# Gaussian Mixture Models

David Mareček

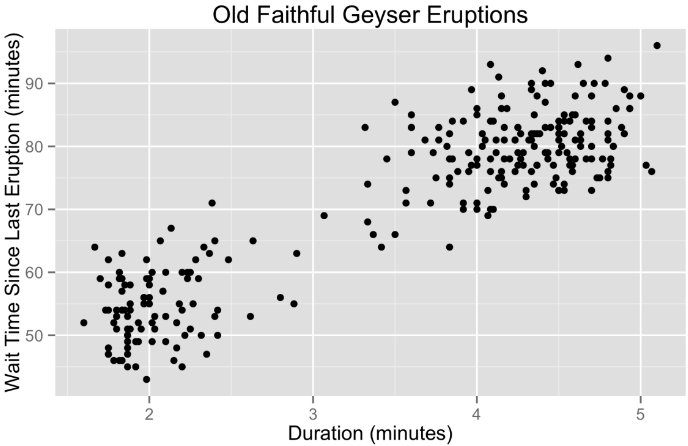
November 26, 2024

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Many of the slides in this presentation were taken from the presentations of David Rosenberg (New York University)

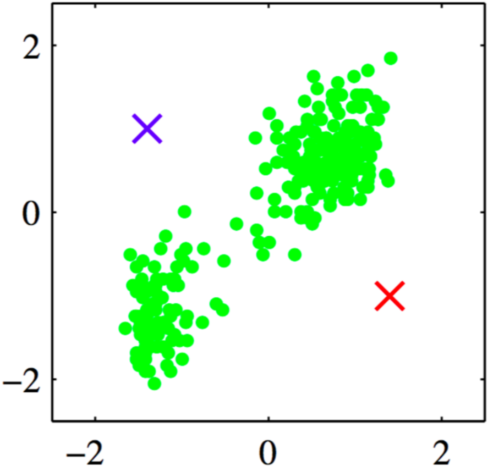
# Example: Old Faithful Geyser




- Looks like two clusters.
- How to find these clusters algorithmically?

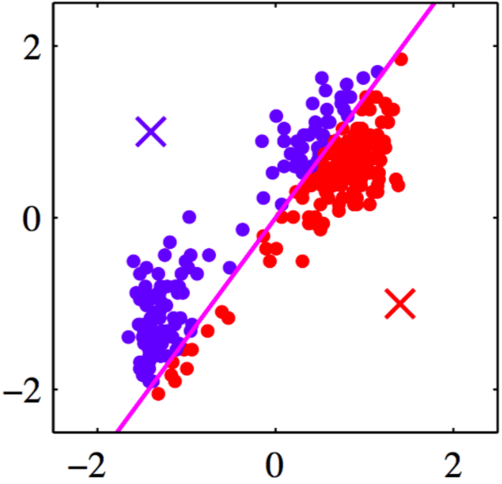
# k-Means clustering

- Standardize the data (equal mean and variance).
- Choose two cluster centers.



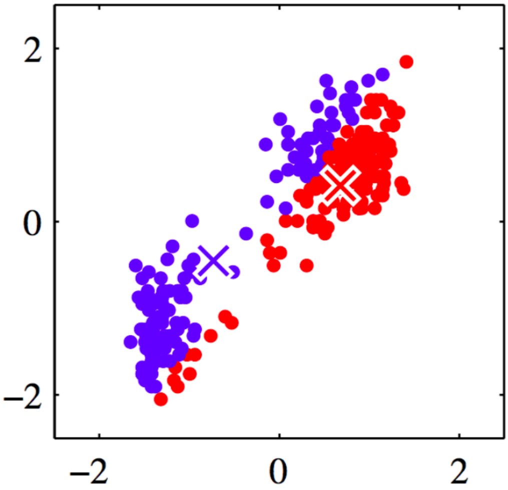
# k-Means clustering

- Assign each point to the closest center.



