# CSCE 633: Machine Learning

Lecture 36: Unsupervised Learning: Gaussian Mixture Models and Expectation Maximization

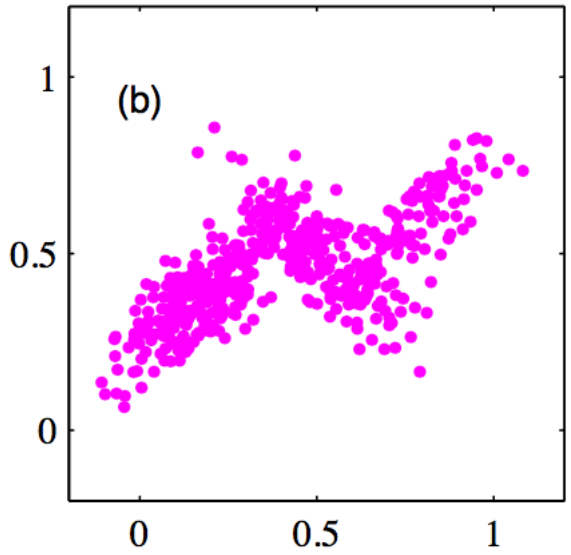
Texas A&M University
Bobak Mortazavi

# Review

- What are the key differences between k-means and hierarchical clustering?
- What are methods by which we define distances?
- Are there other ways in which we might define cluster membership?

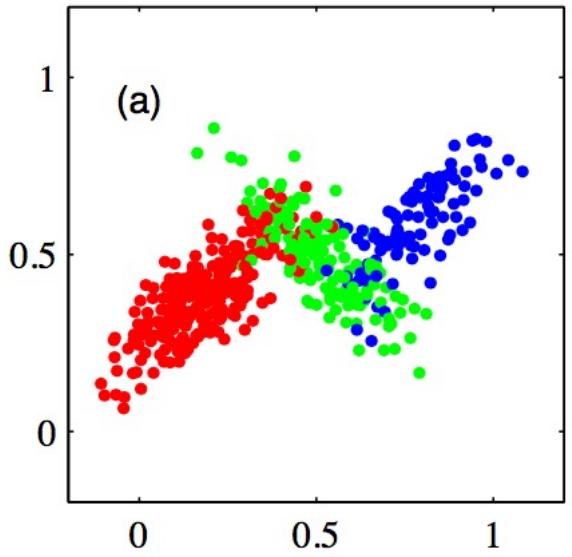
# Probabilistic Interpretation of Clustering

- We want to find a probability p(x) that best describes our data
- The data points here seem to form how many clusters?
- Cannot model a single p(x) with a simple, known distribution (e.g. one Gaussian distribution)



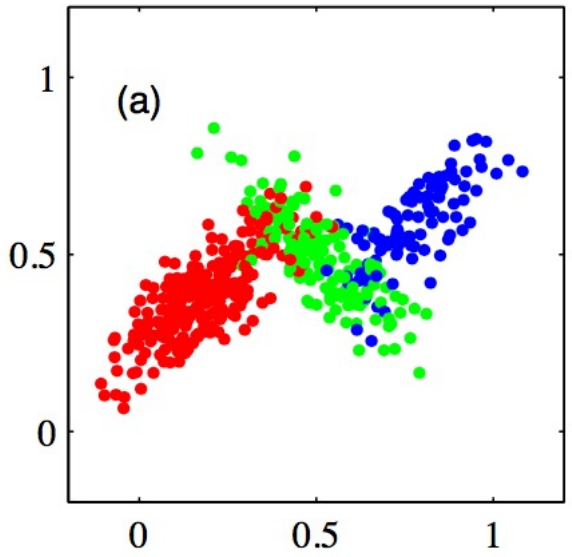
# Probabilistic Interpretation of Clustering

- Instead, we model each region with it's own Gaussian Distribution
- We will then mix these together – called a Gaussian Mixture Model (GMM)



