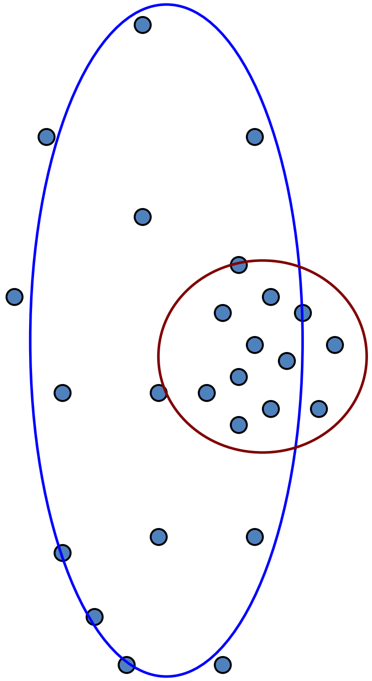
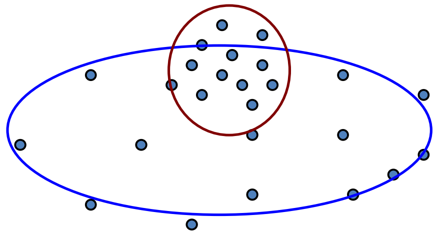
# CSE 446
# Expectation Maximization

# (One) bad case for "hard assignments"?



- Clusters may overlap
- Some clusters may be "wider" than others
- Distances can be deceiving!

# Probabilistic Clustering



- We can use a probabilistic model!
  - allows overlaps, clusters of different size, etc.
- Can tell a *generative story* for data
  - P(X|Y) P(Y) is common
- Challenge: we need to estimate model parameters without labeled Ys

| Y | $X_1$ | $X_2$ |
|---|---|---|
| ?? | 0.1 | 2.1 |
| ?? | 0.5 | -1.1 |
| ?? | 0.0 | 3.0 |
| ?? | -0.1 | -2.0 |
| ?? | 0.2 | 1.5 |
| … | … | … |

# What Model Should We Use?

- Depends on X!
- Here, maybe Gaussian Naïve Bayes?
  - Multinomial over clusters Y, Gaussian over each $X_i$ given Y

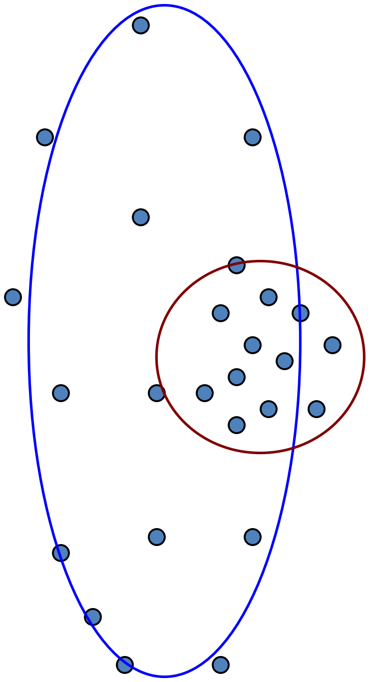
$$p(Y_i = y_k) = \theta_k$$

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

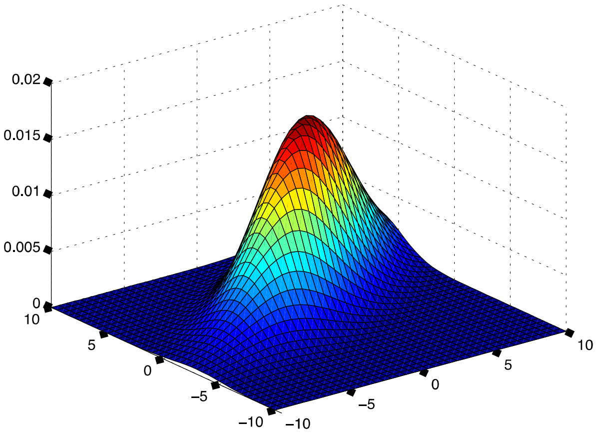
| Y | $X_1$ | $X_2$ |
|---|---|---|
| ?? | 0.1 | 2.1 |
| ?? | 0.5 | -1.1 |
| ?? | 0.0 | 3.0 |
| ?? | -0.1 | -2.0 |
| ?? | 0.2 | 1.5 |
| … | … | … |

# Geometric Interpretation

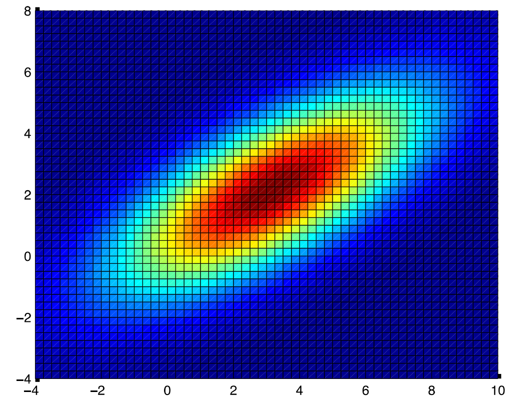
$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$