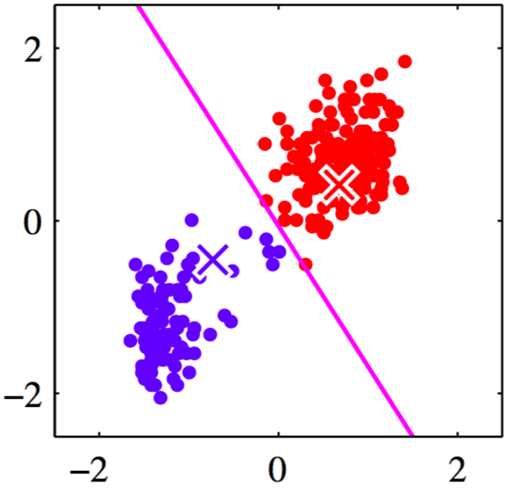
# k-Means clustering

- Compute new cluster centers.



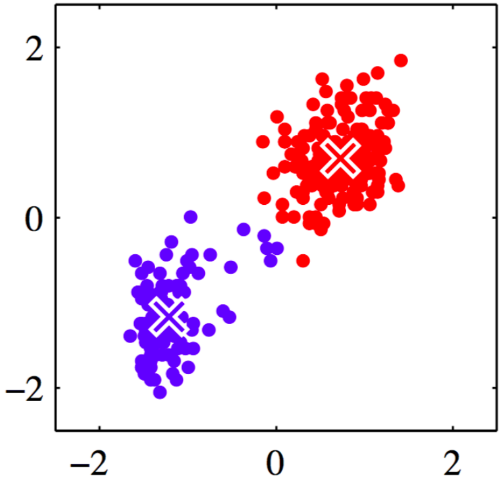
# k-Means clustering

- Assign each point to the closest center.



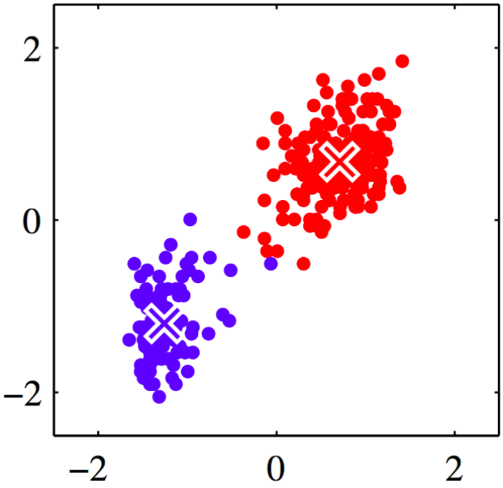
# k-Means clustering

- Compute new cluster centers.



# k-Means clustering

- Iterate until convergence.



# k-Means: Formalization

- Dataset $D = x_1, \dots, x_n \in \mathbb{R}^d$
- Goal (version 1): Partition data into $k$ clusters.
- Goal (version 2): Partition $\mathbb{R}^d$ into $k$ regions.
- Let $\mu_1, \dots \mu_k$ denote cluster centers.

# k-Means: Formalization

- For each $x_i$, use a **one-hot encoding** to designate membership:

$$r_i = (0, 0, \dots, 0, 0, 1, 0, 0)$$

- Let $r_{ic} = 1$ if $x_i$ is assigned to cluster $c$.
- Then,

$$r_i = (r_{i1}, r_{i2}, \dots, r_{ik}).$$

# k-Means: Objective function

- Find cluster centers and cluster assignments minimizing:

$$J(r, \mu) = \sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - x_c||^2.$$

- Is this objective function convex?
- What is the domain of $J$?
- $r \in \{0, 1\}^{n \times k}$, which is not a convex set...
- So domain of $J$ is not convex $\implies J$ is not a convex function.
- We should expect local minima.
- Could replace $|| \cdot ||^2$ with something else, e.g. using $|| \cdot ||$ gives k-**medoids**.

# k-Means: Algorithm

- For fixed $r$ (cluster assignments), minimizing over $\mu$ is easy:

$$J(r,\mu) = \sum_{i=1}^{n}\sum_{c=1}^{k} r_{ic}||x_i - \mu_c||^2$$

$$= \sum_{c=1}^{k}\sum_{i=1}^{n} r_{ic}||x_i - \mu_c||^2 = \sum_{c=1}^{k} J_c$$

$$J_c(\mu_c) = \sum_{i|x_i \text{ belongs to } c} ||x_i - \mu_c||^2$$

- $J_c$ is minimized by

$$\mu_c = \text{mean}(\{x_i | x_i \text{ belongs to cluster } c\})$$

# k-Means: Algorithm

- For fixed $\mu$ (cluster centers), minimizing over $r$ is easy:

$$J(r, \mu) = \sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic} ||x_i - \mu_c||^2$$

