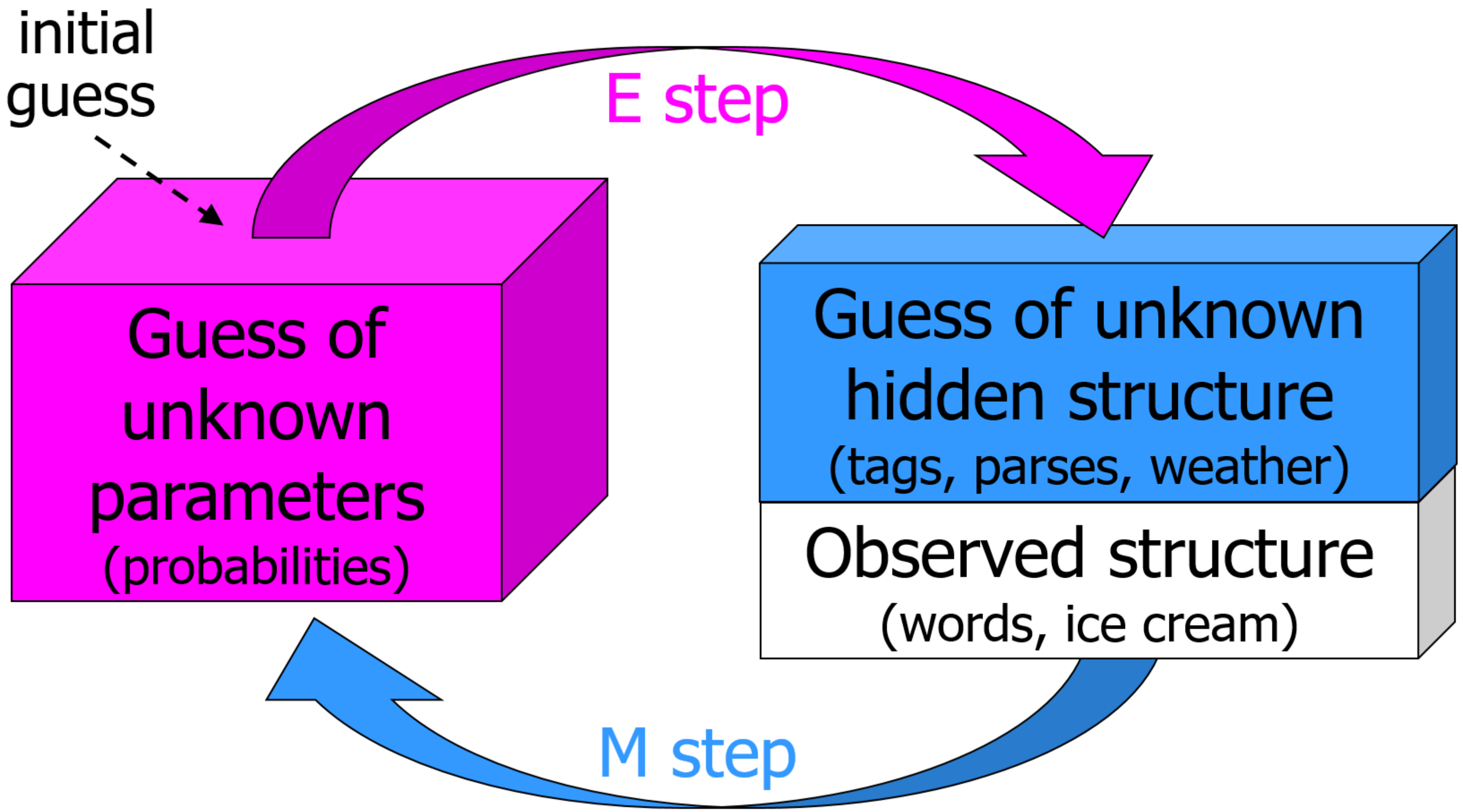
LAB 5 : GMM and EM

Expectation Maximization (EM)

- Iterative method for parameter estimation where you have missing data
- Has two steps: Expectation (E) and Maximization (M)
- Applicable to a wide range of problems
- Old idea (late 50’s) but formalized by Dempster, Laird and Rubin in 1977
- Subject of much investigation. See McLachlan & Krishnan book 1997.