# What if the Clusters are not Axis-Aligned?



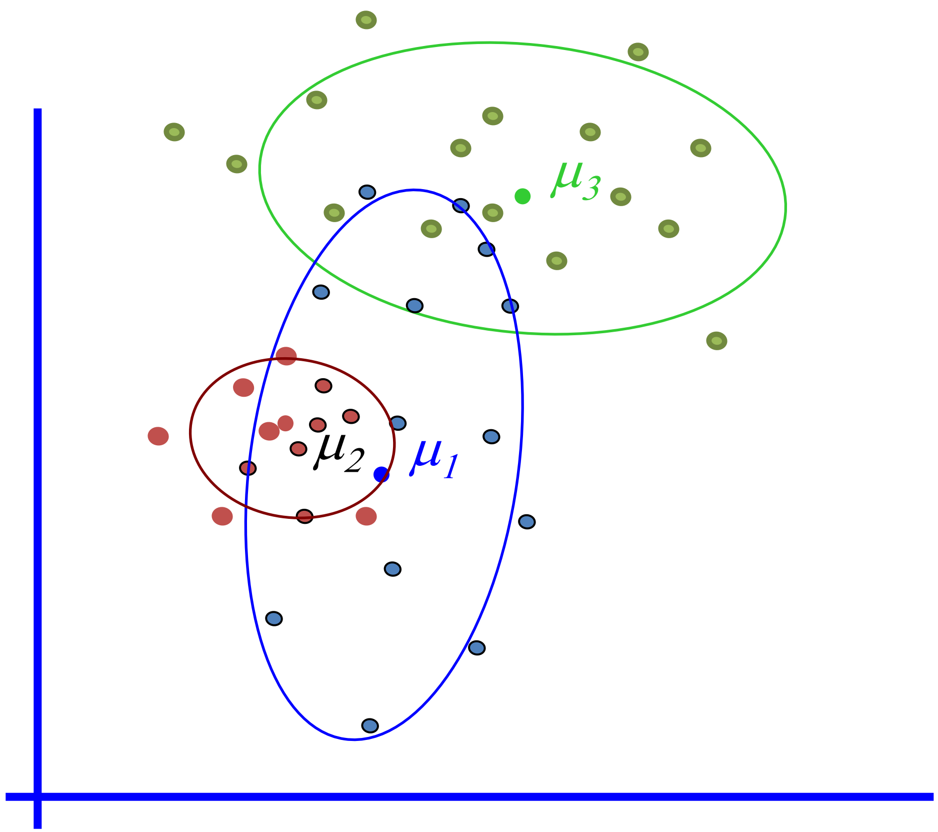

- What if the input dimensions $X_i$ co-vary
- Gaussian Mixture Models
  - Assume m-dimensional data points
  - P(Y) still multinomial, with K classes
  - P(**X**|Y=i), i=1..K are K multivariate Gaussians
    - mean $\mu_i$ is m-dimensional vector
    - variance $\Sigma_i$ is m by m matrix
    - |x| is the determinate of matrix x




$$P(X = x|Y = i) = \frac{1}{\sqrt{(2\pi)^m |\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right)$$

# Multivariate Gaussians

$$P(X = x|Y = k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left(-\frac{1}{2}(x - \mu_k)^T\Sigma_k^{-1}(x - \mu_k)\right)$$



# Multivariate Gaussians: MLE

$$P(X=x) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\Sigma_{j,j} = \frac{1}{N}\sum_{i=1}^{N}(x_{i,j}-\mu_j)^2$$

$$\Sigma_{j,k} = \frac{1}{N}\sum_{i=1}^{N}(x_{i,j}-\mu_j)\cdot(x_{i,k}-\mu_k)$$

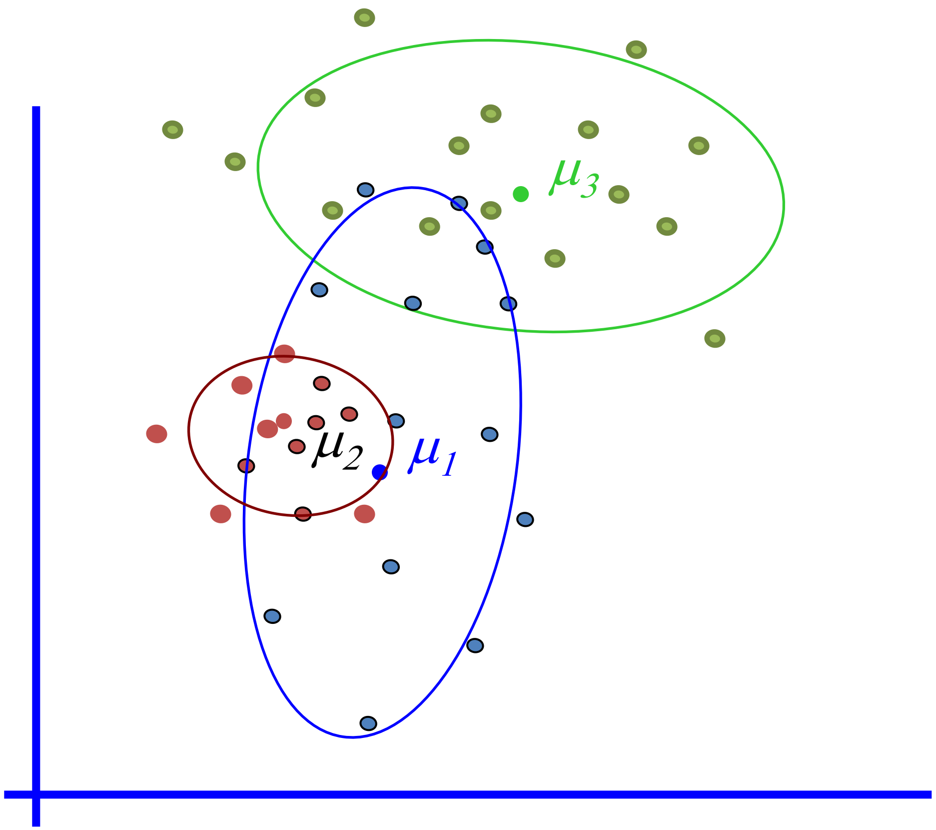
$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T$$

# The General GMM assumption

- P(Y): There are k components
- P(X|Y): Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\Sigma_i$