- For each $i$, exactly one of the following terms is nonzero:

$$r_{i1}||x_i - \mu_1||^2, \; r_{i2}||x_i - \mu_2||^2, \; \ldots, r_{ik}||x_i - \mu_k||^2$$

- Take

$$r_{ic} = 1 \quad \text{if} \quad c = \arg\min_j ||x_i - \mu_j||^2$$

- That is, assign $x_i$ to cluster $c$ with minimum distance

$$||x_i - \mu_c||^2$$

# k-Means algorithm: summary

We will use an **alternating minimization** algorithm:

- Choose initial cluster centers $\mu = (\mu_1, \dots, \mu_k)$.
  - e.g. choose $k$ randomly chosen data points
- Repeat
  - For given cluster centers, find optimal cluster assignments:

$$r_{ic}^{new} = 1 \quad \text{if} \quad c = \arg\min_j ||x_i - \mu_j||^2$$

  - Given cluster assignments, find optimal cluster centers:

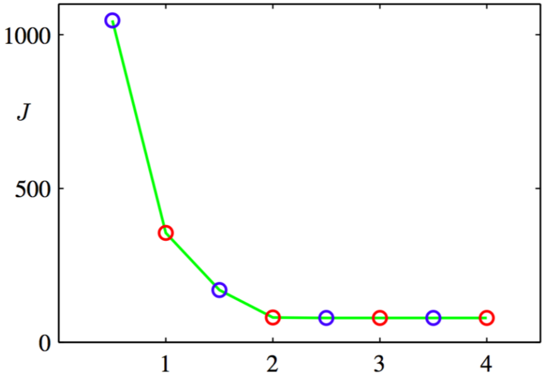
$$\mu_c^{new} = \arg\min_{m \in \mathbb{R}^d} \sum_{i | r_{ic} = 1} ||x_i - \mu_c||^2$$

# k-Means algorithm: convergence

- Note: objective value never increases in an update.
  - (Obvious: worst case, everything stays the same)
- Consider the sequence of objective values: $J_1, J_2, J_3, \ldots$
  - monotonically decreasing
  - bounded below by zero
- Therefore, k-means objective value converges to $\inf_t J_t$.
- **Reminder:** This is convergence to a **local** minimum.
- Best to repeat k-means several times, with different starting points.

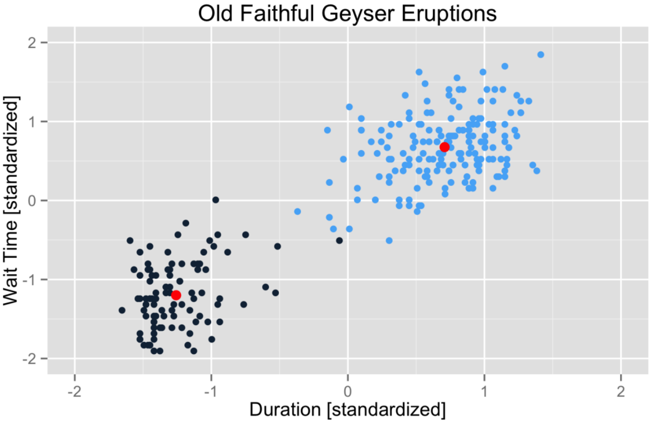
# k-Means: Objective Function Convergence

- Blue circles after **"E" step**: assigning each point to a cluster.
- Red circles after **"M" step**: recomputing the cluster centers.



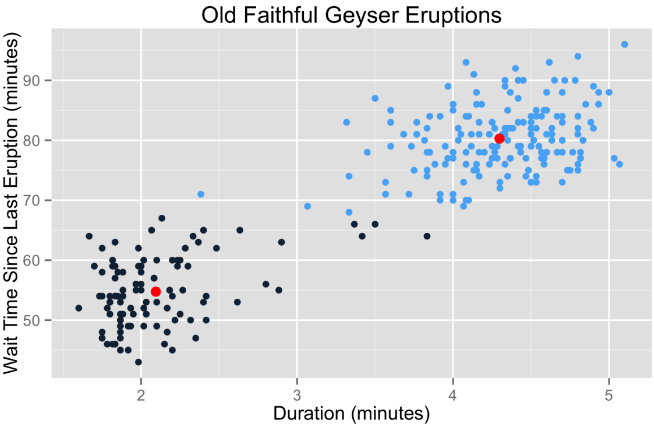
# k-Means Algorithm: Standardizing the data