EM

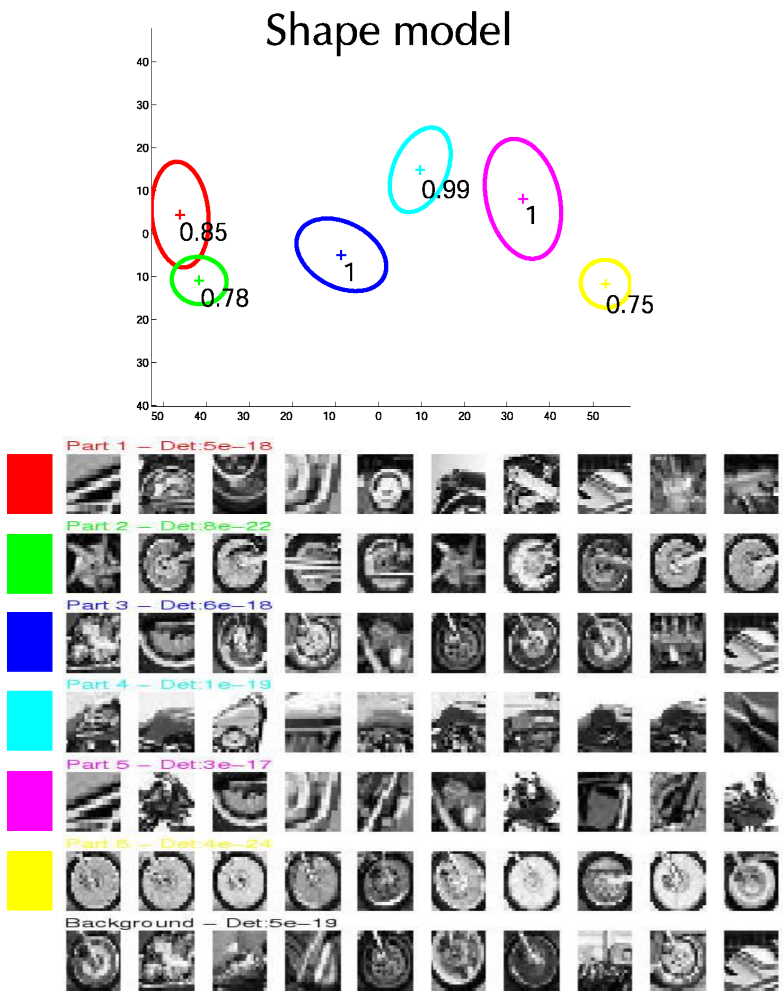

initial guess
E step
Guess of unknown parameters (probabilities)
Guess of unknown hidden structure (tags, parses, weather)
Observed structure (words, ice cream)
M step

Applications of EM

- Automatic segmentation of layers in video
- Probabilistic Latent Semantic Analysis (pLSA)
  - Technique from text community