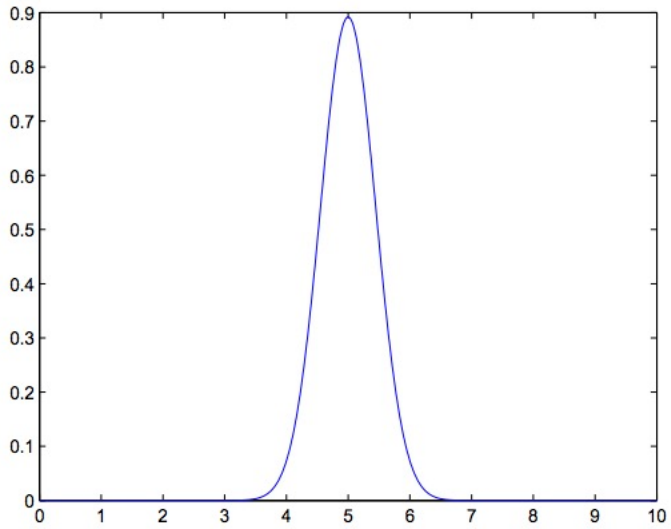
# Probabilistic Interpretation of Clustering

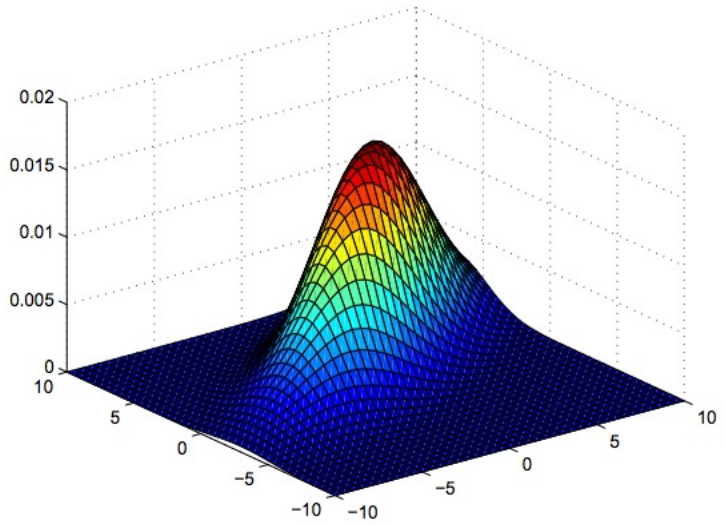
- Instead, we model each region with it's own Gaussian Distribution
- We will then mix these together – called a Gaussian Mixture Model (GMM)
- Questions that come up
  - How do we know which color (region) a data point comes from?
  - What are the parameters of the distributions of each region?



# Multivariate Gaussian Distributions

- Univariate Gaussian Distribution: $p(x;\ \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-\frac{1}{2\sigma^2}(x-\mu)^T)}$



# Multivariate Gaussian Distributions

- Univariate Gaussian Distribution: $p(x;\ \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{1}{2\sigma^2}(x-\mu)^2)}$
- Multivariate Gaussian Distributions (for a vector of p dimensions):

$$p(x;\ \mu, \Sigma) = \frac{1}{\sqrt{2\pi}^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} e^{(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))}$$

# Covariance Matrix

- The covariance between two random variables X and Y is

$$Cov(X,Y) = \mathbb{E}\big(X - \mathbb{E}(Y)\big)\big(Y - \mathbb{E}(Y)\big) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$$

- The covariance matrix provides a way to summarize the covariances of all pairs of variables

