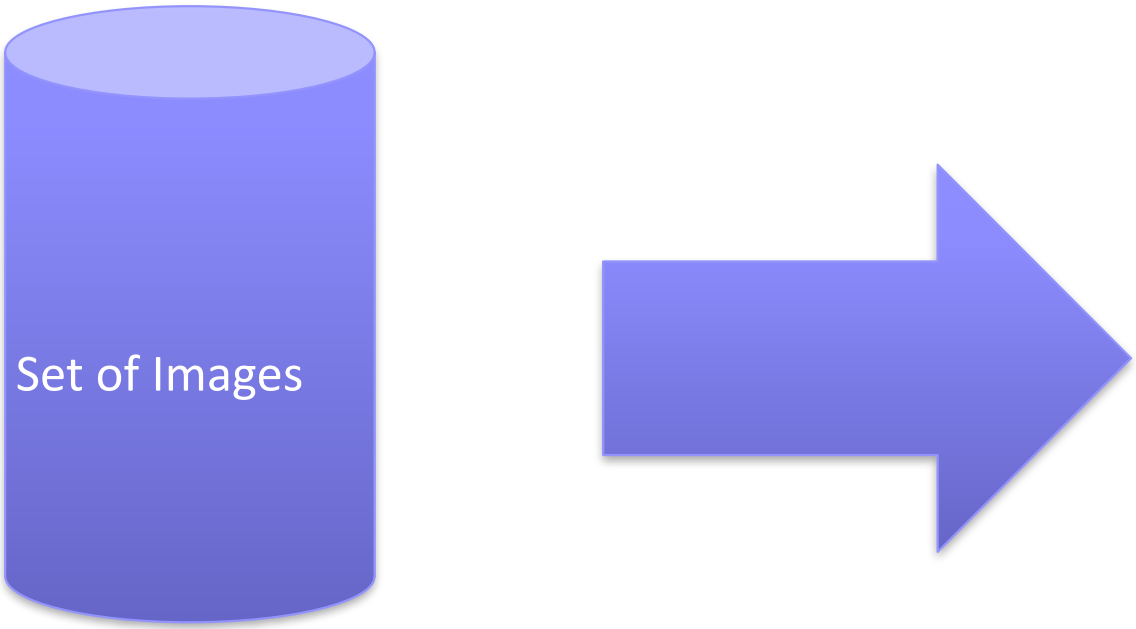
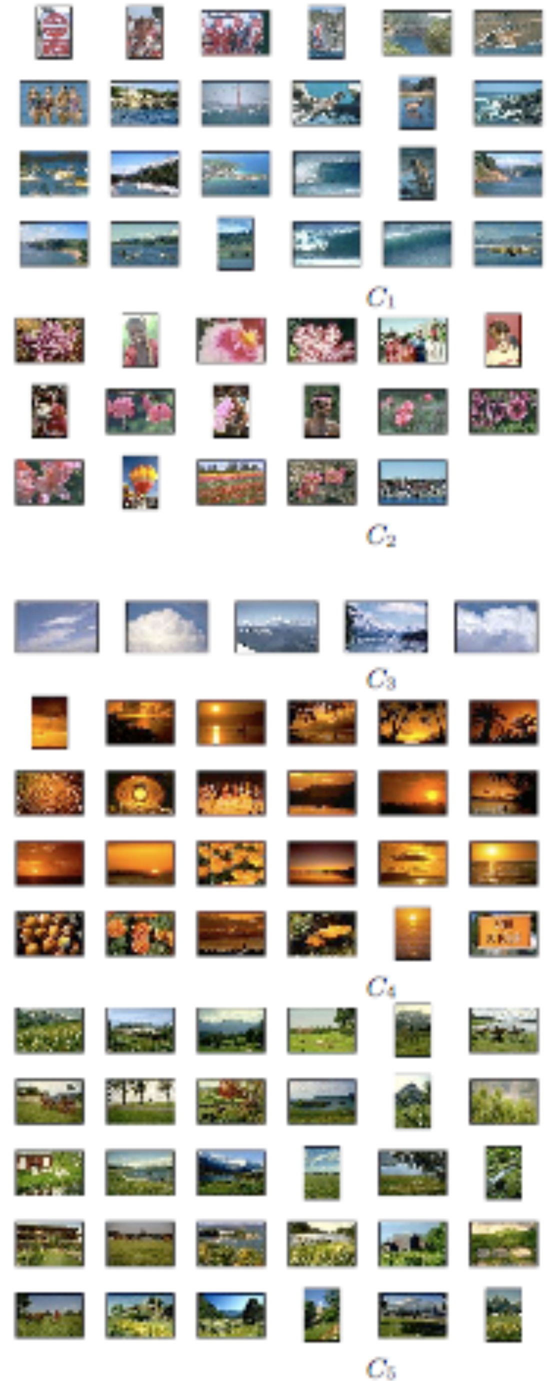
Clustering
K-means

