EM and Gaussian Mixture Models

CS444

Some material on these is slides borrowed from Andrew Moore's machine learning tutorials located at:

http://www.cs.cmu.edu/~awm/tutorials/

Parameterized Probability Distributions

- Parameterized probability distribution:

$$P(X) = P(X;\theta)$$

- θ - The parameters for the distribution.
- Trivial discrete example: X is a Boolean random variable θ indicates the probability that it will be true.

$\theta = .6$

$$p(X = TRUE; \theta = .6) = .6$$
$$p(X = FALSE; \theta = .6) = .4$$

$\theta = .1$

$$p(X = TRUE; \theta = .1) = .1$$
$$p(X = FALSE; \theta = .1) = .9$$

Fitting a Distribution to Data

- Assume we have a set of data points x_1 to x_N.
- The goal is to find a distribution that fits that data. I.e. that could have generated the data.

Maximum Likelihood Learning

- We will assume that x_1 to x_N are **iid** – independent and identically distributed.
- So we can write our problem like this (factorization):

$$\hat{\theta} = \underset{\theta}{argmax} \prod_{i=1}^{N} P(x_i; \theta)$$

- Taking the log gives us log likelihood:

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1}^{N} \log(P(x_i | \theta)) = \underset{\theta}{argmax}\, L$$

Silly Example

- Parameterized coin: Theta – probability of heads:
- ***d*** -- vector of toss data, *h* number of heads, *t* number of tails.

$$P(\boldsymbol{d};\theta) = \prod_{i=1}^{N} P(d_i;\theta) = \theta^h (1-\theta)^t$$

$$L(\boldsymbol{d};\theta) = log(P(\boldsymbol{d};\theta)) = h\,log\,\theta + t\,log(1-\theta)$$

$$\frac{\partial L}{\partial \theta} = \frac{h}{\theta} - \frac{t}{1-\theta} = 0 \quad \rightarrow \quad \theta = \frac{h}{h+t}$$

Remember: $\frac{d}{dx} log(x) = 1/x$

Example borrowed from Russel & Norvig

Maximizing Log Likelihood

- Just another instance of function maximization.
- One approach, set the partial derivatives to 0 and solve:

$$\frac{\partial L}{\partial \theta_1} = 0$$

$$\frac{\partial L}{\partial \theta_2} = 0$$

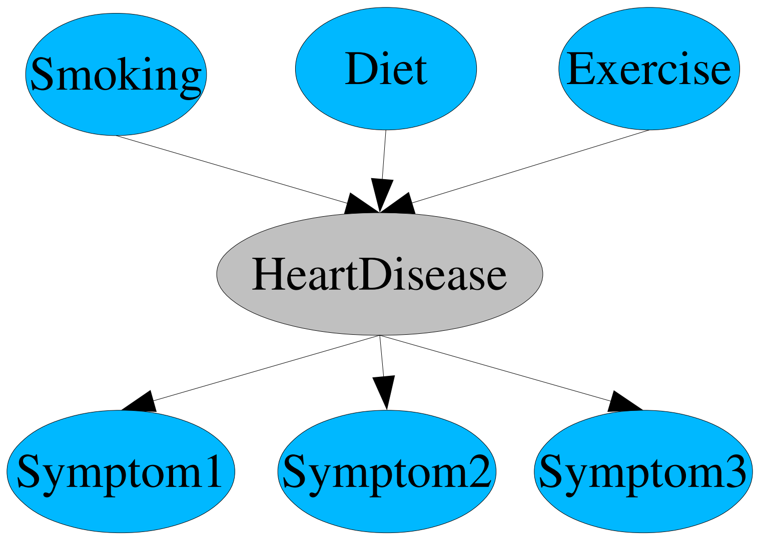
$$\dots$$

$$\frac{\partial L}{\partial \theta_K} = 0$$

- If you can't solve it, gradient descent, or your favorite search algorithm.

Learning With Hidden Variables

- Let's say we have want to learn the parameters of the following Bayes' net:



- We have a database of patient information that includes the lifestyle variables, and the symptom variables, but not a heart disease diagnosis.

Our Dilemma

- Chicken and egg problem:
  - If we knew the parameters of the Bayes' net, we could estimate the probability of the HeartDisease variable.
  - If we know the value of the HeartDisease variable, we could use it to learn the Bayes' net parameters.
- We don't have either.
- The Solution: Expectation Maximization

Expectation Maximization

- Assume that our hidden variables are Z, observed variables are X.
- Guess an assignment to our parameters $\hat{\theta}$.
- Expectation-Step:
  - Compute the expected value of our hidden variables E[Z].
- Maximization-Step
  - Pretend that E[Z] is the true value of Z and use ML to calculate a new $\hat{\theta}$

$$\hat{\theta} = \underset{\hat{\theta}}{argmax} \sum_{i=1}^{N} \log\left(P\left(x_i, E\left[Z_i\right]; \hat{\theta}\right)\right) = \underset{\hat{\theta}}{argmax}\, LL$$

EM Properties

- EM is guaranteed to converge to a local optimum.
- Guaranteed convergence is good.
- Local optimum is bad – the algorithm is sensitive to our initial guess for θ .