Applications of EM

- Learning parts and structure models

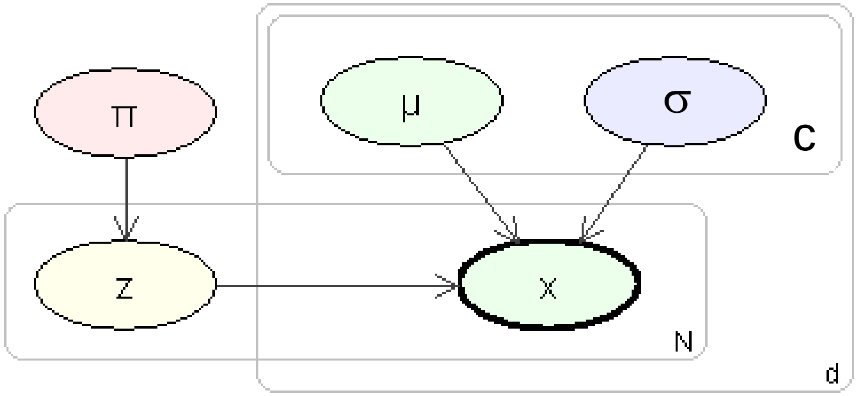


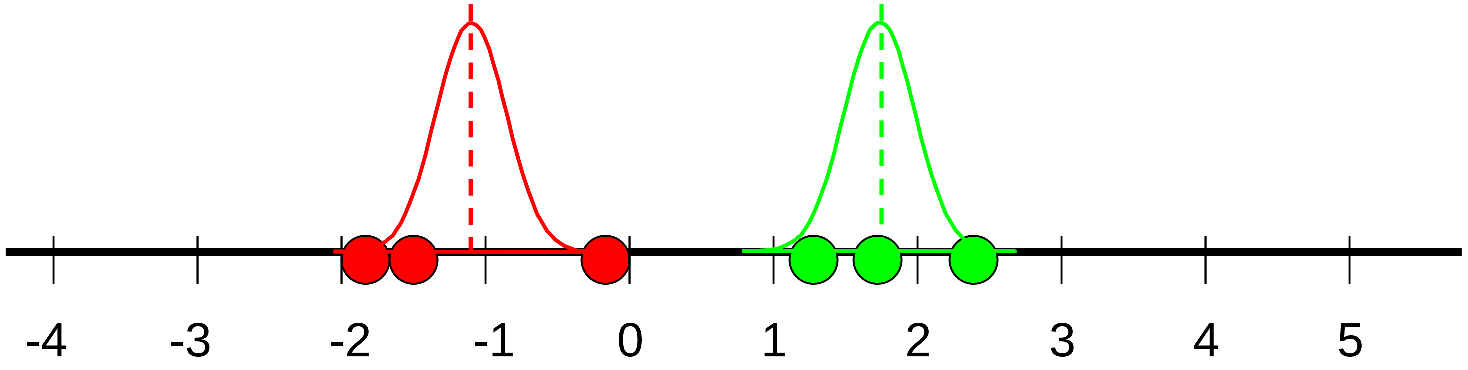
Probabilistic model

Imagine model generating data

Need to introduce label, z, for each data point

Label is called a *latent* variable
also called *hidden, unobserved, missing*



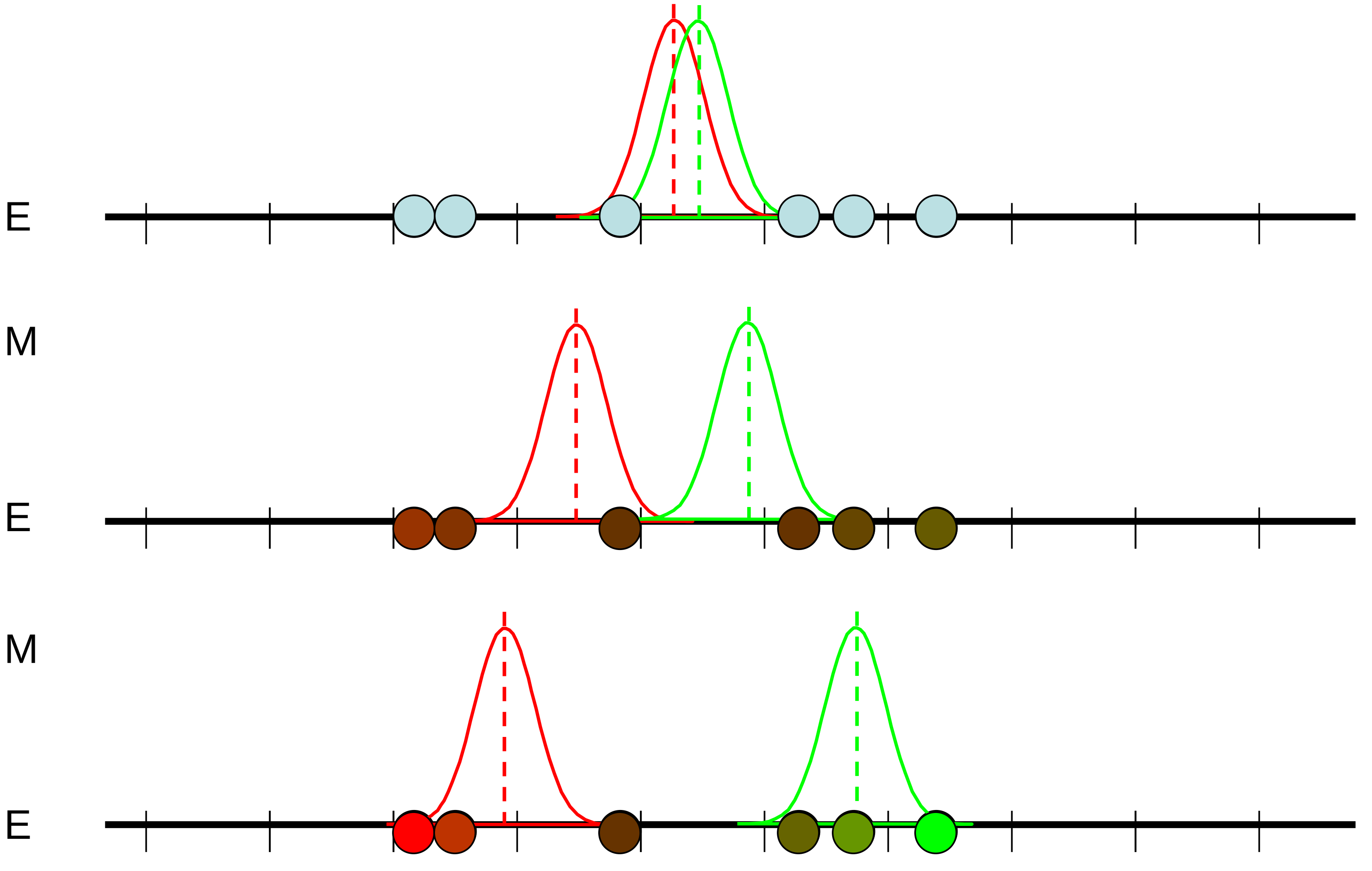


Simplifies the problem:
if we knew the labels, we can decouple the components as estimate parameters separately for each one

Intuition of EM

E-step: Compute a *distribution* on the labels of the points, using current parameters

M-step: Update parameters using current guess of label distribution.



Theory

Repeat until convergence: {

(E-step) For each i, j, set

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

(M-step) Update the parameters:

$$\phi_j := \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)},$$

$$\mu_j := \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}},$$

$$\Sigma_j := \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

}

E-step: Assume our parameters are true. estimate *w*. *w* is Effectively a soft guess for the class labels

M-step: Assume *w* is in fact the correct class label for each *x*

Note that *w* is of length *k* for each data point. That means *w* represents probability of class rather than what is used for something like k-means

Checking for Convergence

- We repeat the expectation and maximization step until we reach “convergence”. But what exactly is convergence? The concept of convergence means that we have a change that is minimal enough for us to consider it to negligible and stop running EM.
- We are fitting trying to fit a GMM to our data, then intuitively **we should have something that measures the fit of our GMM!**
- This called the likelihood and is essentially the fit of your model. What we are asking in layman terms is given model parameters, what is the probability that our data X was generated by them.
- The reason why we do this is because if we simply calculate the likelihood we would end up dealing with very small values which can be problematic.
- So, we take the natural logarithm of the likelihood to circumvent this.
- *The larger the log likelihood = Better the model parameters fit the data*
- So, to test for convergence, we can calculate the log likelihood at the end of each EM step (i.e. model fit with these parameters) and then test whether it has changed “significantly” (defined by the user) from the last EM step. If it has, then we repeat another step of EM. If not, then we consider that EM has converged and then these are our final parameters.

