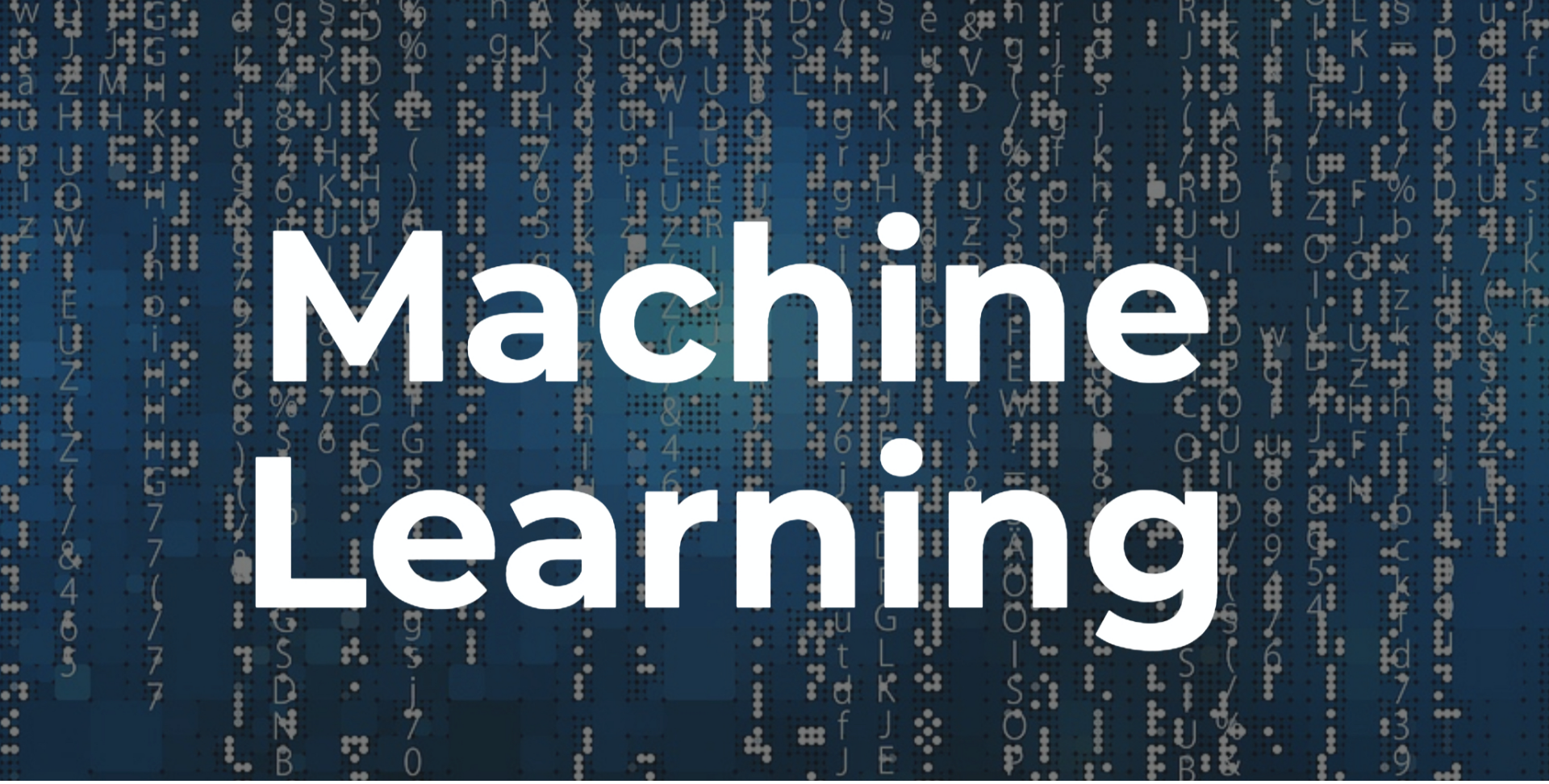


CSCI 5622 Fall 2020

Prof. Claire Monteleoni

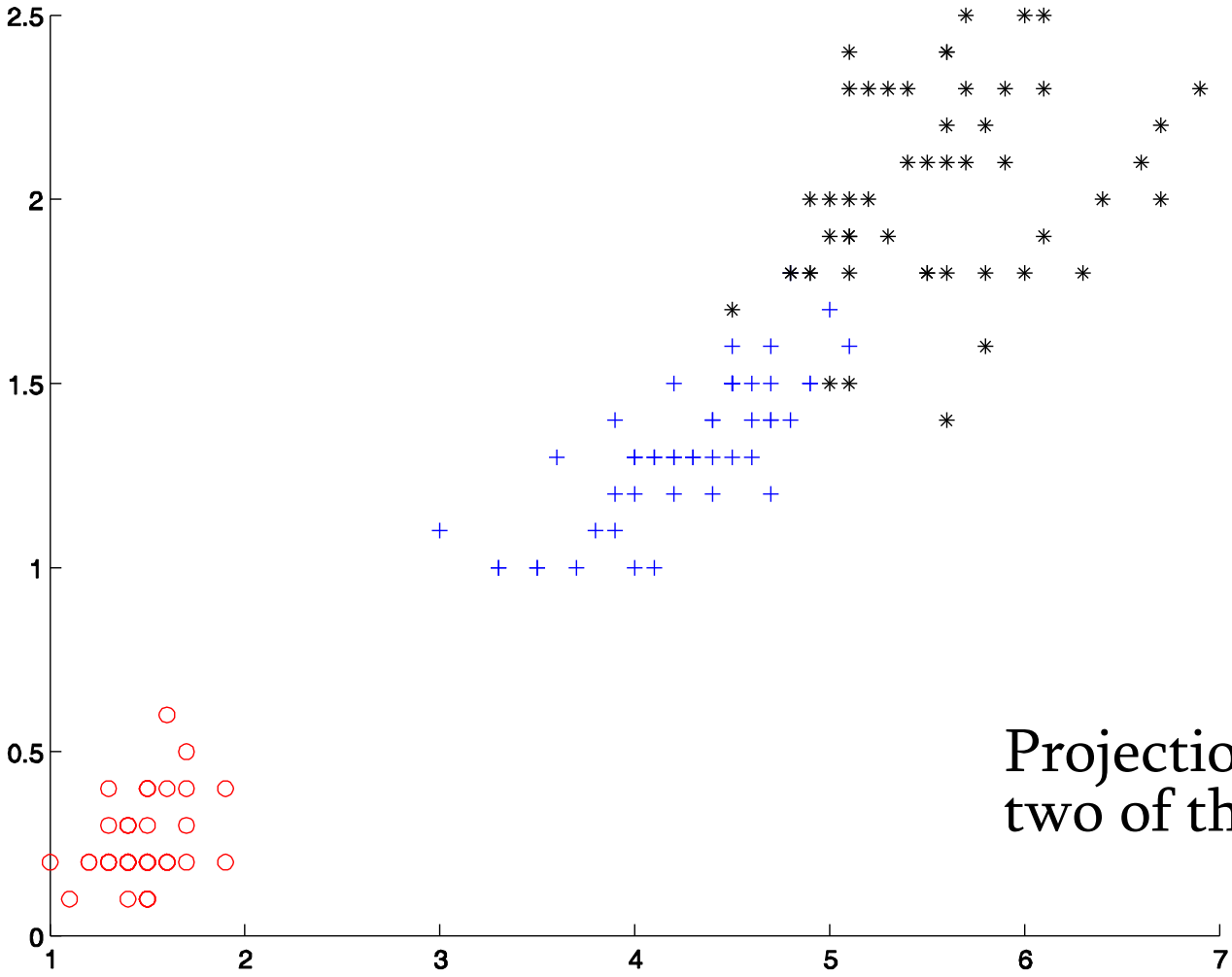
Today

- Intro. to Generative Learning / probabilistic models
  - Maximum likelihood parameter estimation
  - Mixture models
  - Expectation maximization (EM)

with much credit to S. Dasgupta and T. Jaakkola

Iris data revisited

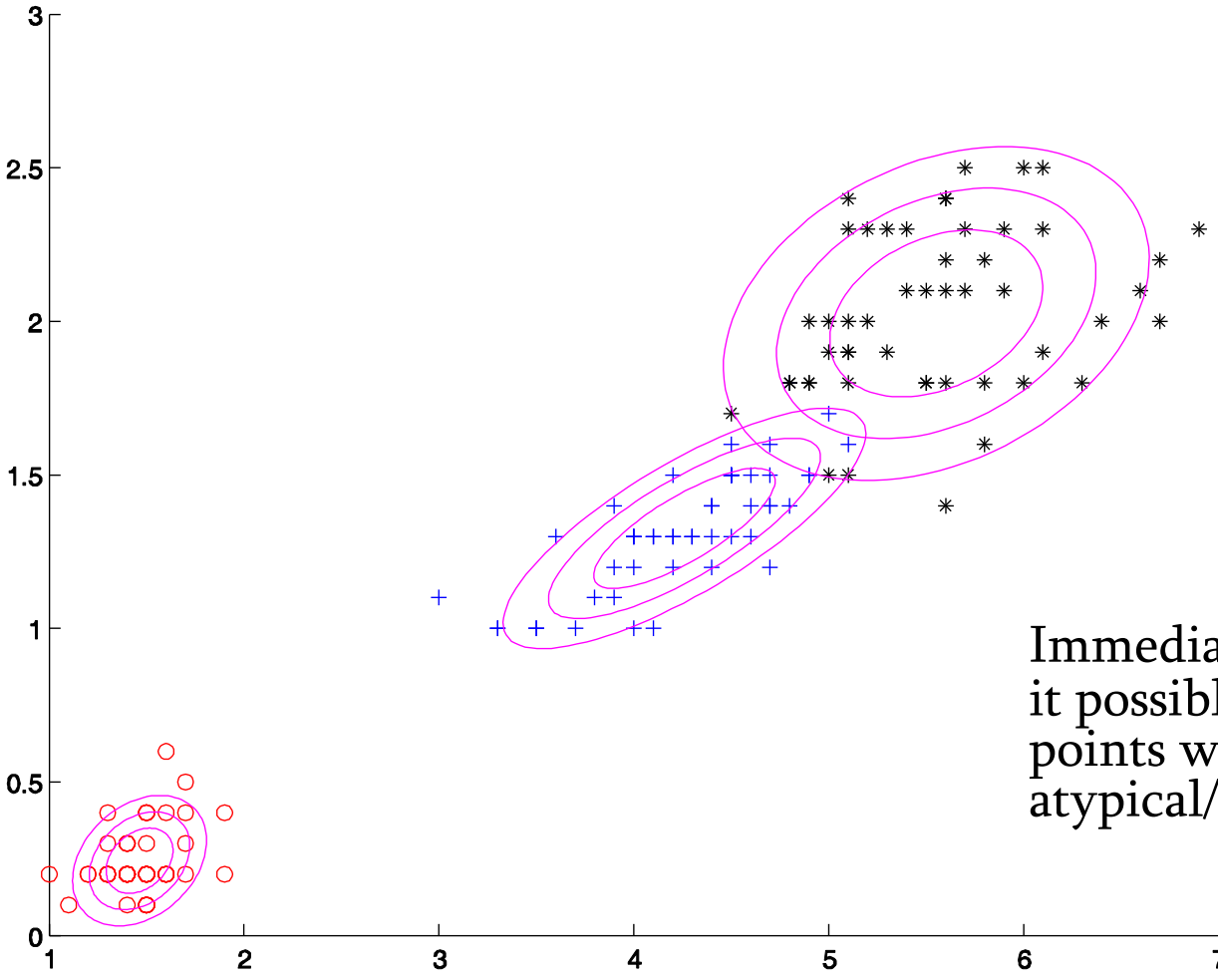
Iris data: 3 classes (setosa, viriginica, versicolor), 4 features



Projection onto two of the features

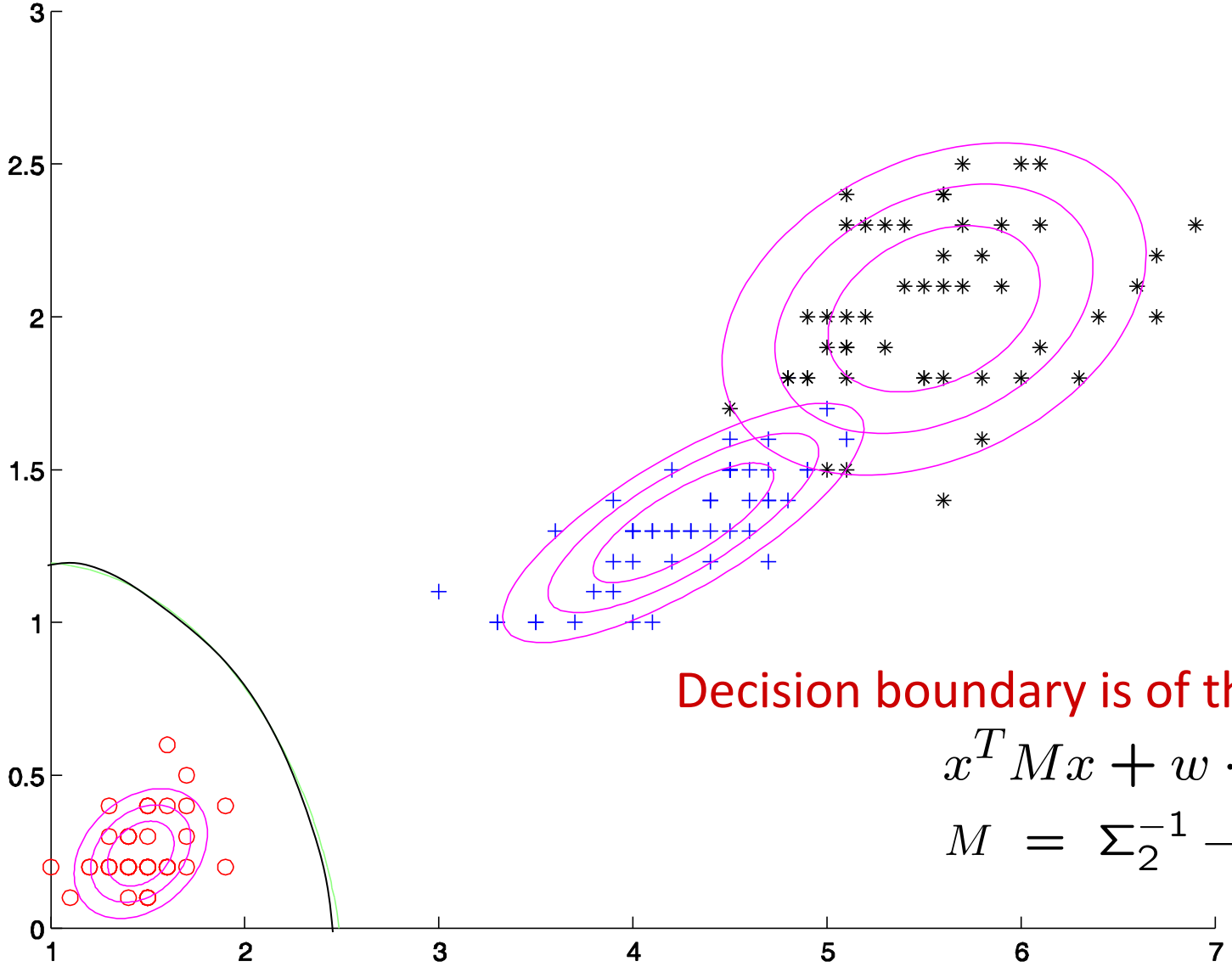
Generative model

Model each class by a Gaussian distribution



Immediately makes it possible to flag points which are atypical/anomalous.