Each data point is sampled from a *generative process*:

1. Pick a component at random: Choose component i with probability *P(y=i)*
2. Datapoint ~ N($\mu_i$, $\Sigma_i$ )



# Gaussian Mixture Model: MLE

$$P(X=x|Y=k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

single Gaussian

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T$$

Gaussian mixture

$$\mu_k = \frac{\sum_{i:y_i=k} x_i}{\mathrm{Count}(y_i=k)}$$

$$\Sigma_k = \frac{\sum_{i:y_i=k}(x_i-\mu_k)(x_i-\mu_k)^T}{\mathrm{Count}(y_i=k)}$$

# Missing Labels

- Problem: the labels y are unknown!
- If we **already** have a trained model, recovering y is just inference.

$$P(Y_i = k | X_i = x) = \frac{P(Y_i = k, X_i = x)}{P(X_i = x)} = \frac{P(X_i = x | Y_i = k) P(Y_i = k)}{P(X_i = x)}$$

$$\propto P(X_i = x | Y_i = k) P(Y_i = k)$$

$$\tilde{w}_{ik} = P(X_i = x | Y_i = k) P(Y_i = k)$$

$$w_{ik} = \frac{\tilde{w}_{ik}}{\sum_{k'=1}^{K} \tilde{w}_{ik'}}$$

# Weighted MLE

- If we have label probabilities, can we refit the model?

$$\tilde{w}_{ik} = P(X_i = x | Y_i = k) P(Y_i = k)$$

$$w_{ik} = \frac{\tilde{w}_{ik}}{\sum_{k'=1}^{K} \tilde{w}_{ik'}}$$

$$\mu_k = \frac{\sum_{i: y_i = k} x_i}{\text{Count}(y_i = k)}$$

$$\mu_k = \frac{\sum_{i=1}^{N} w_{ik} x_i}{\sum_{i=1}^{N} w_{ik}}$$

$$\Sigma_k = \frac{\sum_{i: y_i = k} (x_i - \mu_k)(x_i - \mu_k)^T}{\text{Count}(y_i = k)}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} (x_i - \mu_k)(x_i - \mu_k)^T w_{ik}}{\sum_{i=1}^{N} w_{ik}}$$

# EM Algorithm

- **E**xpectation **M**aximization
- E-step: figure out the probabilities of each label y given the current Gaussians
- M-step: figure out the Gaussian parameters given the probabilities of each label y
- Sound familiar?

**Algorithm 1** EM clustering

Initialize means and covariances (more on this later)
**while** not converged **do**
    E-step: estimate $w_{ik}$ for each datapoint $i$ and each cluster $k$
    M-step: fit $\mu_k$ and $\Sigma_k$ using the weighted MLE fit
**end while**

# EM vs K-Means

- "Hard" **E**xpectation **M**aximization = K-Means
- E-step: figure out the probabilities of each label y given the current Gaussians, clamp to 0 or 1
- M-step: figure out the Gaussian parameters given the probabilities of each label y
- Exactly K-means if we fit only means and set covariances to identity matrix!
- Viewed another way: EM = "soft" k-means!

**Algorithm 2** Hard EM clustering (k-means)

```
Initialize means and covariances (more on this later)
while not converged do
    E-step: estimate y_i = arg max_k w_ik for each datapoint i
    M-step: fit μ_k using the weighted MLE fit, set Σ_k = I
end while
```

Line 3 (E-step): $y_i = \arg\max_k w_{ik}$ for each datapoint $i$. Line 4 (M-step): fit $\mu_k$ using the weighted MLE fit, set $\Sigma_k = \mathbf{I}$.

# What is the objective we want?

- Maximize the probability of the points

$$\mathcal{L} = \prod_{i=1}^{N} p(\mathbf{x}_i)$$

- But we believe the probability of the points depends on the unknown labels, so we marginalize out the unknown labels…

$$\mathcal{L} = \prod_{i=1}^{N} p(\mathbf{x}_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(y_i = k, \mathbf{x}_i)$$

- Which is equal to…

$$\mathcal{L} = \prod_{i=1}^{N} p(\mathbf{x}_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(y_i = k, \mathbf{x}_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(y_i = k) p(\mathbf{x}_i | y_i = k).$$

# What does EM optimize?

- Expected log-likelihood:

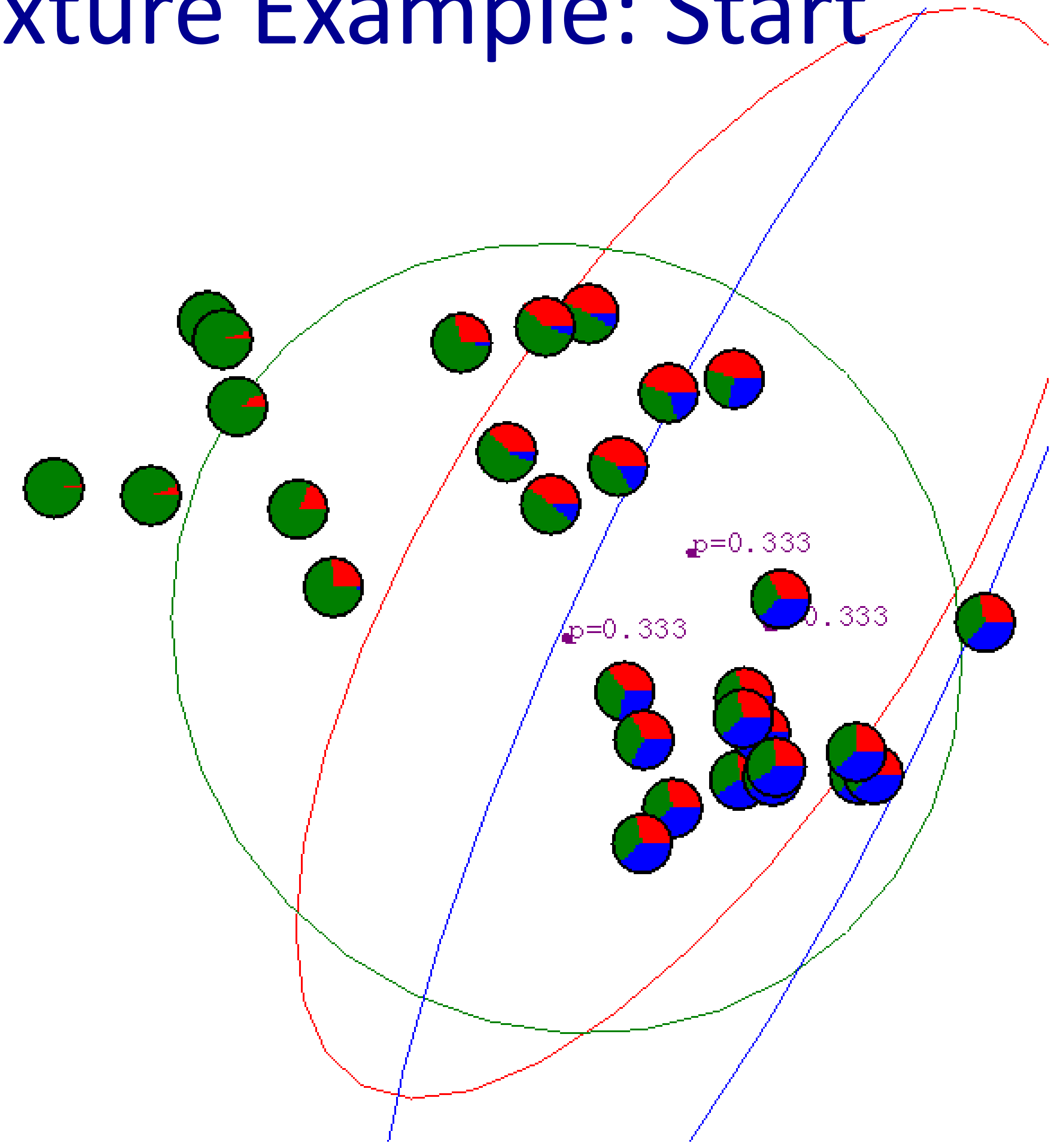
$$\hat{\mathcal{L}} = \sum_{i=1}^{N} \sum_{k=1}^{K} q(y_i = k|\mathbf{x}_i) \log p(y_i = k, \mathbf{x}_i) = \sum_{i=1}^{M} E_q[\log p(y_i = k, \mathbf{x}_i)],$$

- M-step: optimize expected log-likelihood
- E-step: make expected log-likelihood more like the actual likelihood by changing q
- Will see connection to marginal likelihood later

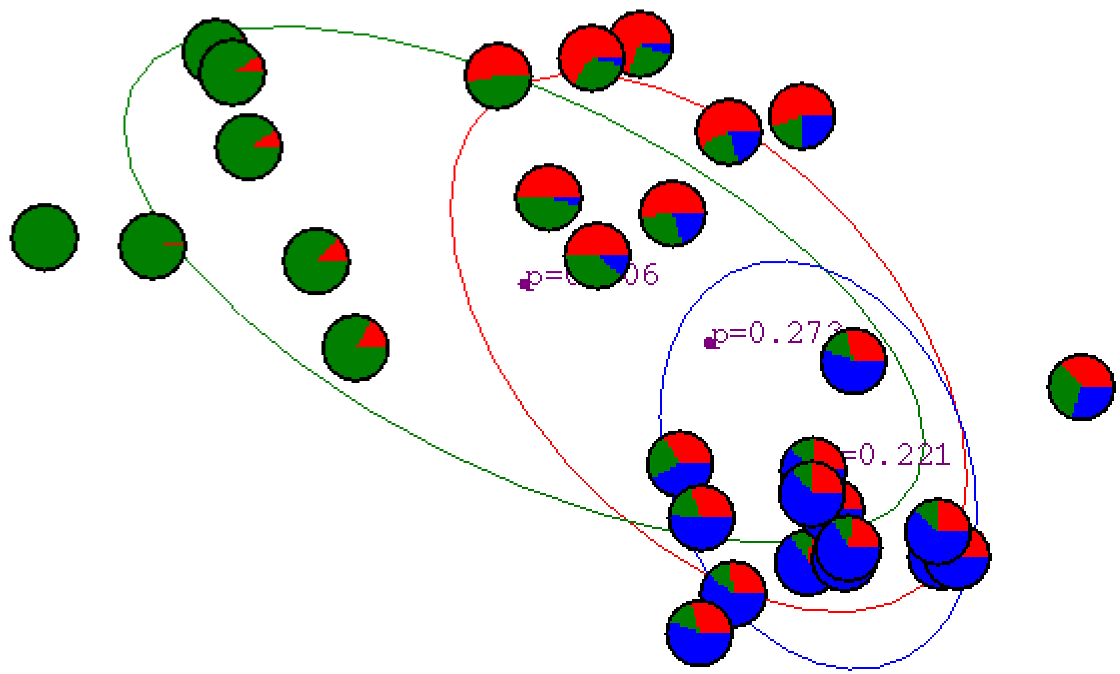
# EM in Practice

- Avoiding getting stuck
  - Random restarts
  - Take restart with best objective value (expected likelihood)
- Initialization
  - Random assignments:
    - Assign points to clusters at random (choose random yi)
    - Compute initial mean and covariance for each cluster for random assignment
  - Random means
    - Set the means to be randomly chosen datapoints
    - Set covariances to be identity
  - Use k-means