What EM won’t do

Pick structure of model
 # components
 graph structure

Find global maximum

Always have nice closed-form updates
 optimize within E/M step

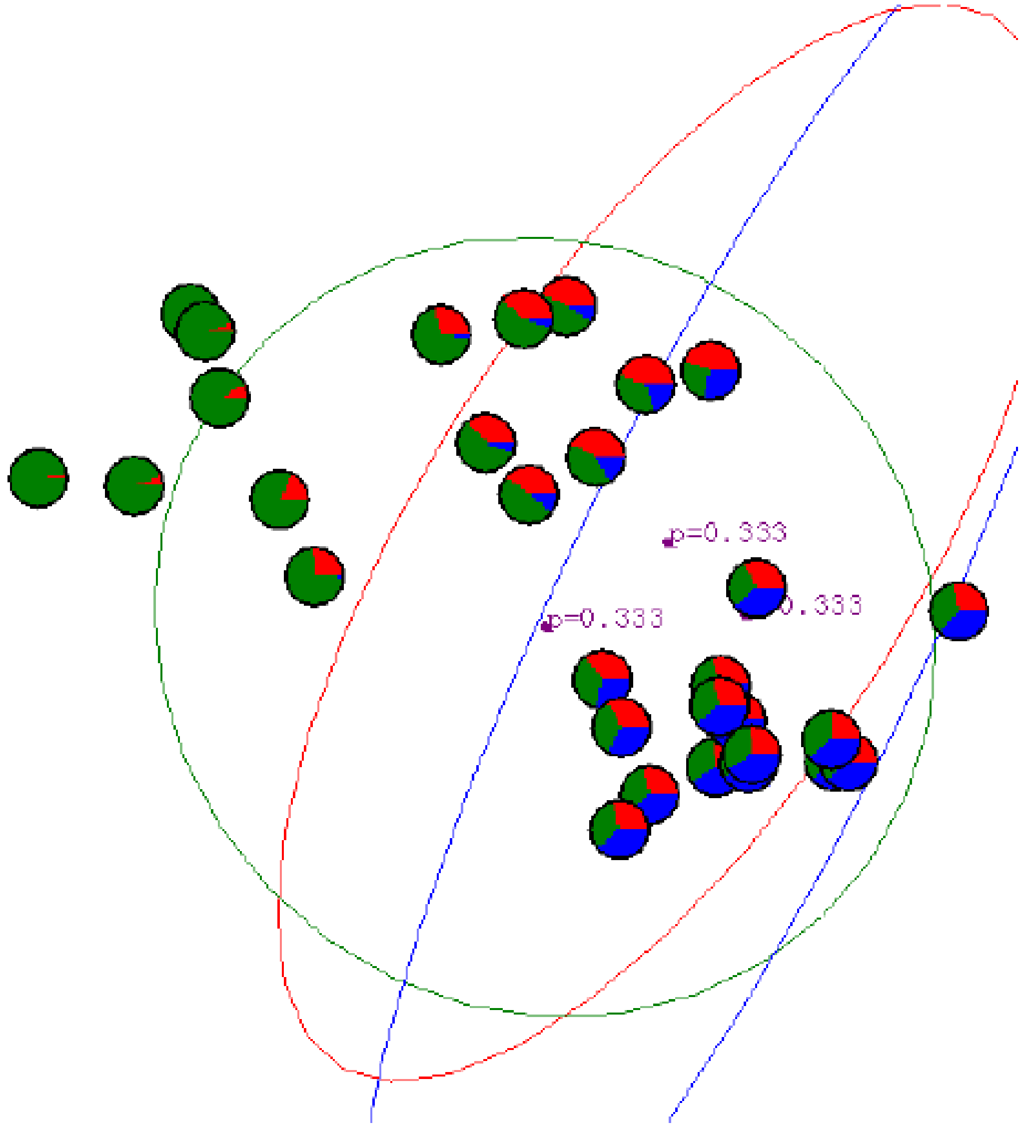
Avoid computational problems
 sampling methods for computing expectations

Why not use standard optimization methods such as gradient descent or Newton's method?

In favor of EM:

- No step size
- Works directly in parameter space model, thus parameter constraints are obeyed
- Fits naturally into graphically model frame work
- Supposedly faster