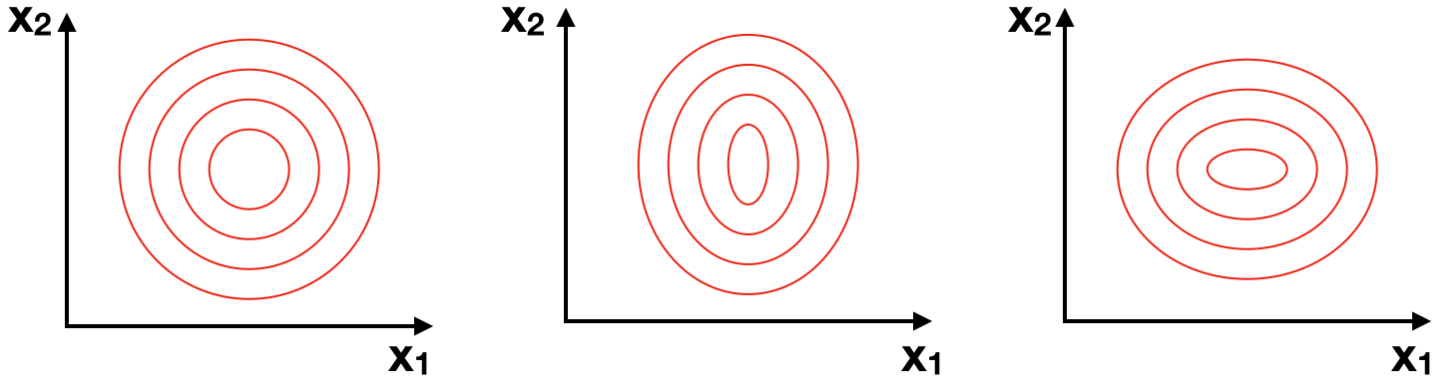
$$\Sigma_{ij} = Cov(X_i, X_j)$$

- Where $\Sigma$ is always positive definite

# Distributions with different Covariance Matrices

- The diagonal covariance case

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

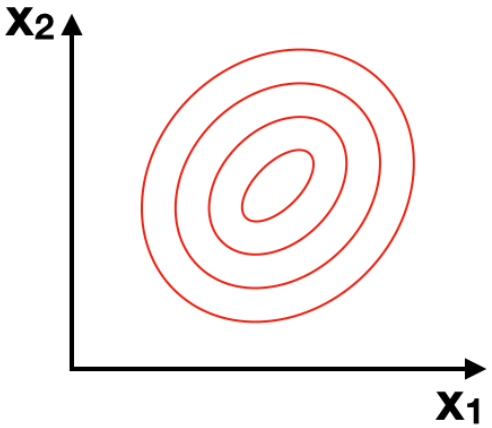



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

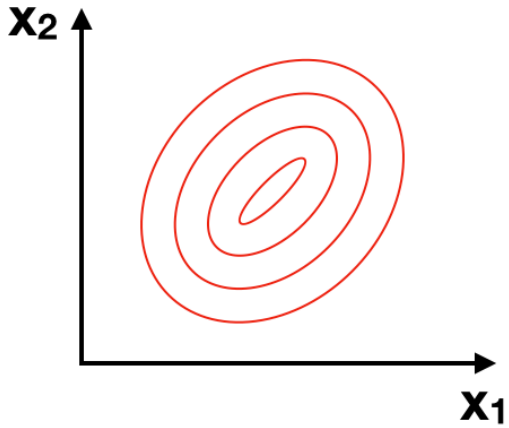
# Distributions with different Covariance Matrices

- The full covariance case

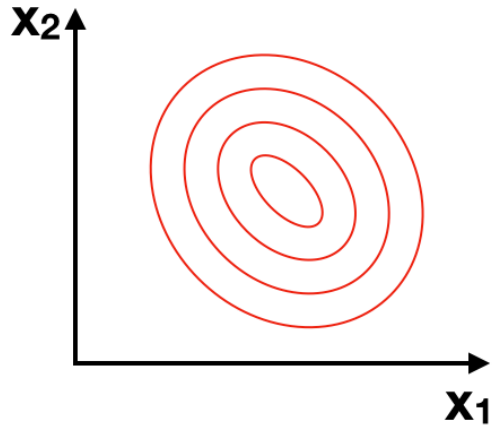
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$




$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$




$$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$




$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

# Gaussian Mixture Models: Formal Definition

- A Gaussian Mixture Model has the following density function for a vector x

$$p(x) = \sum_{k=1}^{K} w_k N(x; \mu_k, \Sigma_k)$$

- K: Number of Gaussians
- $\mu_k, \Sigma_k$ are the mean and covariance of the kth component
- $w_k$ component weight, how much component k contributes to the final distribution where

$w_k > 0$ for all k and $\sum_{k=1}^{K} w_k = 1$

$w_k$ can be represented by the prior distribution $w_k = p(z = k)$ which decides which mixture to use

# GMMs as the marginal distribution of a joint distribution

- Consider the following joint distribution p(x,z) = p(z) p(x|z)
- Z is a discrete random variable between 1 and K which "selects" a specific Gaussian component
- We denote the prior $w_k = p(z = k)$
- Assume Gaussian conditional distributions