- With standardising:



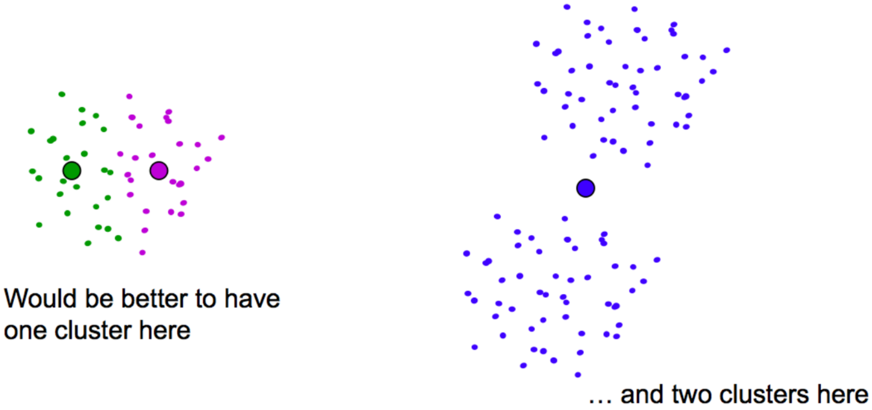
# k-Means Algorithm: Standardizing the data

- Without standardising:



# k-Means: Suboptimal Local Minimum

- The clustering for $k = 3$ below is a local minimum, but suboptimal:



# Cases, in which k-Means cannot succeed

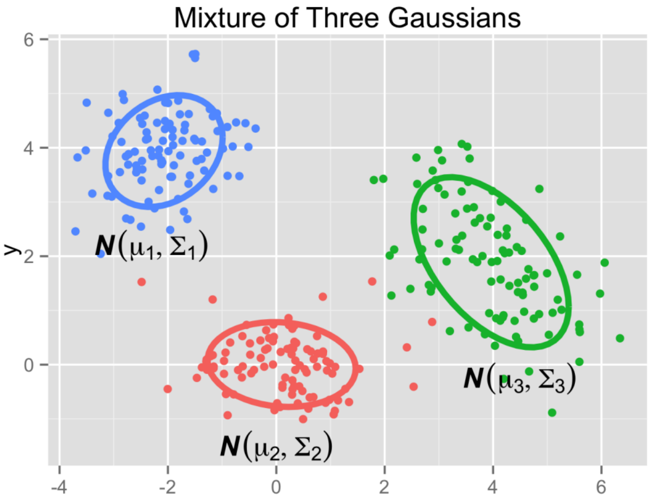
- Disadvantages of k-means: different cluster shapes and variances.
- TODO...

# Probabilistic Model for Clustering

- Let's consider a **generative model** for the data.
- Suppose
  - There are $k$ clusters.
  - We have a probability density for each cluster.
- Generate a point as follows:
  - Choose a random cluster $z \in \{1, 2, \dots, k\}$.
    - $Z \sim Multi(\pi_1, \dots, \pi_k)$.
  - Choose a point from the distribution for cluster $z$.
    - $X|Z = z \sim p(x|z)$.

# Gaussian Mixture Model ($k = 3$)

- Choose $Z \in \{1, 2, 3\} \sim \text{Multi}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.
- Choose $X|Z \sim \mathcal{N}(X|\mu_z, \Sigma_z)$

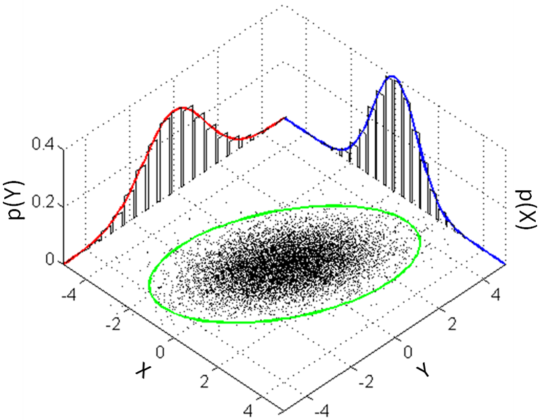


# Gaussian distribution and its multivariate generalization

**Gaussian distribution** with mean $\mu$ and variance $\sigma^2$: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{x-\mu}{\sigma}}$