An Aside: Covariance

- Remember variance, the expected squared difference from the mean:

$$\sigma^2 = Var[X] = E[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx$$

- Now consider two continuous random variables x_1 and x_2, covariance is defined as:

$$cov(x_{1,} x_2) = E[(x_1-\mu_1)(x_2-\mu_2)]$$

Properties of Covariance

- Covariance is symmetric: $\mathrm{cov}(x_1, x_2) = \mathrm{cov}(x_2, x_1)$.
- If x_1 and x_2 are independent, then $\mathrm{cov}(x_1, x_2) = 0$.
- If $\mathrm{cov}(x_1, x_2) > 0$ then x_2 tends to increase as x_1 increases.
- If $\mathrm{cov}(x_1, x_2) < 0$ then x_2 tends to decrease as x_1 increases.
- Correlation is defined as follows:

$$cor(x_{1,}x_2)=\frac{cov(x_{1,}x_2)}{\sigma_1\sigma_2}$$

- Just covariance rescaled: $-1 \leq cor(x_{1,}x_2) \leq 1$

Random Vectors

- Consider a random vector $\boldsymbol{X}$.
- The expectation is:

$$\boldsymbol{M} = E[\boldsymbol{X}] = \int_{-\infty}^{\infty} \boldsymbol{X}\, p(\boldsymbol{X})\, d\boldsymbol{X}$$

- The covariance matrix is:

$$\Sigma = E[(\boldsymbol{X} - \boldsymbol{M})(\boldsymbol{X} - \boldsymbol{M})^T]$$

- Sample mean and covariance matrix:

$$\hat{\boldsymbol{M}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{X}_{\boldsymbol{i}} \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{X}_{\boldsymbol{i}} - \hat{\boldsymbol{M}})(\boldsymbol{X}_{\boldsymbol{i}} - \hat{\boldsymbol{M}})^T$$

Covariance Matrix?

- For a random vector $\boldsymbol{X}$ with N dimensions, the covariance matrix is a $N \times N$ matrix where entry (i,j) is $\mathrm{cov}(x_i, x_j)$.
- For a two dimensional vector:

$$cov(\boldsymbol{X}) = \begin{bmatrix} cov(x_{1,} x_1) & cov(x_{1,} x_2) \\ cov(x_{2,} x_1) & cov(x_{2,} x_2) \end{bmatrix}$$

- Things to notice
  - The matrix is symmetric.
  - The values on the diagonal are just the variance of the i'th dimension.

Multi-Dimensional Gaussians

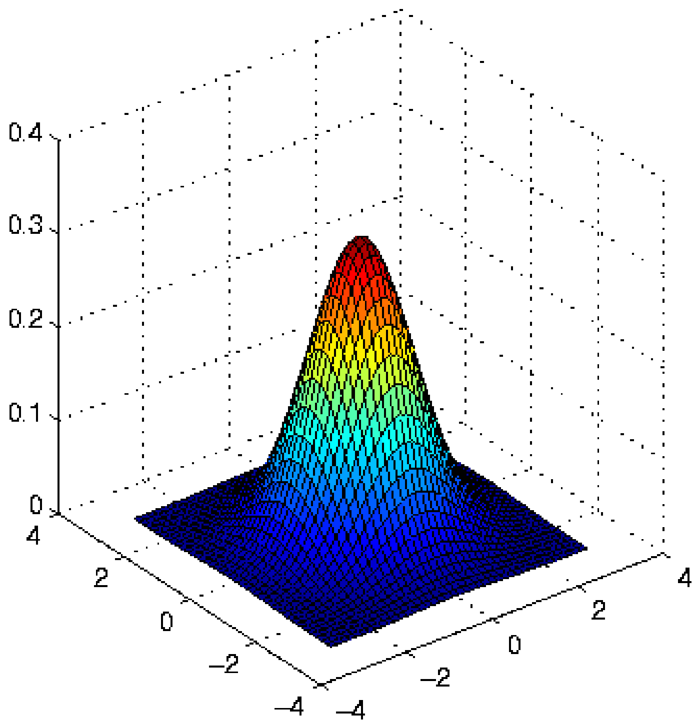
- Here is the formula for an N dimensional Gaussian. It's not as bad as it looks.

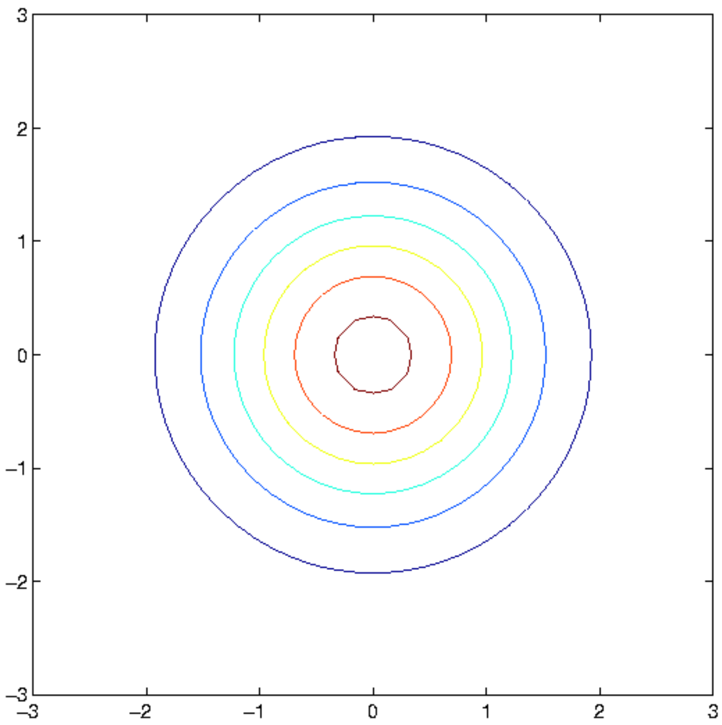
$$p(\boldsymbol{X}) = \left(\frac{1}{(2\pi)^{\frac{N}{2}} \left|\Sigma\right|^{\frac{1}{2}}} \right) e^{\left(-\frac{1}{2} (\boldsymbol{X}-\boldsymbol{M})^T \Sigma^{-1} (\boldsymbol{X}-\boldsymbol{M}) \right)}$$