Boundary: circle/plus

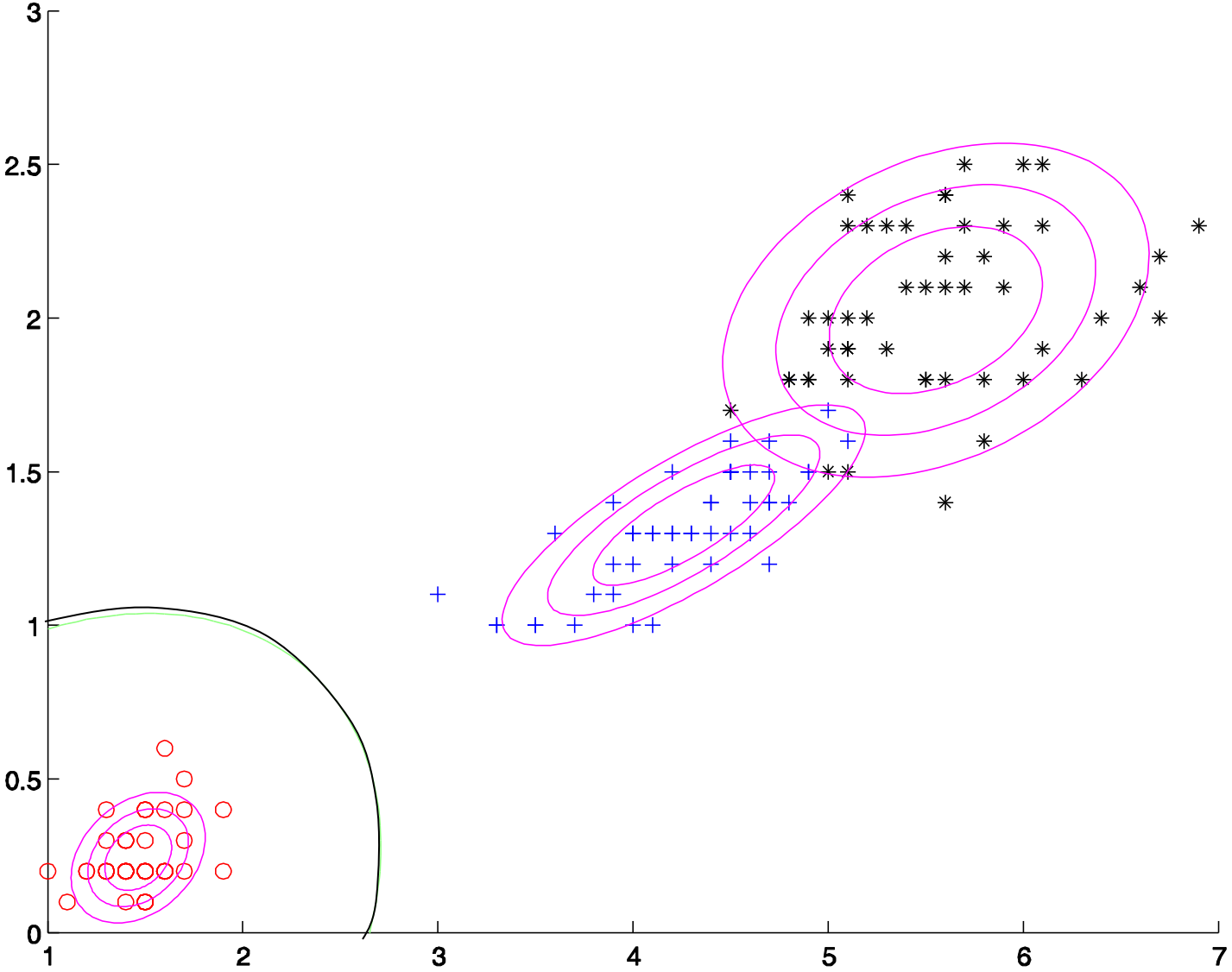


Decision boundary is of the form:

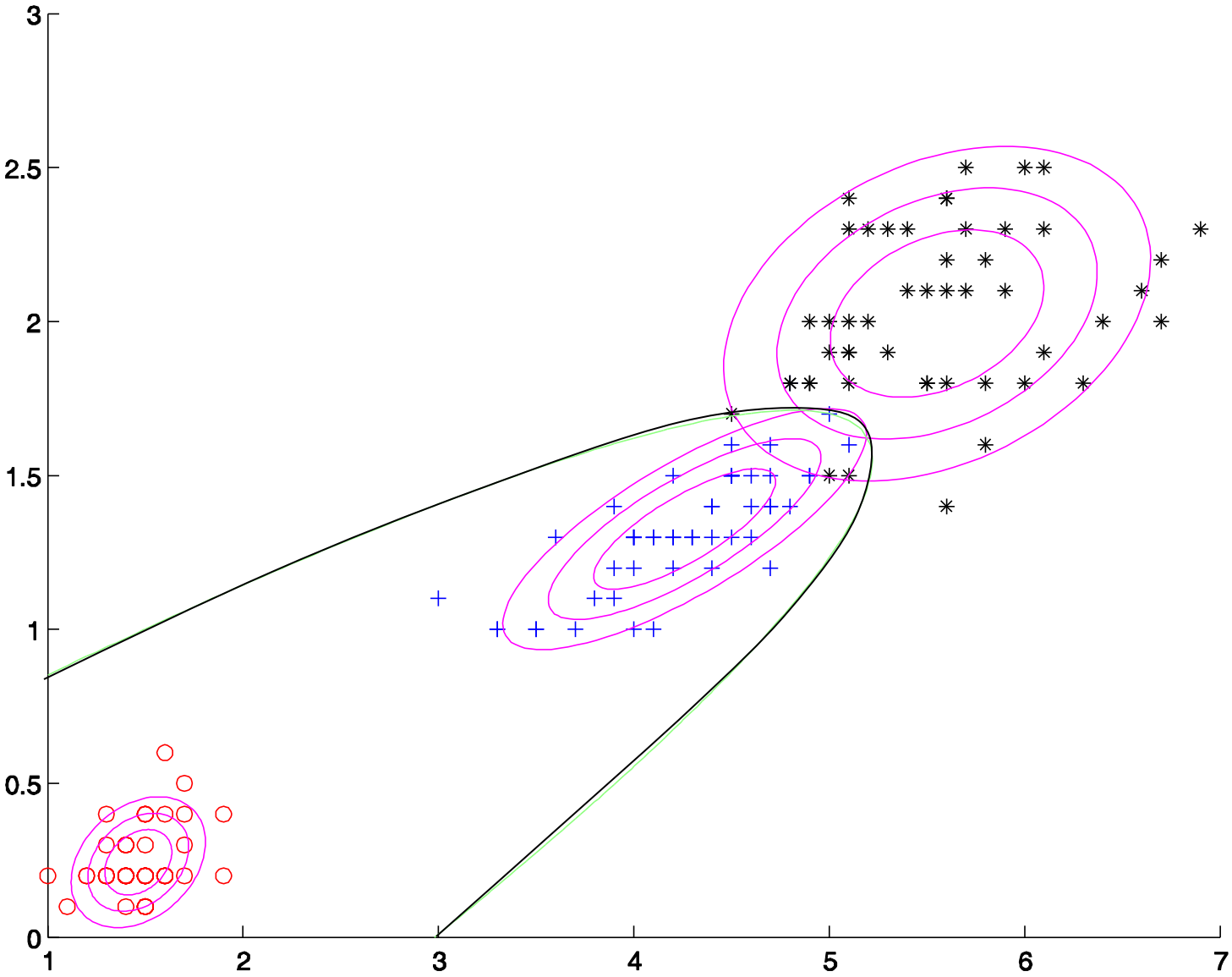
$$x^T M x + w \cdot x \geq \theta$$

$$M = \Sigma_2^{-1} - \Sigma_1^{-1}$$

Boundary: circle/star

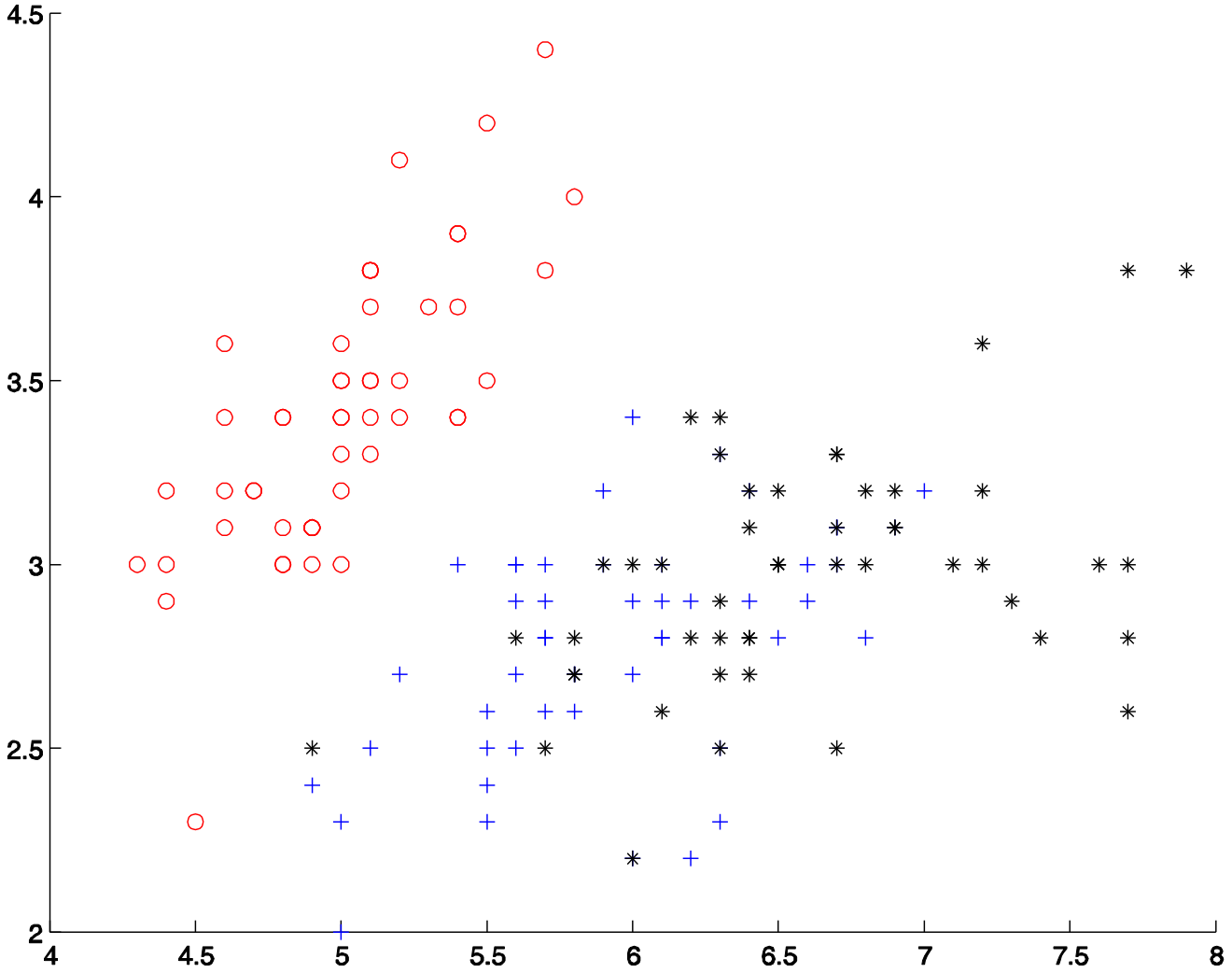


Boundary: plus/star

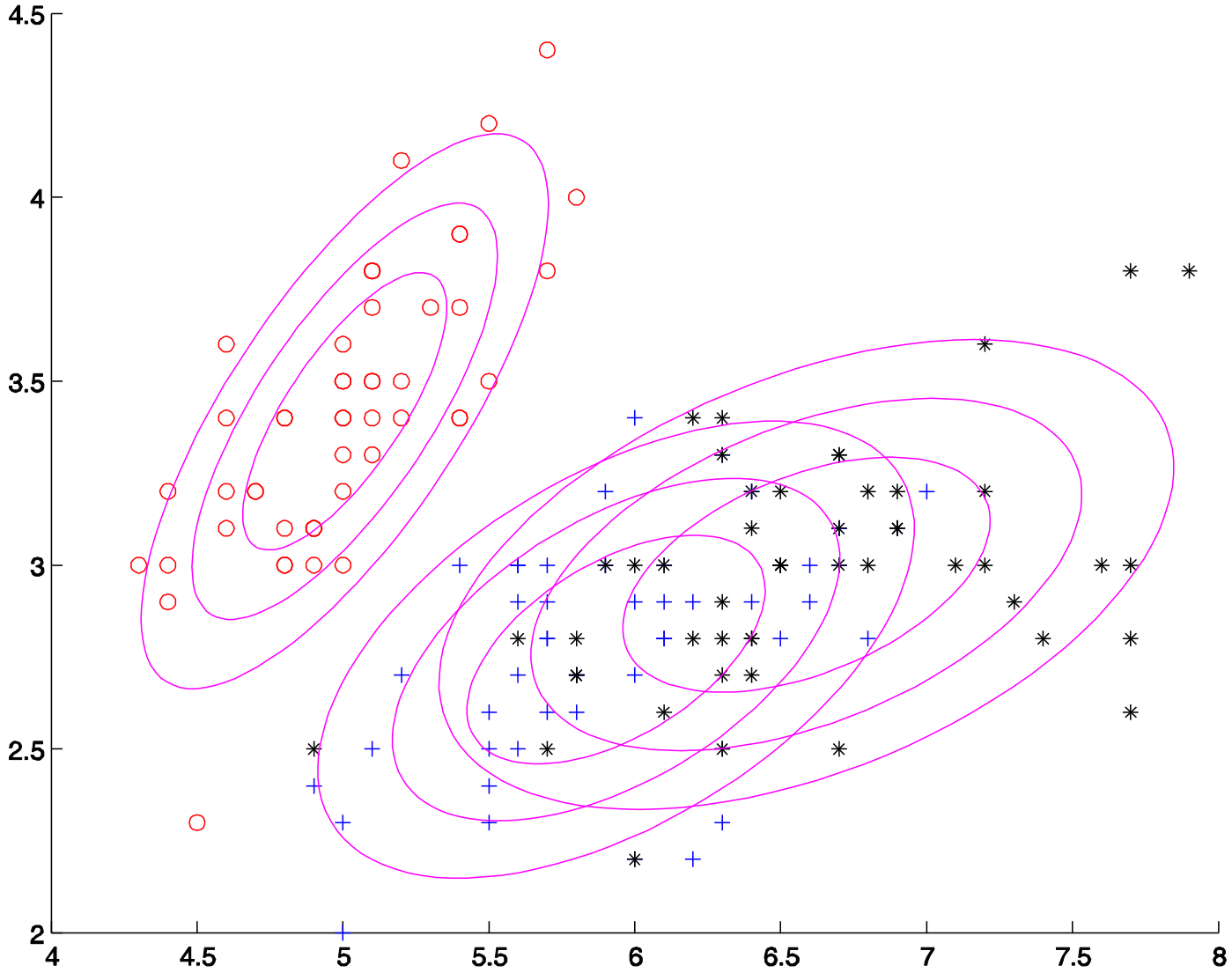


The other two features

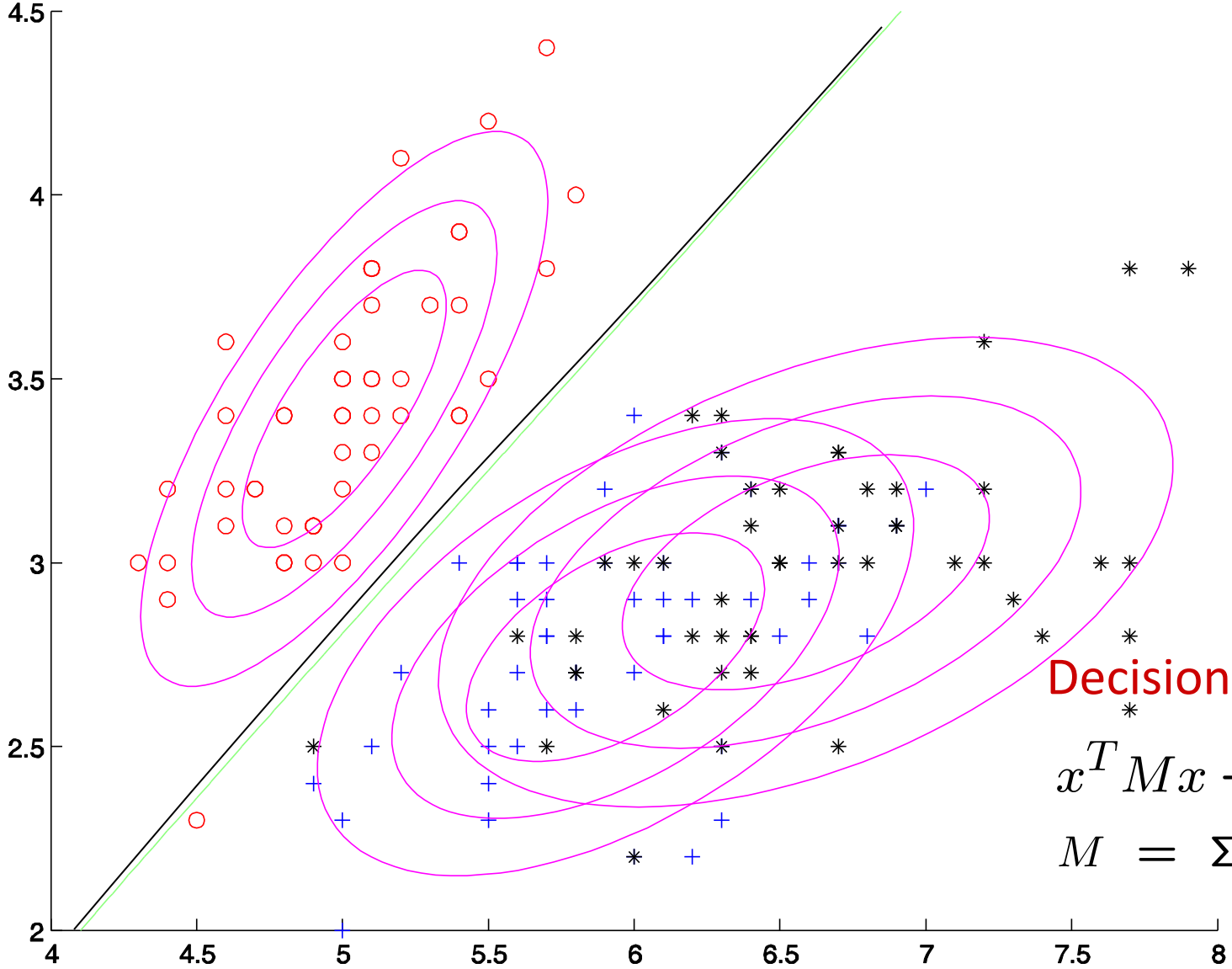
(less informative than the previous pair)



Generative model



Boundary: circle/plus

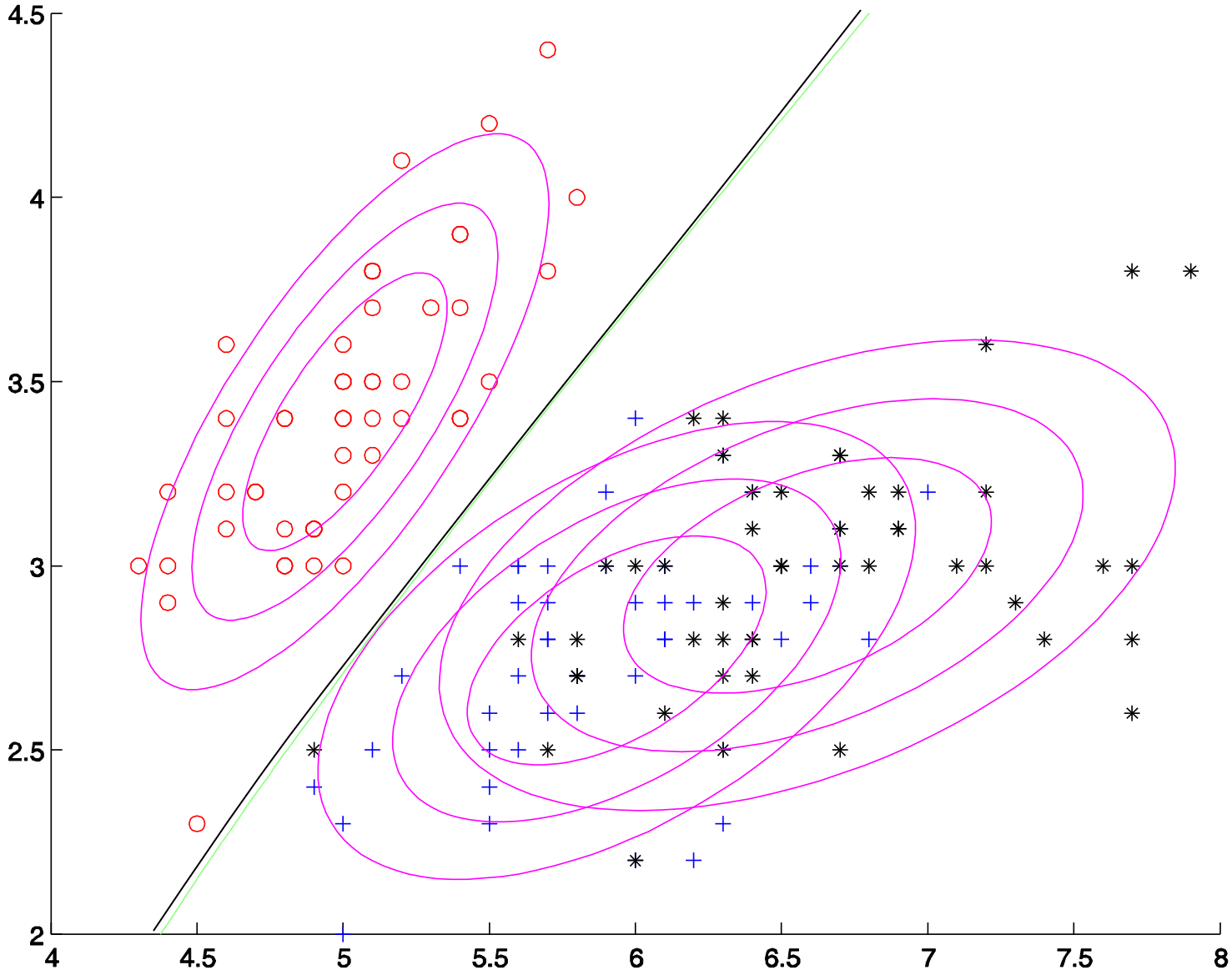


Decision Boundary:

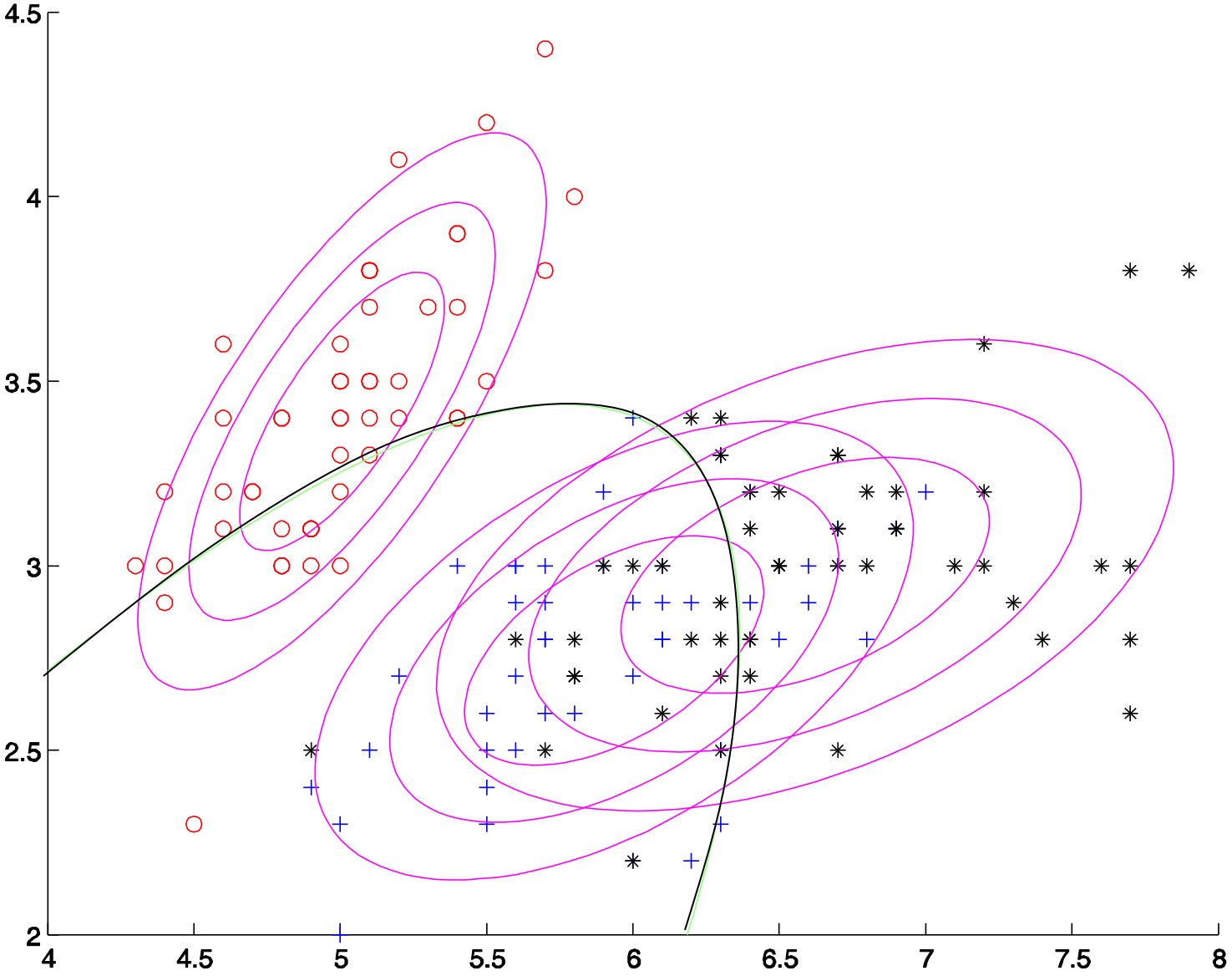
$$x^T M x + w \cdot x \geq \theta$$

$$M = \Sigma_2^{-1} - \Sigma_1^{-1}$$

Boundary: circle/star



Boundary: plus/star



Generative models

To each class, fit:

a density model (eg. Gaussian)

a mixing weight (proportion)

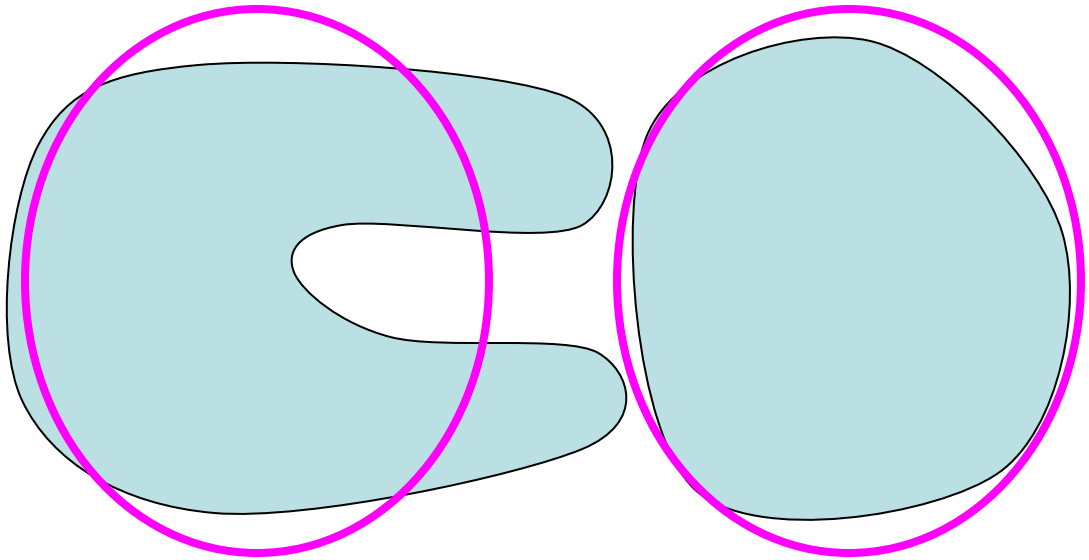
Advantages

1. multiclass is a breeze
2. robustly handles misclassified points and noise
3. can identify anomalous points
4. special density models (eg. Bayes nets, HMMs) can model temporal and other dependencies
5. returns not just a classification but also a confidence $P(y \mid x)$