**Multivariate Gaussian distribution** with mean $\vec{\mu}$ and covariance matrix $\Sigma$:

$$f(\vec{x}) = \frac{1}{\sqrt{\det(\Sigma)} \cdot \sqrt{(2\pi)^k}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^\top \Sigma^{-1} (\vec{x}-\vec{\mu})}$$

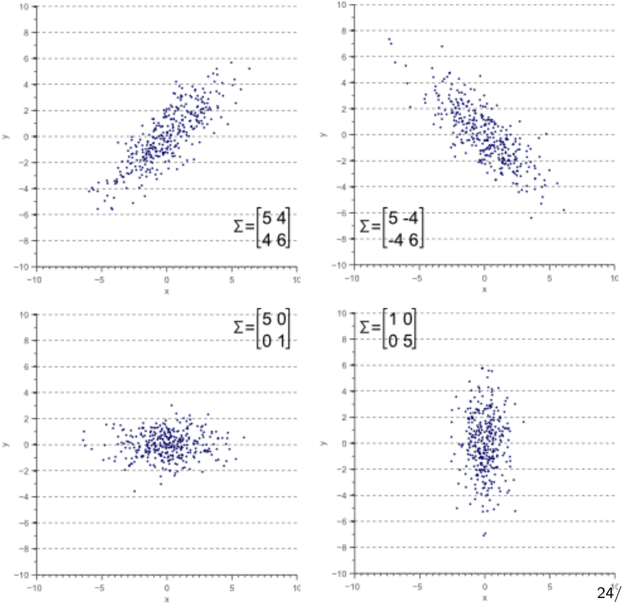


# Covariance matrix

- generalizes the notion of variance to multiple dimensions
- must be symmetric

$$\Sigma_{X_i,X_j} = cov(X_i, X_j)$$
$$= E((X_i - E(X_i))(X_j - E(X_j)))$$

$$\Sigma_{X_i,X_i} = var(X_i) = E((X_i - E(X_i))^2)$$



# Gaussian Mixture Model: Generative view

- We are generating each point together with its cluster using the joint distribution:

$$p(x, z) = p(z)p(x|z) = \pi_z \mathcal{N}(x|\mu_z, \Sigma_z)$$

- $\pi_z$ is probability of choosing cluster $z$.
- $x|z$ (a point $x$ assigned to cluster $z$) has a multivariate normal distribution $\mathcal{N}(\mu_z, \Sigma_z)$.
- a cluster $z$ corresponding to the point $x$ is the true cluster assignment.

# Latent Variable Model

- Back in the reality, we observe $X$, not $(X, Z)$.
- Cluster assignment $Z$ is called a **hidden variable**.
- Gausssian mixture model is a **Latent Variable Model**, i.e. a probability model for which certain variables are never observed.

# Model-Based Clustering

- We observe $X = x$.
- The conditional distribution of the cluster $z$ given (assigned to) the point $x$ is

$$p(z|x) = p(x, z)/p(x)$$

- The conditional distribution is a **soft assignment** to clusters.
- A **hard assignment** is

$$z^* = \underset{z \in \{1,\ldots,k\}}{\arg\max} P(Z = z | X = x)$$

.

- So if we have the model, clustering is trivial.

# Estimating/Learning the Gaussian Mixture Model

- We'll use the common acronym **GMM**.
- What does it mean to "have" or "know" the GMM?
- It means knowing the parameters:

$$\begin{aligned} \text{Cluster probabilities:} \quad & \pi = (\pi_1, \ldots, \pi_k) \\ \text{Cluster means:} \quad & \mu = (\mu_1, \ldots, \mu_k) \\ \text{Cluster covariance matrices} \quad & \Sigma = (\Sigma_1, \ldots, \Sigma_k) \end{aligned}$$

- We have a probability model: let's find the MLE.
- Suppose we have data $D = \{x_1, \ldots, x_n\}$.
- We need the model likelihood for $D$.