- Notice: $\sum$ and $\boldsymbol{M}$ uniquely describe a multivariate Gaussian.

Examples:

$$\boldsymbol{M} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

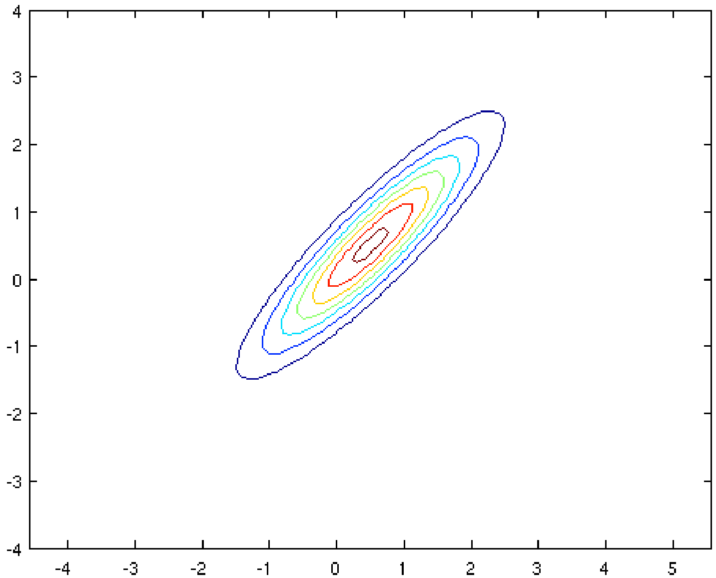


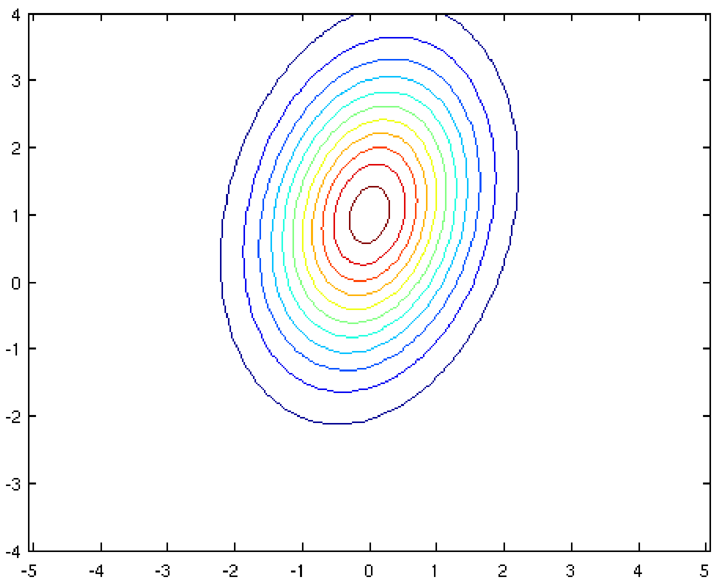


Examples

$$\boldsymbol{M} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

$$\boldsymbol{M} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & .3 \\ .3 & 1 \end{bmatrix}$$





EM for Gaussian Mixture Models

- Data points are generated from one of K Gaussians, each of which may have a different mean and covariance.
- Our hidden variables are K variables W_1 through W_K. W_j=1 if a point was generated by component j.
- Parameters:
  - Let π_j be the prior probability that a given point comes from mixture component j: $P(W_j=1) = \pi_j$.
  - There is a μ and Σ for each mixture component. μ_1 through μ_K and Σ_1 through Σ_K.
- The goal is to recover these parameters from unlabeled data points.

