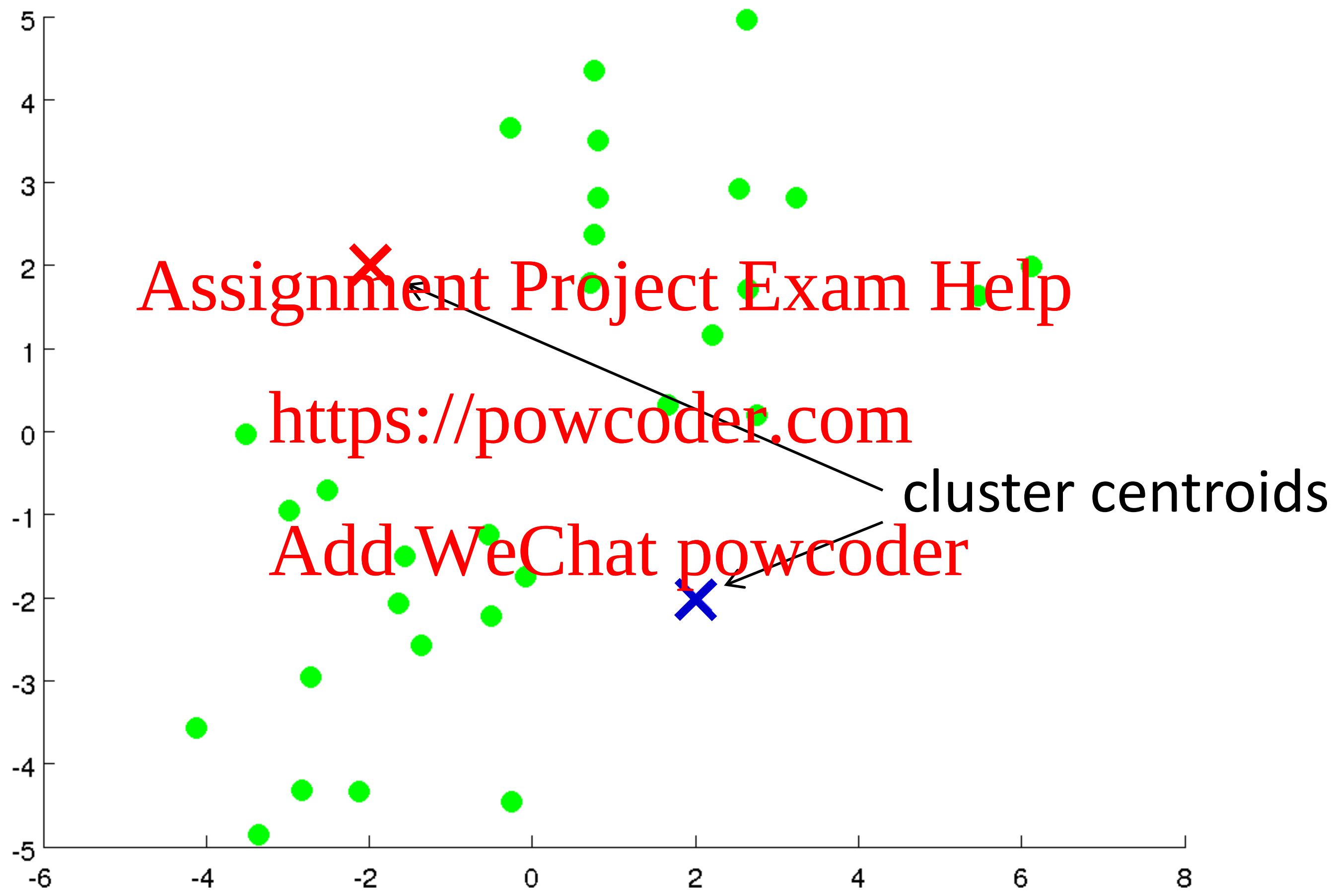


Unsupervised Learning I

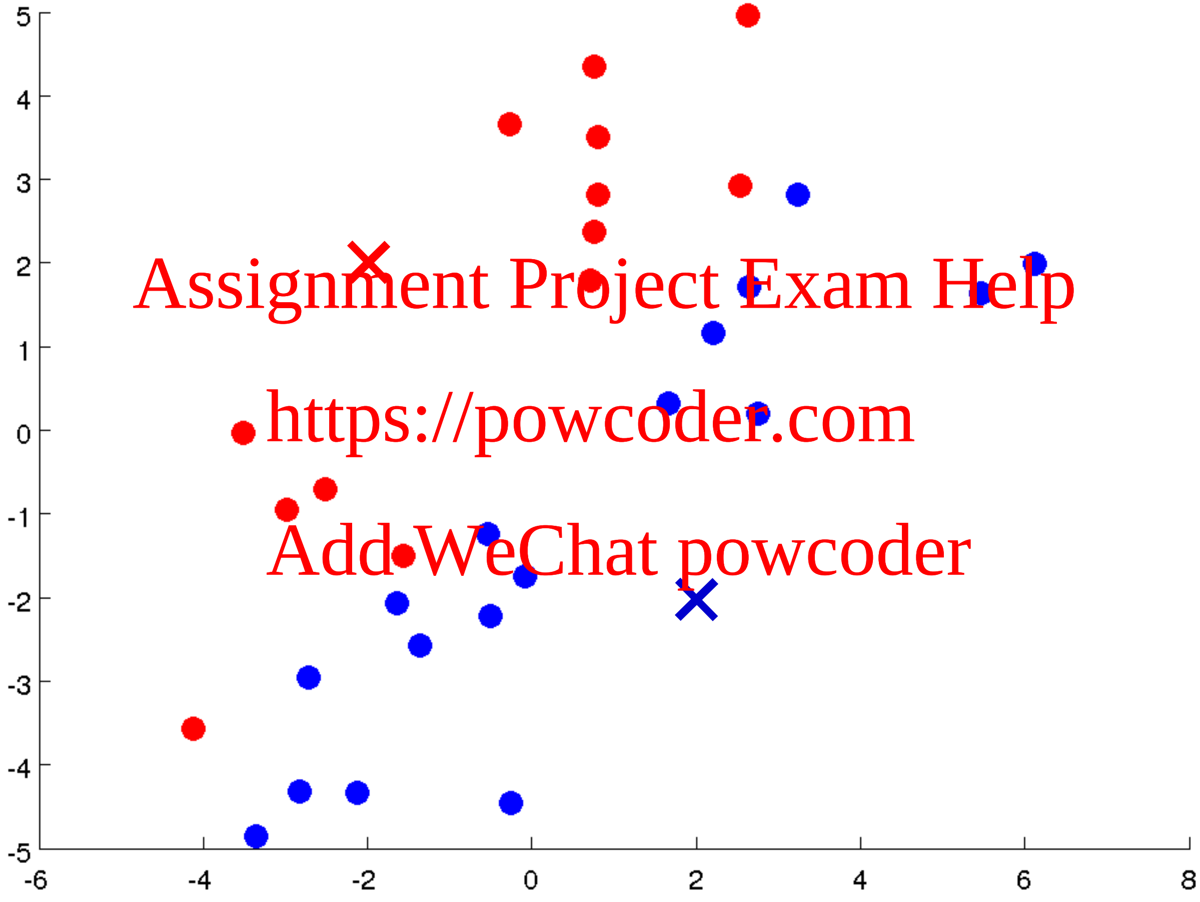
K-means Algorithm



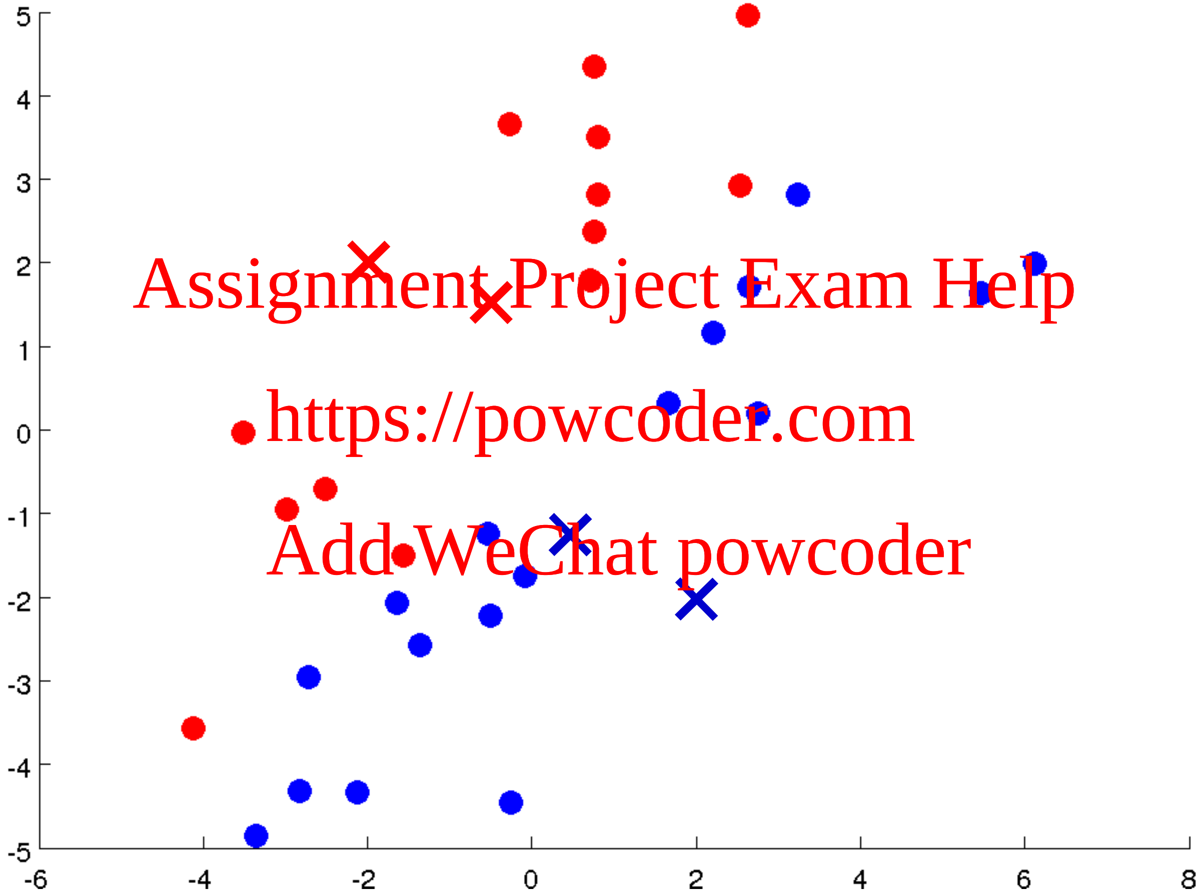
slide credit: Andrew Ng



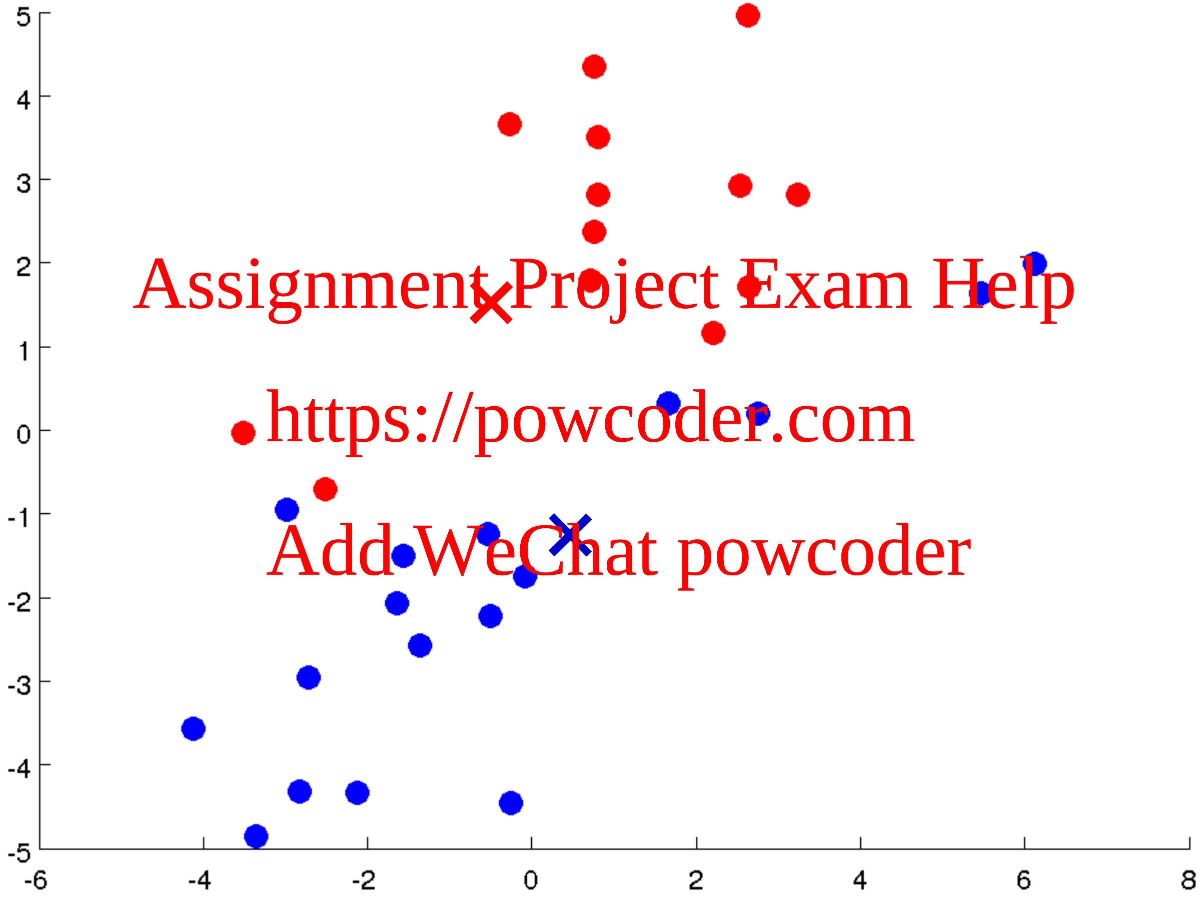
slide credit: Andrew Ng



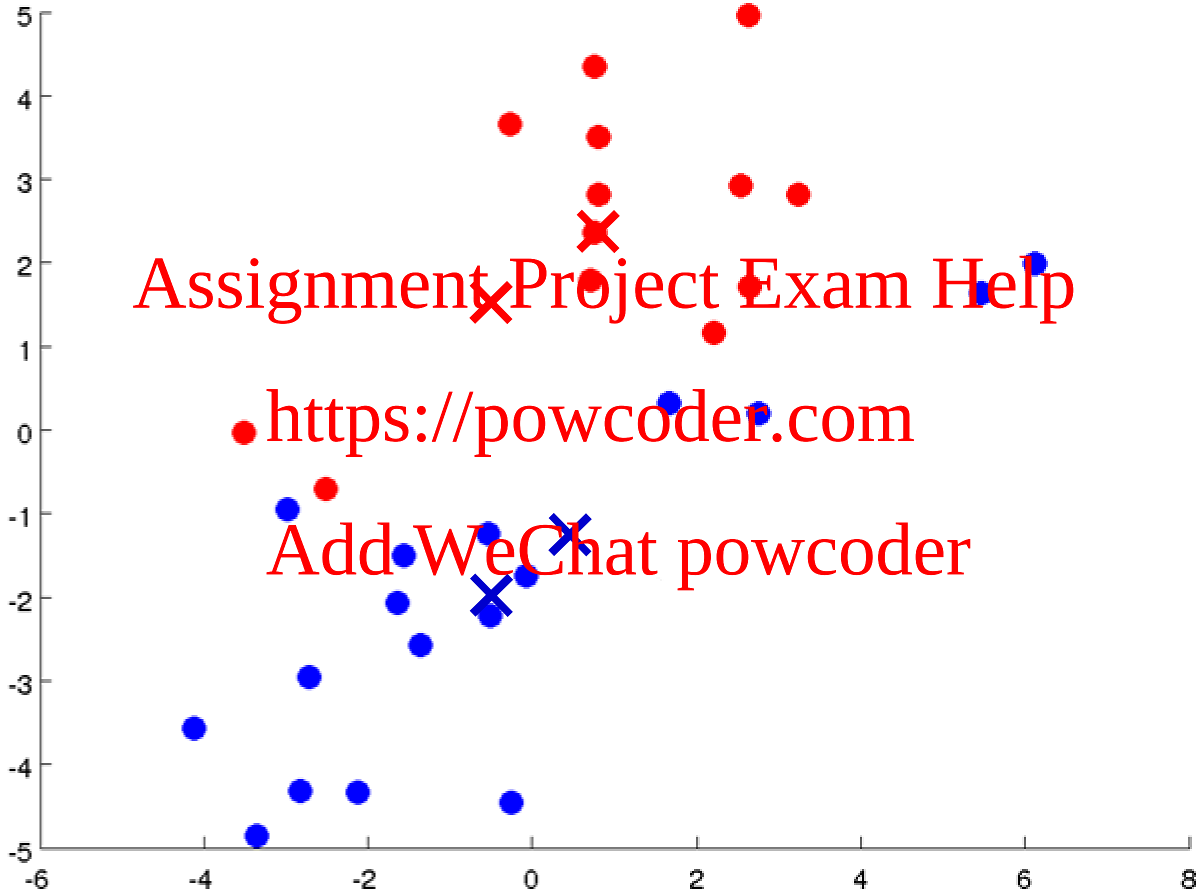
slide credit: Andrew Ng



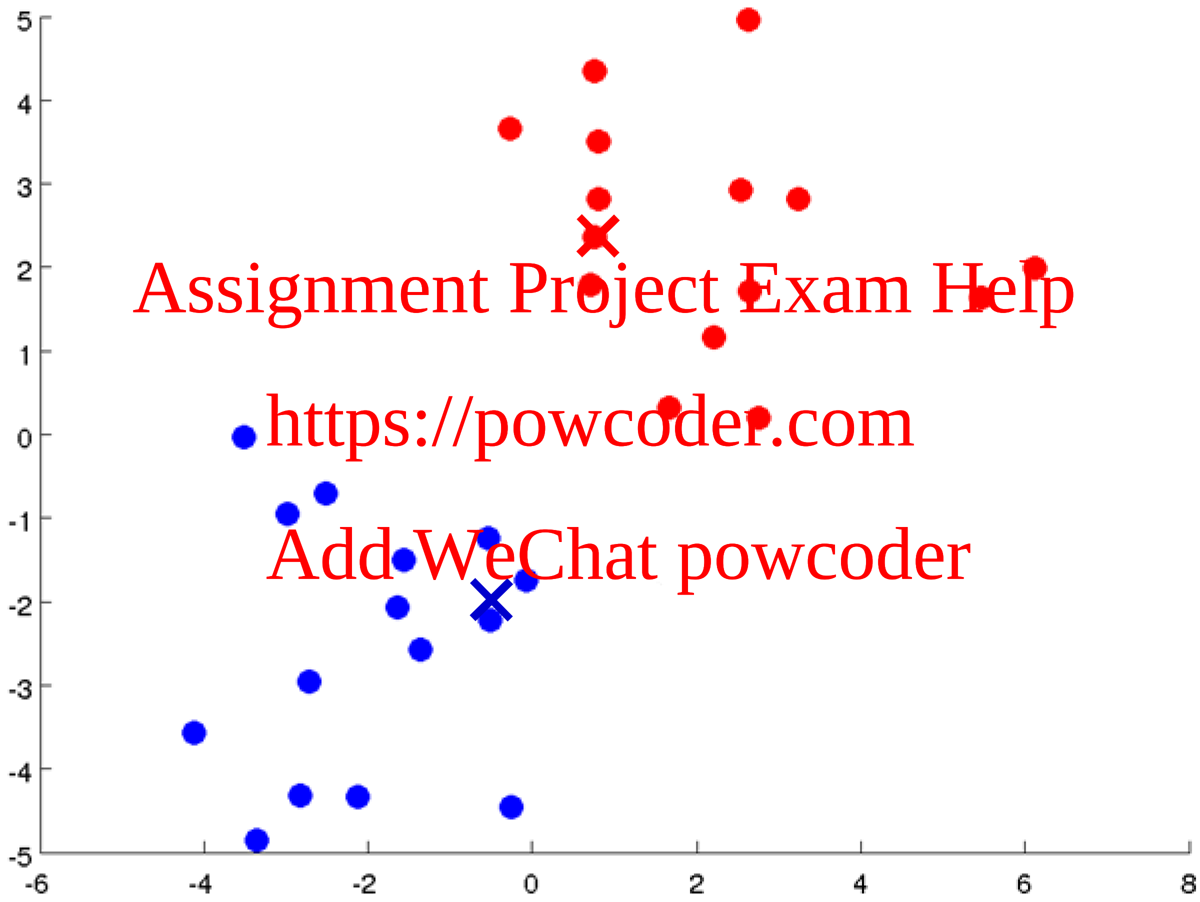