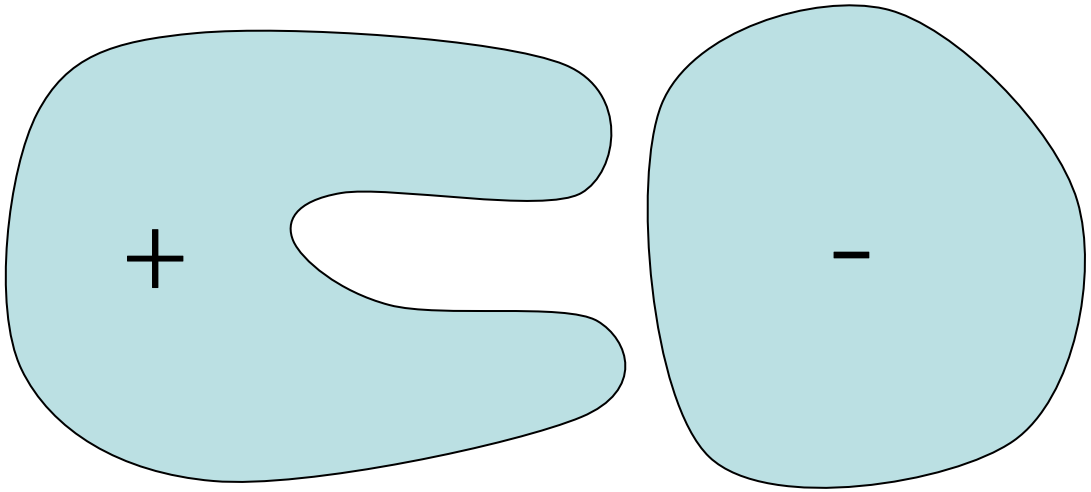
Generative models

Disadvantages

1. formula for P(y | x) assumes the density models are perfectly correct, but this is never true
2. if we are only interested in classification, shouldn't we model interclass boundaries rather than the individual classes?



Generative vs. discriminative



Generative:

Learn density models: $P^+ = P(x \mid y = +1)$, $P^- = P(x \mid y = -1)$,

weights: $\pi^+ = P(y = +1)$, $\pi^- = P(y = -1)$

This gives a full model of the joint PDF, P(x,y)

Use Bayes rule to get P(y|x)

Discriminative:

Learn P(y|x) directly

Or simply learn the decision boundary between X and Y

The two approaches

- There are two broad approaches to classification problems:

Discriminative (so far)

- model = a set of classifiers $\mathcal{F}$
- choose $\hat{f} \in \mathcal{F}$ that classifies training examples well
- label new inputs $\underline{x}$ based on $\hat{y} = \text{sign}(\, \hat{f}(\underline{x}) \,)$

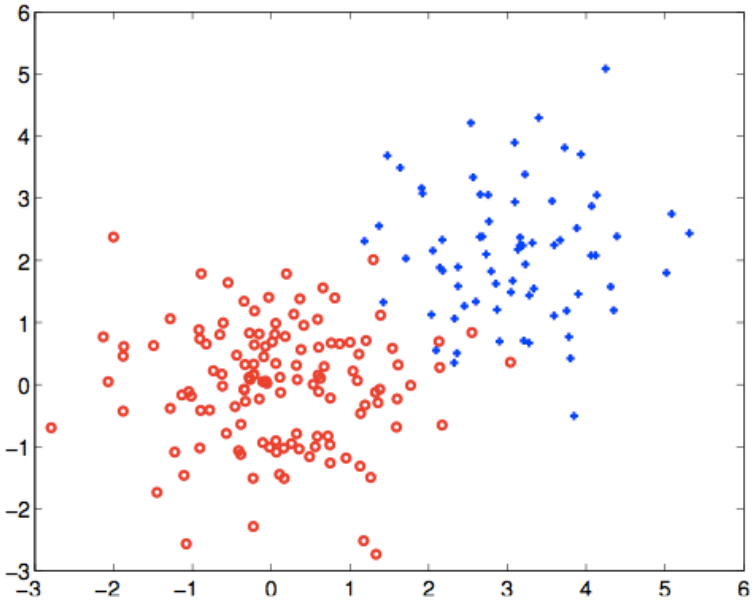
Generative (preview)

- model = a set of distributions $P(\underline{x}, y; \theta)$, $\theta \in \Theta$
- choose $P(\underline{x}, y; \hat{\theta})$ such that training examples are likely samples from this distribution
- label new inputs $\underline{x}$ based on $\hat{y} = \text{argmax}_y \, P(\underline{x}, y; \hat{\theta})$

Training/test data generation

- We assume that the training (and test) examples are drawn as samples (generated) from some unknown distribution

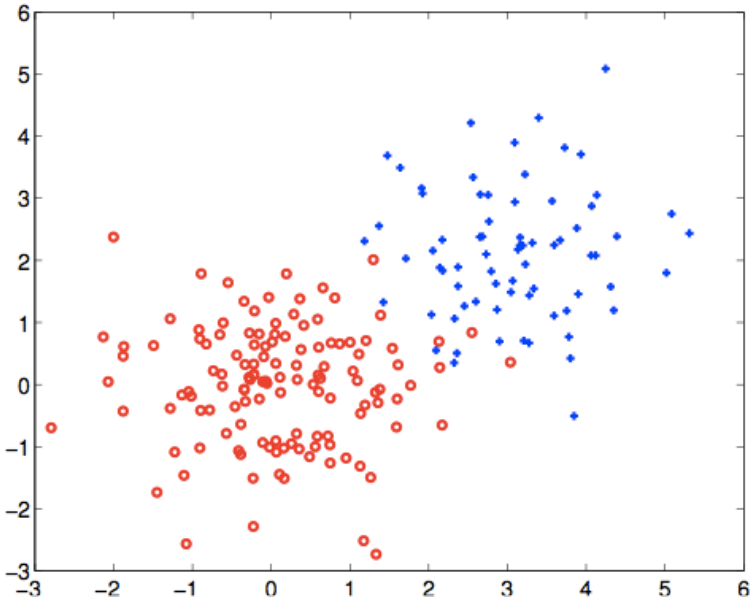
$$(\underline{x}, y) \sim P(\underline{x}, y)$$



Training/test data generation

- We assume that the training (and test) examples are drawn as samples (generated) from some unknown distribution

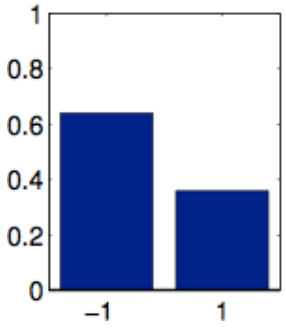
$$(\underline{x}, y) \sim P(\underline{x}, y) = P(\underline{x})P(y|\underline{x}) = P(\underline{x}|y)P(y)$$



- We can always think of these samples $(\underline{x}, y)$ as having been generated in two steps: first y, then $\underline{x}$ given y

$$y \sim P(y), \quad \underline{x} \sim P(\underline{x}|y)$$

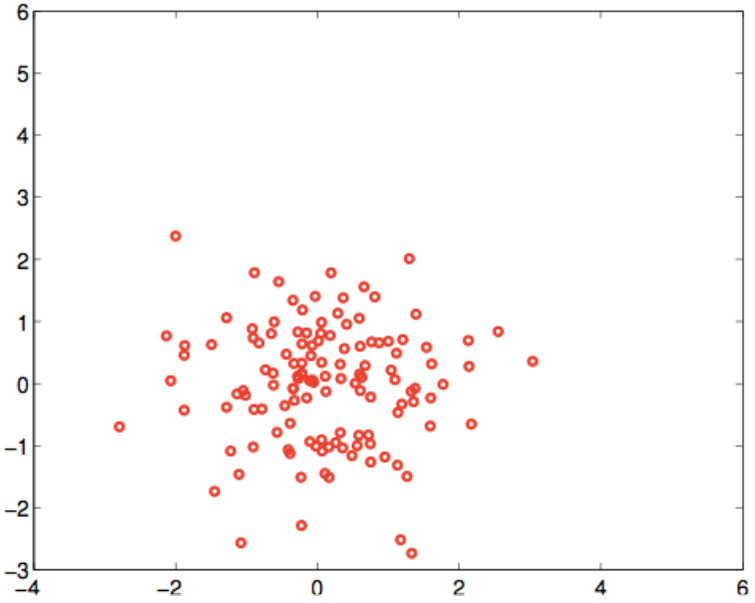
Training/test data generation

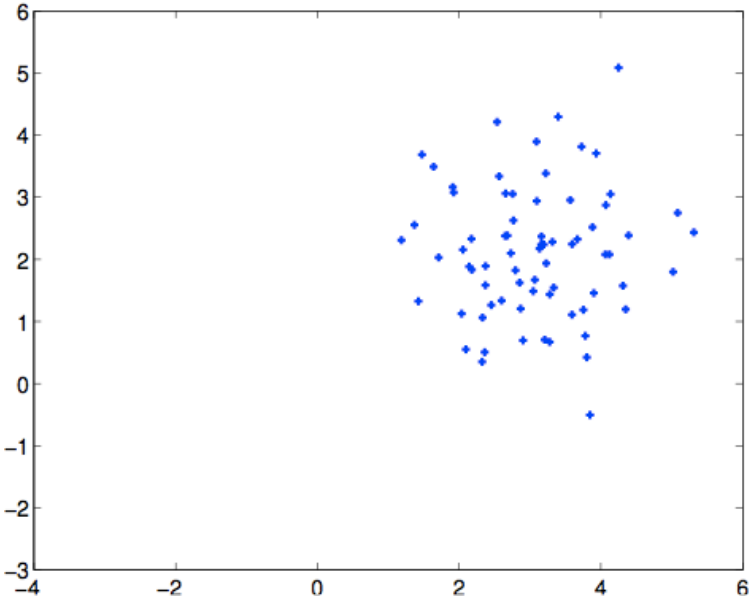


$P(y)$

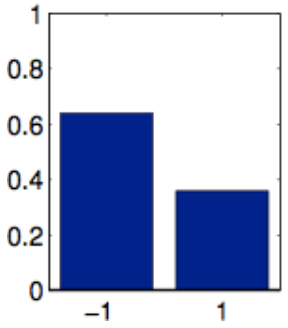
$P(\underline{x}|y=-1)$

$P(\underline{x}|y=1)$





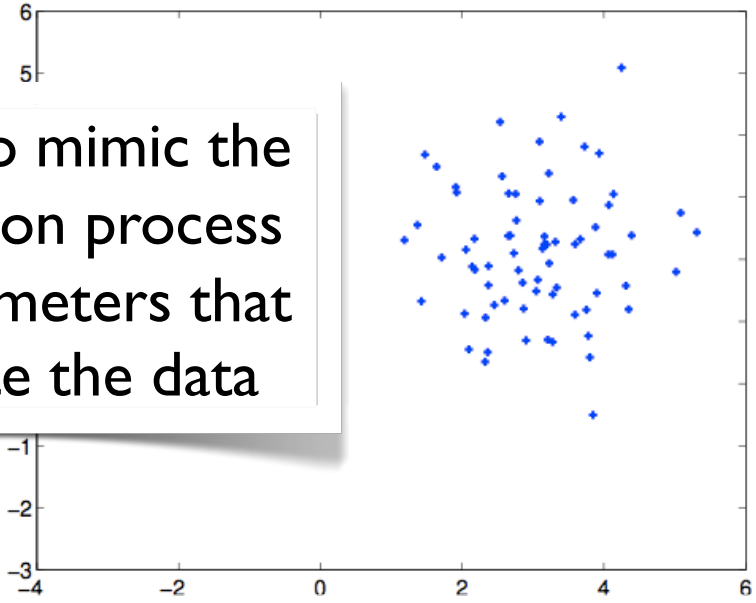
Generative modeling



$P(y;\hat{\theta})$

$P(\underline{x}|y=-1;\hat{\theta})$

$P(\underline{x}|y=1;\hat{\theta})$



Our goal is to mimic the data generation process and find parameters that best recreate the data

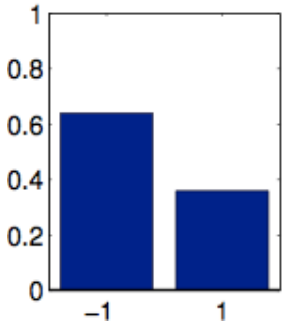
Generative modeling

- The label selection is simply a biased coin flip (Bernoulli distribution)

$$P(y = 1; \theta) = q$$

$$P(y = -1; \theta) = 1 - q$$

where q is included in θ (parameters that define the full distribution)

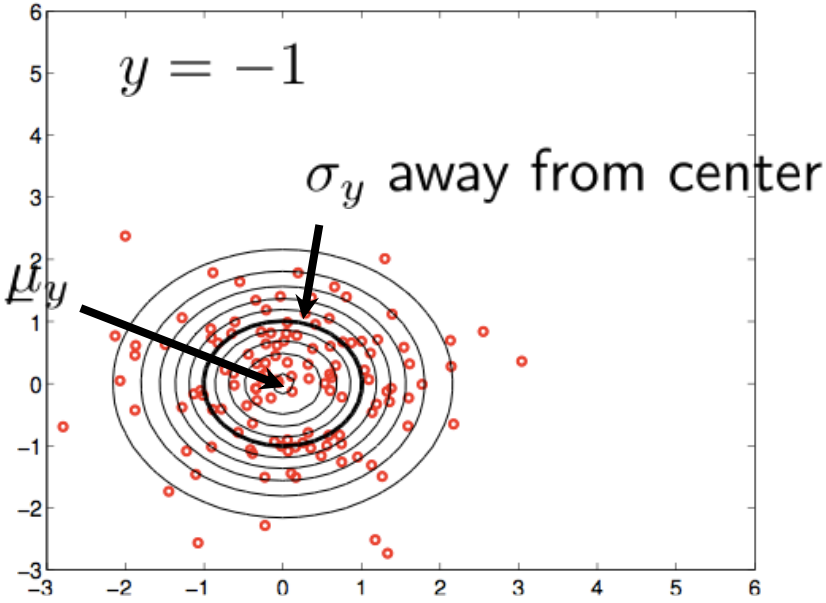


Generative modeling

- We can use simple spherical Gaussian models for the class-conditional distributions

$$P(\underline{x}|y;\theta) = N(\underline{x};\, \underline{\mu}_y, \sigma_y^2 I)$$

$$= \frac{1}{(2\pi\sigma_y^2)^{d/2}} \exp\left\{ -\frac{1}{2\sigma_y^2}\|\underline{x} - \underline{\mu}_y\|^2 \right\}$$