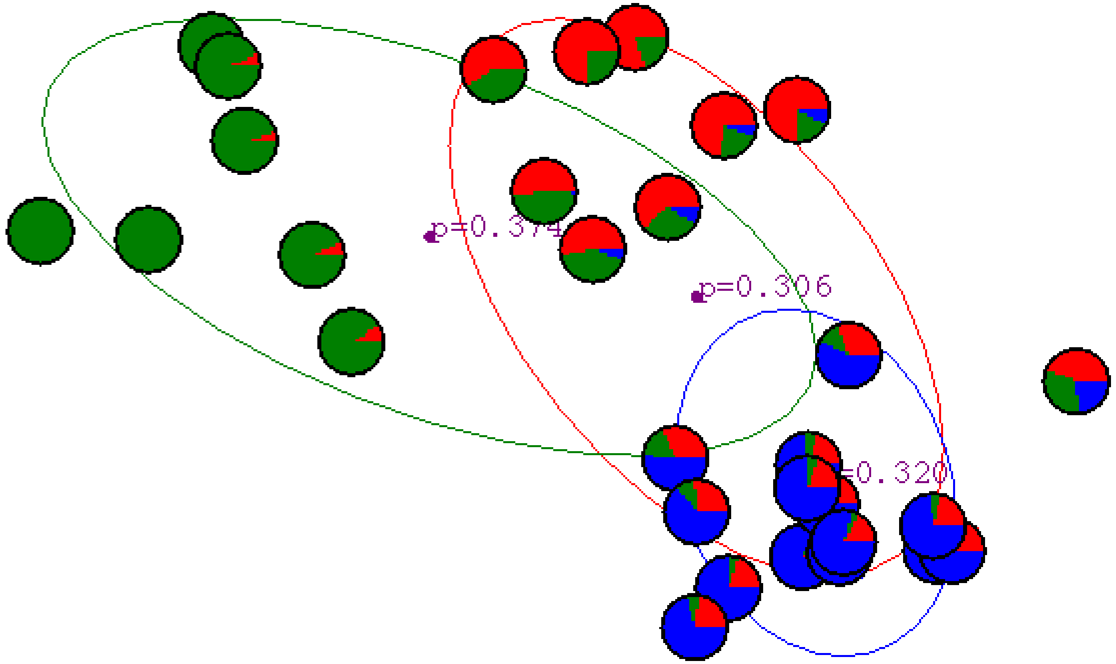
# Gaussian Mixture Example: Start



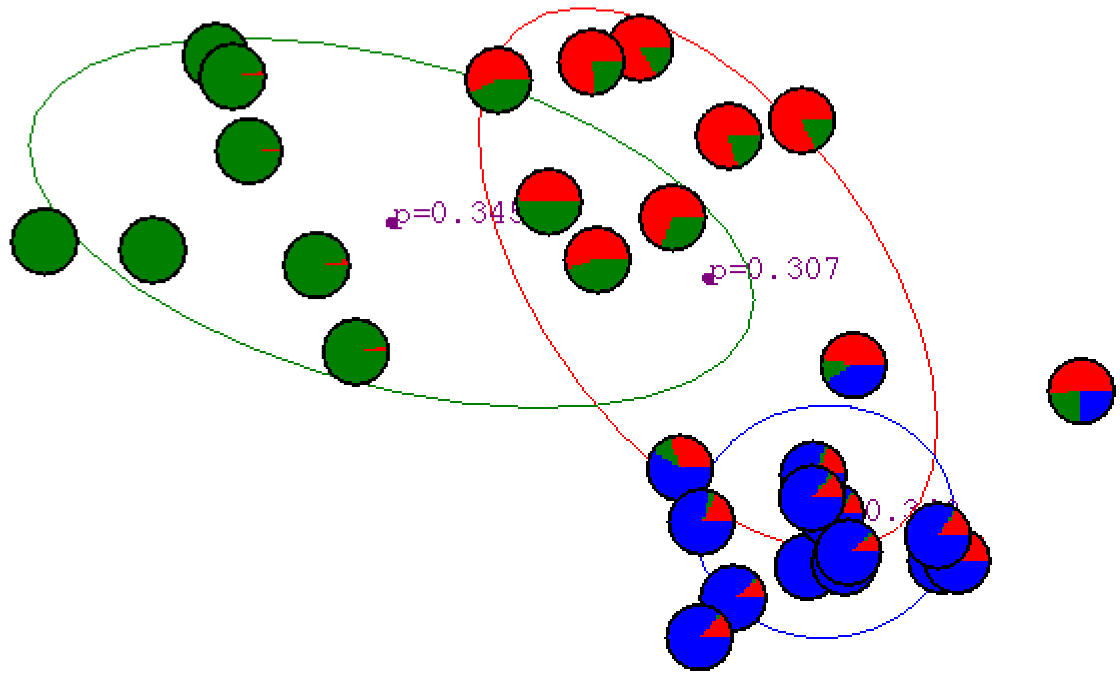
# After first iteration