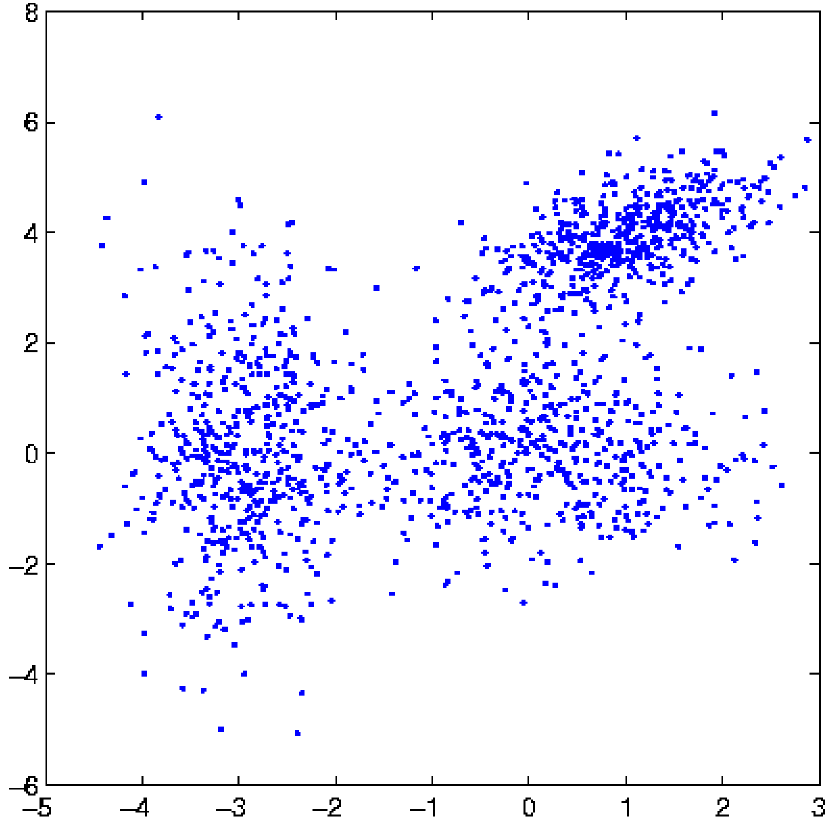
More on GMM

- The pdf: $p(x)=\sum_{i=1}^{K} \pi_i \, p(x|\mu_i, \Sigma_i)$

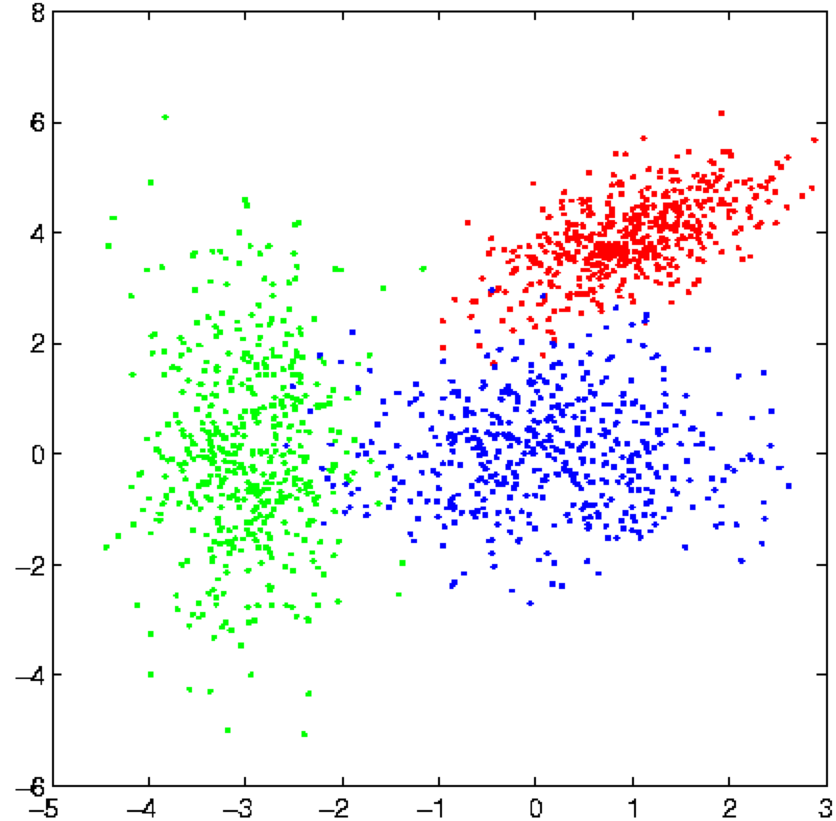
- To generate a data point:
  - First select a mixture component according to P(W).
  - Then generate a point from the Gaussian associated with that mixture component.

Gaussian Mixture Example

We have this:



Life would be easier if we had this:



EM for GMM

- E-Step ($p_{i,j}$ is the probability that point i was generated by mixture component j) $P(W_j=1|x_i,\theta)$. This is the expected value of W_j. (W_j is an indicator variable.)

$$p_{i,j}=\frac{p(x_i|\mu_j,\Sigma_j)\pi_j}{\sum_{k=1}^{K} p(x_i|\mu_k,\Sigma_k)\pi_k}$$

EM for GMM

- M Step: Update the parameters:

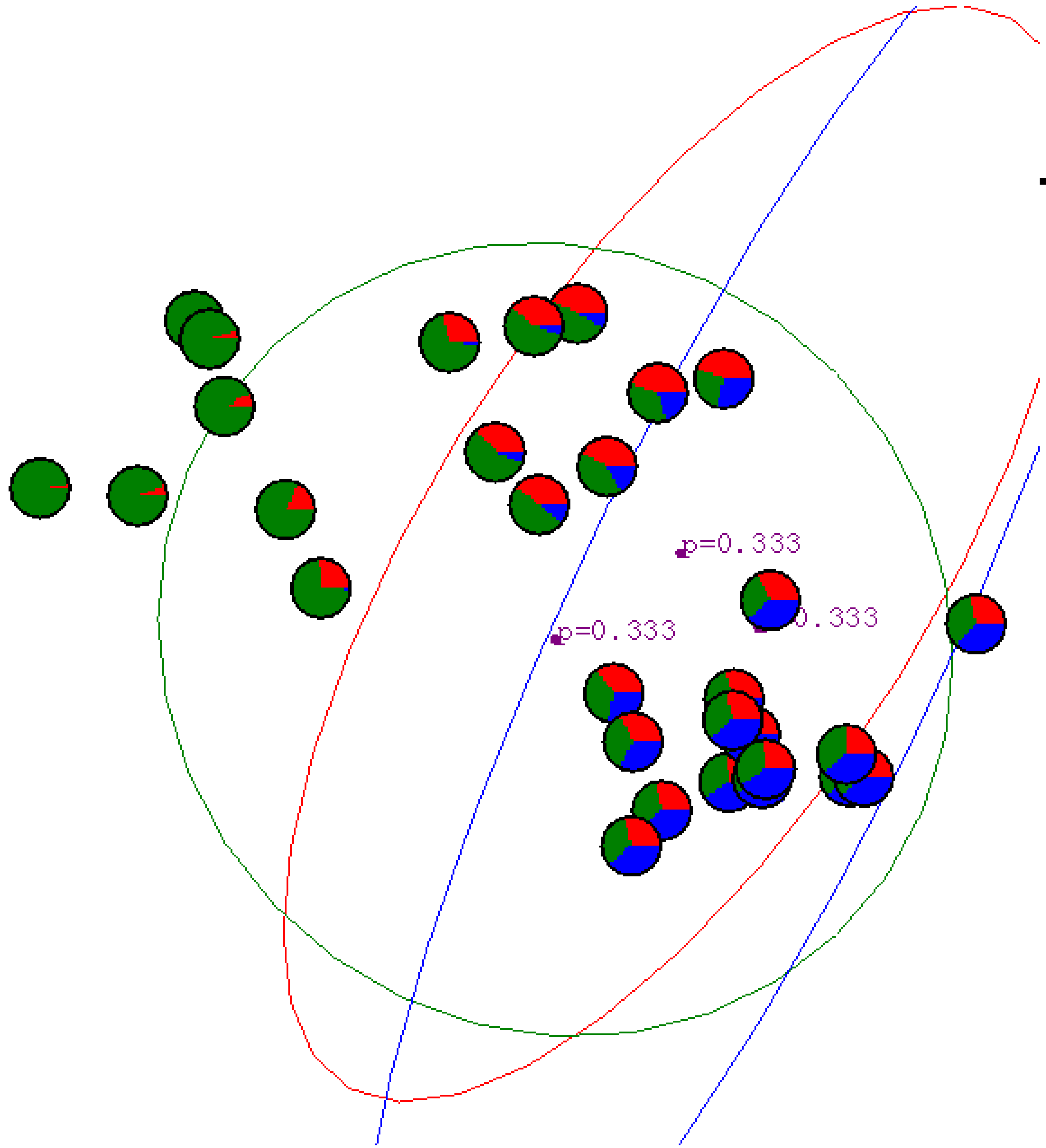
$$\mu_j = \frac{\sum_{i=1}^{N} p_{i,j} x_i}{\sum_{i=1}^{N} p_{i,j}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} p_{i,j} (\boldsymbol{x_i} - \mu_j)(\boldsymbol{x_i} - \mu_j)^T}{\sum_{i=1}^{N} p_{i,j}}$$