Generative modeling

- We can also use full Gaussian models $(d = dim(\underline{x}))$

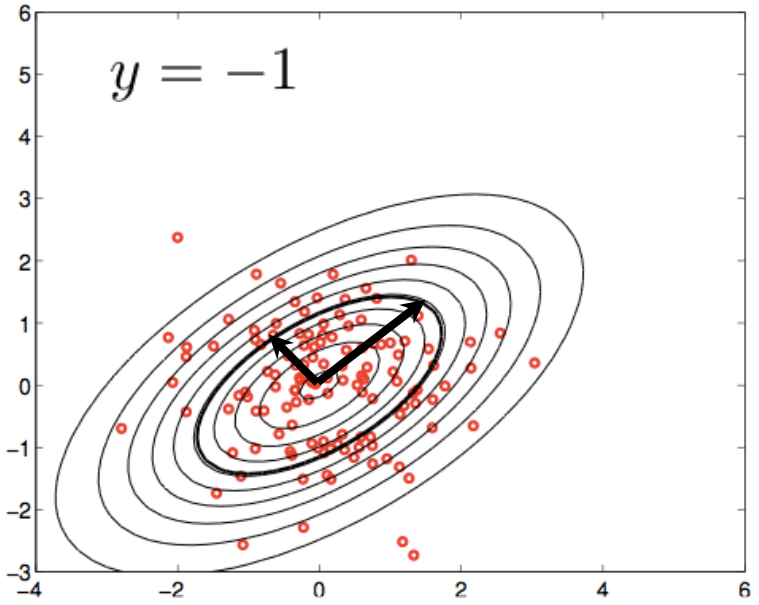
$$P(\underline{x}|y;\theta) = N(\underline{x};\ \underline{\mu}_y, \Sigma_y)$$

$$= \frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}} \exp\left\{ -\frac{1}{2}(\underline{x} - \underline{\mu}_y)^T \Sigma_y^{-1} (\underline{x} - \underline{\mu}_y) \right\}$$

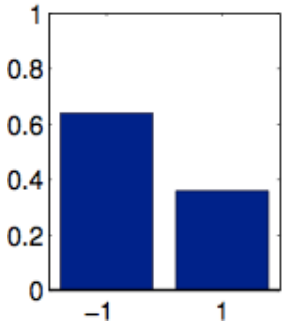
$$\Sigma_y = R \begin{bmatrix} \sigma_{y1}^2 & 0 \\ 0 & \sigma_{y2}^2 \end{bmatrix} R^T$$

rotation matrix

variances along the two principal axis



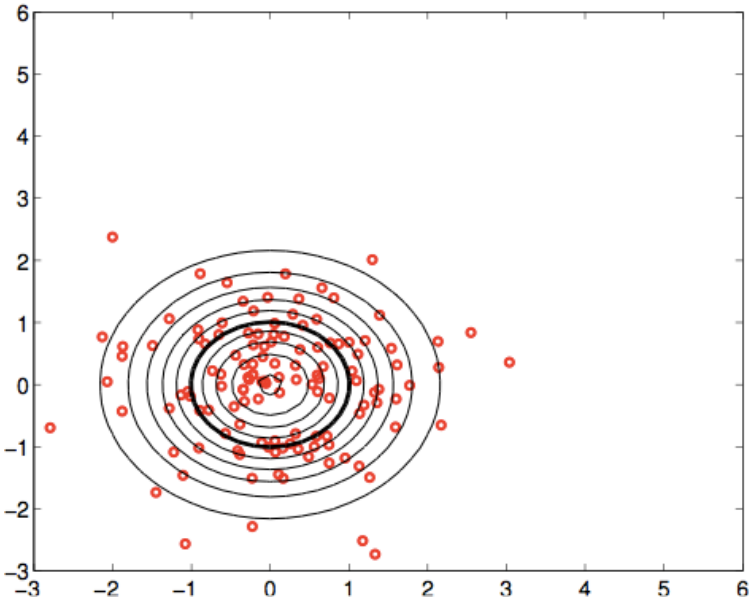
Generative modeling: estimation

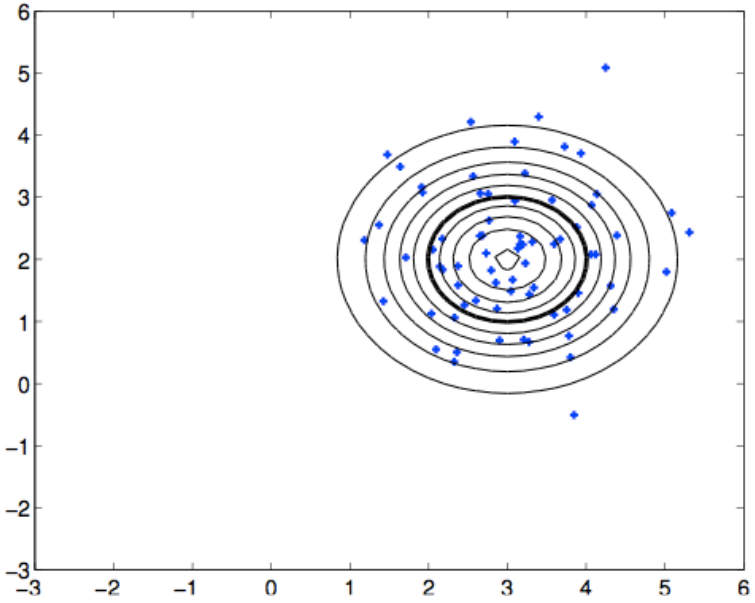


$P(y;\hat{\theta})$

$P(\underline{x}|y=-1;\hat{\theta})$

$P(\underline{x}|y=1;\hat{\theta})$





Maximum likelihood estimation

- Our parameterized Gaussian model is

$$P(\underline{x}, y; \theta) = P(\underline{x}|y; \theta)P(y; \theta) = N(\underline{x};\, \mu_y, \sigma_y^2 I)\, P(y; \theta)$$

- We find parameters $\theta = (\underline{\mu}_+, \underline{\mu}_-, \sigma_+^2, \sigma_-^2, q)$ that maximize the log-likelihood of the training data (examples and labels)

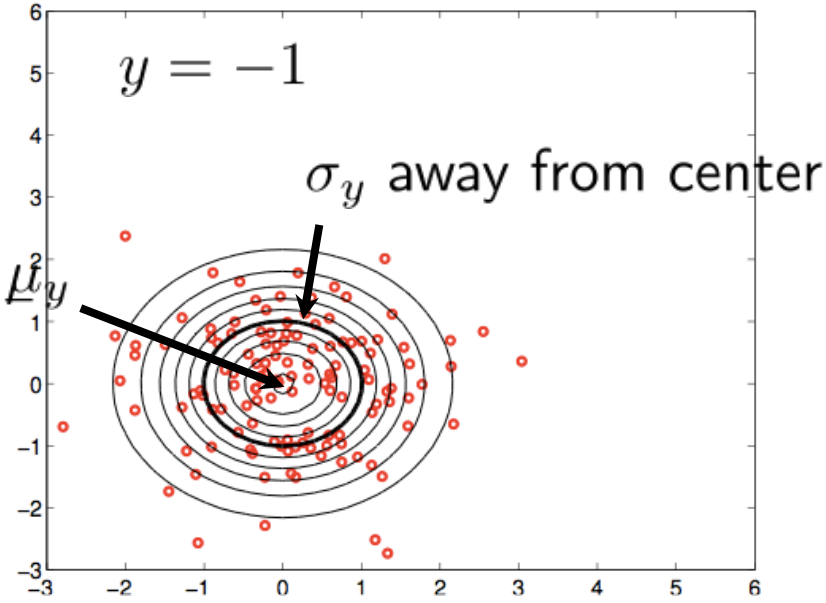
$$l(D; \theta) = \sum_{i=1}^{n} \log P(\underline{x}_i, y_i; \theta) = \sum_{i=1}^{n} \left[\log P(\underline{x}_i|y_i; \theta) + \log P(y_i; \theta) \right]$$

Generative modeling

- We can use simple spherical Gaussian models for the class-conditional distributions

$$P(\underline{x}|y;\theta) = N(\underline{x};\, \underline{\mu}_y, \sigma_y^2 I)$$

$$= \frac{1}{(2\pi\sigma_y^2)^{d/2}} \exp\left\{ -\frac{1}{2\sigma_y^2}\|\underline{x} - \underline{\mu}_y\|^2 \right\}$$



Maximum likelihood estimation

- Our parameterized Gaussian model is

$$P(\underline{x}, y; \theta) = P(\underline{x}|y; \theta)P(y; \theta) = N(\underline{x};\, \mu_y, \sigma_y^2 I)\, P(y; \theta)$$

- We find parameters $\theta = (\underline{\mu}_+, \underline{\mu}_-, \sigma_+^2, \sigma_-^2, q)$ that maximize the log-likelihood of the training data (examples and labels)

$$l(D; \theta) = \sum_{i=1}^{n} \log P(\underline{x}_i, y_i; \theta) = \sum_{i=1}^{n} \left[\log P(\underline{x}_i|y_i; \theta) + \log P(y_i; \theta) \right]$$

$$= \sum_{i=1}^{n} \left[-\frac{d}{2} \log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2} \|\underline{x}_i - \mu_{y_i}\|^2 \right] +$$

Maximum likelihood estimation

- Our parameterized Gaussian model is

$$P(\underline{x}, y; \theta) = P(\underline{x}|y; \theta)P(y; \theta) = N(\underline{x};\, \mu_y, \sigma_y^2 I)\, P(y; \theta)$$

- We find parameters $\theta = (\underline{\mu}_+, \underline{\mu}_-, \sigma_+^2, \sigma_-^2, q)$ that maximize the log-likelihood of the training data (examples and labels)

$$l(D; \theta) = \sum_{i=1}^{n} \log P(\underline{x}_i, y_i; \theta) = \sum_{i=1}^{n} \left[\log P(\underline{x}_i|y_i; \theta) + \log P(y_i; \theta) \right]$$

$$= \sum_{i=1}^{n} \left[-\frac{d}{2} \log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2} \|\underline{x}_i - \mu_{y_i}\|^2 \right] +$$

$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1) \log q + \delta(y_i, -1) \log(1 - q) \right]$$

indicator function

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right] + \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial q} l(D;\theta) = \frac{\sum_{i=1}^{n} \delta(y_i, 1)}{q} - \frac{\sum_{i=1}^{n} \delta(y_i, -1)}{1-q} = 0$$

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right]$$

$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial q} l(D;\theta) = \frac{\sum_{i=1}^{n} \delta(y_i, 1)}{q} - \frac{\sum_{i=1}^{n} \delta(y_i, -1)}{1-q} = 0$$

$$\Rightarrow \hat{q} = \frac{1}{n}\sum_{i=1}^{n} \delta(y_i, 1)$$

fraction of points labeled +1

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right]$$

$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial \underline{\mu}_y} l(D;\theta) =$$

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right]$$

$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial \underline{\mu}_y} l(D;\theta) = \sum_{i=1}^{n} \delta(y, y_i)\frac{1}{\sigma_y^2}(\underline{x}_i - \mu_y) = 0$$

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right]$$

$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial \underline{\mu}_y} l(D;\theta) = \sum_{i=1}^{n} \delta(y, y_i)\frac{1}{\sigma_y^2}(\underline{x}_i - \mu_y) = 0$$

$$\Rightarrow \hat{\mu}_y = \frac{1}{\sum_{i=1}^{n} \delta(y, y_i)} \sum_{i=1}^{n} \delta(y, y_i)\, \underline{x}_i$$

average of points in class y

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right]$$

$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial \sigma_y^2} l(D;\theta) = \sum_{i=1}^{n} \delta(y, y_i) \left[-\frac{d}{2\sigma_y^2} + \frac{1}{2\sigma_y^4}\|\underline{x}_i - \hat{\mu}_y\|^2 \right] = 0$$

Maximum likelihood estimation

$$l(D;\theta) = \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi\sigma_{y_i}^2) - \frac{1}{2\sigma_{y_i}^2}\|\underline{x}_i - \mu_{y_i}\|^2 \right]$$

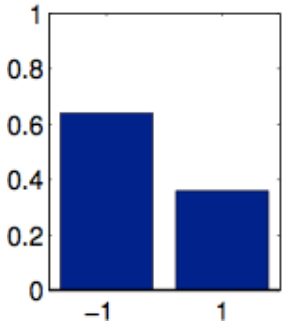
$$+ \sum_{i=1}^{n} \left[\delta(y_i, 1)\log q + \delta(y_i, -1)\log(1-q) \right]$$

$$\frac{\partial}{\partial \sigma_y^2} l(D;\theta) = \sum_{i=1}^{n} \delta(y, y_i) \left[-\frac{d}{2\sigma_y^2} + \frac{1}{2\sigma_y^4}\|\underline{x}_i - \hat{\mu}_y\|^2 \right] = 0$$

$$\Rightarrow \hat{\sigma}_y^2 = \frac{1}{d\sum_{i=1}^{n}\delta(y, y_i)} \sum_{i=1}^{n} \delta(y, y_i)\,\|\underline{x}_i - \hat{\mu}_y\|^2$$