UNIVERSITY of WASHINGTON

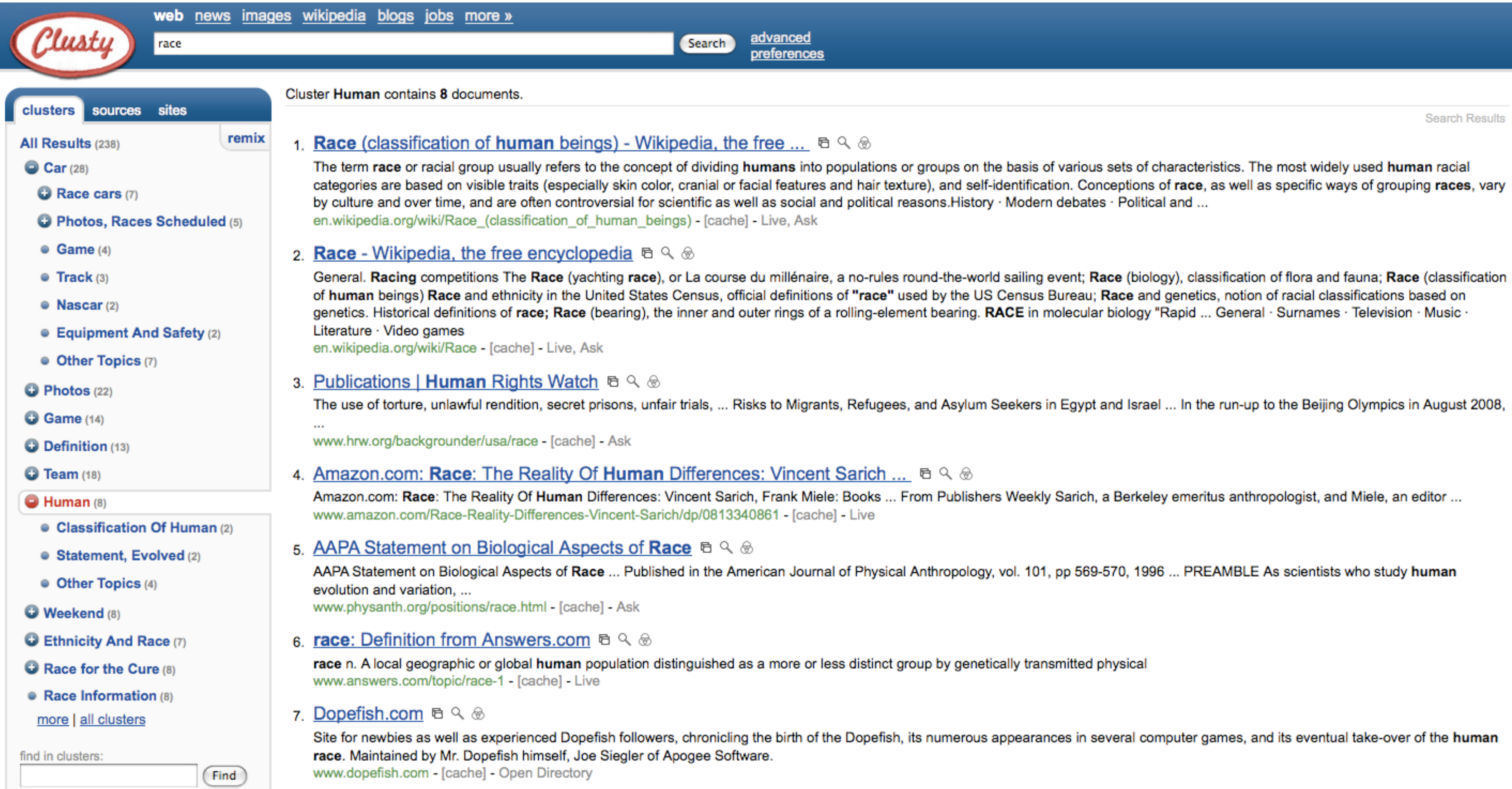
Clustering images



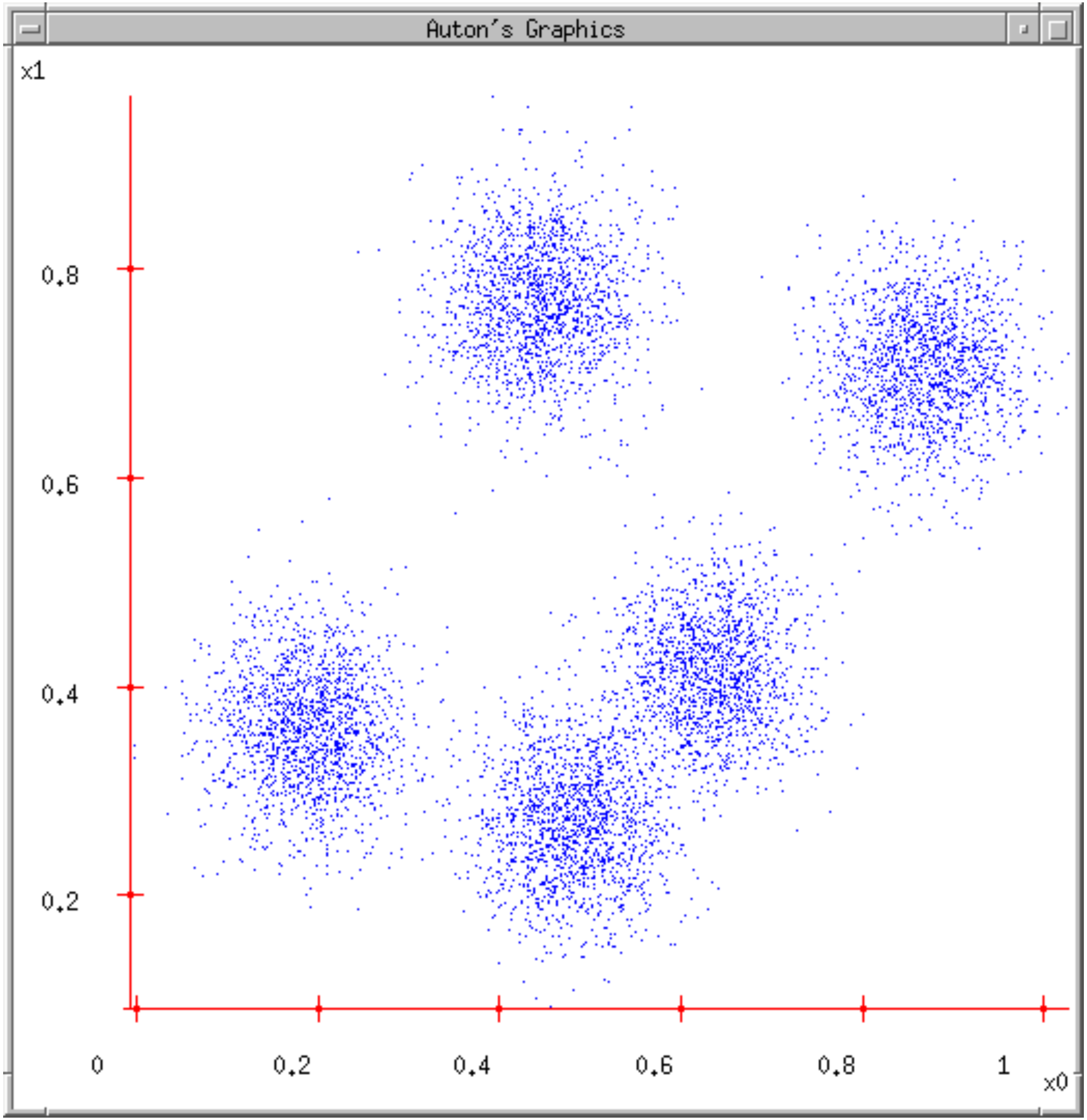


[Goldberger et al.]

Clustering web search results

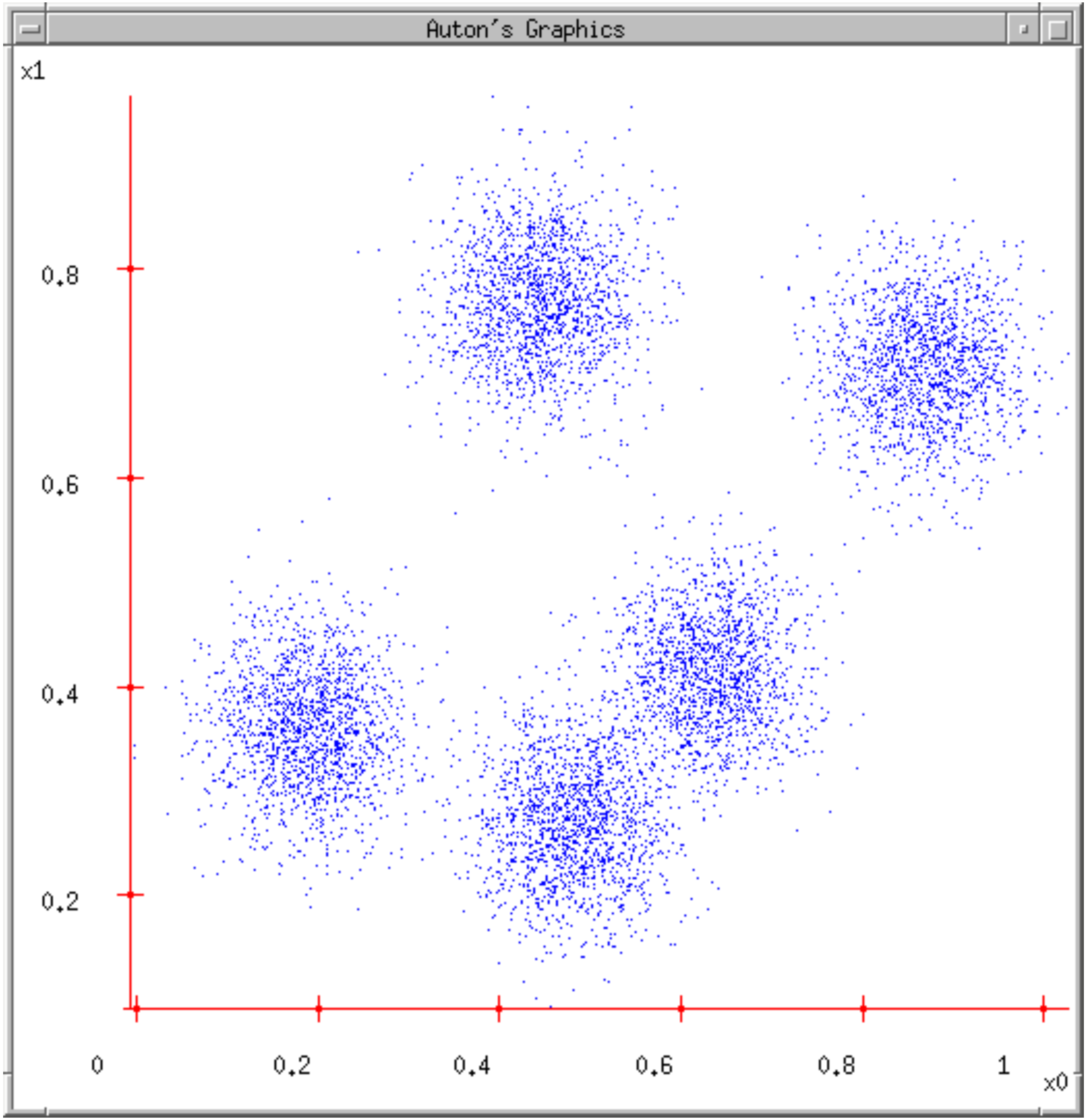


Some Data



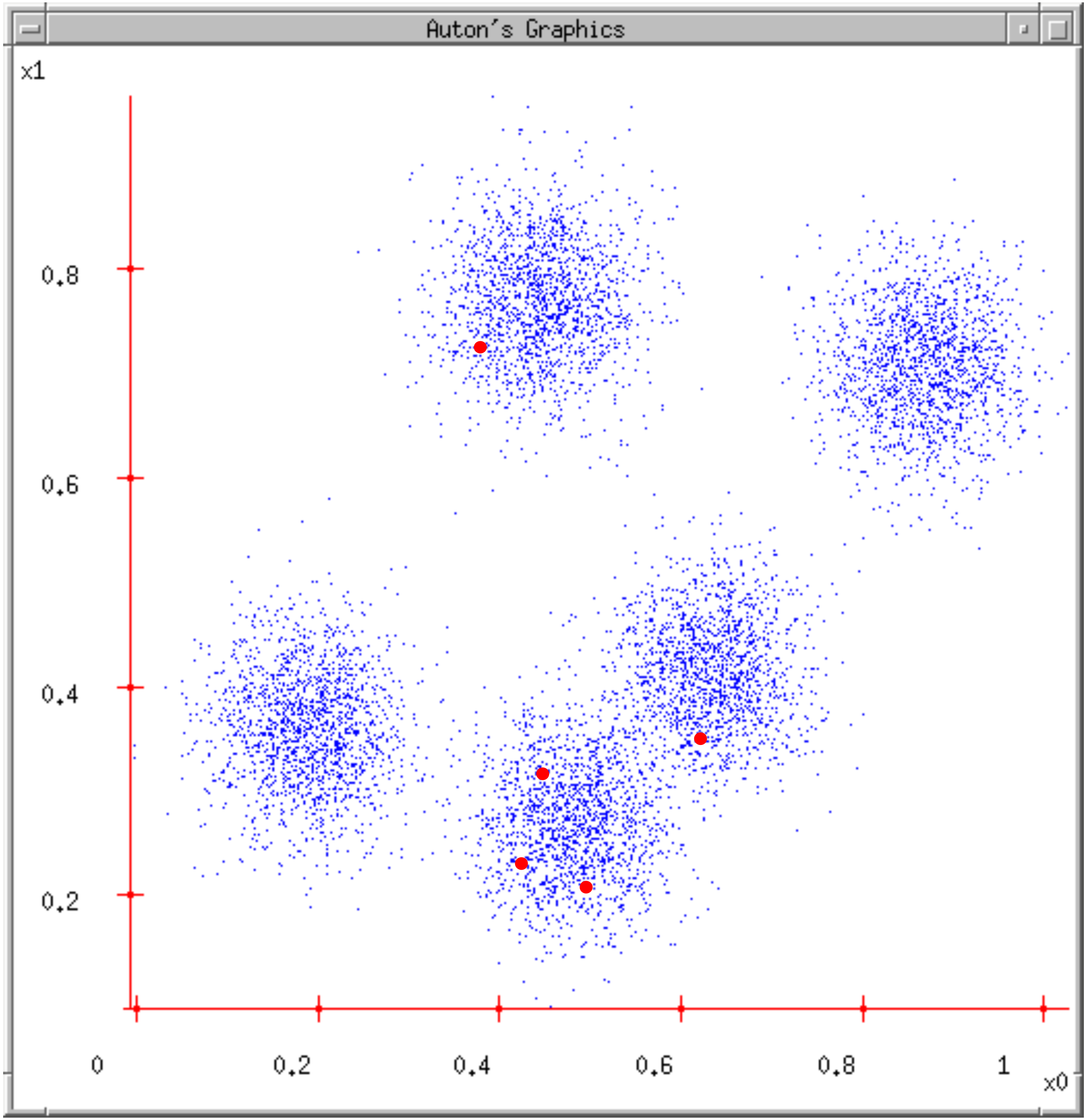
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*