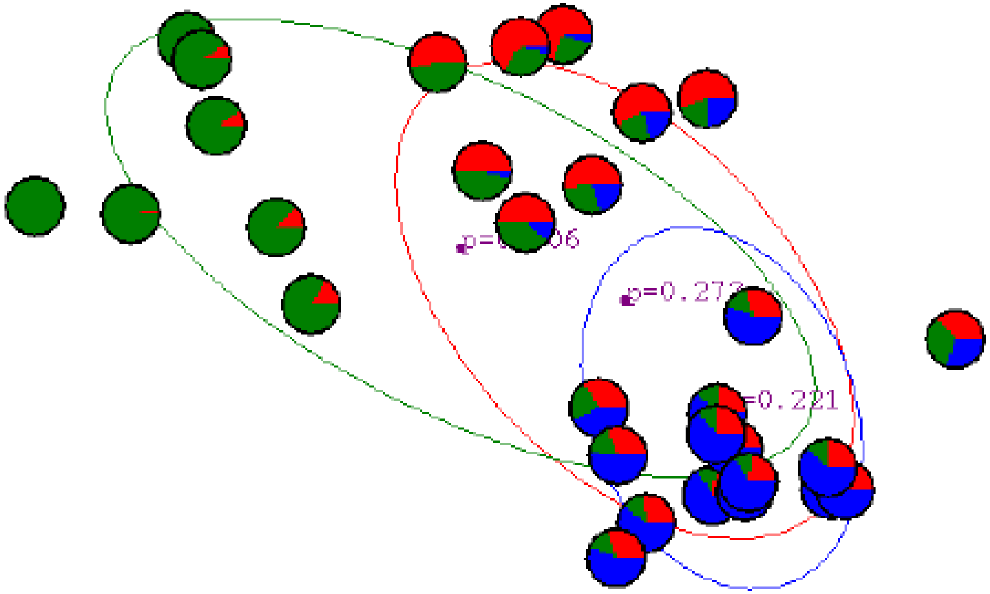
Gaussian Mixture Example: Start



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

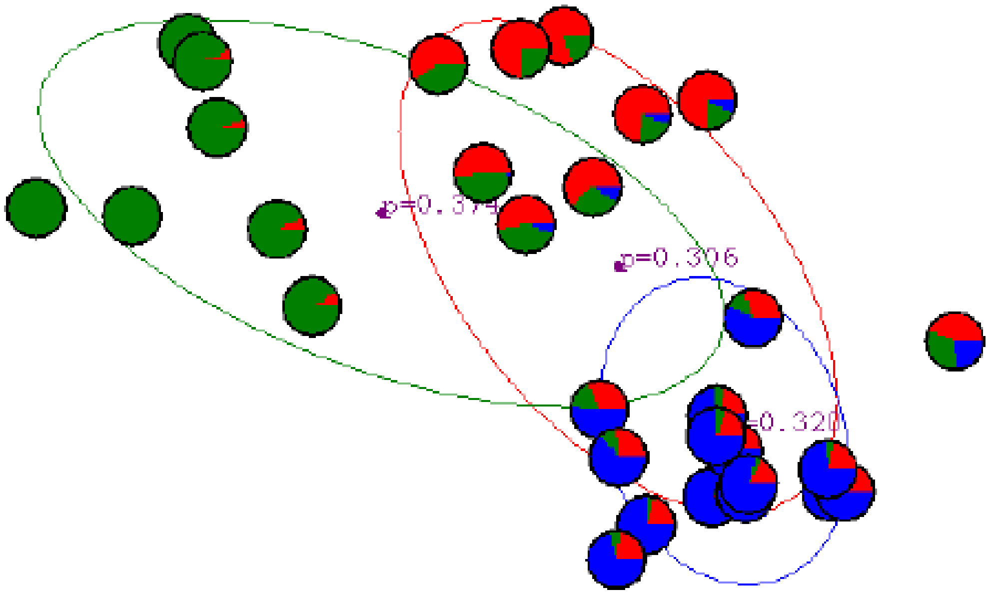
Most slides from
http://www.autonlab.org/tutorials/

After first iteration



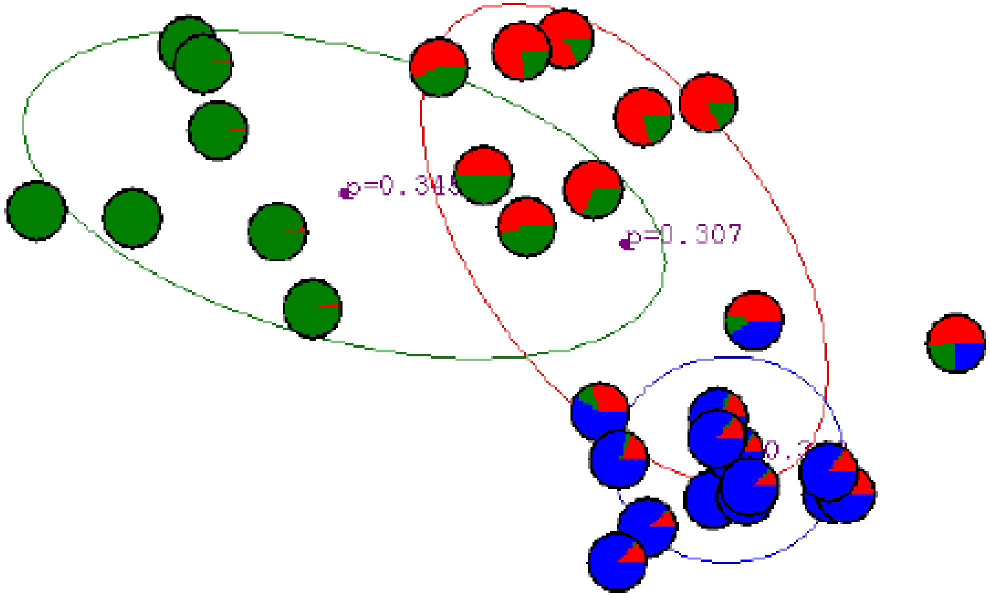
Most slides from
http://www.autonlab.org/tutorials/