# Gaussian Mixture Model: Marginal Distribution

- Since we only observe $X$, we need the **marginal distribution**:

$$p(x) = \sum_{z=1}^{k} p(x, z) = \sum_{z=1}^{k} \pi_z \mathcal{N}(x|\mu_z, \Sigma_z)$$

- Note that $p(x)$ is a convex combination of probability densities.
- This is a common form for a probability model...

# Mixture Distributions (or Mixture Models)

**Definition:**
A probability density $p(x)$ represents a **mixture distribution** or **mixture model**, if we can write it as a **convex combination** of probability densities. That is,

$$p(x) = \sum_{i=1}^{k} w_i p_i(x),$$

where $w_i \geq 0$, $\sum_{i=1}^{k} w_i = 1$, and each $p_i$ is a probability density.

- In our Gaussian mixture model, $X$ has a **mixture distribution**.
- More constructively, let $S$ be a set of probability distributions:
  - Choose a distribution randomly from $S$.
  - Sample $X$ from the chosen distribution.
- Then $X$ has a mixture distribution.

# Estimating/Learning the Gaussian Mixture Model

- The model likelihood for $D = \{x_1, \dots, x_n\}$ is

$$L(\pi, \mu, \Sigma) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \sum_{z=1}^{k} \pi_z \mathcal{N}(x_i | \mu_z, \Sigma_z).$$

- As usual, we'll take our objective function to be the log of this:

$$J(\pi, \mu, \Sigma) = \sum_{i=1}^{n} \log\{\sum_{z=1}^{k} \pi_z \mathcal{N}(x_i | \mu_z, \Sigma_z)\}$$

# Properties of the GMM Log-Likelihood

- GMM log-likelihood:

$$J(\pi, \mu, \Sigma) = \sum_{i=1}^{n} \log\{\sum_{z=1}^{k} \pi_z \mathcal{N}(x_i|\mu_z, \Sigma_z)\}$$

- Let's compare to the log-likelihood for a single Gaussian:

$$\sum_{i=1}^{n} \log \mathcal{N}(x_i|\mu, \Sigma) = -\frac{nd}{2} \log(2\pi) - \frac{n}{2} \log|\Sigma| - \frac{1}{2} \sum_{i=1}^{n} (x_i - \mu)' \Sigma^{-1} (x_i - \mu)$$

- For a single Gaussian, the log cancels the exp in the Gaussian density.
- For the GMM, the sum inside the log prevents this cancellation $\Longrightarrow$ No closed form expression for MLE.

# MLE for Gaussian Model

- Let's start by the MLE for the Gaussian model.
- For data $D = \{x_1, \dots, x_n\}$, the log likelihood is given by

$$\sum_{i=1}^{n} \log \mathcal{N}(x_i|\mu, \Sigma) = -\frac{nd}{2} \log(2\pi) - \frac{n}{2} \log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)' \Sigma^{-1}(x_i - \mu)$$

- With some calculus, we find that the MLE parameters are

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\Sigma_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{MLE})(x_i - \mu_{MLE})^T$$

- For GMM, if we knew the cluster assignment $z_i$ for each $x_i$, we could compute the MLEs for each cluster.

# Cluster Responsibilities: Some New Notation

- Denote the probability that observed value $x_i$ comes from cluster $j$ by

$$\gamma_i^j = P(Z = j | X = x_i).$$

- The **responsibility** that cluster $j$ takes for observation $x_i$.
- Computationally,

$$\gamma_i^j = P(Z = j | X = x_i) = \frac{p(Z = j, X = x_i)}{p(x)} = \frac{\pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)}{\sum_{c=1}^{k} \pi_c N(x_i | \mu_c, \Sigma_c)}$$

- The vector $(\gamma_i^1, \dots, \gamma_i^k)$ is exactly the **soft assignment** for $x_i$.
- Let $n_c = \sum_{i=1}^{n} \gamma_i^c$ be the number of points "soft assigned" to cluster $c$.