average per dimension squared error in class y

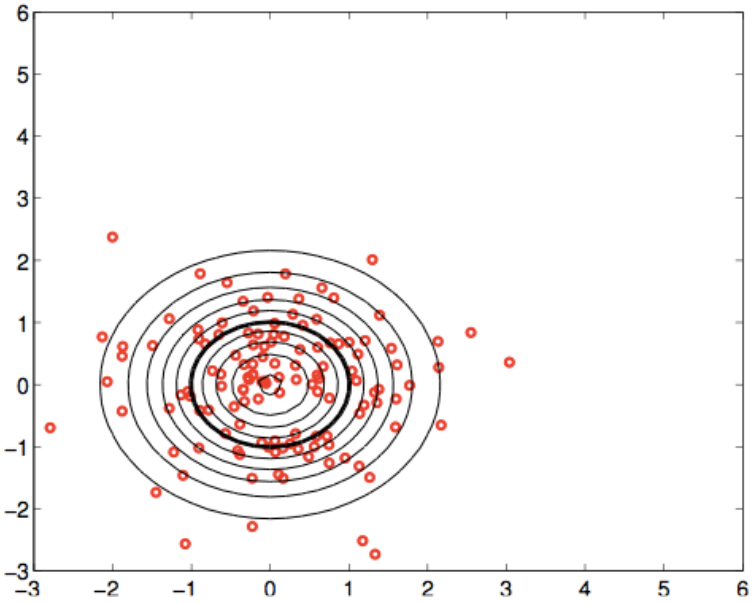
Generative modeling: classification

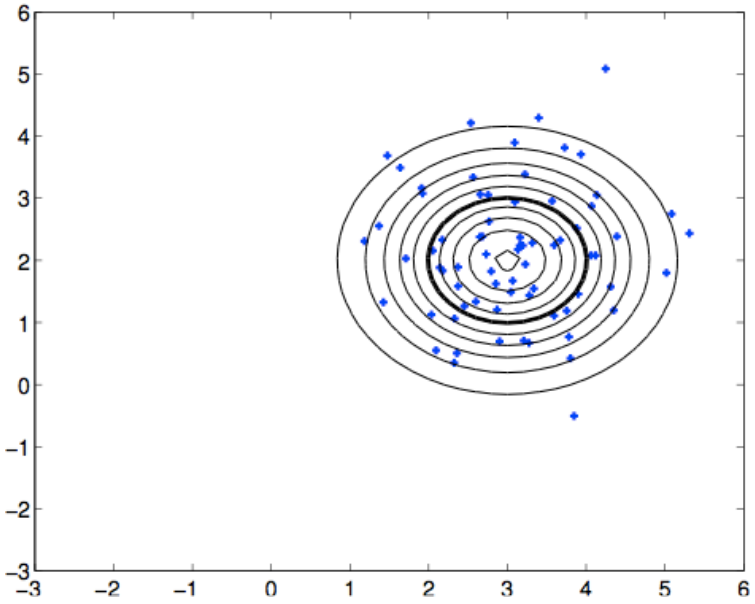


$P(y;\hat{\theta})$

$P(\underline{x}|y=-1;\hat{\theta})$

$P(\underline{x}|y=1;\hat{\theta})$





Decision boundary

- We predict the most likely label for $\underline{x}$ (cf. minimum probability of error classifier)

$$\hat{y} = \arg\max_{y} P(\underline{x}, y; \hat{\theta})$$

- The resulting decision boundary corresponds to all $\underline{x}$ such that

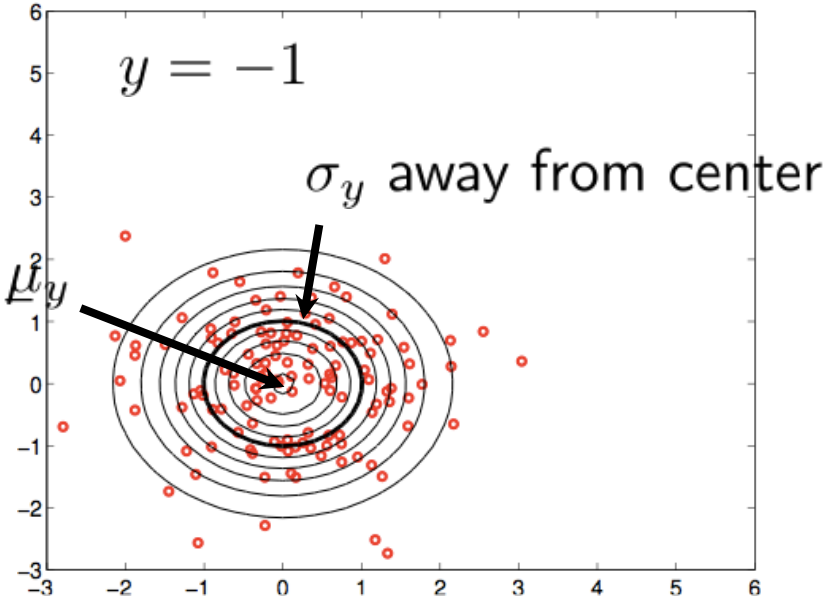
$$\log \frac{P(\underline{x}, y=1; \hat{\theta})}{P(\underline{x}, y=-1; \hat{\theta})} = \log \frac{P(y=1; \hat{\theta})}{P(y=-1; \hat{\theta})} + \log \frac{P(\underline{x}|y=1; \hat{\theta})}{P(\underline{x}|y=-1; \hat{\theta})} = 0$$

Generative modeling

- We can use simple spherical Gaussian models for the class-conditional distributions

$$P(\underline{x}|y;\theta) = N(\underline{x};\, \underline{\mu}_y, \sigma_y^2 I)$$

$$= \frac{1}{(2\pi\sigma_y^2)^{d/2}} \exp\left\{ -\frac{1}{2\sigma_y^2}\|\underline{x} - \underline{\mu}_y\|^2 \right\}$$



Decision boundary

- We predict the most likely label for $\underline{x}$ (cf. minimum probability of error classifier)

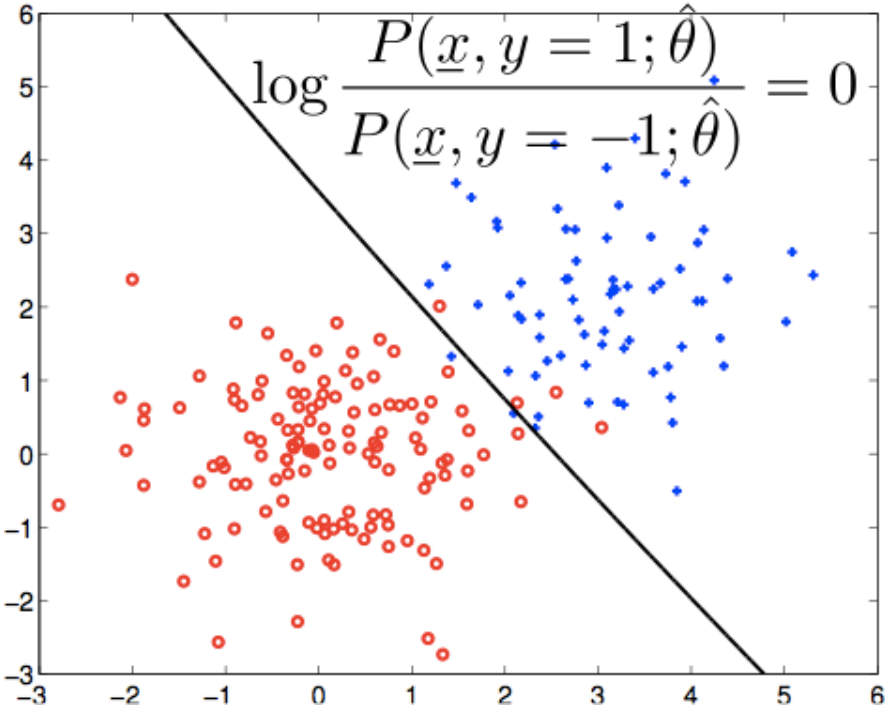
$$\hat{y} = \arg\max_y P(\underline{x}, y; \hat{\theta})$$

- The resulting decision boundary corresponds to all $\underline{x}$ such that

$$\begin{aligned}
\log \frac{P(\underline{x}, y=1; \hat{\theta})}{P(\underline{x}, y=-1; \hat{\theta})} &= \log \frac{P(y=1; \hat{\theta})}{P(y=-1; \hat{\theta})} + \log \frac{P(\underline{x}|y=1; \hat{\theta})}{P(\underline{x}|y=-1; \hat{\theta})} \\
&= \log \frac{\hat{q}}{1-\hat{q}} - \frac{d}{2} \log(\frac{\hat{\sigma}_{+1}^2}{\hat{\sigma}_{-1}^2}) \\
&\quad - \frac{1}{2\sigma_{+1}^2} \|\underline{x} - \underline{\mu}_{+1}\|^2 + \frac{1}{2\sigma_{-1}^2} \|\underline{x} - \underline{\mu}_{-1}\|^2 \\
&= 0
\end{aligned}$$

- This is linear in $\underline{x}$ if $\sigma_{+1}^2 = \sigma_{-1}^2$ (otherwise quadratic)

Decision boundary

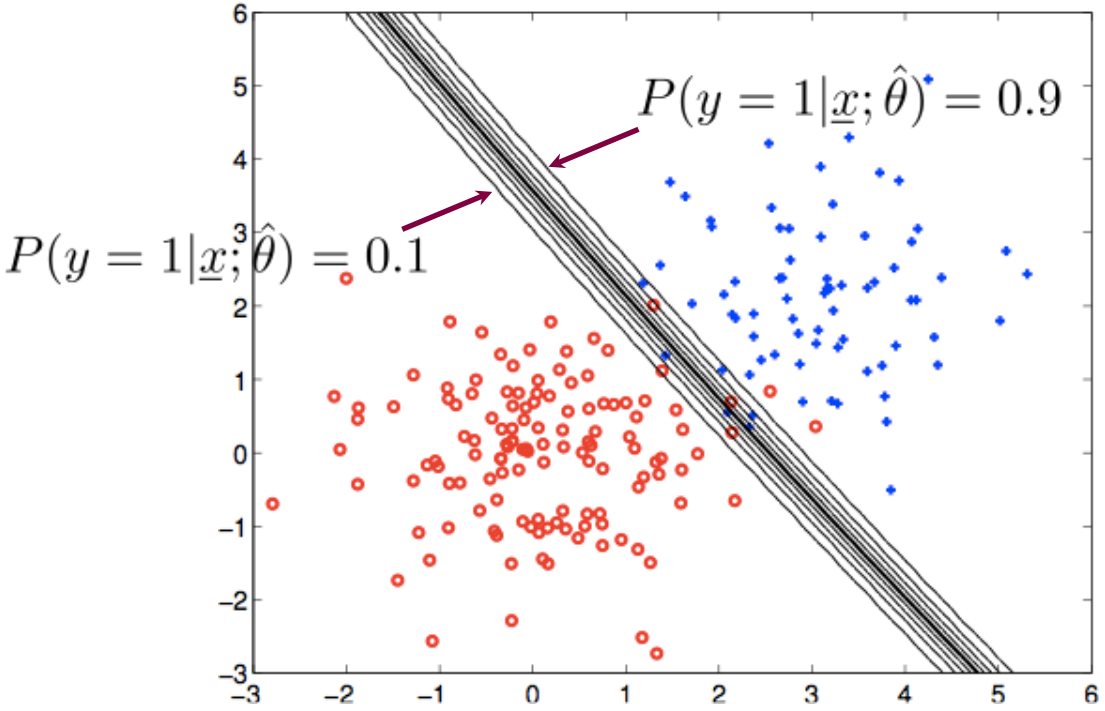


- This is very close to the optimal since the data were generated from two Gaussians with the same variance

Probabilistic predictions

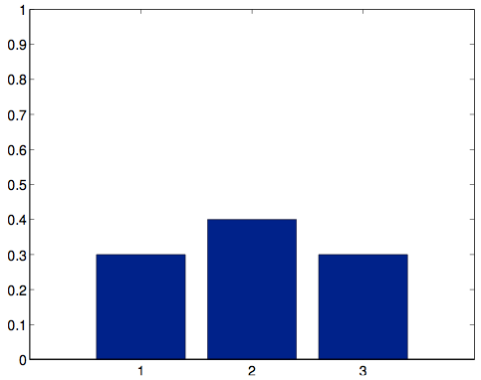
- The model also permits us to evaluate probabilities over the possible class labels such as

$$P(y = 1|\underline{x}; \hat{\theta}) = \frac{P(\underline{x}, y = 1; \hat{\theta})}{\sum_{y' \in \{-1,1\}} P(\underline{x}, y'; \hat{\theta})}$$