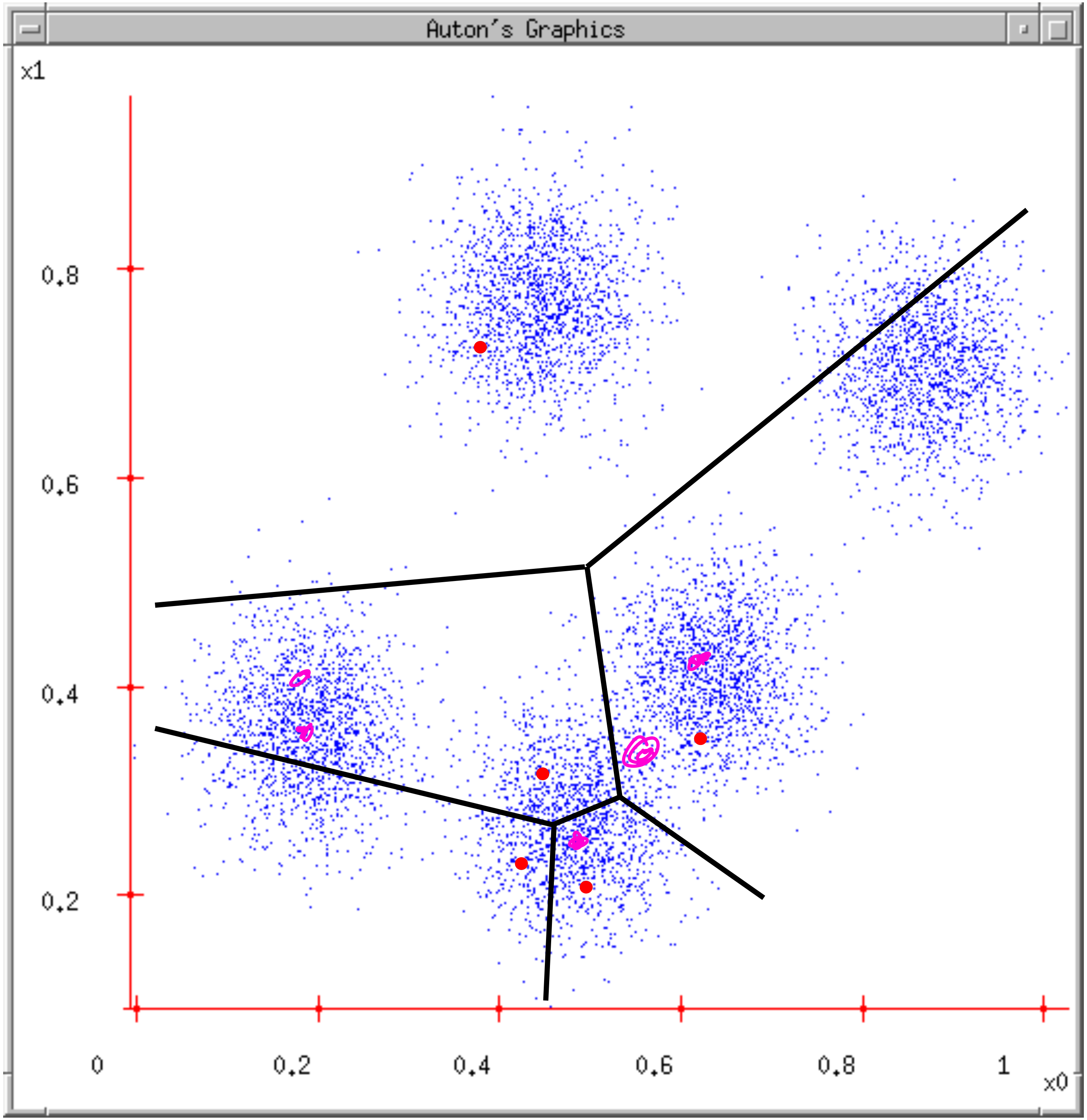
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



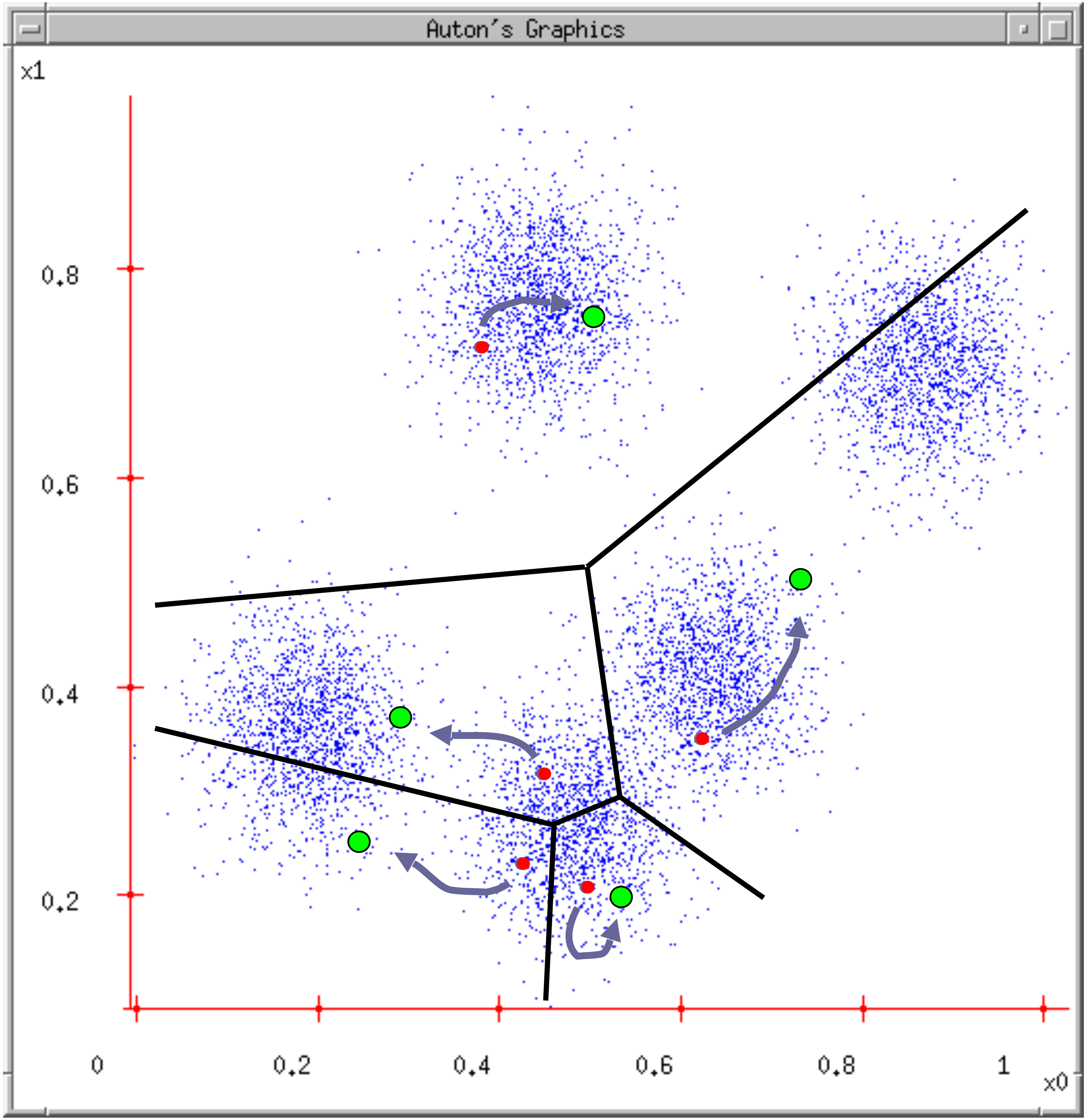
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



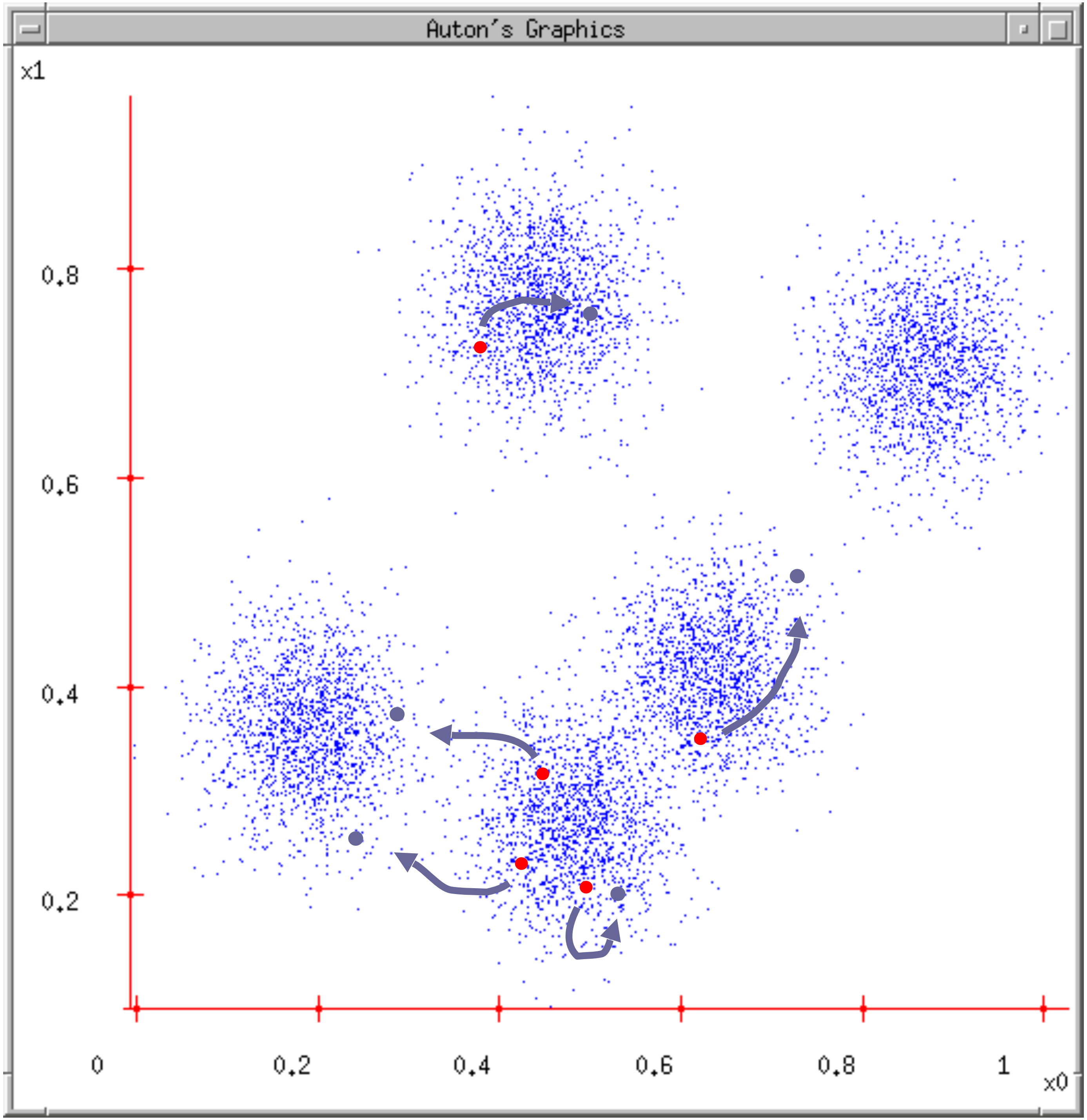
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