$$p(x \mid z = k) = N(x; \mu_k, \Sigma_k)$$

- Then the marginal distribution of x is

$$p(x) = \sum_{k=1}^{K} w_k N(x; \mu_k, \Sigma_k)$$

Known as the Gaussian Mixture Model

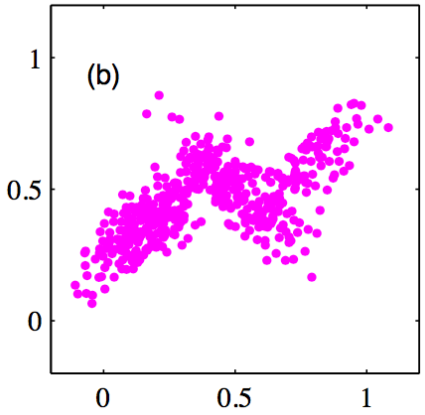
# GMMs for clusters

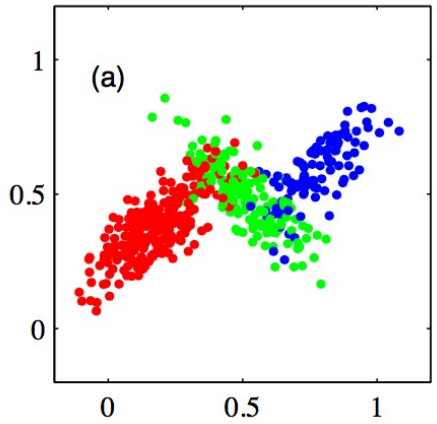
- If we have the joint distribution between x and z (representing a specific color) we have

$$p(x \mid z = red) = N(x; \mu_k, \Sigma_k)$$
$$p(x \mid z = blue) = N(x; \mu_k, \Sigma_k)$$
$$p(x \mid z = green) = N(x; \mu_k, \Sigma_k)$$

And the marginal distribution is then

$$p(x) = p(red)N(x; \mu_k, \Sigma_k) + p(blue)N(x; \mu_k, \Sigma_k) + p(green\ )N(x;\ \mu_k, \Sigma_k)$$



# Parameter Estimation for GMM

- Assume we knew the component each data point belonged to

$$Data\ D = \{(x_1, z_1), \dots, (x_n, z_n)\}$$

For subjects i = 1, … , n, and components 1 to k

- We want to find $\theta = (\mu_k, \Sigma_k, w_k)$ to satisfy

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log(p(D)|\,\theta)$$

# Parameter Estimation for GMM

- Assume we knew the component each data point belonged to

$$Data\ D = \{(x_1, z_1), \dots, (x_n, z_k)\}$$

For subjects i = 1, … , n, and components 1 to k

- We want to find $\theta = (\mu_k, \Sigma_k, w_k)$ to satisfy

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log(p(D)|\,\theta)$$

- The solution:

$$w_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} x_n$$

$$\Sigma_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

Where $\gamma_{nk} = 1$ if $z_n = k$

# Understanding the intuition

$$w_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} x_n$$

$$\Sigma_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

- For $w_k$: count the number of data points whose $z_n$ is k and divide by the total number of datapoints
- For $\mu_k$: get all the data points whose $z_n$ is k and compute their mean
- For $\Sigma_k$: get all the data points whose $z_n$ is k and compute their covariance

# Estimating membership with a soft $\gamma_{nk}$

- Define $\gamma_{nk} = p(z_n = k \mid x_n)$

Still have

$$w_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} x_n$$

$$\Sigma_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

- Only now every data point $x_n$ is assigned to a component fractionally according to $p(z_n = k \mid x_n)$, which we call the responsibility

# Parameter Estimation

- To estimate a soft $\gamma_{nk}$

- Since we do not know $\theta$ to begin with, we cannot compete the soft $\gamma_{nk}$ but we can invoke an iterative procedure and alternate between estimating $\gamma_{nk}$ and computing $\theta = (\mu_k, \Sigma_k, w_k)$
- Step 0: Guess $\theta$ with initial values
- Step 1: compute $\gamma_{nk}$ given current $\theta$
- Step 2: Update $\theta$ using the computed $\gamma_{nk}$
- Step 3: Go back to step 1

Questions: i) is this procedure correct, for example, in optimizing a sensible set of criteria?

ii) Practically, will this procedure ever stop iterating?

The answer lies in the expectation maximization (EM) algorithm, a powerful procedure for model estimation with unknown data

# EM: Motivation and Setup

- EM is used to estimate parameters for probabilistic models with hidden/latent variables

$$p(x;\ \theta) = \sum_{z} p(x, z;\ \theta)$$

Where x is the observed random variable and z is the hidden

- We are given data $D = \{x_1, \dots, x_n\}$ where the corresponding hidden variable values z are not included
- Our goal then is to obtain the maximum likelihood estimate of $\theta$

$$\theta^* = \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \log p(x_i|\theta) = \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \log \sum_{i=1}^{n} p(x_i, z_i|\theta)$$

- The above objective function is called the incomplete log-likelihood and it is computationally intractable (since it needs to sum over all possible values of z and then take their log)