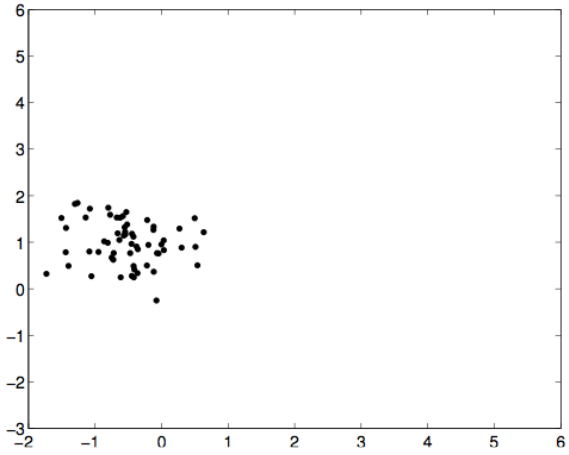
A mixture model

three types $z = 1, 2, 3$

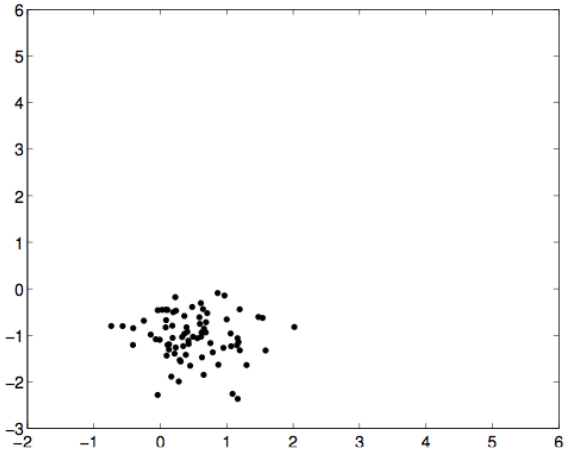


$P(z; \hat{\theta})$

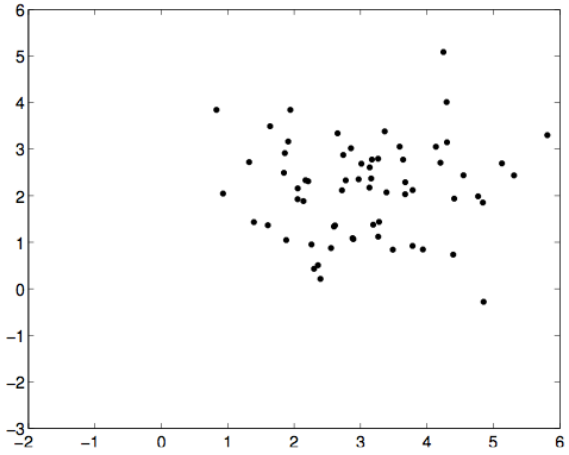
$P(\underline{x}|z = 1; \hat{\theta})$



$P(\underline{x}|z = 2; \hat{\theta})$

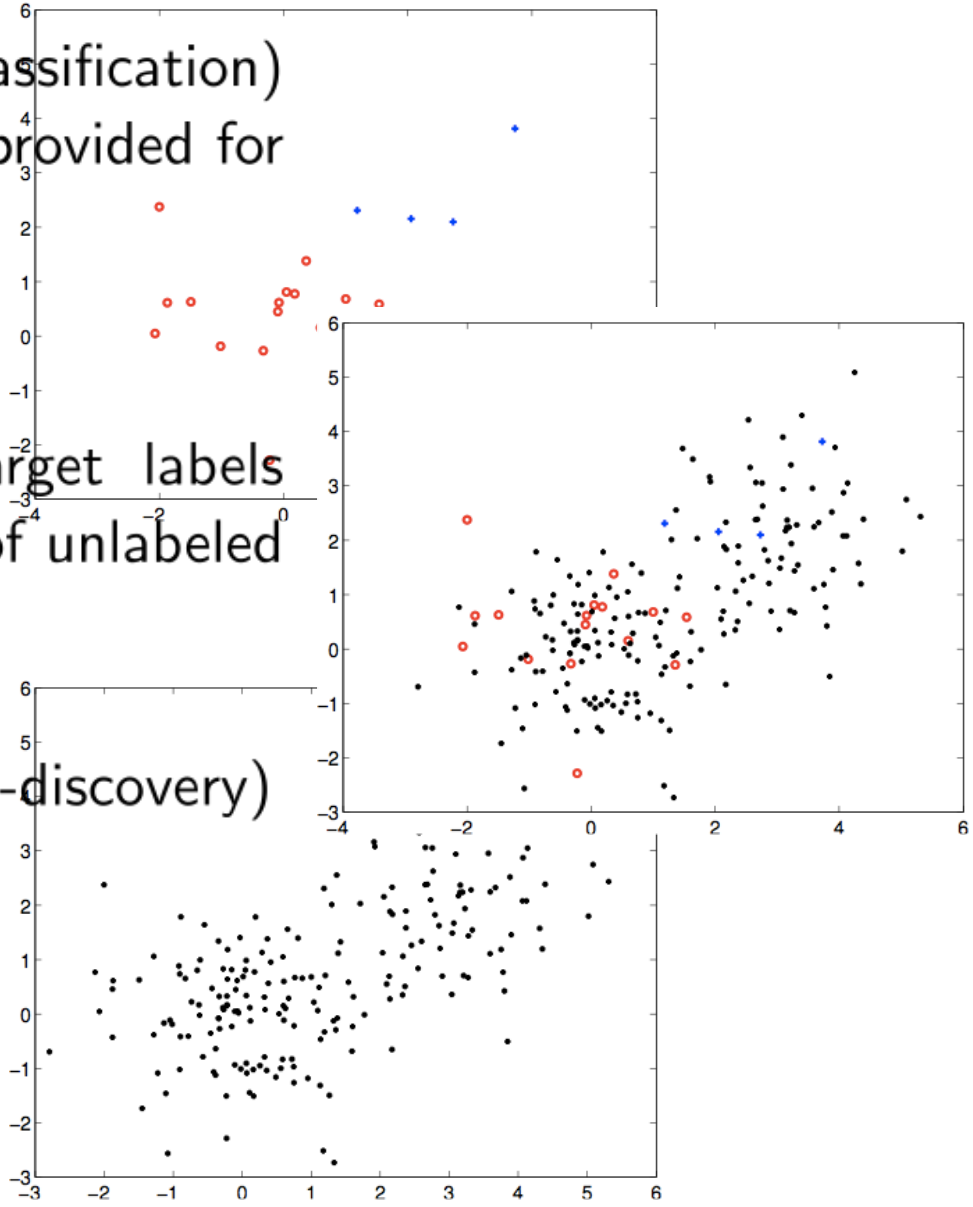


$P(\underline{x}|z = 3; \hat{\theta})$



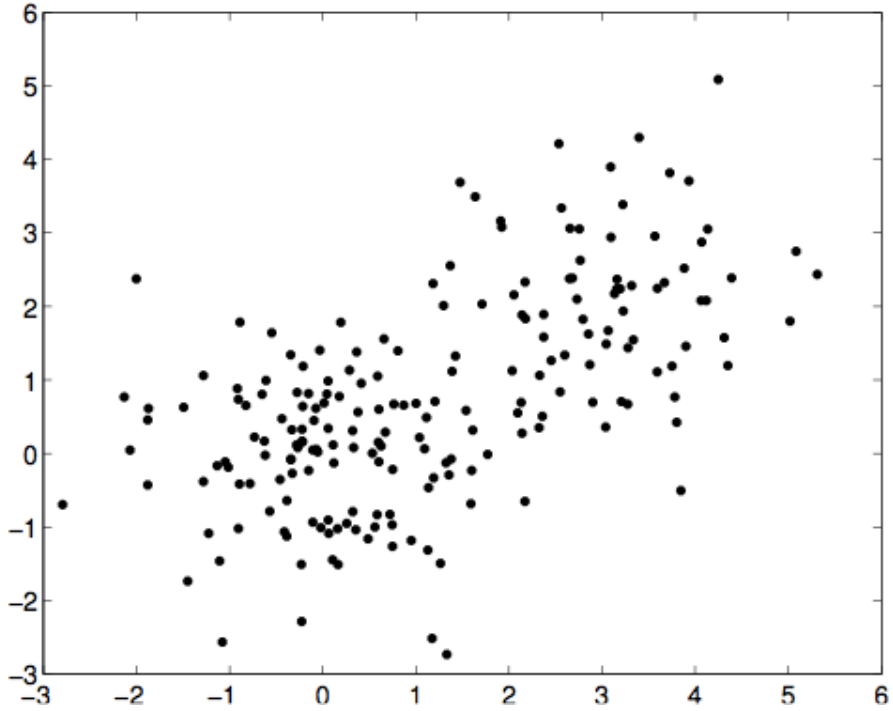
Classification - class discovery

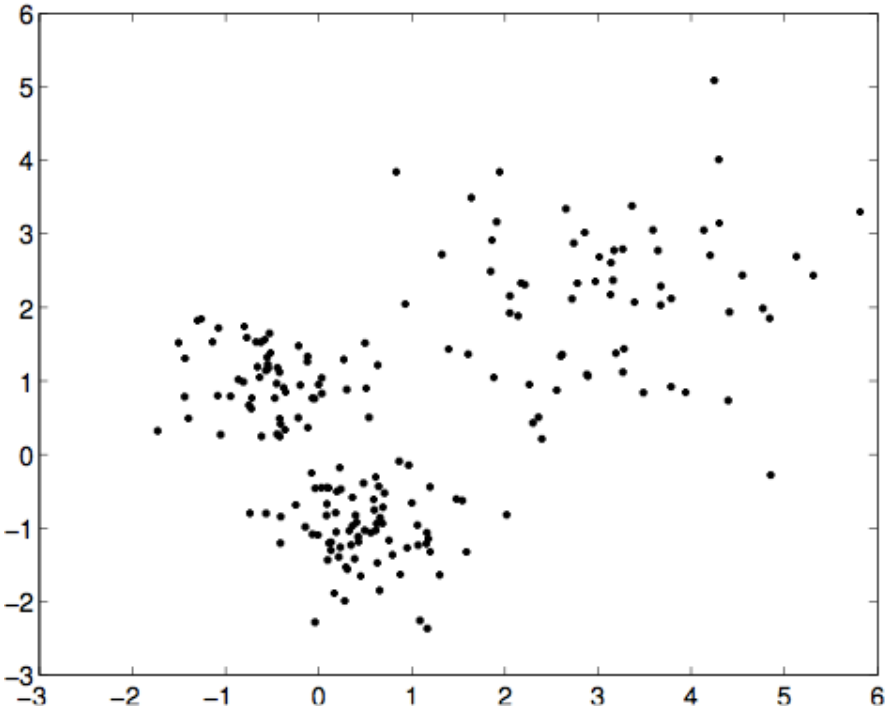
- Supervised learning (e.g., classification)
  - target labels (responses) provided for each example
- Semi-supervised learning
  - in addition to a few target labels we have a large number of unlabeled examples
- Unsupervised learning (class-discovery)
  - all examples are unlabeled



Class or type discovery

- There may be many underlying classes or "types" of examples
- Each class may itself consist of different sub-types
- Our goal is to discover how the data may have been generated from a "mixture" of underlying types





Mixtures and latent variables

- A k-component mixture model with parameters θ

$$P(\underline{x}, z; \theta) = P(\underline{x}|z; \theta)P(z; \theta)$$

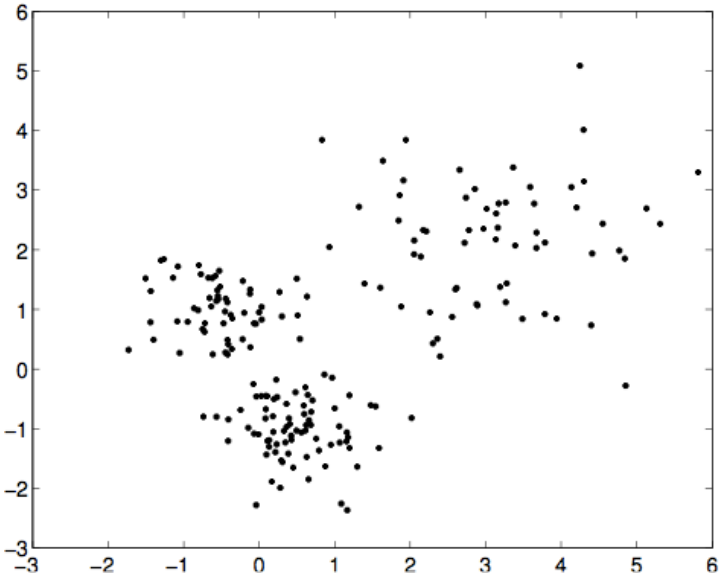
$$z \sim P(z; \theta) \text{ sample type } z = 1, \ldots, k$$

$$\underline{x} \sim P(\underline{x}|z; \theta) \text{ sample } \underline{x} \text{ given type } z$$

- The type z is latent (unobserved). The marginal distribution over $\underline{x}$ is given by the mixture

$$P(\underline{x}; \theta) = \sum_{z=1}^{k} P(\underline{x}|z; \theta)P(z; \theta)$$

$$\underline{x} \sim P(\underline{x}; \theta) \text{ sample } \underline{x}$$



Mixture estimation

- Consider a $k-$component mixture of spherical Gaussians

$$P(z;\theta) = \theta_z, \quad \sum_{z=1}^{k} \theta_z = 1$$

$$P(\underline{x}|z;\theta) = N(\underline{x};\, \underline{\mu}_z, \sigma_z^2 I)$$

where $\theta = \{\underline{\mu}_1, \ldots, \underline{\mu}_k, \sigma_1^2, \ldots, \sigma_k^2, \theta_1, \ldots, \theta_k\}$.

- We estimate the mixture parameters by maximizing the log-likelihood of the observed data $D = \{\underline{x}_1, \ldots, \underline{x}_n\}$

$$l(D;\theta) = \sum_{i=1}^{n} \log P(\underline{x}_i;\theta) = \sum_{i=1}^{n} \log \sum_{z=1}^{k} P(\underline{x}_i|z;\theta) P(z;\theta)$$

- This would be much easier if we knew the value of z corresponding to each $\underline{x}_i$ (complete data).

Mixture estimation: complete data

- If we knew the type z_i corresponding to each $\underline{x}_i$ then we would maximize the *complete log-likelihood*

$$\sum_{i=1}^{n} \log P(\underline{x}_i, z_i; \theta) = \sum_{i=1}^{n} \sum_{z=1}^{k} \delta(z, z_i) \log P(\underline{x}_i, z; \theta)$$

Mixture estimation: complete data

- If we knew the type z_i corresponding to each $\underline{x}_i$ then we would maximize the *complete log-likelihood*

$$\sum_{i=1}^{n} \log P(\underline{x}_i, z_i; \theta) = \sum_{i=1}^{n} \sum_{z=1}^{k} \delta(z, z_i) \log P(\underline{x}_i, z; \theta)$$

- Just as in the classification context, finding the maximizing parameters is simple (based on separately estimating the Gaussians from the points assigned to them)

$$\hat{\theta}_z = \frac{\sum_{i=1}^{n} \delta(z, z_i)}{n}$$

$$\hat{\mu}_z = \frac{1}{\sum_{i=1}^{n} \delta(z, z_i)} \sum_{i=1}^{n} \delta(z, z_i)\, \underline{x}_i$$

$$\hat{\sigma}_z^2 = \frac{1}{d \sum_{i=1}^{n} \delta(z, z_i)} \sum_{i=1}^{n} \delta(z, z_i) \left\| \underline{x}_i - \hat{\mu}_z \right\|^2$$

The EM-algorithm

Initialize: select the initial parameters $\theta^{(0)}$

E-step: fix the current parameters $\theta^{(m)}$, evaluate posterior assignments (divide each point softly across the mixture components)

$$p(z|i) = P(z|\underline{x}_i; \theta^{(m)}), \quad z = 1, \ldots, k, \quad i = 1, \ldots, n$$

M-step: fix the posterior assignments $p(z|i)$, find new parameters $\theta^{(m+1)}$ by maximizing the expected complete log-likelihood

$$\sum_{i=1}^{n} \sum_{z=1}^{k} p(z|i) \log P(\underline{x}_i, z; \theta)$$

with respect to θ

Mixture estimation: E-step

- The difficulty now is that we have to *complete* the data by inferring values for the missing types z_i

Mixture estimation: E-step

- The difficulty now is that we have to *complete* the data by inferring values for the missing types z_i
- Given the mixture parameters θ our best guess about the value of z_i corresponding to $\underline{x}_i$ is given by the posterior

$$p(z|i) = P(z|\underline{x}_i;\theta) = \frac{P(\underline{x}_i|z;\theta)P(z;\theta)}{\sum_{z'=1}^{k} P(\underline{x}_i|z';\theta)P(z';\theta)}$$

  Note that these "soft" posterior assignments $p(z|i)$ depend on the current setting of the parameters θ

Mixture estimation: E-step

- The difficulty now is that we have to *complete* the data by inferring values for the missing types z_i
- Given the mixture parameters θ, our best guess about the value of z_i corresponding to $\underline{x}_i$ is given by the posterior

$$p(z|i) = P(z|\underline{x}_i;\theta) = \frac{P(\underline{x}_i|z;\theta)P(z;\theta)}{\sum_{z'=1}^{k} P(\underline{x}_i|z';\theta)P(z';\theta)}$$

  Note that these "soft" posterior assignments $p(z|i)$ depend on the current setting of the parameters θ

- The soft assignments help us stochastically fill-in the missing type values. We therefore maximize the *expected log-likelihood* with respect to θ for fixed soft assignments

$$\sum_{i=1}^{n}\sum_{z=1}^{k} p(z|i)\log P(\underline{x}_i, z;\theta)$$

Mixture estimation: M-step

- As we have filled in the missing values, the estimation step is again easy. We simply use the "soft" posterior assignments $p(z|i)$ in place of the indicators $\delta(z, z_i)$

$$\hat{\theta}_z = \frac{\sum_{i=1}^{n} p(z|i)}{n}$$

$$\hat{\mu}_z = \frac{1}{\sum_{i=1}^{n} p(z|i)} \sum_{i=1}^{n} p(z|i)\, \underline{x}_i$$

$$\hat{\sigma}_z^2 = \frac{1}{d \sum_{i=1}^{n} p(z|i)} \sum_{i=1}^{n} p(z|i)\, \|\underline{x}_i - \hat{\underline{\mu}}_z\|^2$$

- These new parameters $\hat{\theta}$ can in turn be used to derive better soft assignments and so on.