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} p_{i,j}$$

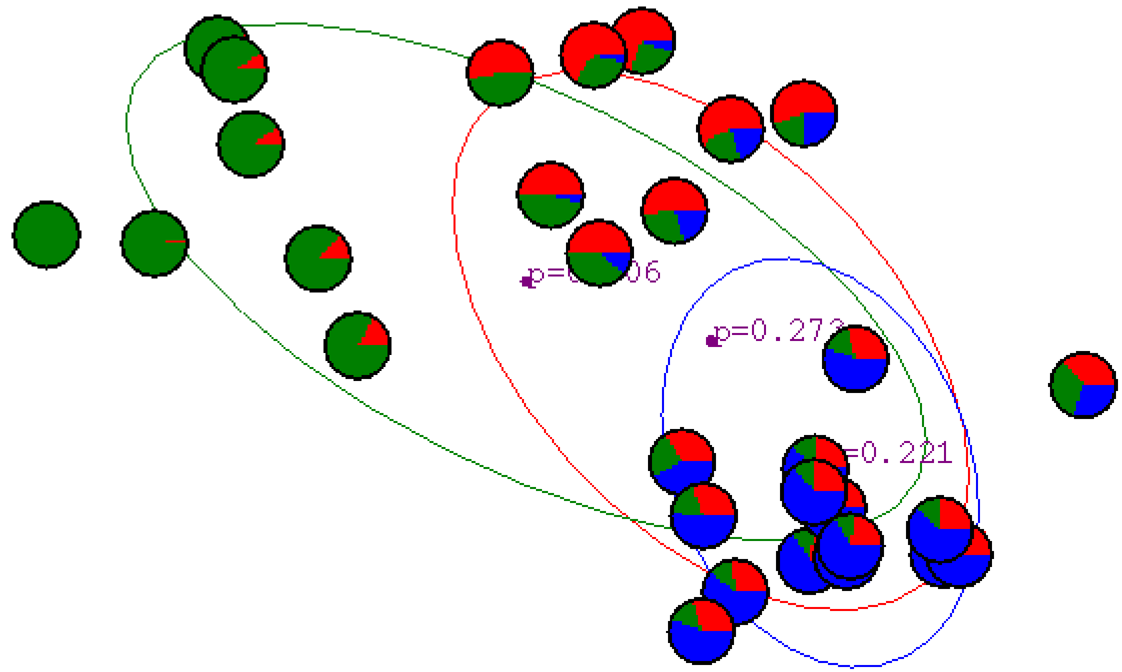
Gaussian Mixture Example: Start

Advance apologies: in Black and White this example will be incomprehensible



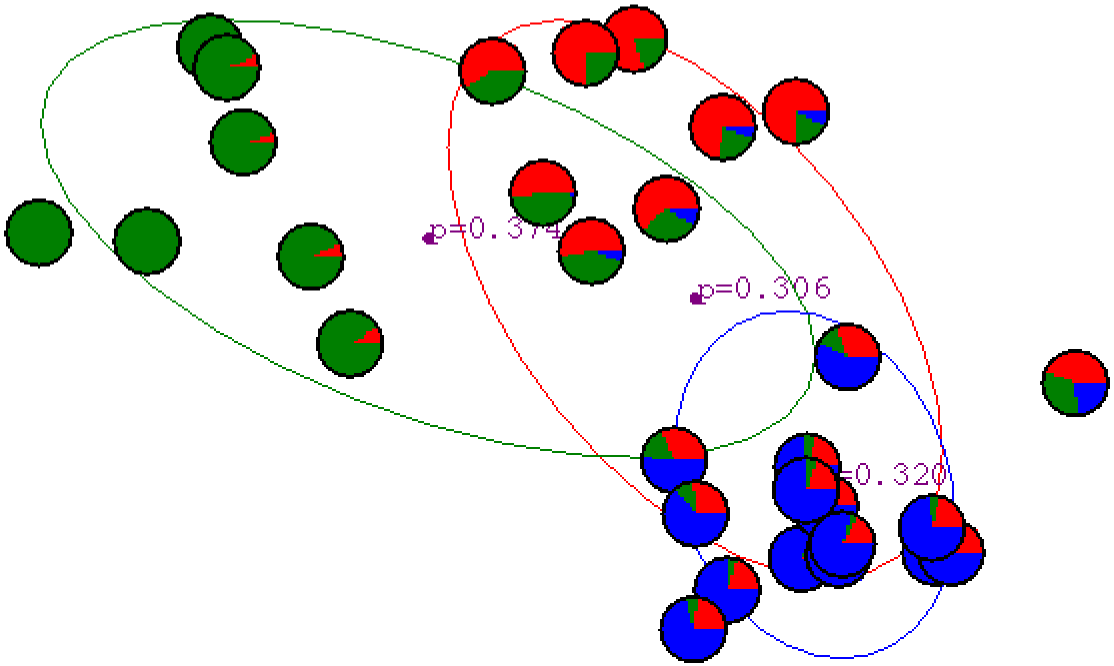
http://www.cs.cmu.edu/~awm/tutorials/

After first iteration