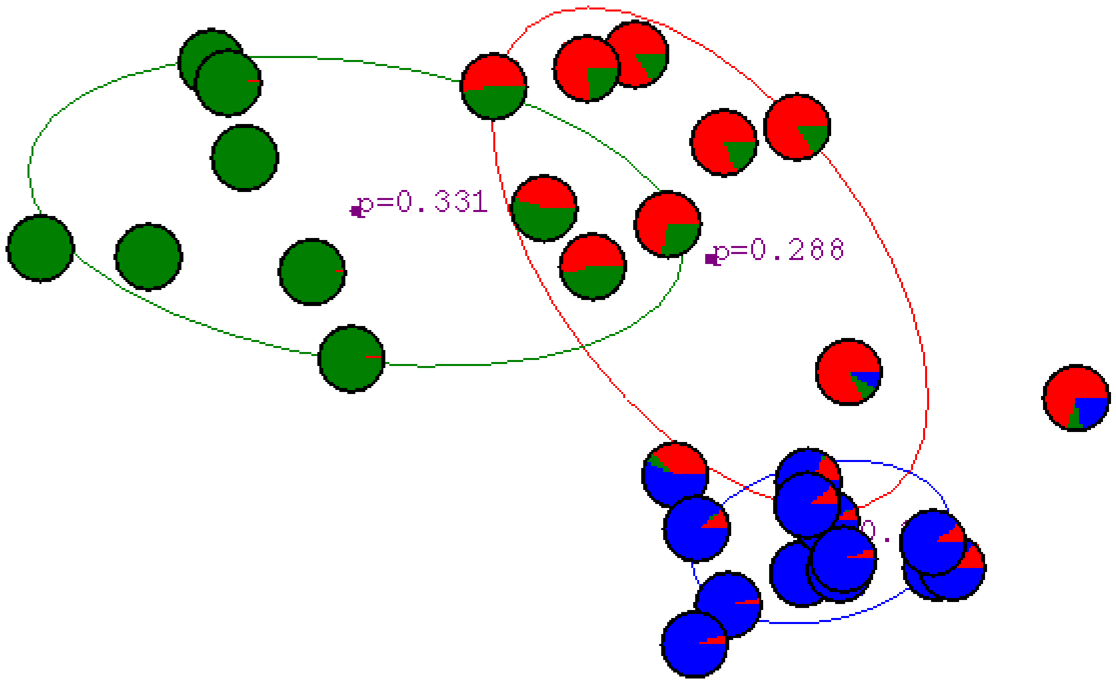
# After 2nd iteration



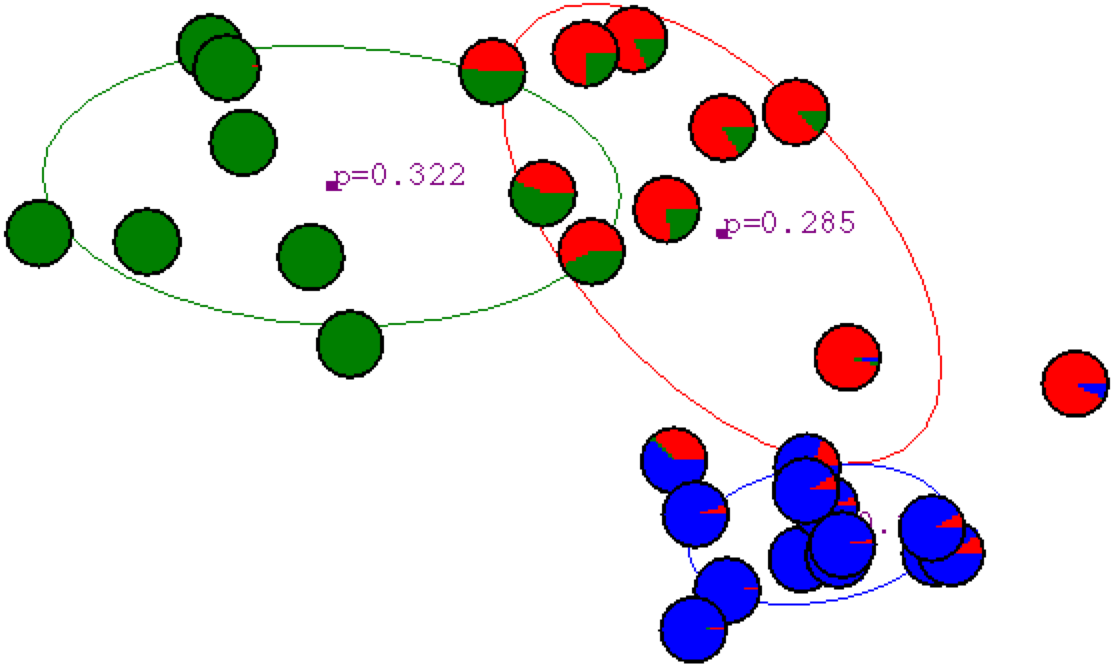
# After 3rd iteration



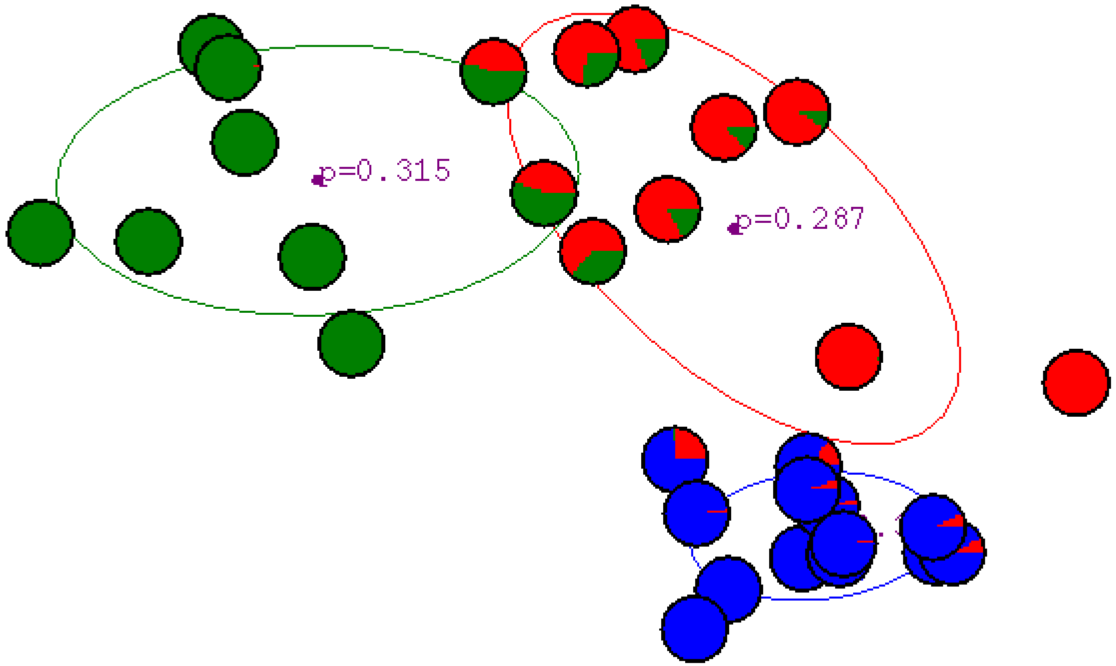