slide credit: Andrew Ng



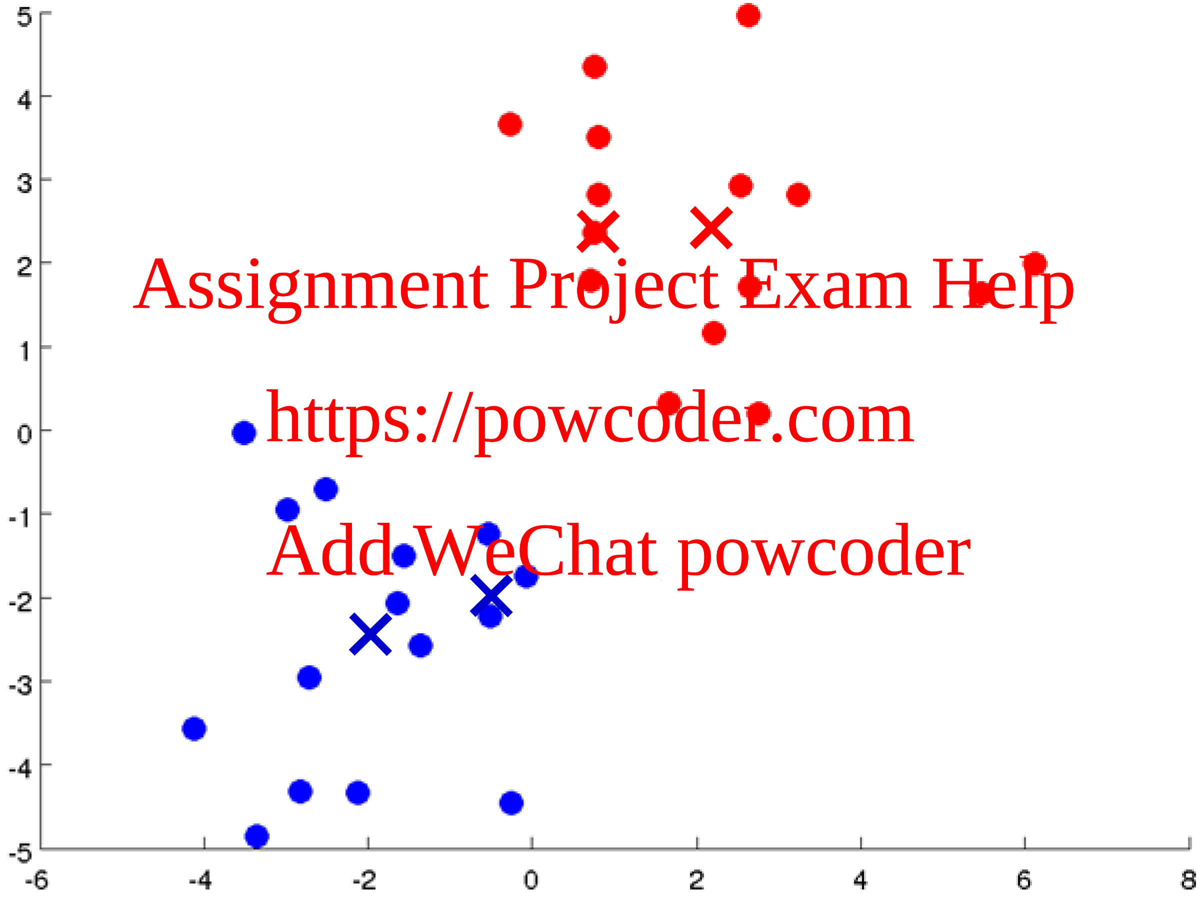
slide credit: Andrew Ng



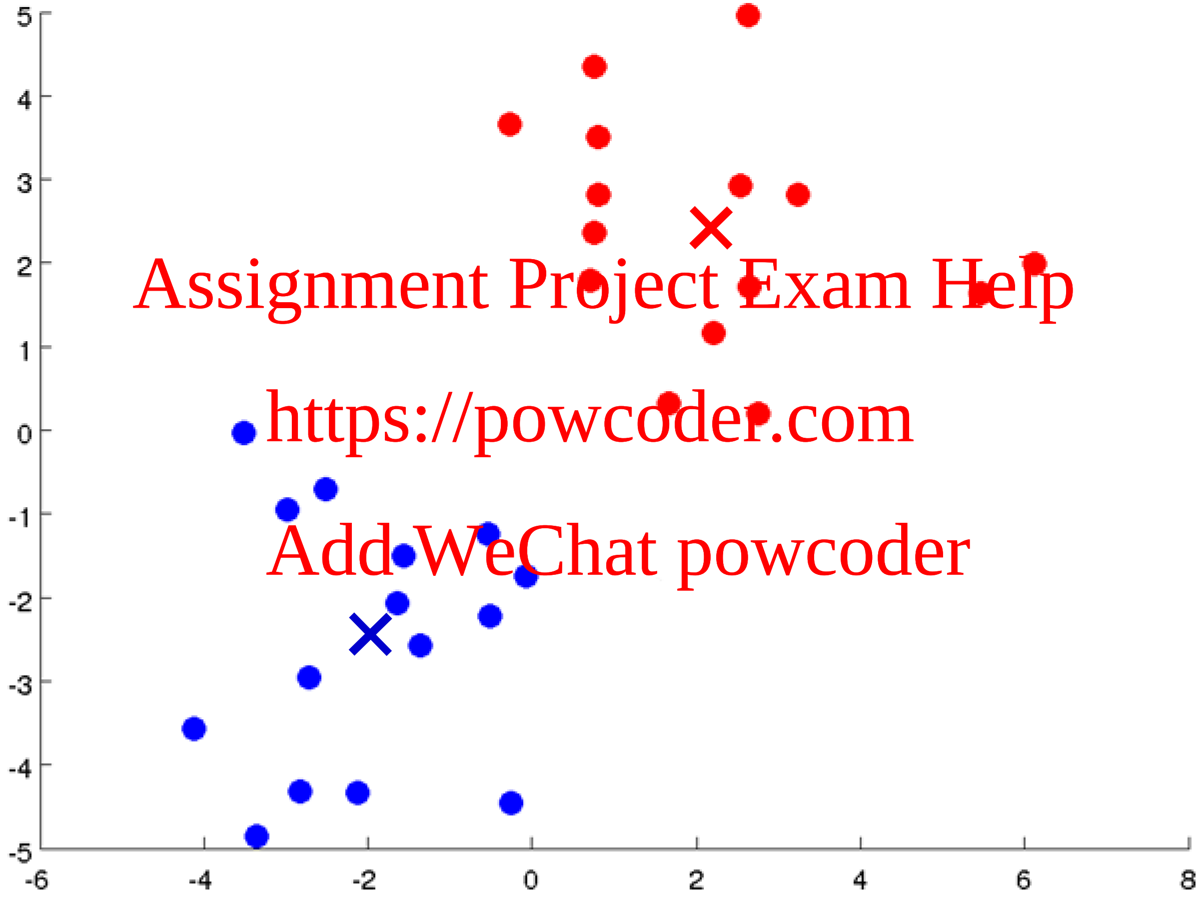
slide credit: Andrew Ng



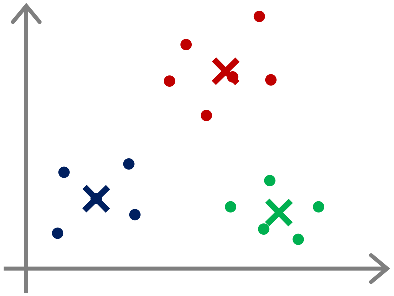
slide credit: Andrew Ng



slide credit: Andrew Ng



slide credit: Andrew Ng



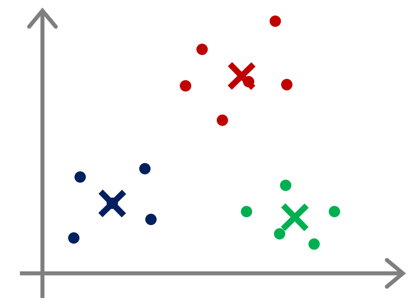
K-means algorithm