K-means

- Randomly initialize k centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

K-means ++

- Classify: Assign each point $j \in \{1, \ldots N\}$ to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$
- Recenter: μ_i becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C(j)=i} ||\mu - x_j||^2$

Does K-means converge??? Part 1

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Fix μ, optimize C

Does K-means converge??? Part 2

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Fix C, optimize μ

Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

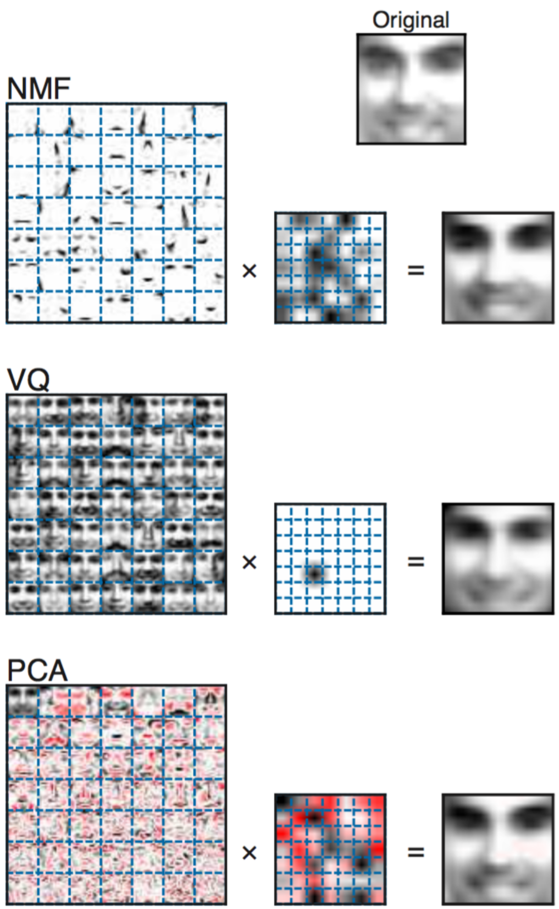
FIGURE 14.9. *Sir Ronald A. Fisher* (1890 − 1962) *was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at* 8 *bits per pixel. The center image is the result of* 2×2 *block VQ, using* 200 *code vectors, with a compression rate of* 1.9 *bits/pixel. The right image uses only four code vectors, with a compression rate of* 0.50 *bits/pixel*

Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at 8 bits per pixel. The center image is the result of* 2×2 *block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*



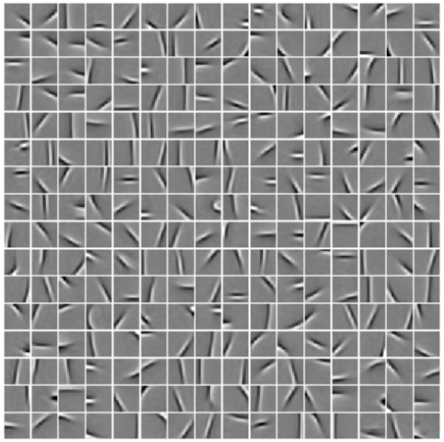
Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at 8 bits per pixel. The center image is the result of* 2×2 *block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

Typical output of
on patches



Similar reduced representation can be used as a feature vector

Coates, Ng, *Learning Feature Representations with K-means,* 2012

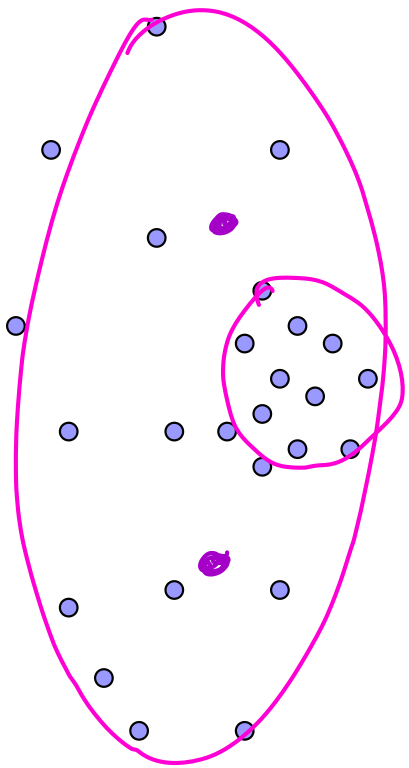
Mixtures of Gaussians

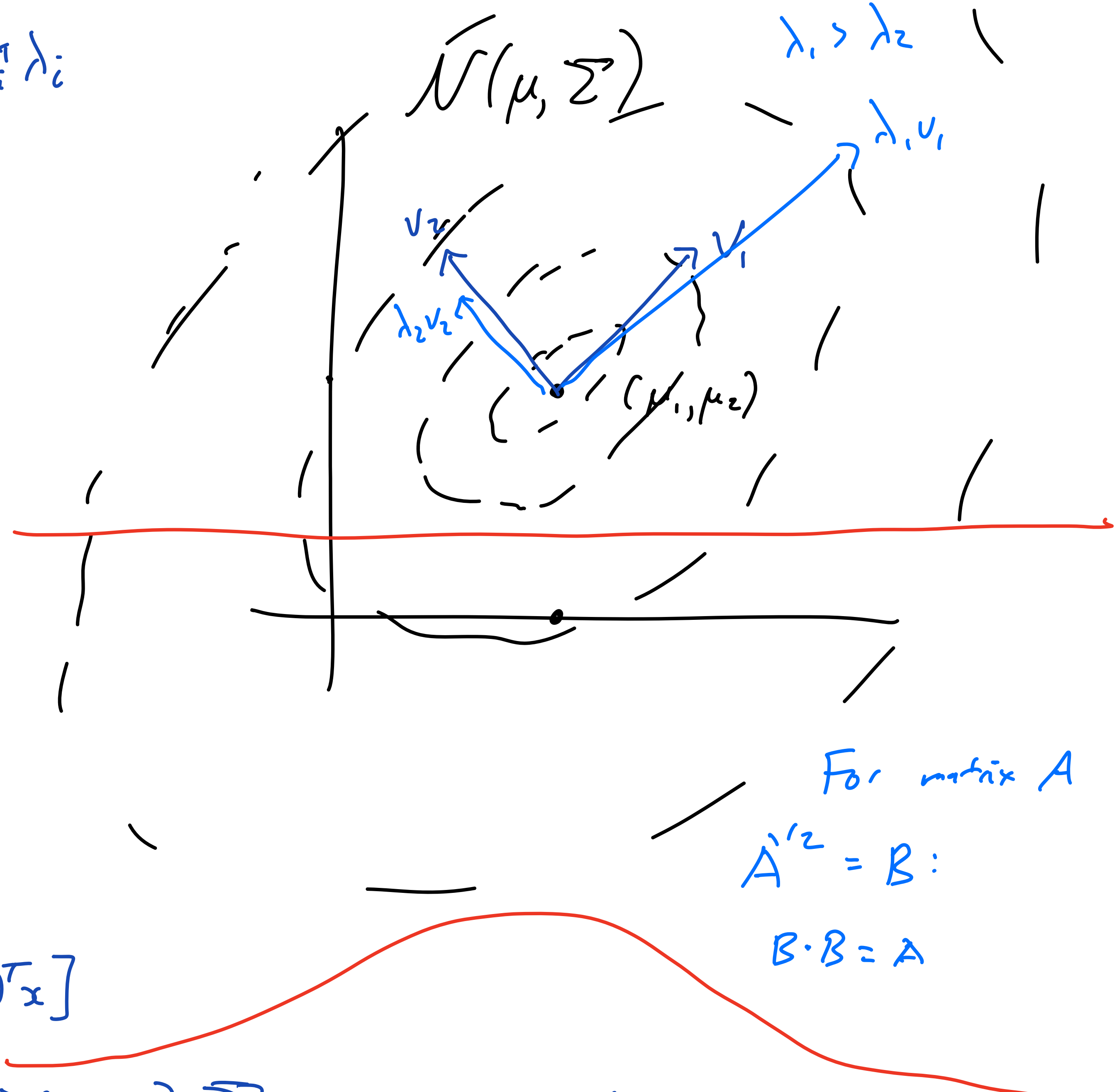
(One) bad case for k-means

$$\Sigma = \sum_{i=1}^{d} v_i v_i^T \lambda_i$$

$$\Sigma = \mathbb{E}[(X-\mu)(X-\mu)^T]$$

$X \sim N(\mu, \Sigma)$





For matrix A

$A^{1/2} = B$:

$B \cdot B = A$

$$x^T \Sigma x = \mathbb{E}[x^T (X-\mu)(X-\mu)^T x]$$

$$= \mathbb{E}[|x^T(X-\mu)|^2] \geq 0 \Rightarrow \Sigma \text{ is positive semi-definite.}$$

$X \sim \mathcal{N}(\mu, \Sigma)$

$Z \sim \mathcal{N}(0, I)$

$X' = \mu + \Sigma^{1/2} Z$

$\mathbb{E}[X'] = \mu$

$\mathbb{E}[(X'-\mu)(X-\mu)^T] = \mathbb{E}[\Sigma^{1/2} Z Z^T (\Sigma^{1/2})^T] = \Sigma^{1/2} \cdot \Sigma^{1/2}$