# After 4th iteration



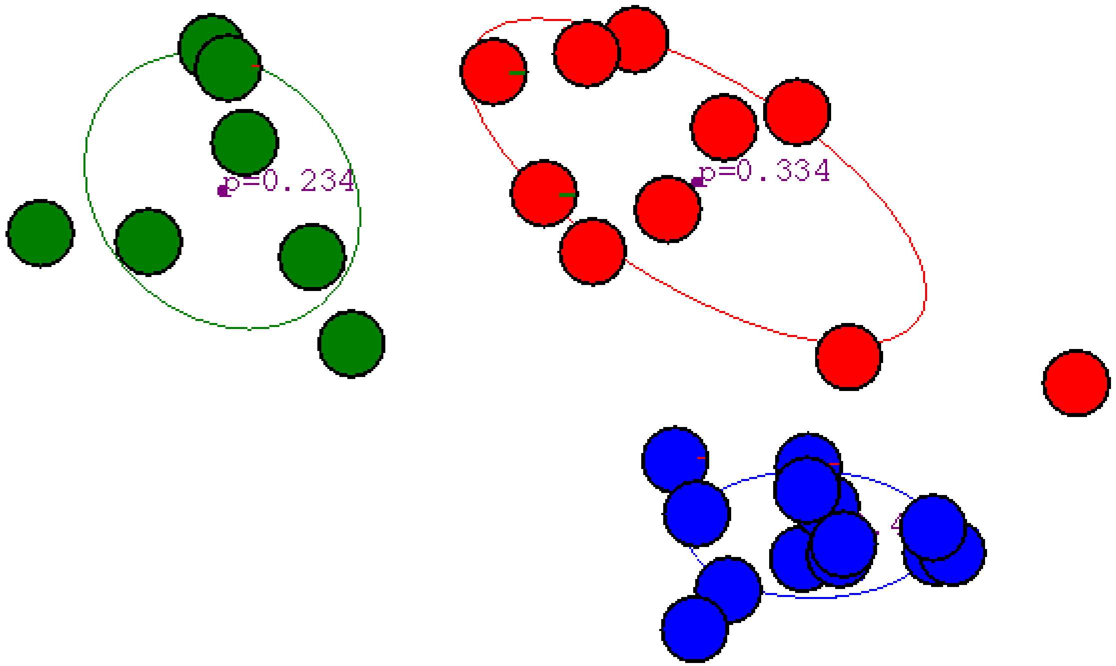
# After 5th iteration



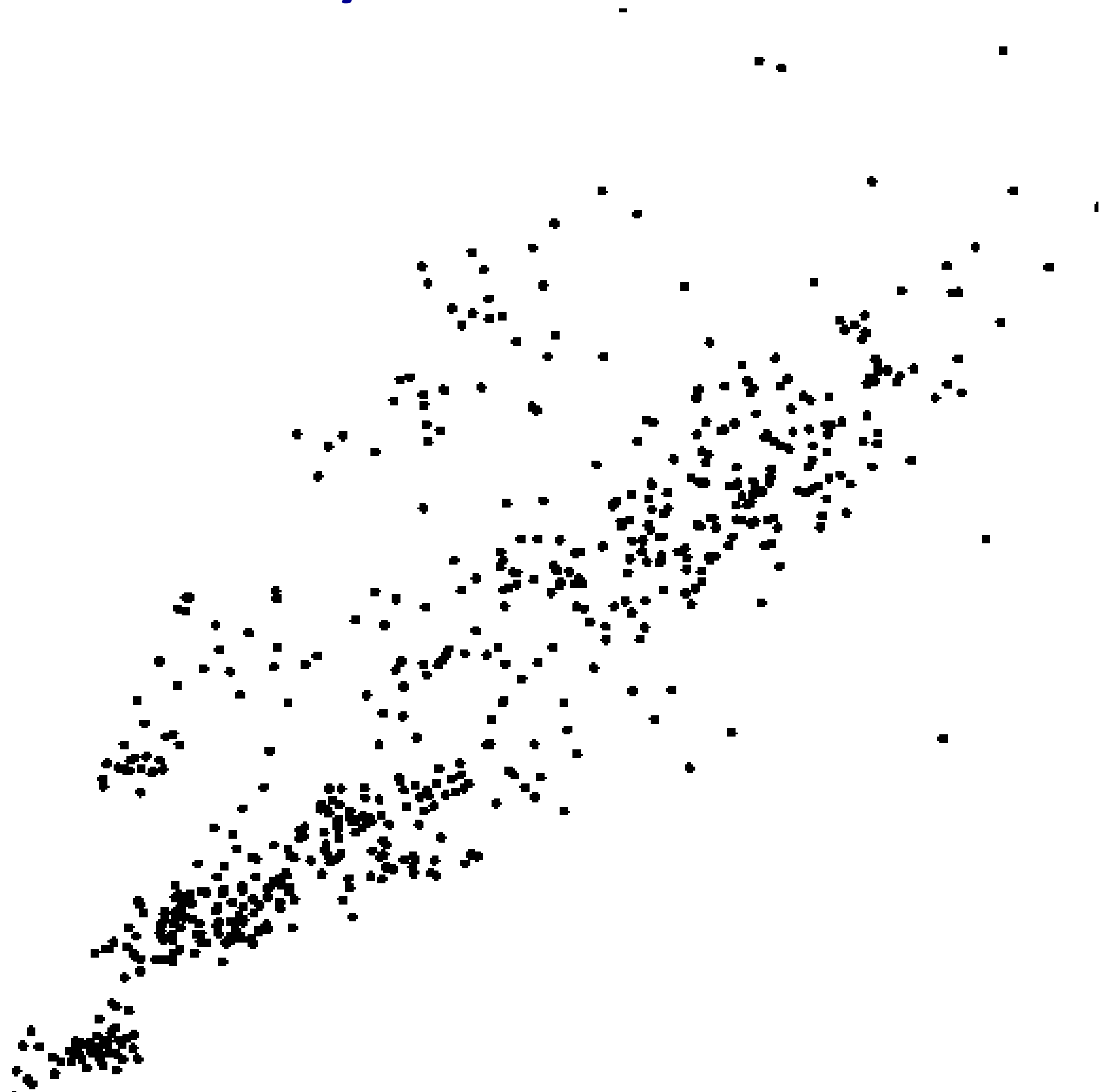
# After 6th iteration