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention)

K-means algorithm



Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat {

for i = 1 to m

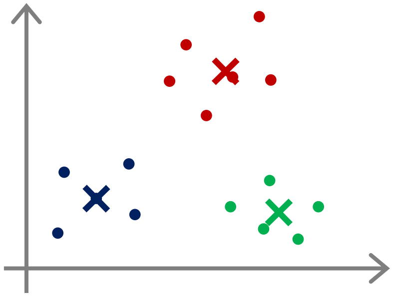
$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

slide credit: Andrew Ng



K-means Cost Function

$c^{(i)}$ = index of cluster (1,2,…,K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

Optimization cost: “distortion”

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

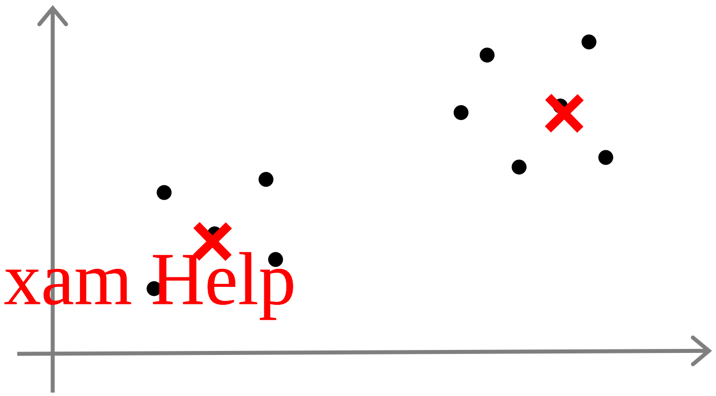
$$\min_{\substack{c^{(1)}, \ldots, c^{(m)}, \\ \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

slide credit: Andrew Ng

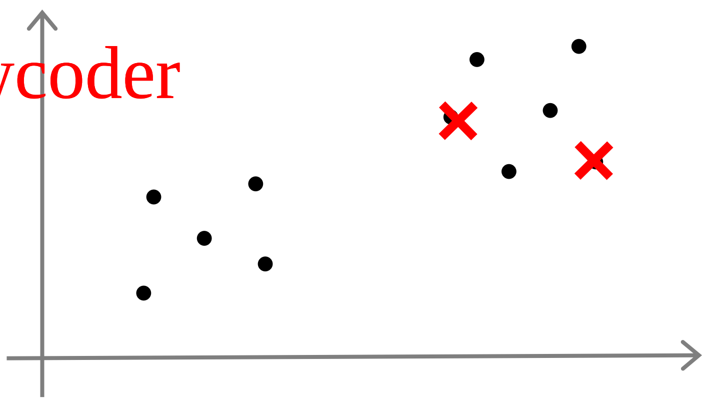
Random initialization

Should have $K < m$

Randomly pick K training examples.