# EM: The complete log-likelihood

- If the incomplete log-likelihood is

$$l(\theta) = \sum_{i=1}^{n} \log \sum_{n} p(x_i, z_i|\theta)$$

- EM uses a clever trick to change this into a sum-log form by defining

$$Q_q(\theta) = \sum_{i=1}^{n} \log \mathbb{E}_{z_i \sim q(z_i)} p(x_i, z_i|\theta) = \sum_{n} \sum_{z} q(z_n) \log p(x_n, z_n|\theta)$$

Where q(z) is distributed over the possible z

- The above is called the expected (complete) log-likelihood, since it takes the form of sum log which is computational tractable

# EM: Choice of q(z)

- We will chose a special $q(z) = p(z\,|x, \theta)$ i.e. the posterior probability of z
- We can show that if

$$Q(\theta) = Q_{z \sim p(z|x,\theta)}(\theta)$$

then

$$l(\theta) = Q(\theta) + \sum_{i=1}^{n} \mathbb{H}\left(p(z|x_i, \theta)\right)$$

Where $\mathbb{H}$ is the entropy of the probabilistic distribution $p(z|x, \theta)$

$$\mathbb{H}\left(p(z|x\ \ ,\theta)\right) = -\sum_{z} p(z|x,\theta)\log p(z|x,\theta)$$

# EM

- As before, $Q(\theta)$ cannot be computed, as $p(z|x,\theta)$ depends on the unknown parameter values $\theta$
- Instead we will use a known value of $\theta^{old}$ to compute the expected likelihood

$$Q(\theta,\theta^{old}) = \sum_n \sum_z p(z_n|x_n,\theta^{old}) \log p(x_n,z_n|\theta)$$

- In the above the variable is $\theta$ , while $\theta^{old}$ is known
- We will show that

$$l(\theta) \geq Q(\theta,\theta^{old}) + \sum_n \mathbb{H}\left(p(z|x_n,\theta^{old})\right)$$

- Thus, $Q(\theta)$ is better than $Q(\theta,\theta^{old})$ but we cannot compute the former

# EM Algorithm

- Suppose we have an initial guess $\theta^{old}$ and we maximize the auxiliary function

$$\theta^{new} = \underset{\theta}{\operatorname{argmax}}\, A(\theta, \theta^{old})$$

With the new guess we will have

$$l(\theta^{new}) \geq A\left(\theta^{new}, \theta^{old}\right) \geq l(\theta^{old})$$

By maximizing the auxiliary function, we will keep increasing the likelihood (which is the core of the EM algorithm)

# EM Algorithm: Outline

- Step 0: Initialize $\theta$ with $\theta^{(0)}$
- Step 1 (The E-Step): Compute the auxiliary function using the current value of $\theta$

$$A\left(\theta \ , \theta^{(t)}\right)$$

- Step 2 (The M-Step): Maximize the auxiliary function such that

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, A(\theta, \theta^{(t)})$$

- Step 3: Increase t to t + 1 and go back to step 1, or stop if $l\left(\theta^{(t+1)}\right)$ does not improve much

# EM Algorithm: Remarks

- EM converges only to a local optimum: a global optimum is not guaranteed
- The E-step depends on computing the posterior probability $p(z|x, \theta^{(t)})$
- The M-step does not depend on the entropy term, so we need only to do the following

$$\theta^{(t+1)} = \operatorname*{argmax}_{\theta} A(\theta, \theta^{(t)}) = \operatorname*{argmax}_{\theta} Q(\theta, \theta^{(t)})$$

We often call the last term the Q-function

# EM for GMMs

- Hidden variable follows a multinomial distribution such that $z_n \sim Multinomial(w_1, \dots, w_k)$
- The likelihood of observation $x_n$ equals the probability of a corresponding component $p(x_n|z_n, \theta) = \prod_{k=1}^{K} N(x_n|\mu_k, \Sigma_k)^{\gamma_{nk}}$
- We can calculate the complete data log-likelihood as

$$Q(\theta) = \mathbb{E}\left[\sum_n \log p(x_n, z_n|\theta)\right] = \mathbb{E}\left[\sum_n \log(p(z_n|\theta)p(x_n|z_n, \theta))\right]$$

$$= \mathbb{E}\left[\sum_n \log\left(\prod_k w_k^{\gamma_{nk}} \prod_k N(x_n|\mu_k, \Sigma_k)^{\gamma_{nk}}\right)\right]$$

$$= \mathbb{E}\left[\sum_n \sum_k \gamma_{nk}(\log w_k + \log N(x_n|\mu_k, \Sigma_k))\right]$$

$$= \sum_n \sum_k \mathbb{E}[\gamma_{nk}|x_n, \mu_k, \Sigma_k]\gamma_{nk}(\log w_k + \log N(x_n|\mu_k, \Sigma_k))$$