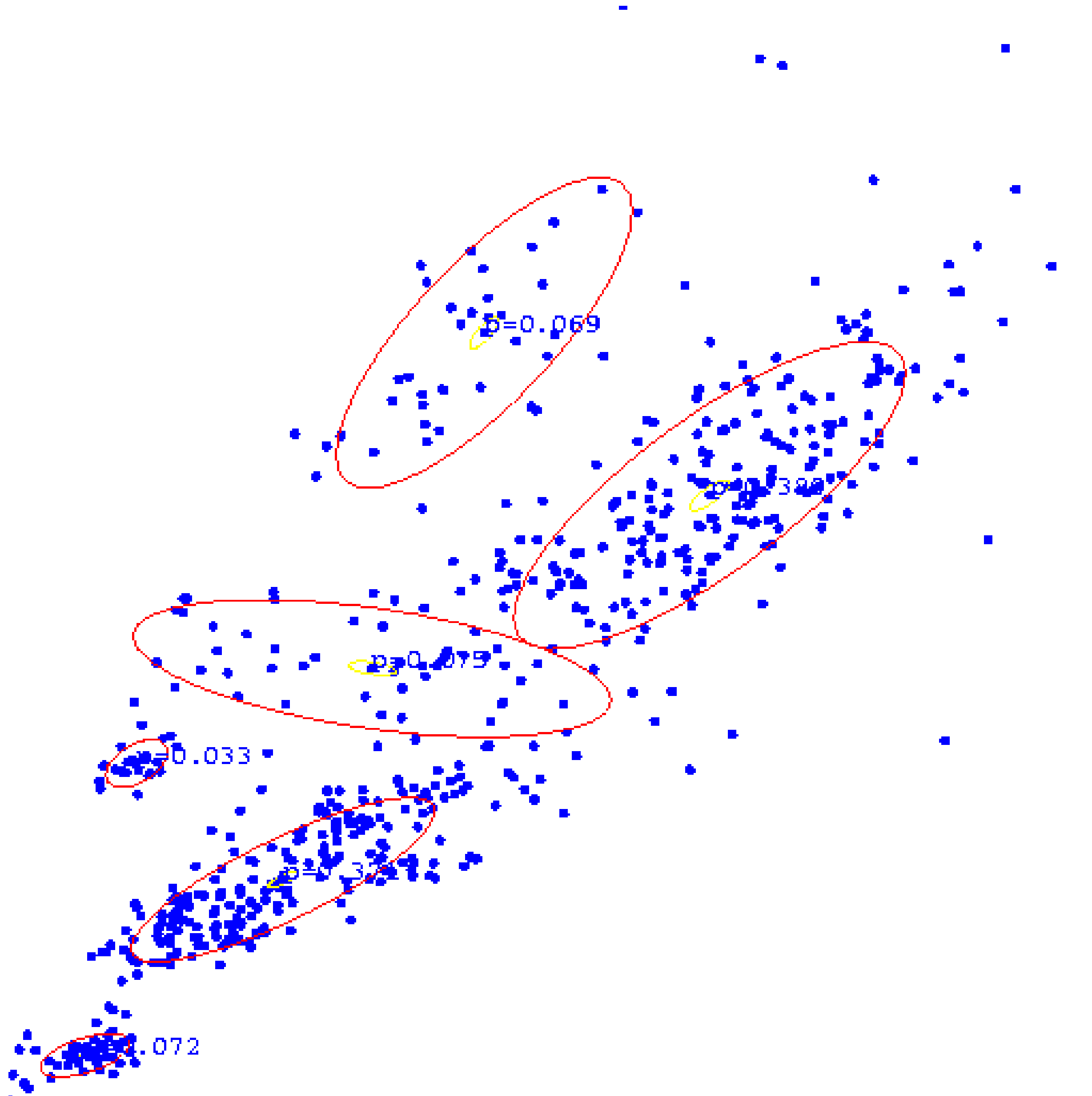
# After 20th iteration



# Some Bio Assay data



# GMM clustering of the assay data



# Resulting Density Estimator