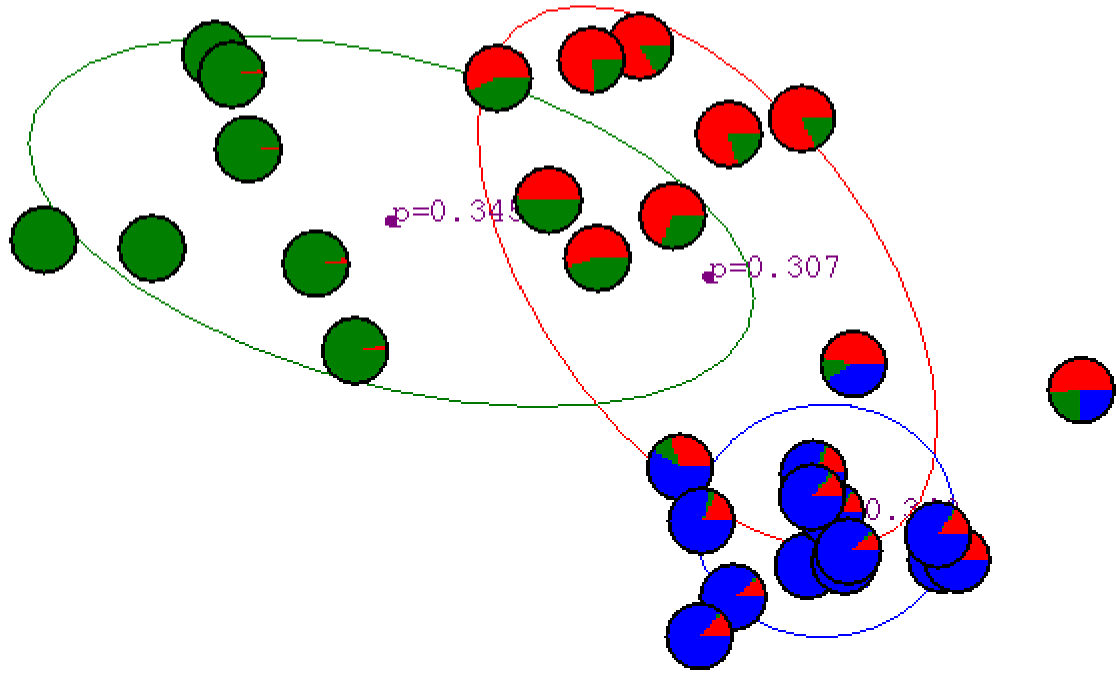
http://www.cs.cmu.edu/~awm/tutorials/

After 2nd iteration



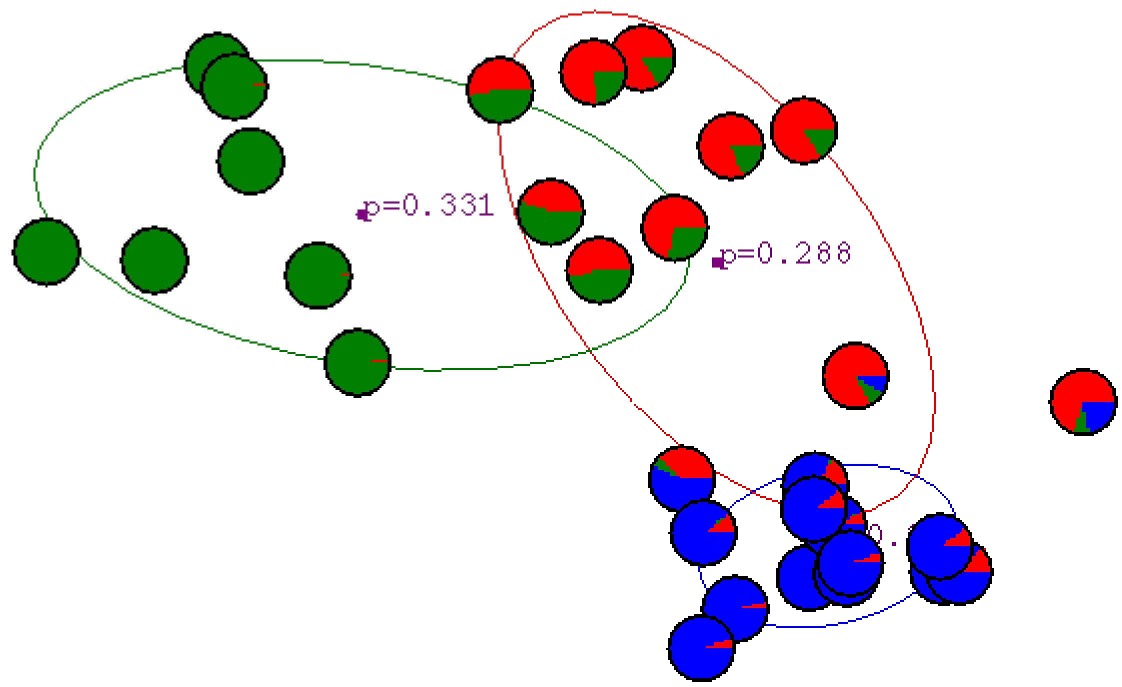
http://www.cs.cmu.edu/~awm/tutorials/

After 3rd iteration



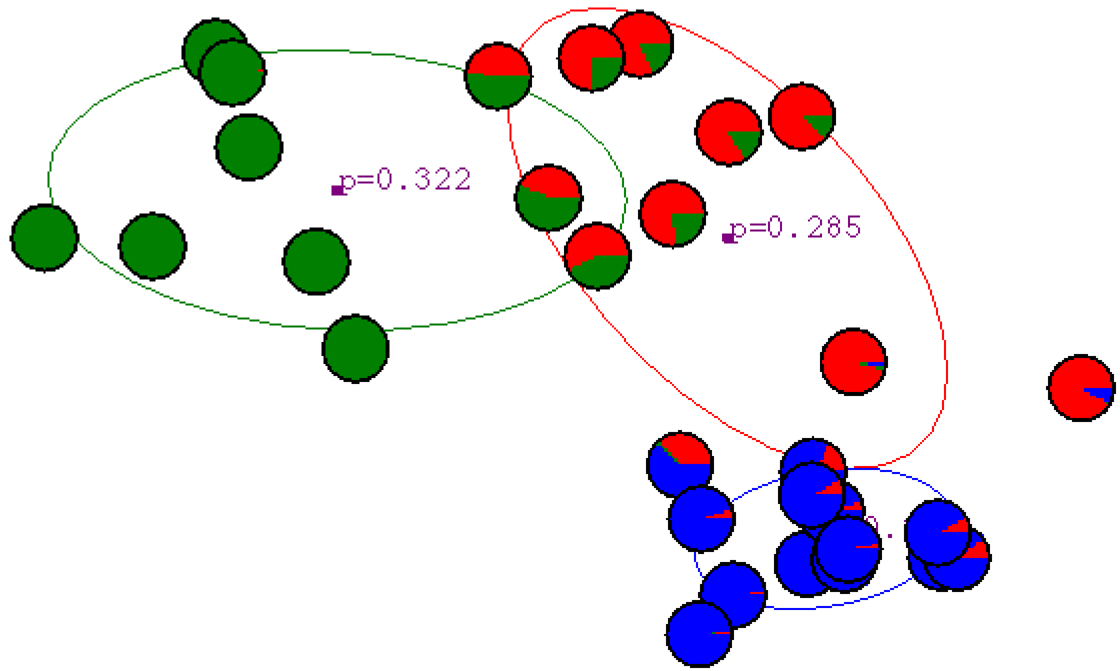
http://www.cs.cmu.edu/~awm/tutorials/

After 4th iteration