$= \Sigma$

$[Z Z^T]_{i,j} = z_i z_j$

$\mathbb{E}[Z Z^T] = I$

$\Sigma = V \Lambda V^T$

$= \sum_i v_i v_i^T \lambda_i$

$\Sigma^{1/2} = \sum_i v_i v_i^T \sqrt{\lambda_i}$

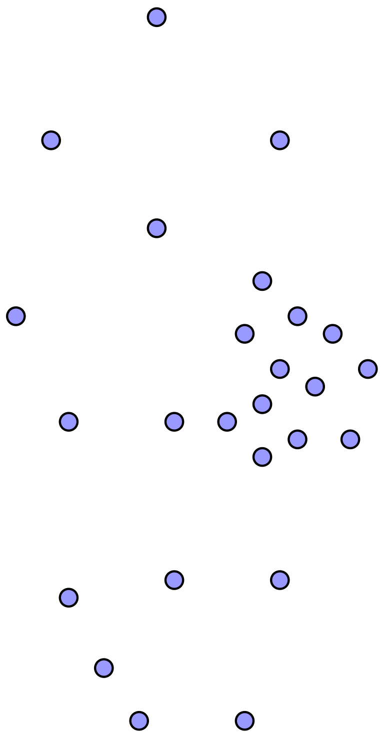
$\Sigma^{1/2} \cdot \Sigma^{1/2} = \sum v_i v_i^T \lambda_i$

$v_i^T v_j = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{o.w.} \end{cases}$

$\Longrightarrow X' = X$ ∎

(One) bad case for k-means

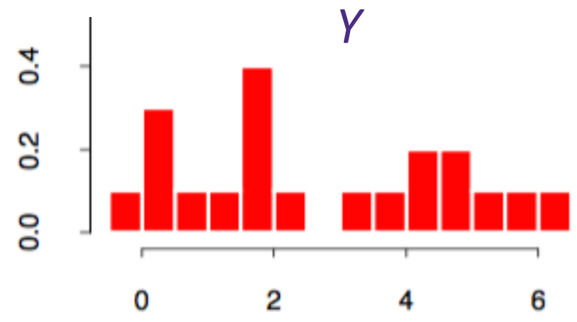
- Clusters may overlap
- Some clusters may be “wider” than others



Mixture models

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

$\Delta \in \{0,1\}$ with $\Pr(\Delta = 1) = \pi$



$\mathbf{Z} = \{y_i\}_{i=1}^n$ is observed data

If $\phi_\theta(x)$ is Gaussian density with parameters $\theta = (\mu, \sigma^2)$ then

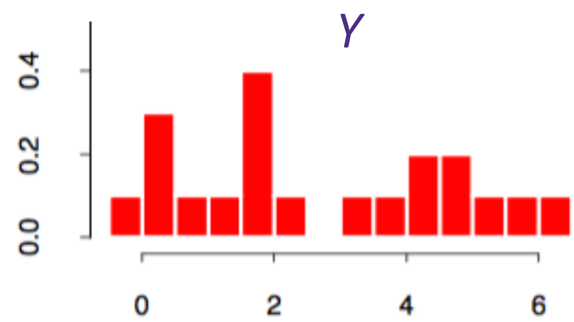
$$\ell(\theta; \mathbf{Z}) = \sum_{i=1}^n \log[(1-\pi)\phi_{\theta_1}(y_i) + \pi\phi_{\theta_2}(y_i)]$$

Mixture models

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

$\Delta \in \{0, 1\}$ with $\Pr(\Delta = 1) = \pi$

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$



$\mathbf{Z} = \{y_i\}_{i=1}^n$ is observed data

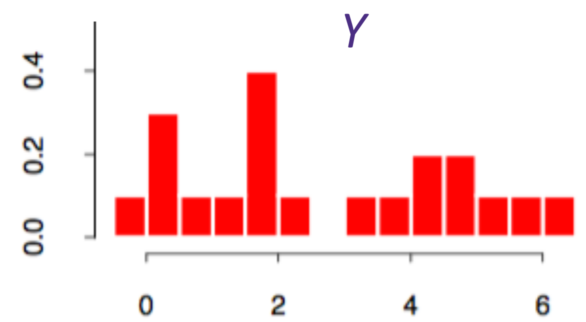
$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$ is unobserved data

If $\phi_\theta(x)$ is Gaussian density with parameters $\theta = (\mu, \sigma^2)$ then

$$\ell(\theta; y_i, \Delta_i = 0) =$$

$$\ell(\theta; y_i, \Delta_i = 1) =$$

Mixture models



$$\begin{aligned} Y_1 &\sim N(\mu_1, \sigma_1^2), \\ Y_2 &\sim N(\mu_2, \sigma_2^2), \\ Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2, \end{aligned}$$

$\Delta \in \{0,1\}$ with $\Pr(\Delta = 1) = \pi$

$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$

$\mathbf{Z} = \{y_i\}_{i=1}^n$ is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$ is unobserved data

If $\phi_\theta(x)$ is Gaussian density with parameters $\theta = (\mu, \sigma^2)$ then

$$\ell(\theta; \mathbf{Z}, \mathbf{\Delta}) = \sum_{i=1}^{n} (1-\Delta_i) \log[(1-\pi)\phi_{\theta_1}(y_i)] + \Delta_i \log(\pi \phi_{\theta_2}(y_i)]$$

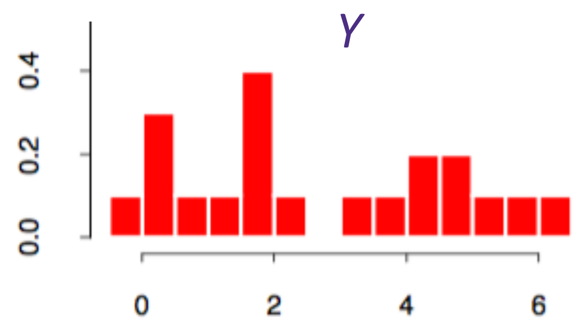
If we knew Δ, how would we choose θ?

Mixture models

$$\begin{aligned} Y_1 &\sim N(\mu_1, \sigma_1^2), \\ Y_2 &\sim N(\mu_2, \sigma_2^2), \\ Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2, \end{aligned}$$

$\Delta \in \{0,1\}$ with $\Pr(\Delta = 1) = \pi$

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$



$\mathbf{Z} = \{y_i\}_{i=1}^n$ is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$ is unobserved data

If $\phi_\theta(x)$ is Gaussian density with parameters $\theta = (\mu, \sigma^2)$ then

$$\ell(\theta; \mathbf{Z}, \mathbf{\Delta}) = \sum_{i=1}^n (1-\Delta_i)\log[(1-\pi)\phi_{\theta_1}(y_i)] + \Delta_i \log(\pi \phi_{\theta_2}(y_i)]$$

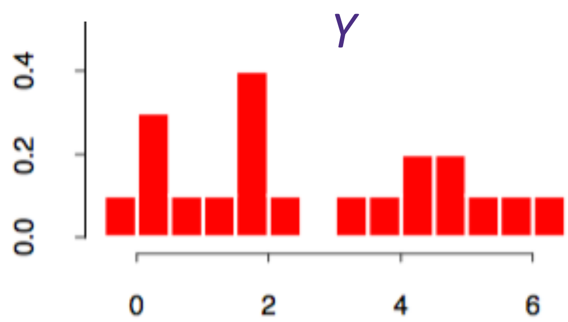
If we knew θ, how would we choose $\mathbf{\Delta}$?

Mixture models

$$\begin{aligned} Y_1 &\sim N(\mu_1, \sigma_1^2), \\ Y_2 &\sim N(\mu_2, \sigma_2^2), \\ Y &= (1-\Delta)\cdot Y_1 + \Delta \cdot Y_2, \end{aligned}$$

$\Delta \in \{0,1\}$ with $\Pr(\Delta = 1) = \pi$

$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$



$\mathbf{Z} = \{y_i\}_{i=1}^n$ is observed data

$\mathbf{\Delta} = \{\Delta_i\}_{i=1}^n$ is unobserved data

If $\phi_\theta(x)$ is Gaussian density with parameters $\theta = (\mu, \sigma^2)$ then

$$\ell(\theta; \mathbf{Z}, \mathbf{\Delta}) = \sum_{i=1}^{n} (1-\Delta_i)\log[(1-\pi)\phi_{\theta_1}(y_i)] + \Delta_i \log(\pi\phi_{\theta_2}(y_i)]$$

$$\gamma_i(\theta) = \mathbb{E}[\Delta_i | \theta, \mathbf{Z}] =$$

Mixture models

Algorithm 8.1 *EM Algorithm for Two-component Gaussian Mixture.*

---

1. Take initial guesses for the parameters $\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$ (see text).

2. *Expectation Step*: compute the responsibilities

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, \quad i = 1, 2, \ldots, N. \tag{8.42}$$