# E-step for GMM

$$\mathbb{E}[\gamma_{nk}|x_n, \mu_k, \Sigma_k] = p(\gamma_{nk} = 1\,|x_n, \theta_k)$$

$$= \frac{p(x_n|\gamma_{nk} = 1, \theta_k)p(\gamma_{nk} = 1|\theta_k)}{\sum_k p(x_n|\gamma_{nk} = 1, \theta_k)\, p(\gamma_{nk} = 1|\theta_k)}$$

We compute the probability

$$\gamma_{nk} = p(z = k|x_n, \theta^{(t)})$$

$$= \frac{w_k p(x_n|\mu_k^{(t-1)}\Sigma_k^{(t-1)})}{\sum_k w_k p(x_n|\mu_k^{(t-1)}\Sigma_k^{(t-1)})}$$

# M-step for GMM

Maximize the auxiliary function

$$Q(\theta, \theta^{(t)}) = \sum_n \sum_k p(z = k | x_n, \theta^{(t)}) \log p(x_n, z = k | \theta)$$

$$= \sum_n \sum_k \gamma_{nk} \log p(x_n, z = k | \theta)$$

$$= \sum_n \sum_k \gamma_{nk} \log(p(z = k)\, p(x_n | z = k)$$

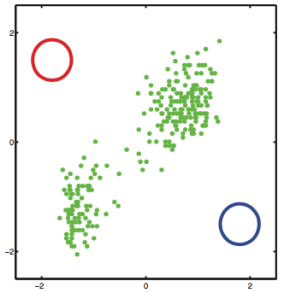
$$= \sum_n \sum_k \gamma_{nk} [\log w_k + \log N(x_n | \mu_k, \Sigma_k)]$$

Which yields

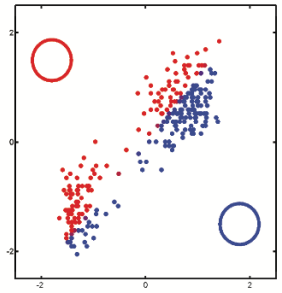
$$w_k = \frac{\sum_n \gamma_{nk}}{N}, \mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk}\, x_n$$

$$\Sigma_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk}\, (x_n - \mu_k)(x_n - \mu_k)^T$$

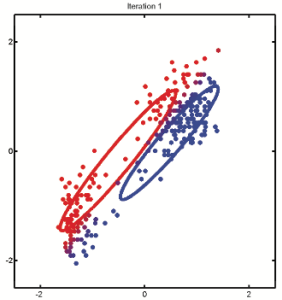
# EM for GMMs



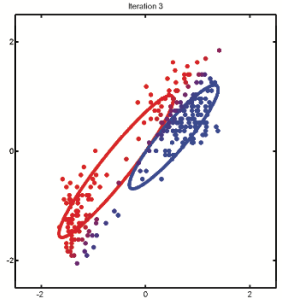
(a)



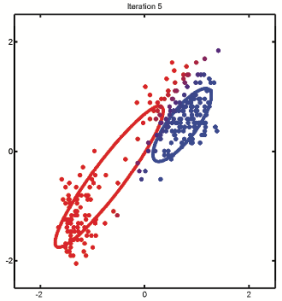
(b)

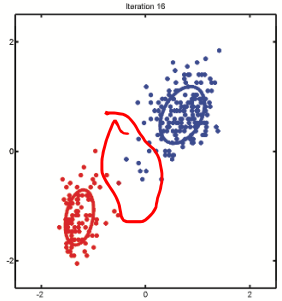



(c)




(d)



# GMM and K-Means

- GMMs provide probabilistic interpretation for k-means
- Assume all Gaussian components have $\sigma^2 I$ for their covariance matrices
- Assume further that those approach 0
- Then we only need to estimate $\mu_k$, i.e. the means
- Thus, EM for GMM is a parameter estimation that simplifies to K-Means
- For this reason, k-means is often called a Hard GMM or GMM is called a Soft k-means. The soft posterior provides a probabilistic assignment to a cluster k represented by the corresponding Gaussian distribution
- In both cases – how do you find the right number of components, k?

# Takeaways

- Goals:
  - Understanding why we use unsupervised learning
  - Understanding the key differences between k-Means, hierarchical clustering, and GMM
  - Understanding the Expectation Maximization approach to training