After 2nd iteration



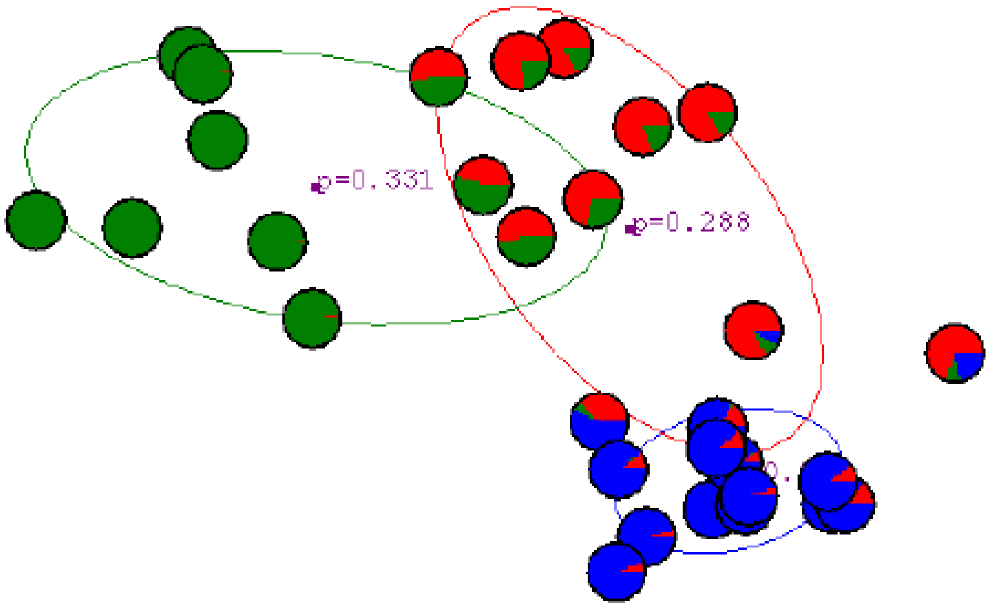
Most slides from
http://www.autonlab.org/tutorials/

After 3rd iteration



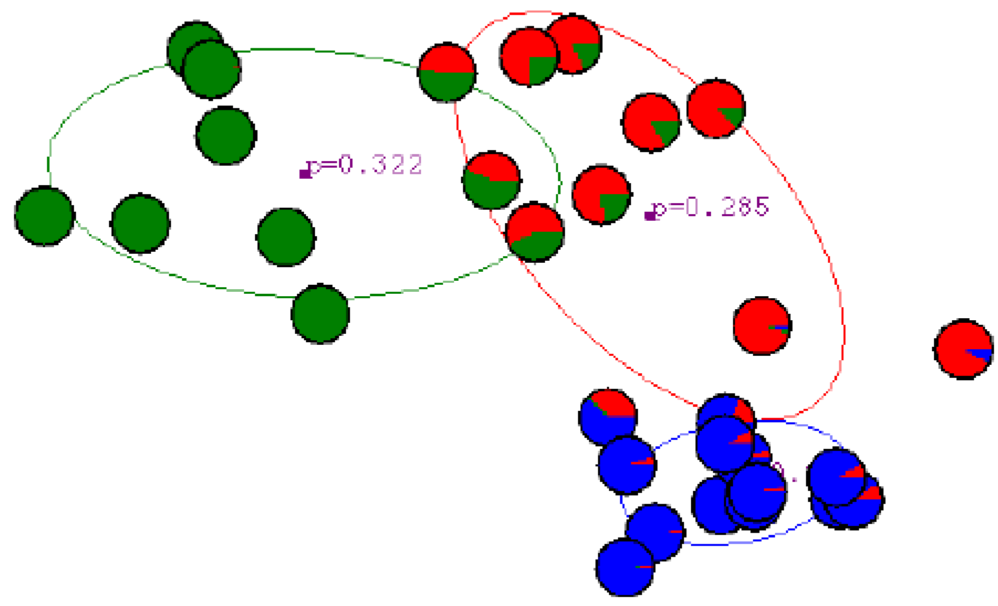
Most slides from
http://www.autonlab.org/tutorials/