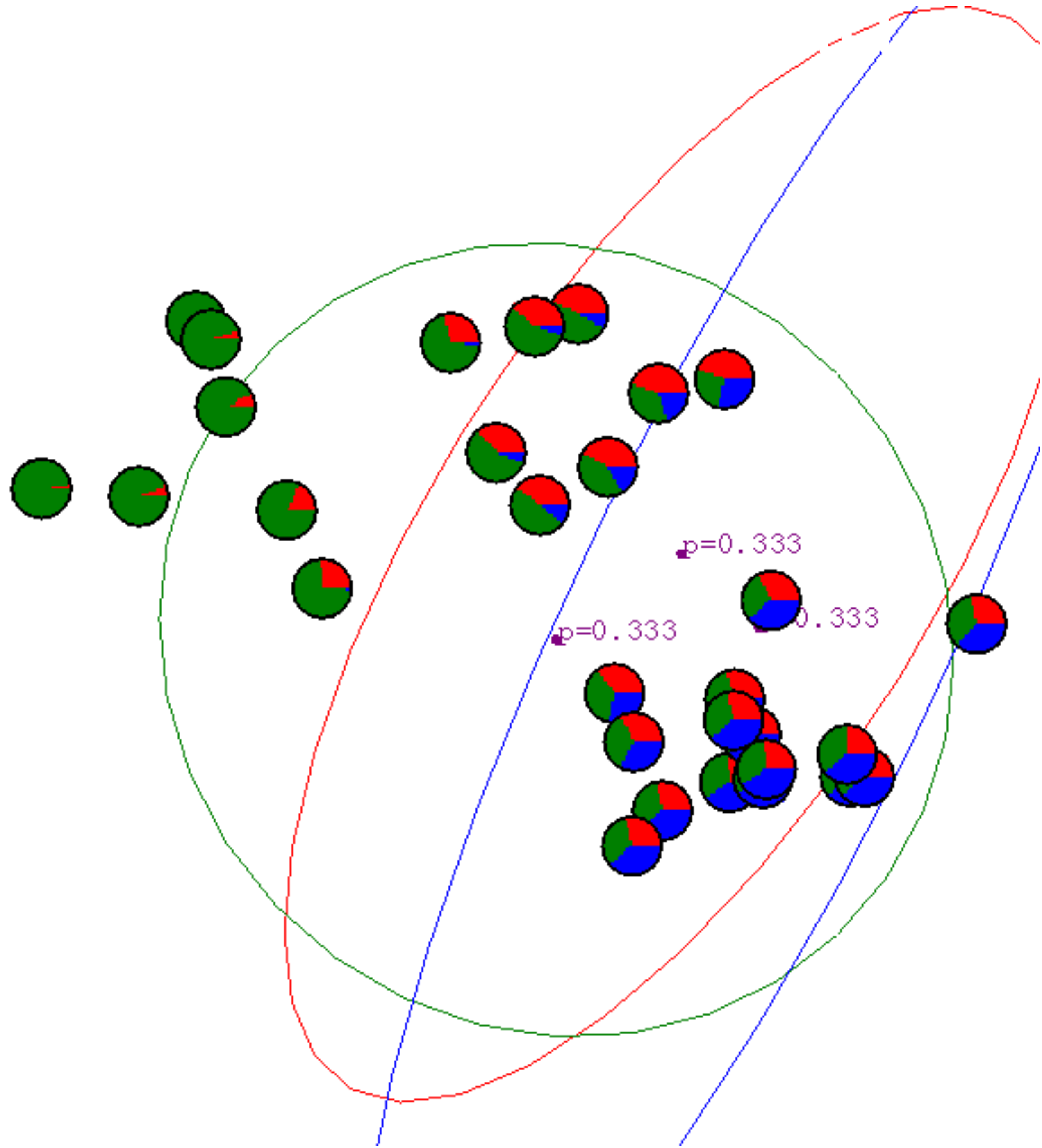
3. *Maximization Step*: compute the weighted means and variances:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)},$$

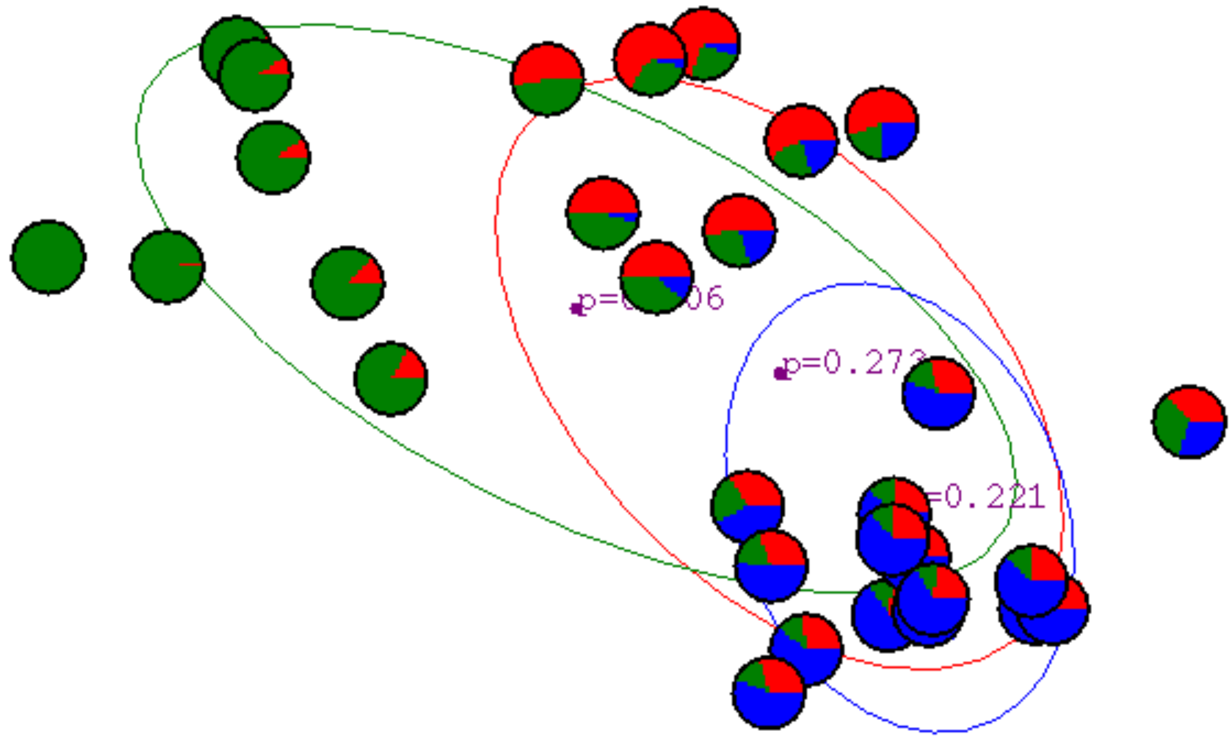
$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i},$$

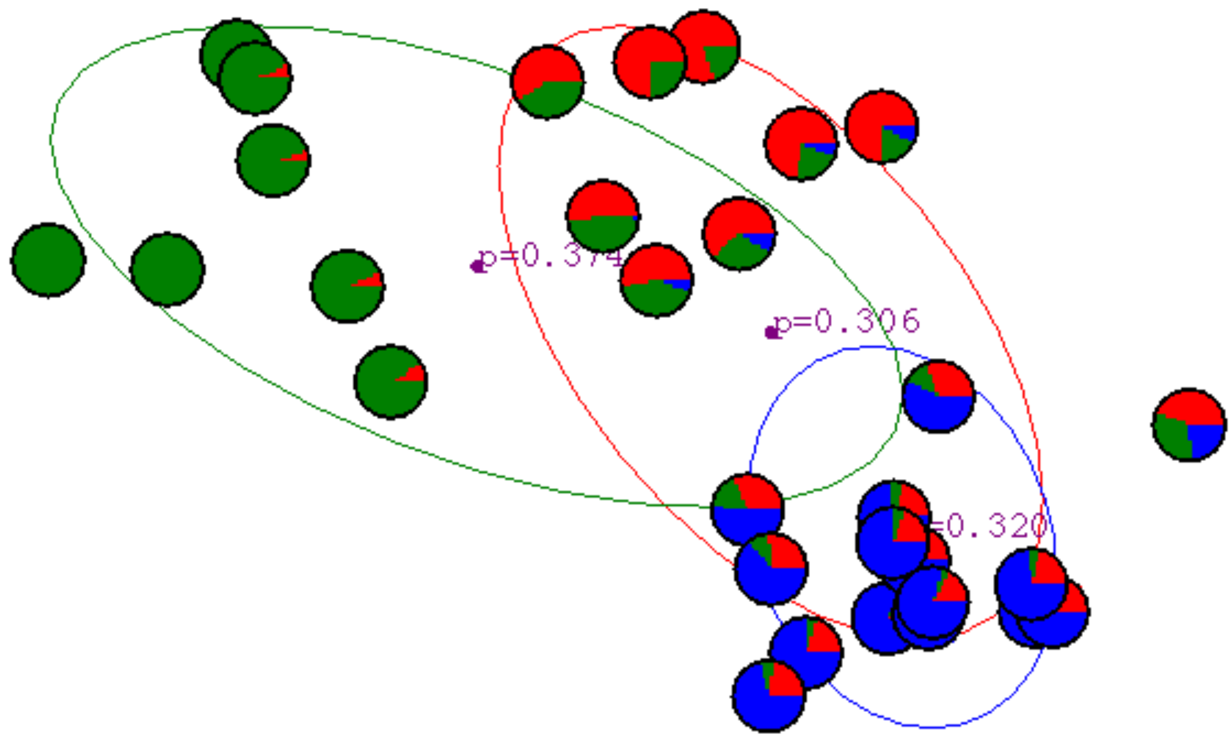
   and the mixing probability $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$.