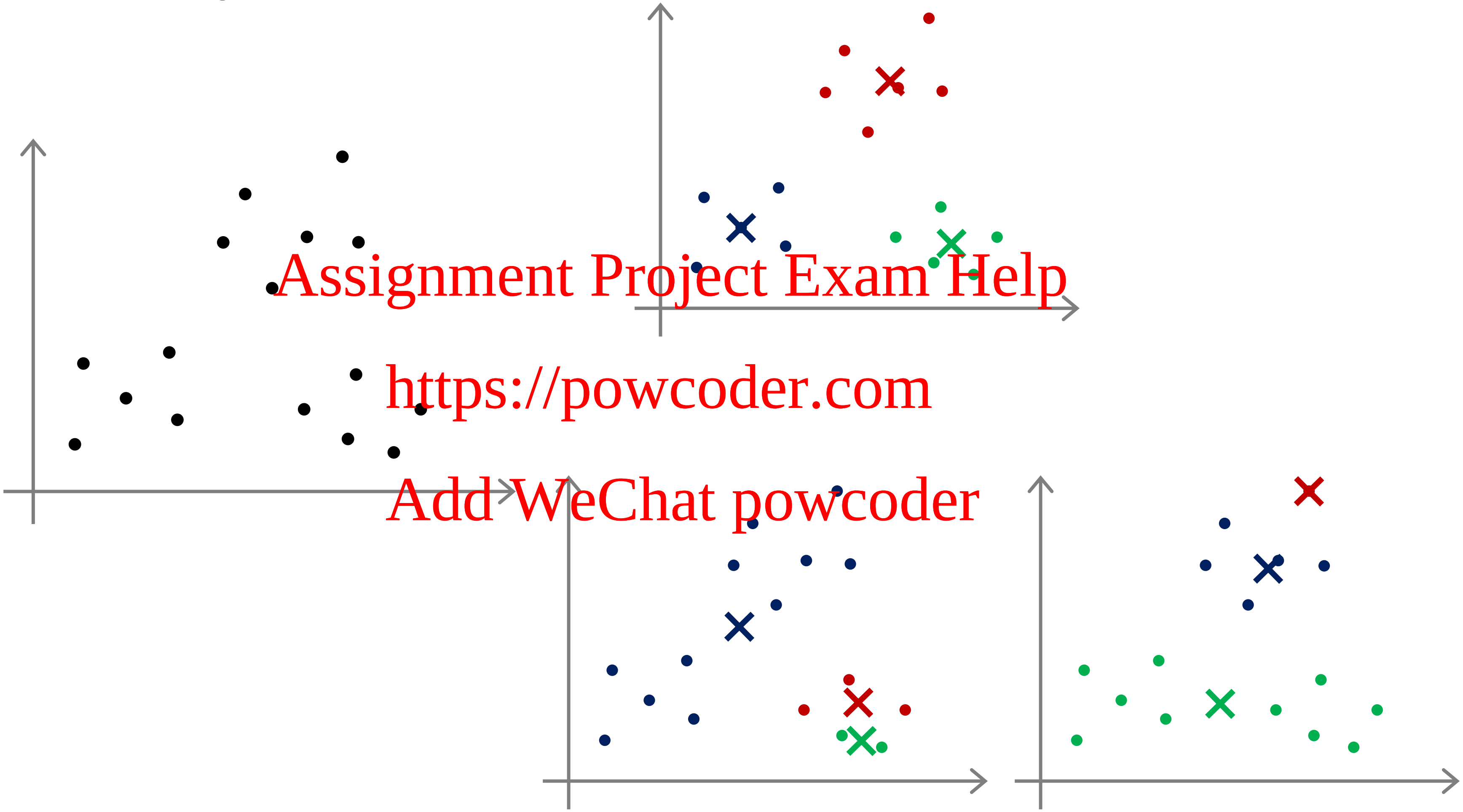


Set $\mu_1, \ldots, \mu_K$ equal to these K examples.



slide credit: Andrew Ng

Local Optima



slide credit: Andrew Ng

Avoiding Local Optima with Random Initialization

For i = 1 to 100 {

Randomly initialize K-means.

Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.

Compute cost function (distortion)

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

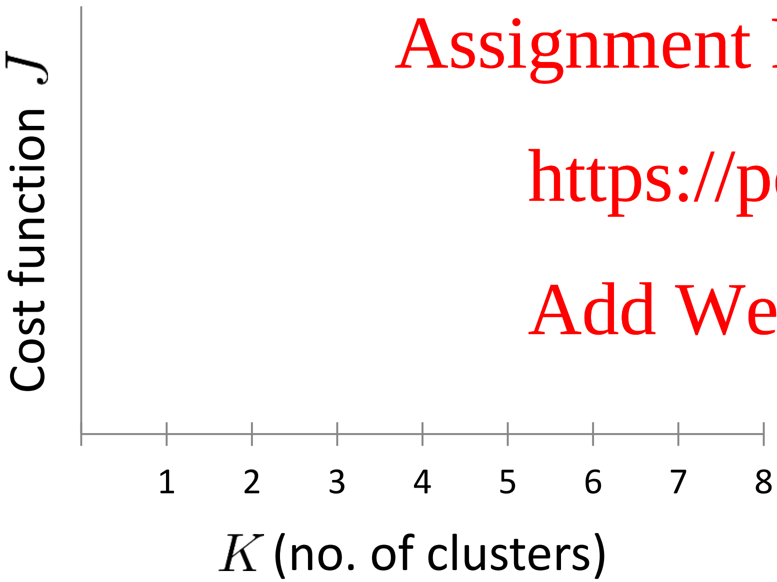
}

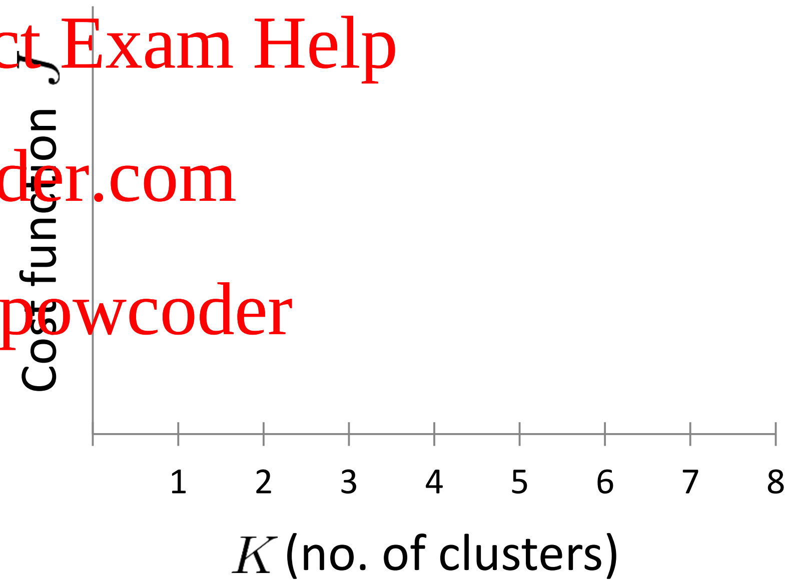
Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

slide credit: Andrew Ng

How to choose K?

Elbow method:



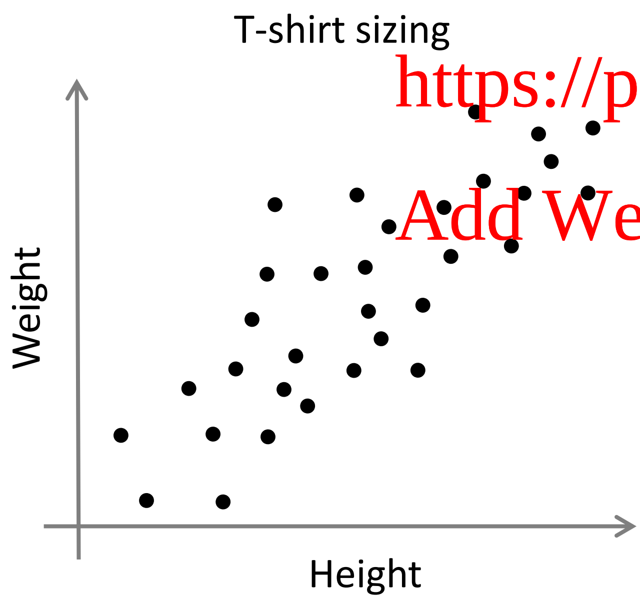


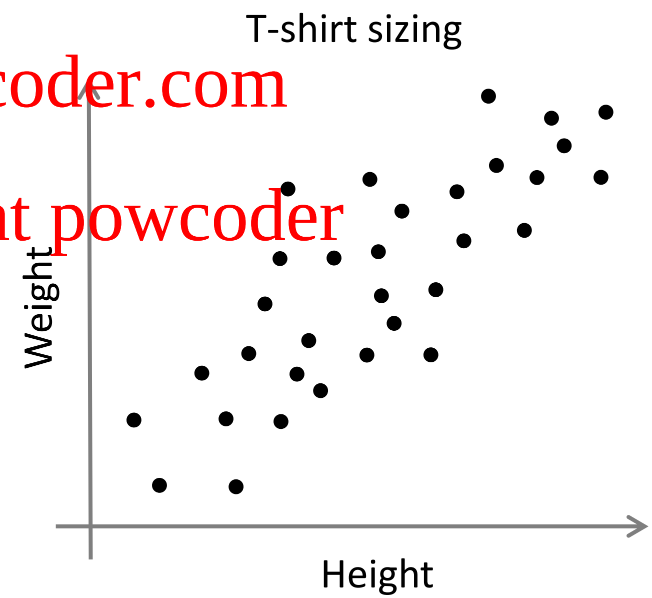
slide credit: Andrew Ng

How to choose K?

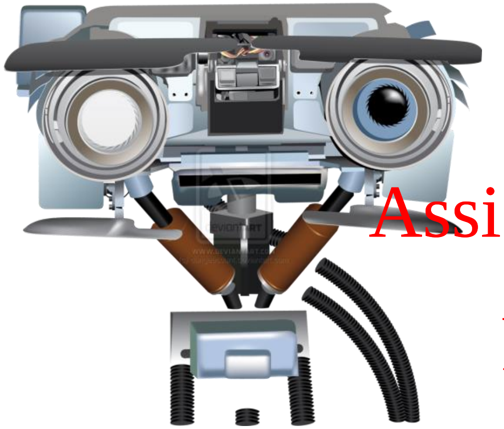
Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

E.g.





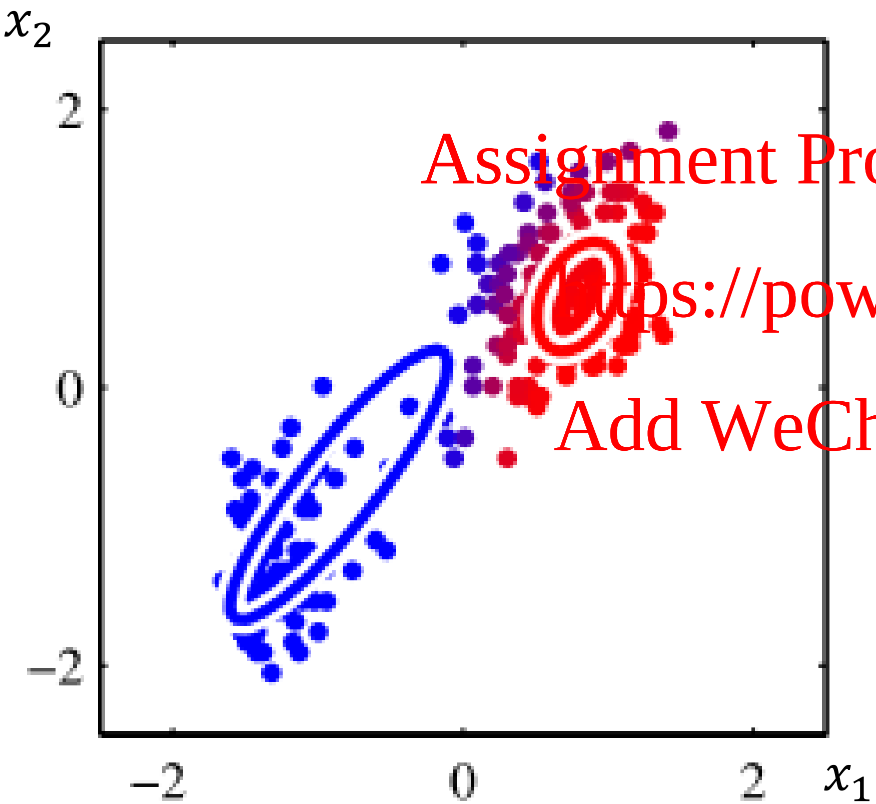
slide credit: Andrew Ng



Unsupervised Learning I

Mixtures of Gaussians

Mixtures of Gaussians: Intuition



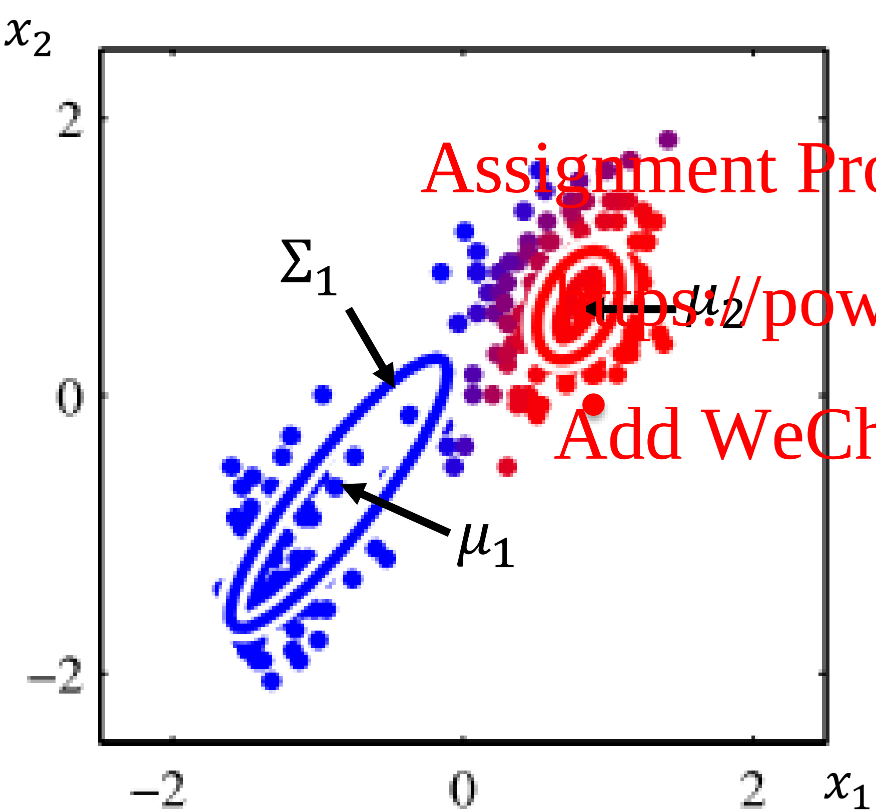
"Soft" cluster membership

Define a distribution over x :

To generate each point x,

- Choose its cluster component z
- Sample x from the Gaussian distribution for that component

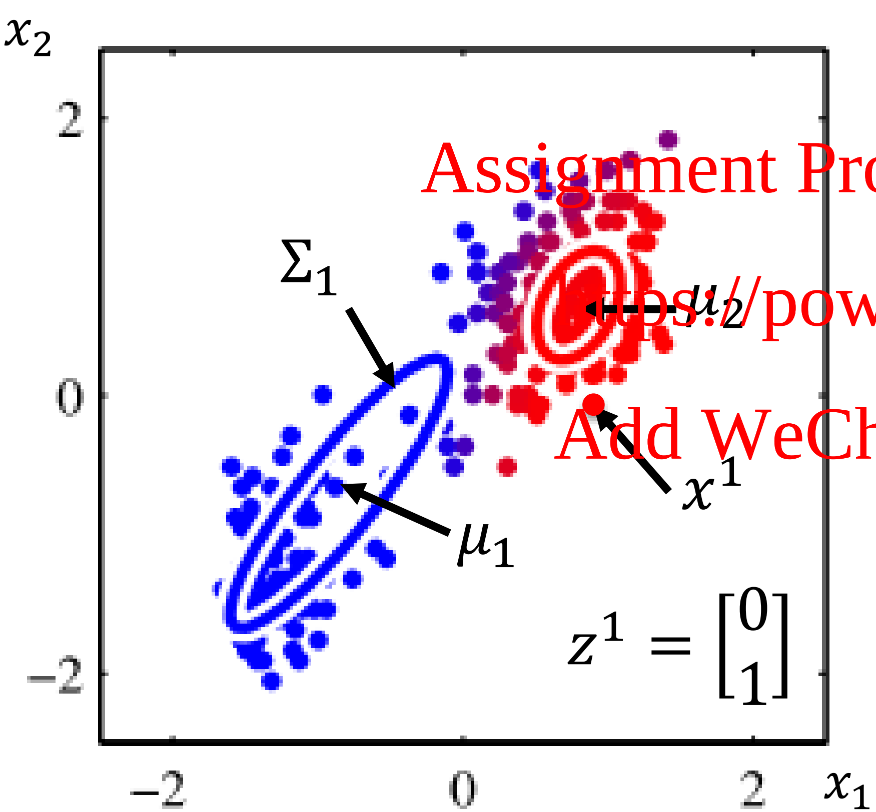
Mixtures of Gaussians:
component membership variable z