# EM Algorithm for GMM: Overview

1. Initialize parameters $\mu, \Sigma, \pi$
2. "E step". Evaluate the responsibilities using current parameters:

$$\gamma_i^j = \frac{\pi_j N(x_i|\mu_j, \Sigma_j)}{\sum_{c=1}^{k} \pi_c N(x_i|\mu_c, \Sigma_c)},$$

   for $i = 1, \ldots, n$ and $j = 1, \ldots, k$.
3. "M step". Re-estimate the parameters using responsibilities:

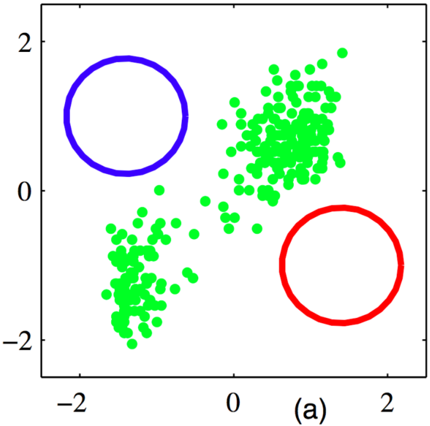
$$\mu_c^{new} = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_i^c x_i$$

$$\Sigma_c^{new} = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_i^c (x_i - \mu_{MLE})(x_i - \mu_{MLE})^T$$

$$\pi_c^{new} = \frac{n_c}{n},$$

4. Repeat from Step 2, until log-likelihood converges.

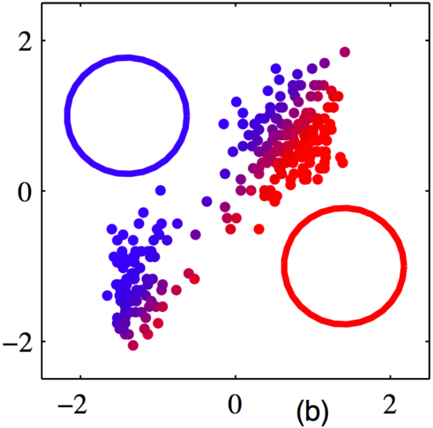
- Initialization:



(a)

# EM for GMM

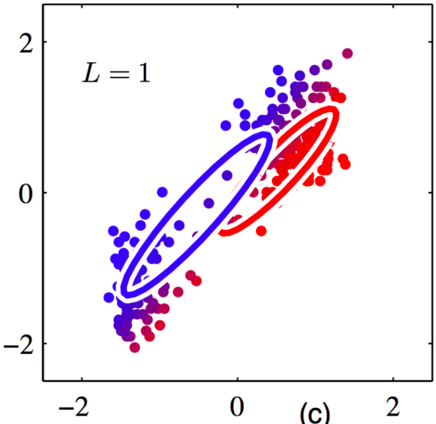
- First soft assignment:




(b)

# EM for GMM

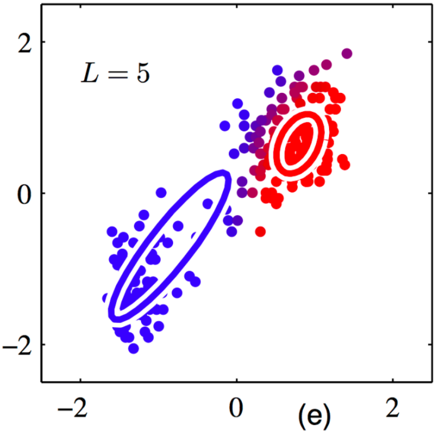
- First soft assignment:



(c)

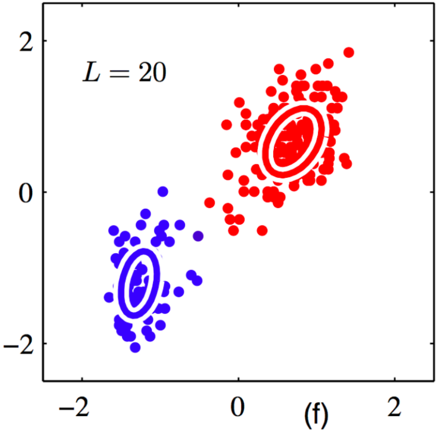
# EM for GMM

- After 5 rounds of EM:



# EM for GMM

- After 20 rounds of EM:



# Relation to K-Means

- EM for GMM seems a little like k-means
- In fact, there is a precise correspondence.
- First, fix each cluster covariance matrix to be $\sigma^2 I$
- As we take $\sigma^2 \leftarrow 0$, the update equations converge to doing $k$-means.
- If you do a quick experiment yourself, you'll find
  - Soft assignments converge to hard assignments.
- We can use k-means to initialize parameters of the GMM EM algorithm