4. Iterate steps 2 and 3 until convergence.

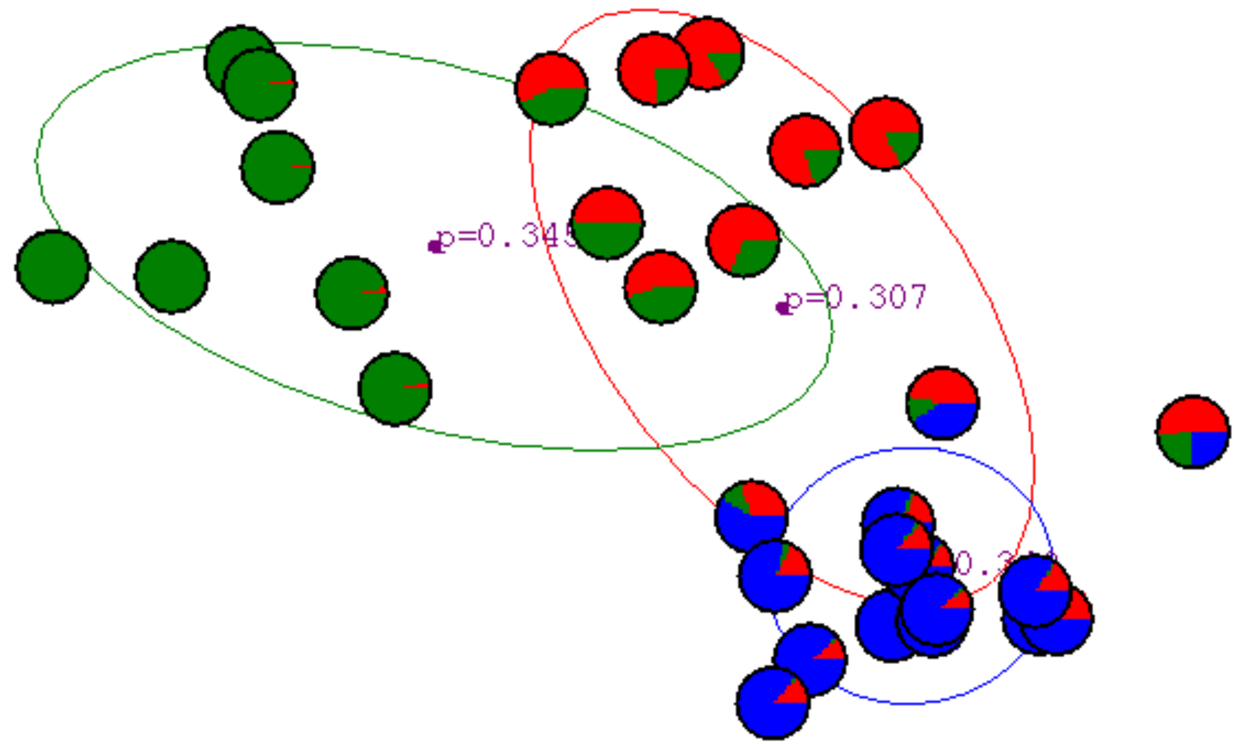


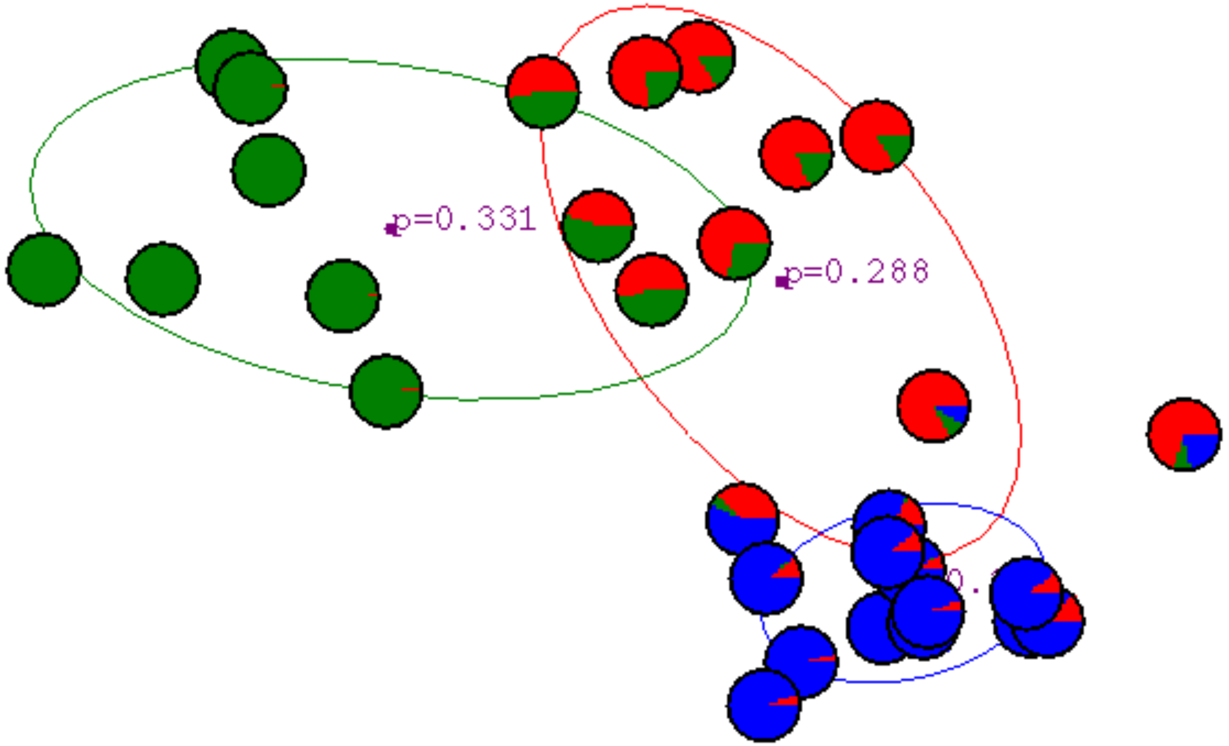
p=0.333
0.333
p=0.333

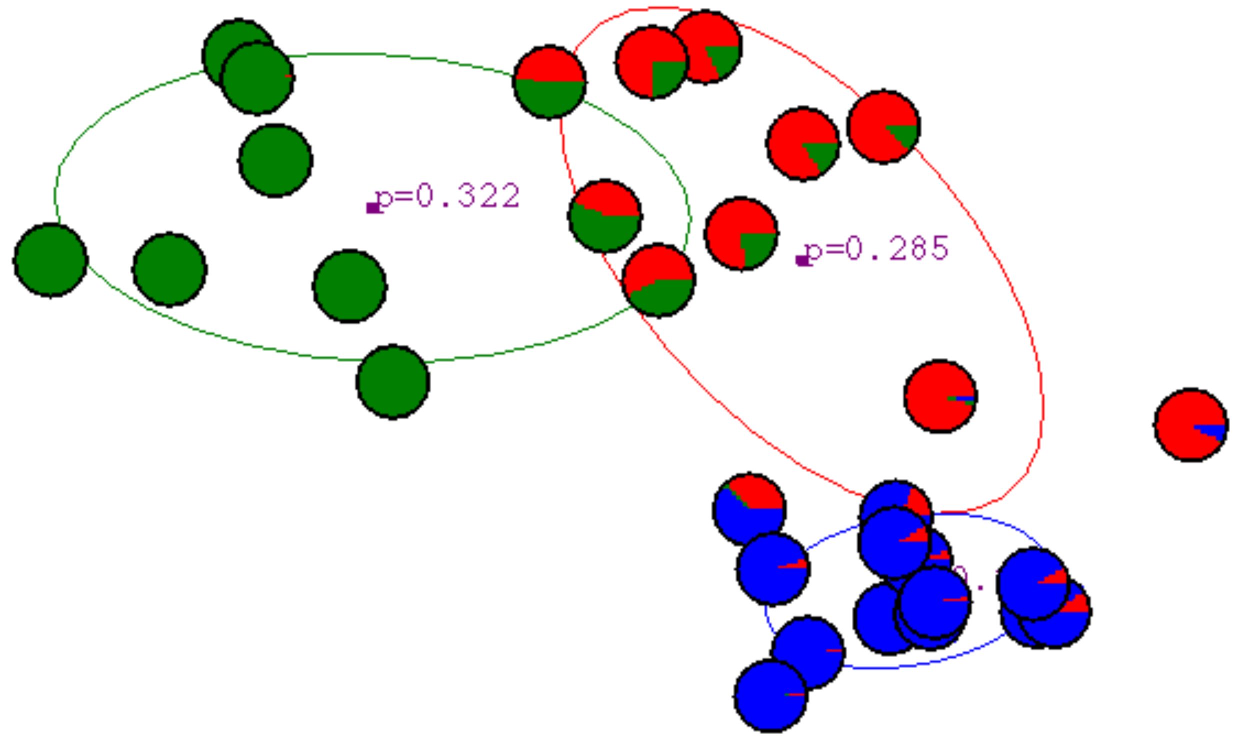


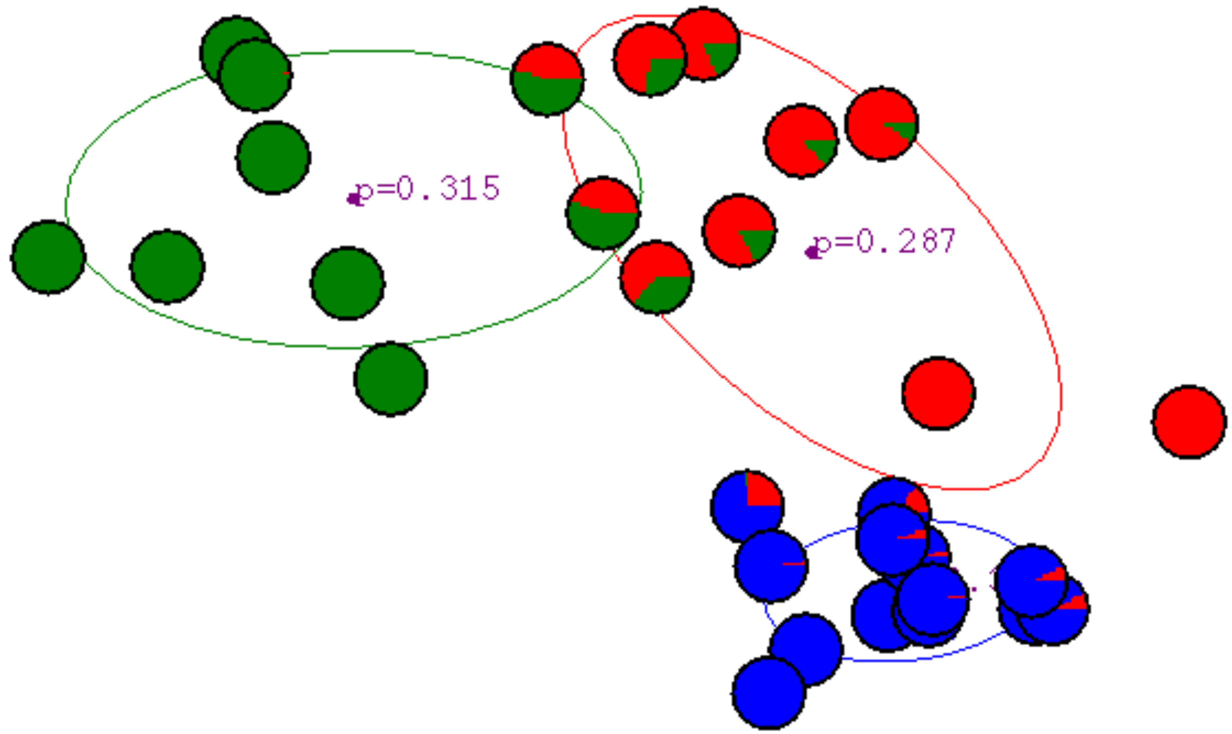


p=0.374