- Assume K components, k-th component is a Gaussian with parameters μ_k, Σ_k
- Introduce discrete r.v. $z \in R^K$ that denotes the component that generates the point
- one element of z is equal to 1 and others are 0, i.e. "one-hot":

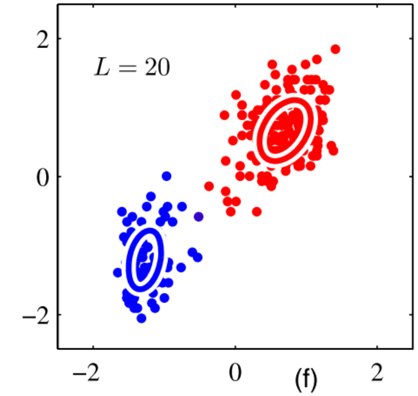
$$z_k \in \{0,1\} \text{ and } \sum_k z_k = 1$$

Mixtures of Gaussians:
Data generation example



- Suppose $K = 2$ components, k-th component is a Gaussian with parameters μ_k, Σ_k
- To sample i-th data point:
  - Pick component z^i with $\mathrm{p}(z_k = 1) = \pi_k$ (parameter)
  - for example, $\pi_k = 0.5$, and we picked $z^1 = [0, 1]^T$
  - Pick data point x^i with probability $\mathrm{N}(\mathrm{x}; \mu_k, \Sigma_k)$

Mixtures of Gaussians



- $z_k \in \{0,1\}$ and $\sum_k z_k = 1$
- K components, k-th component is a Gaussian with parameters μ_k, Σ_k
- define the joint distribution $p(\mathbf{x}, \mathbf{z})$ in terms of a marginal distribution $p(\mathbf{z})$ and a conditional distribution $p(\mathbf{x}|\mathbf{z})$

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- where

$$p(z_k = 1) = \pi_k \qquad 0 \leqslant \pi_k \leqslant 1 \qquad \sum_{k=1}^{K} \pi_k = 1$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

Substitute and simplify

Maximum Likelihood Solution for Mixture of Gaussians

- This distribution is known as a Mixture of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- We can estimate parameters using Maximum Likelihood, i.e. maximize

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) =$$

$$\ln p(x^1, x^2, \dots, x^N | \pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K)$$

- This algorithm is called Expectation Maximization (EM)
- Very similar to soft version of K-Means!

Expectation Maximization

- We can estimate parameters using Maximum Likelihood, i.e. minimize neg. log likelihood

$$-\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Problem: don't know values of *"hidden"* (or "*latent*") variable z, we don't observe it

- Solution: treat z^i as parameters and use coordinate descent

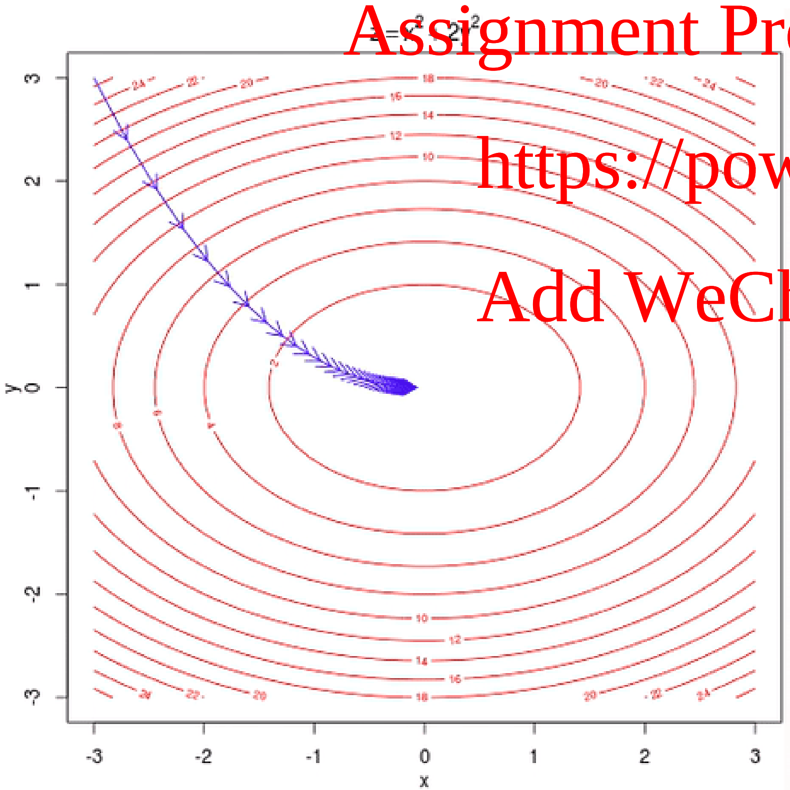
Coordinate Descent

gradient descent:

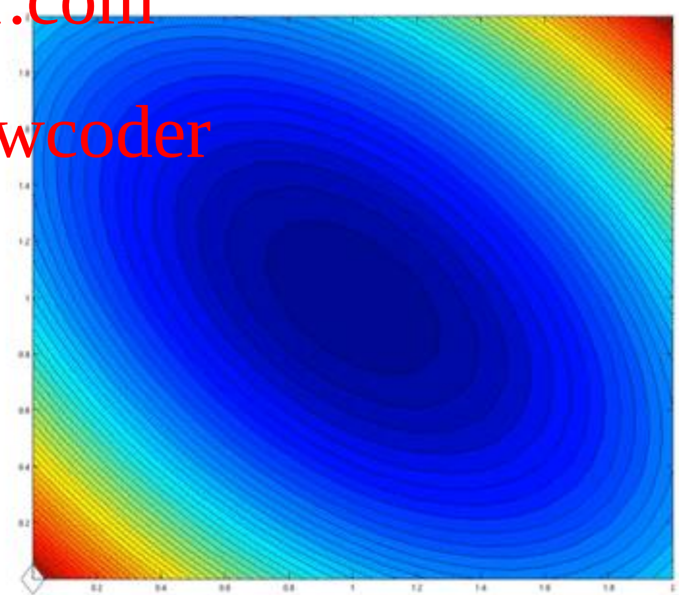
- Minimize w.r.t all parameters at each step

coordinate descent:

- fix some coordinates, minimize w.r.t. the rest
- alternate



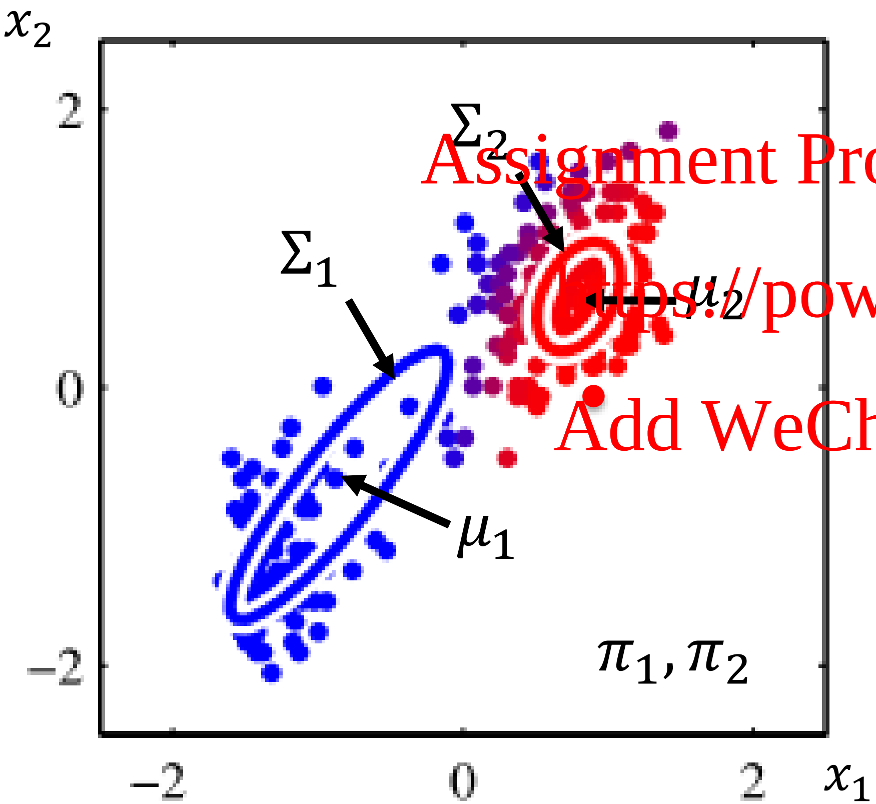
$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



Credit: Martin Takac

Credit: http://vis.supstat.com/2013/03/gradient-descent-algorithm-with-r/

Expectation Maximization



Coordinate descent for Mixtures of Gaussians:

Alternate

- fix $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$, update z^i
- fix z^i, update $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

Expectation Maximization Algorithm

- A general technique for finding maximum likelihood estimators in latent variable models
- Initialize and iterate until convergence:

E-Step: estimate posterior probability of the latent variables $p(z_k|x)$, holding parameters fixed

M-Step: maximize likelihood w.r.t parameters (here $\mu_k, \Sigma_\mathrm{k}, \pi_k$) using latent probabilities from E-step

EM for Gaussian Mixtures Example

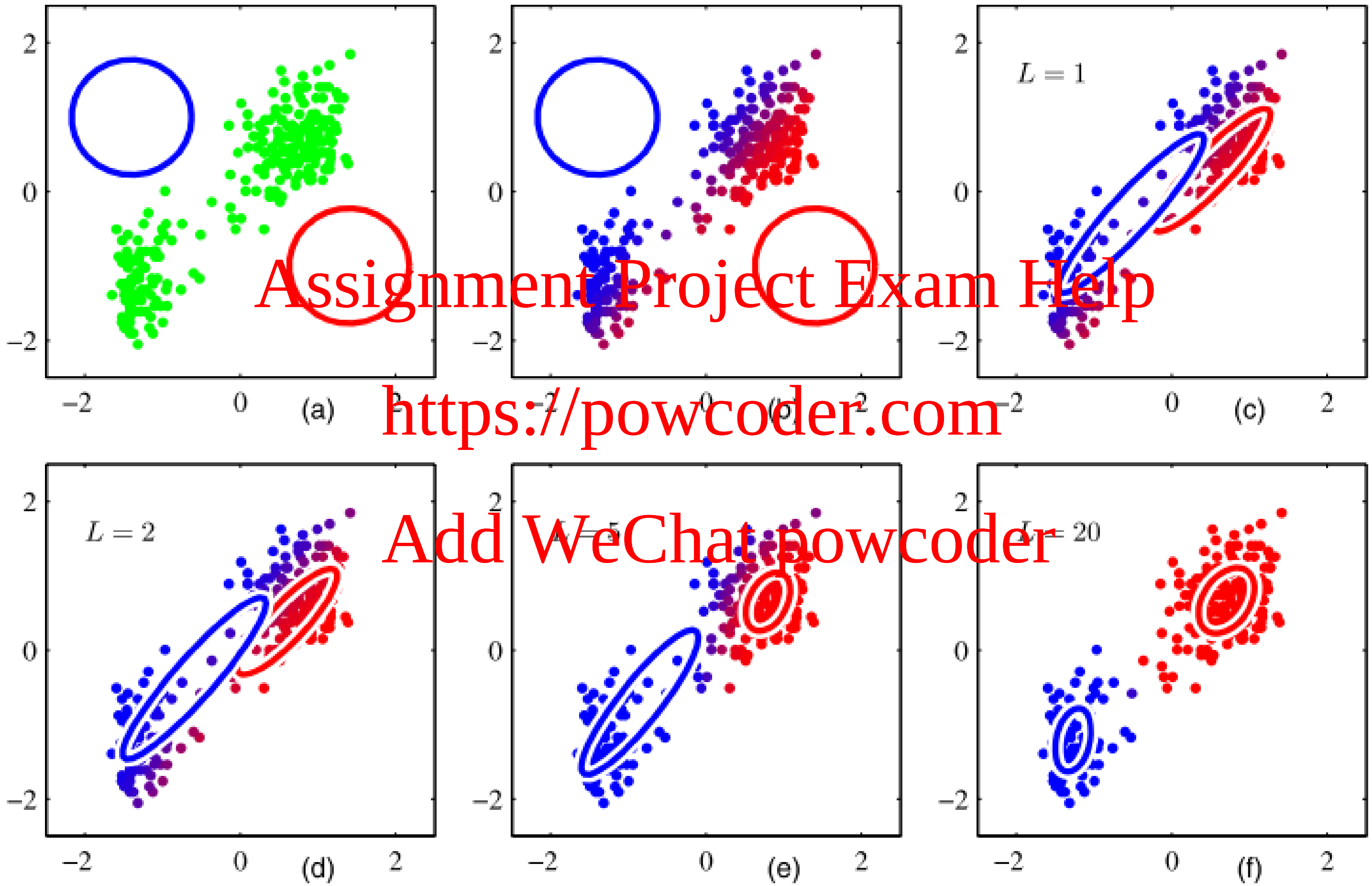


Figure 9.8 Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

EM for Gaussian Mixtures

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{9.23}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \qquad N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

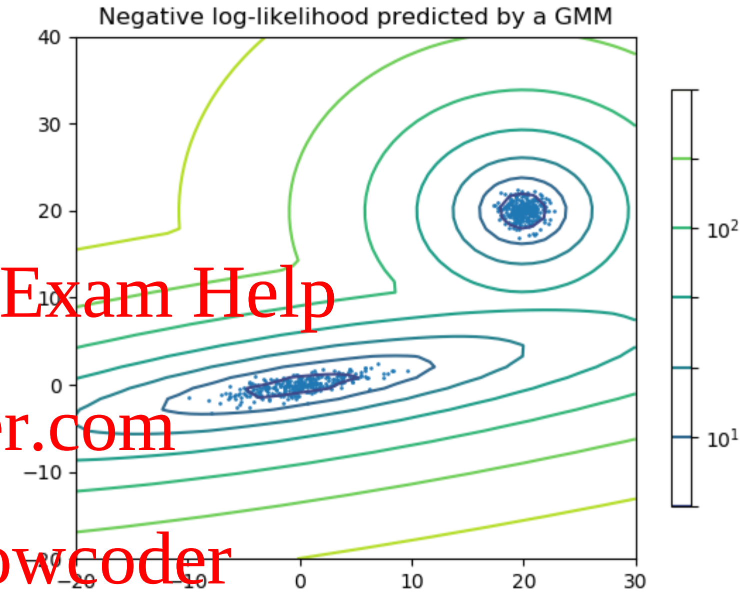
$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

see Bishop Ch. 9.2

Gaussian Mixtures

Data: $X = \{x_n\}$

Parameters: π_k, μ_k, Σ_k



$$- \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

How many possible solutions for *K* clusters? K^N

Is the cost function convex? no

Summary

- Unsupervised learning

- Discrete latent variables:
  - K-Means clustering

  - Gaussian Mixture clustering

- Next time: Continuous latent variables
  - Principal components analysis

Next Class

Unsupervised Learning I: PCA:

dimensionality reduction, PCA

Reading: Bishop 12.1