After 4th iteration



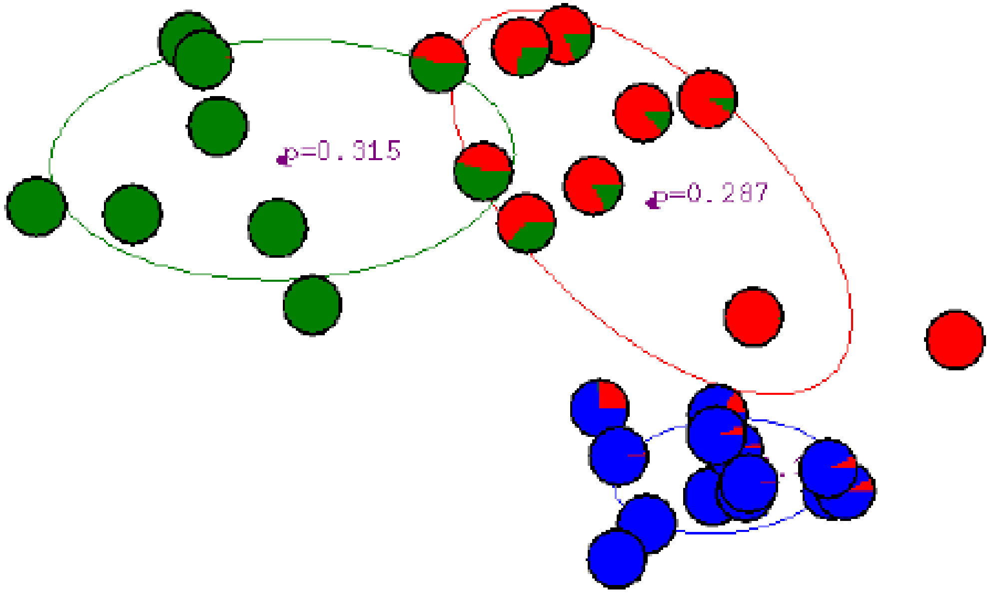
Most slides from
http://www.autonlab.org/tutorials/

After 5th iteration



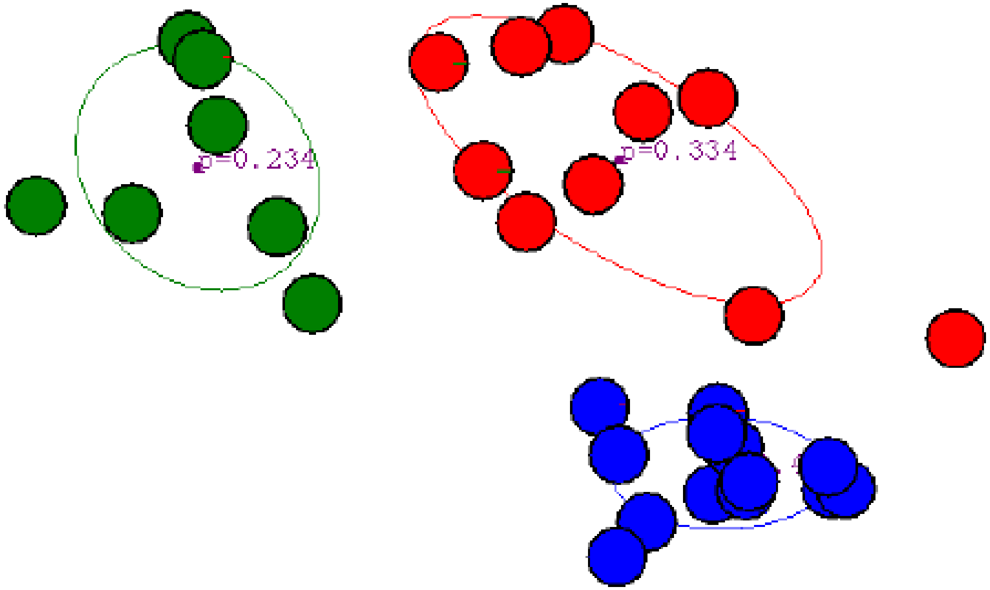
Most slides from
http://www.autonlab.org/tutorials/

After 6th iteration



Most slides from
http://www.autonlab.org/tutorials/

After 20th iteration



Most slides from
http://www.autonlab.org/tutorials/

References

- www.cs.northwestern.edu/~ddowney/courses/395_Winter2010/em.ppt
- people.csail.mit.edu/fergus/research/tutorial_em.ppt
- http://tinyheero.github.io/2016/01/03/gmm-em.html
- https://www.cs.jhu.edu/~jason/465/PowerPoint/lect26-em.ppt