p=0.306
=0.320

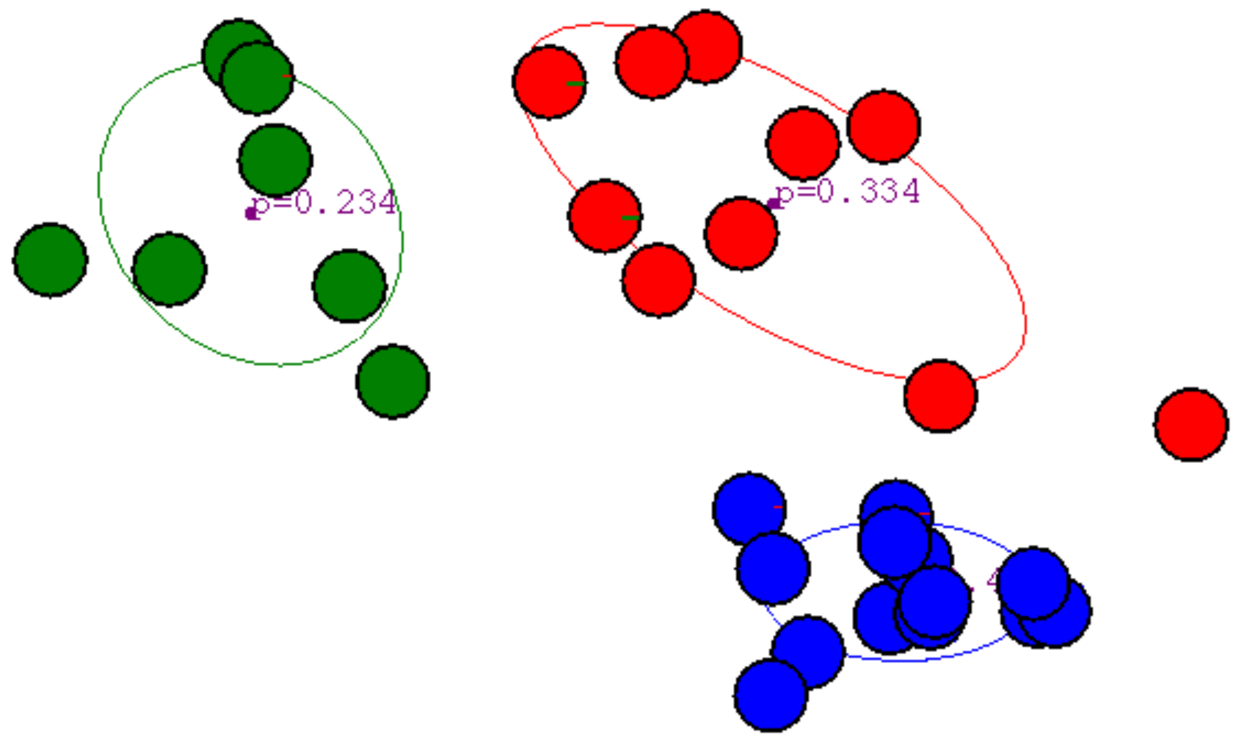

p=0.307

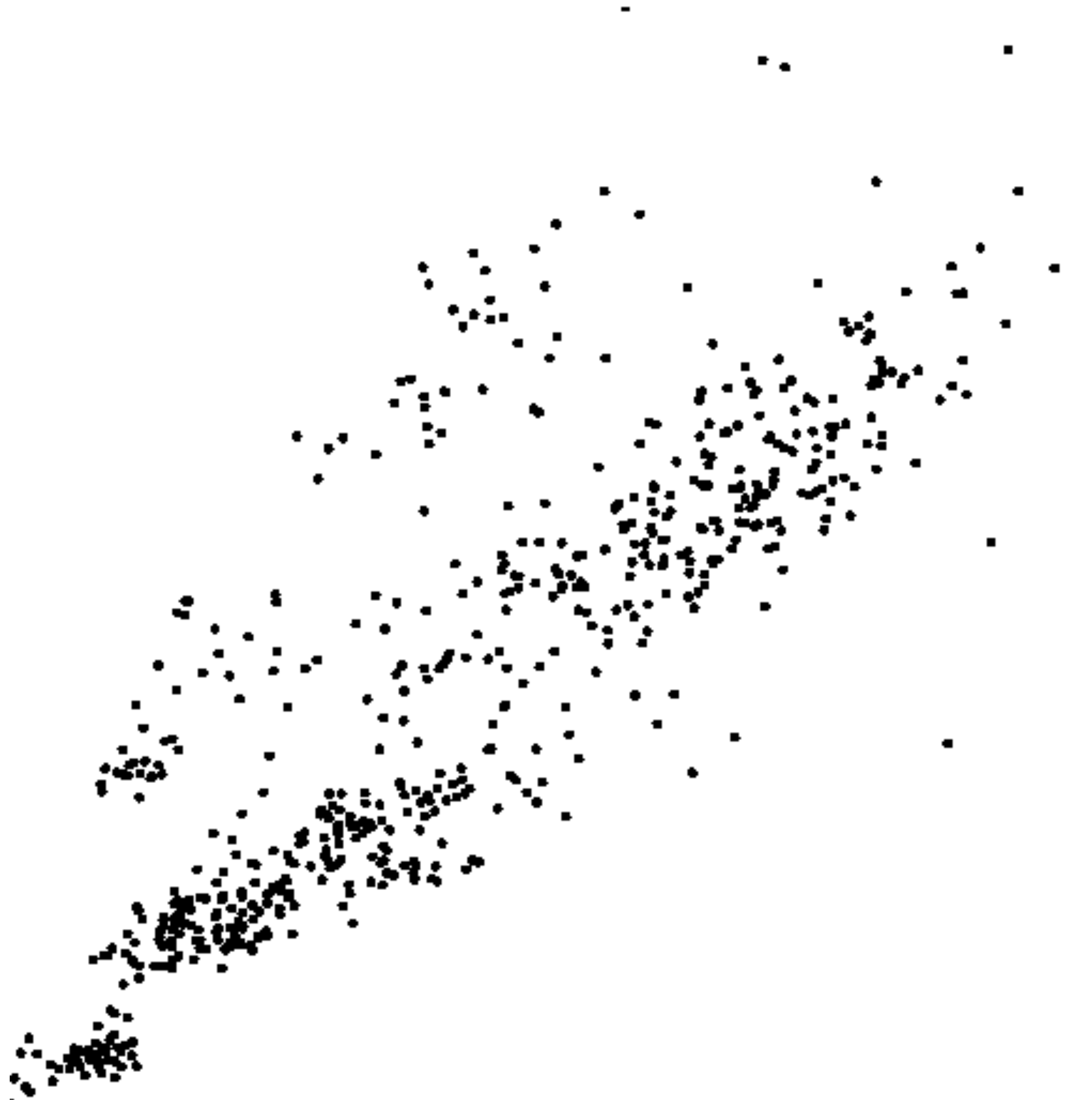

p=0.331
p=0.288

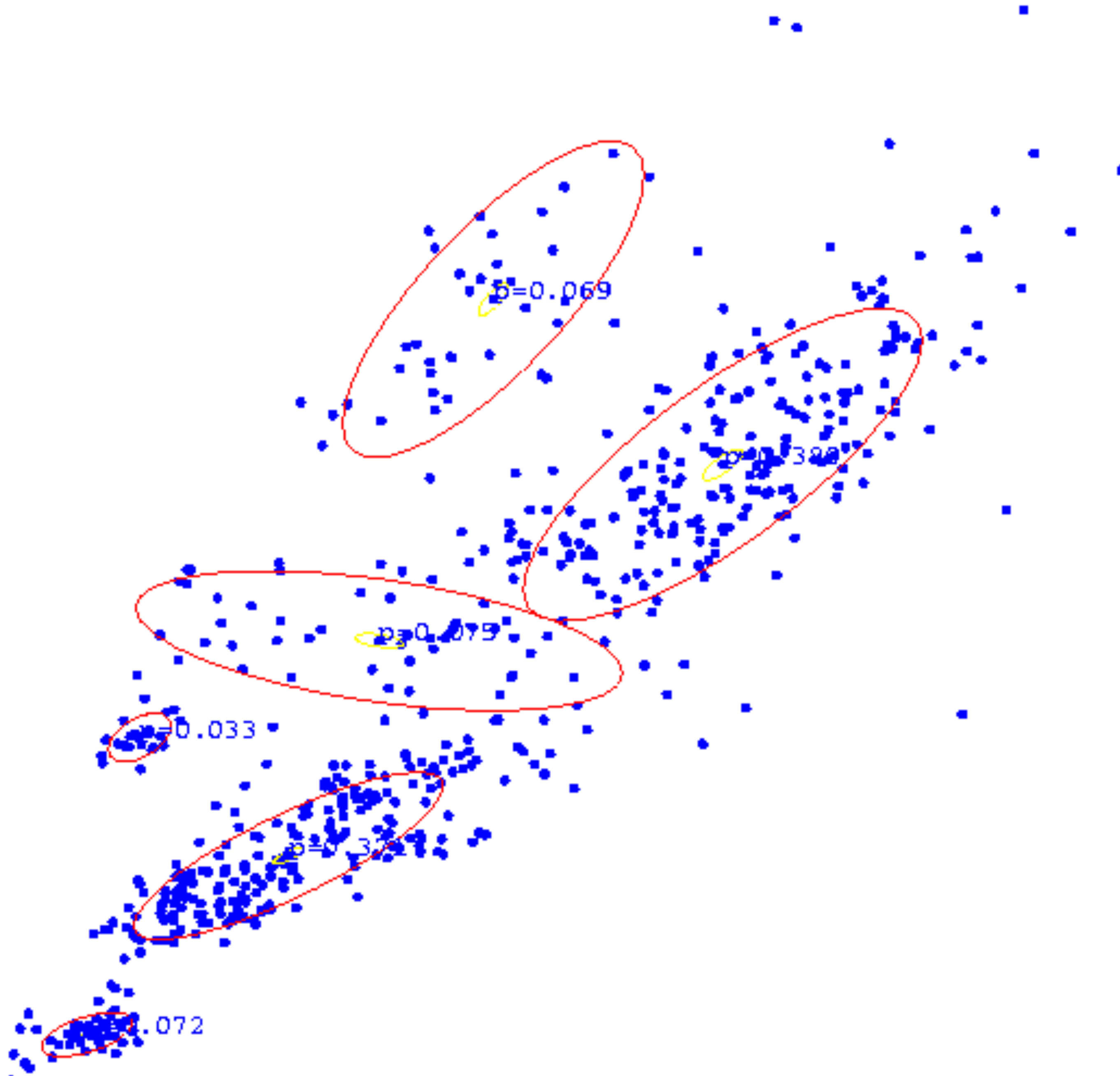

p=0.322
p=0.285



p=0.315
p=0.287


p=0.234
p=0.334




p=0.069
0.033
.072