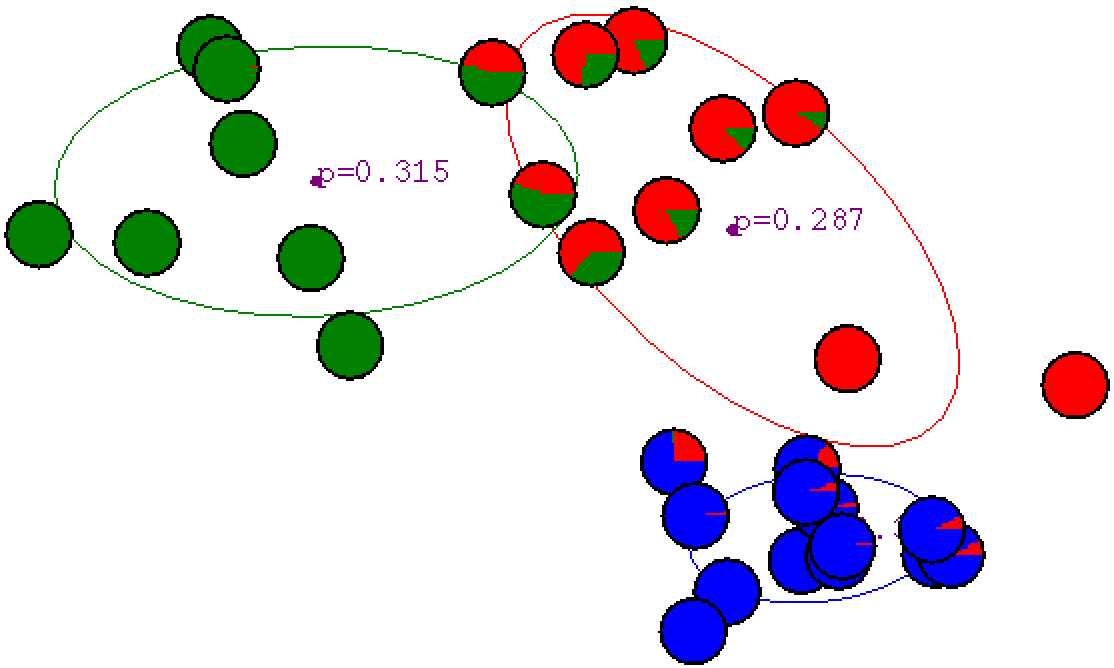
http://www.cs.cmu.edu/~awm/tutorials/

After 5th iteration



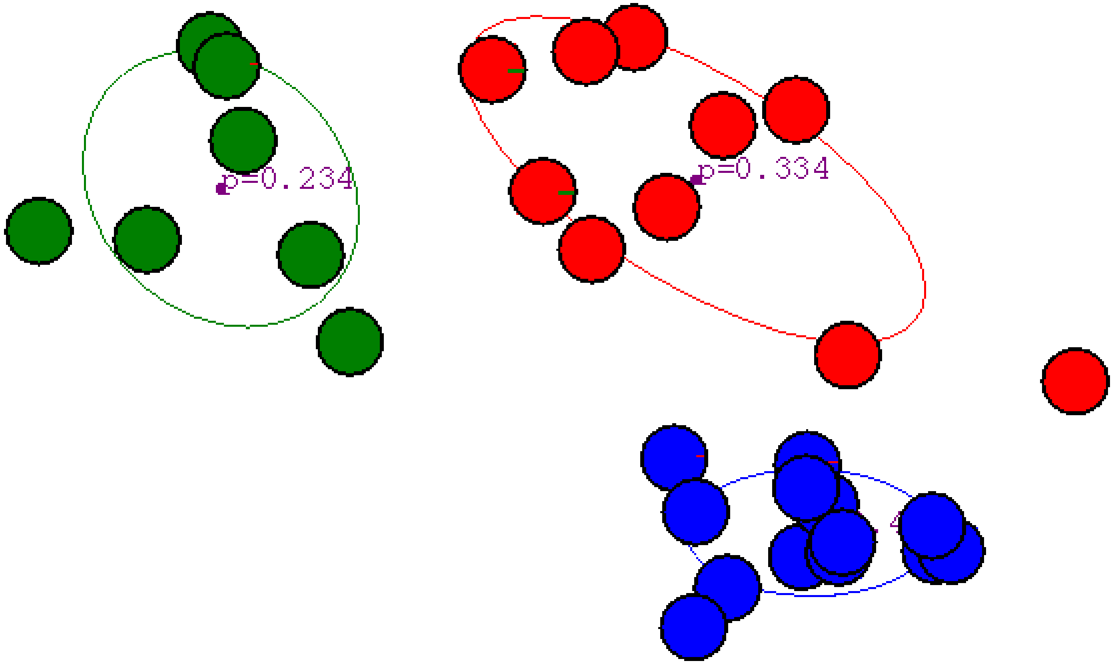
http://www.cs.cmu.edu/~awm/tutorials/

After 6th iteration



http://www.cs.cmu.edu/~awm/tutorials/

After 20th iteration



http://www.cs.cmu.edu/~awm/tutorials/