[E-step]

The EM-algorithm

Initialize: select the initial parameters $\theta^{(0)}$

E-step: fix the current parameters $\theta^{(m)}$, evaluate posterior assignments (divide each point softly across the mixture components)

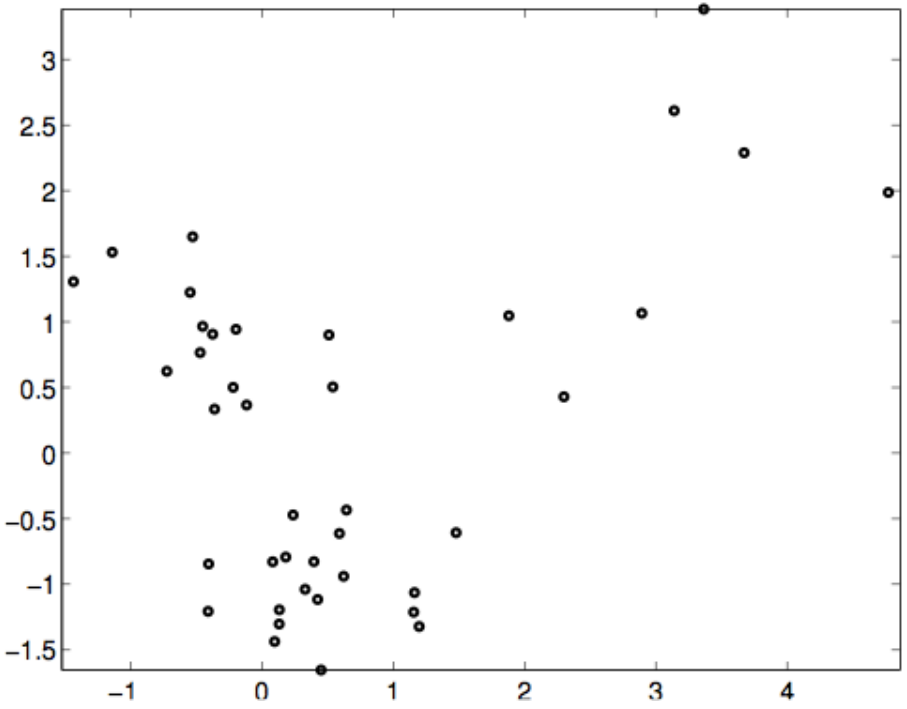
$$p(z|i) = P(z|\underline{x}_i; \theta^{(m)}), \quad z = 1, \ldots, k, \quad i = 1, \ldots, n$$

M-step: fix the posterior assignments $p(z|i)$, find new parameters $\theta^{(m+1)}$ by maximizing the expected complete log-likelihood

$$\sum_{i=1}^{n} \sum_{z=1}^{k} p(z|i) \log P(\underline{x}_i, z; \theta)$$

with respect to θ

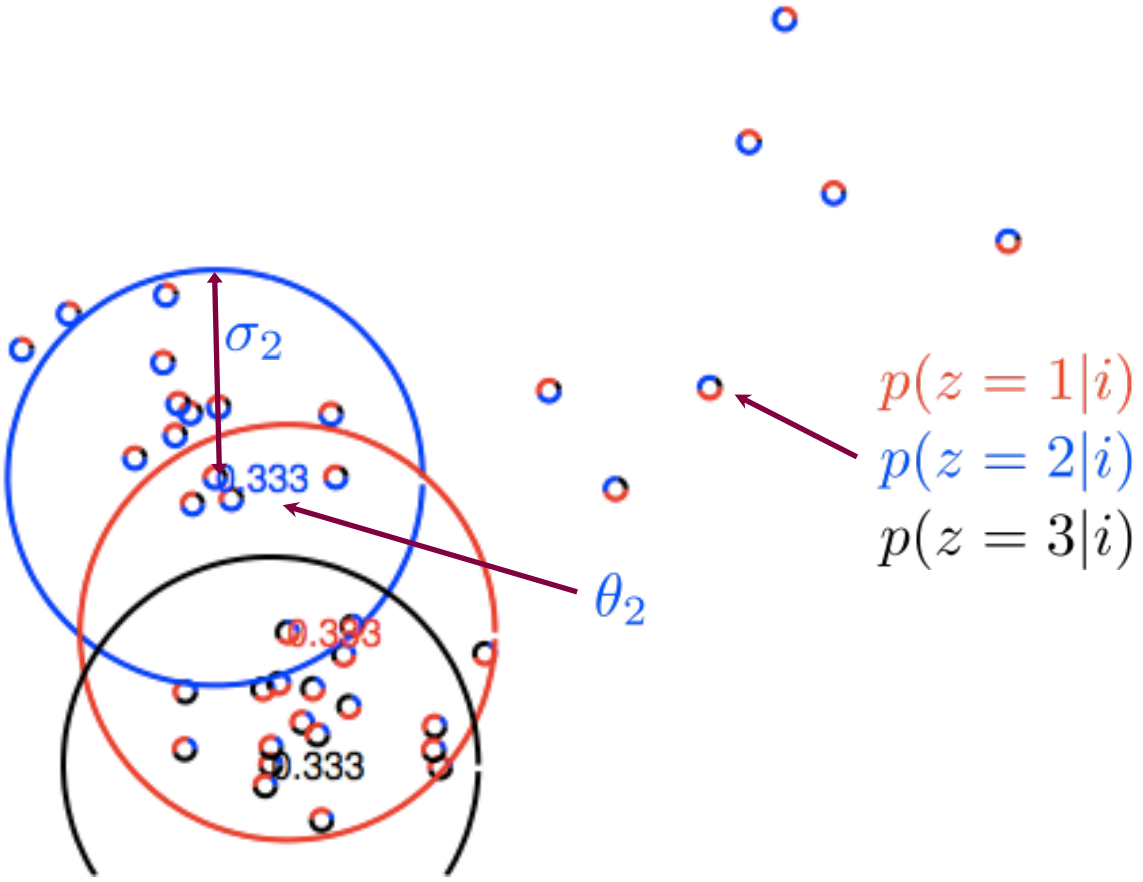
Mixture of Gaussians example



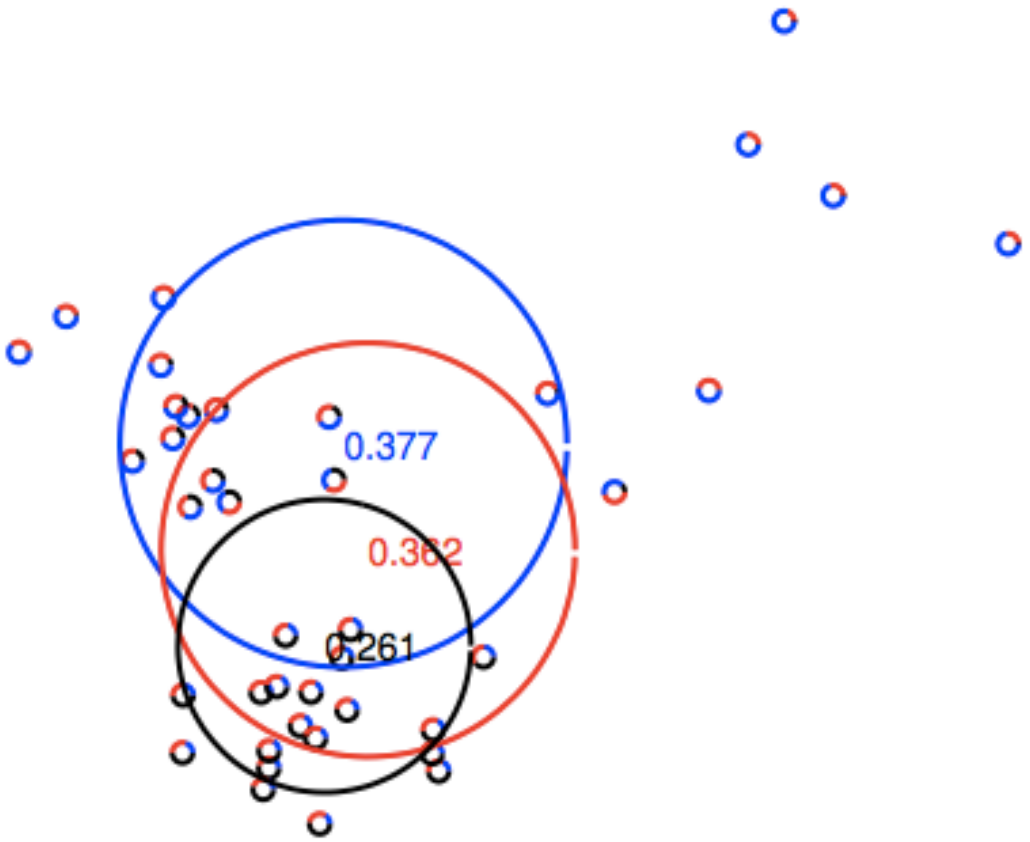
- The EM-algorithm is iterative and requires an initialization
  - $\theta_z = 1/k,\ z = 1, \ldots, k$
  - each μ_z is set to a randomly selected point
  - each σ_z^2 is set to the mean squared distance of points to the overall mean

Mixture of Gaussians example

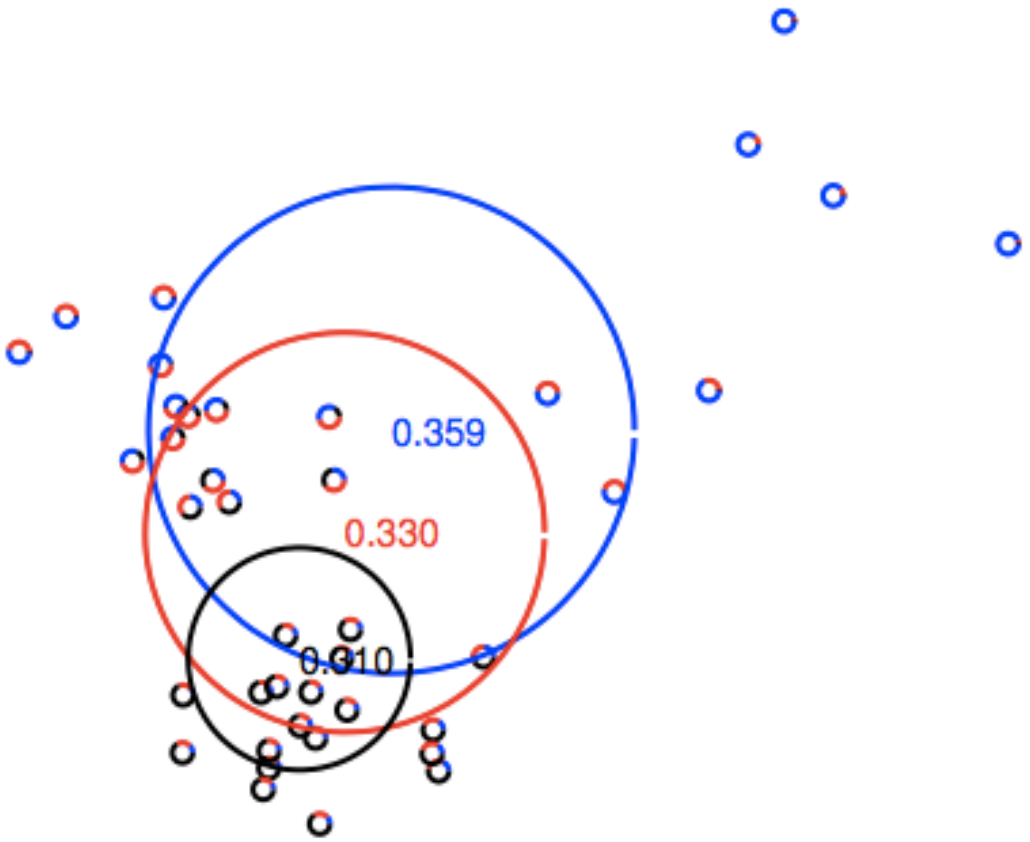
- initial 3-component mixture



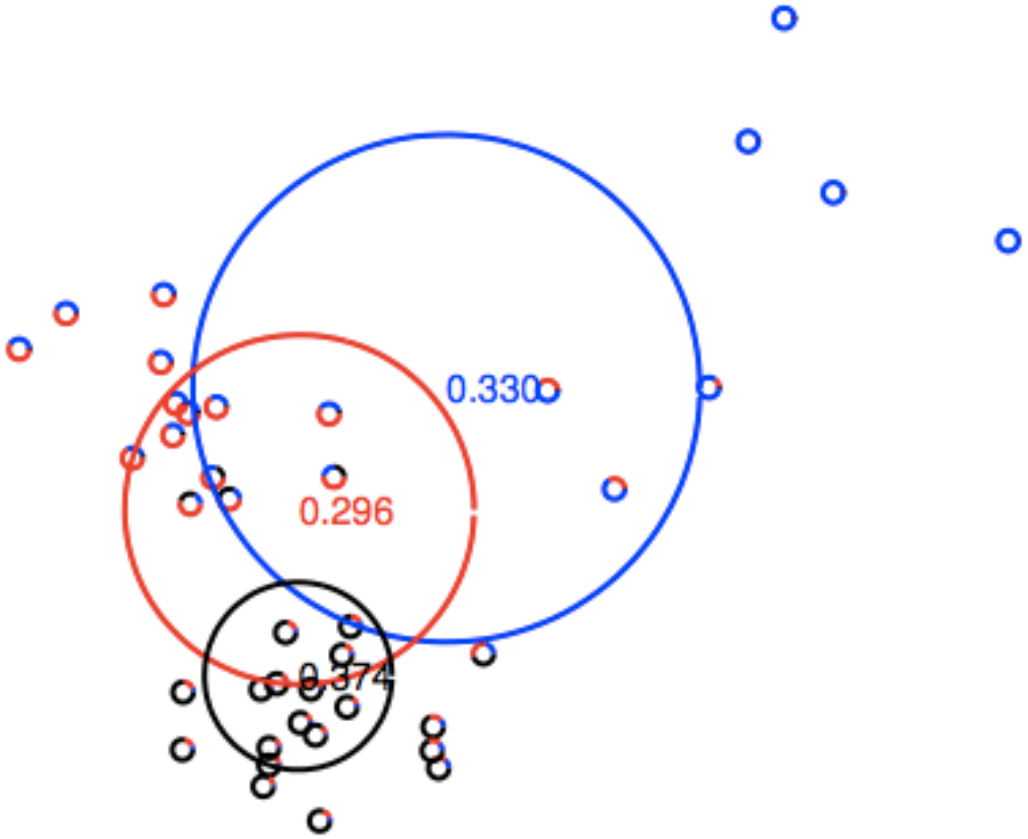
Mixture of Gaussians example



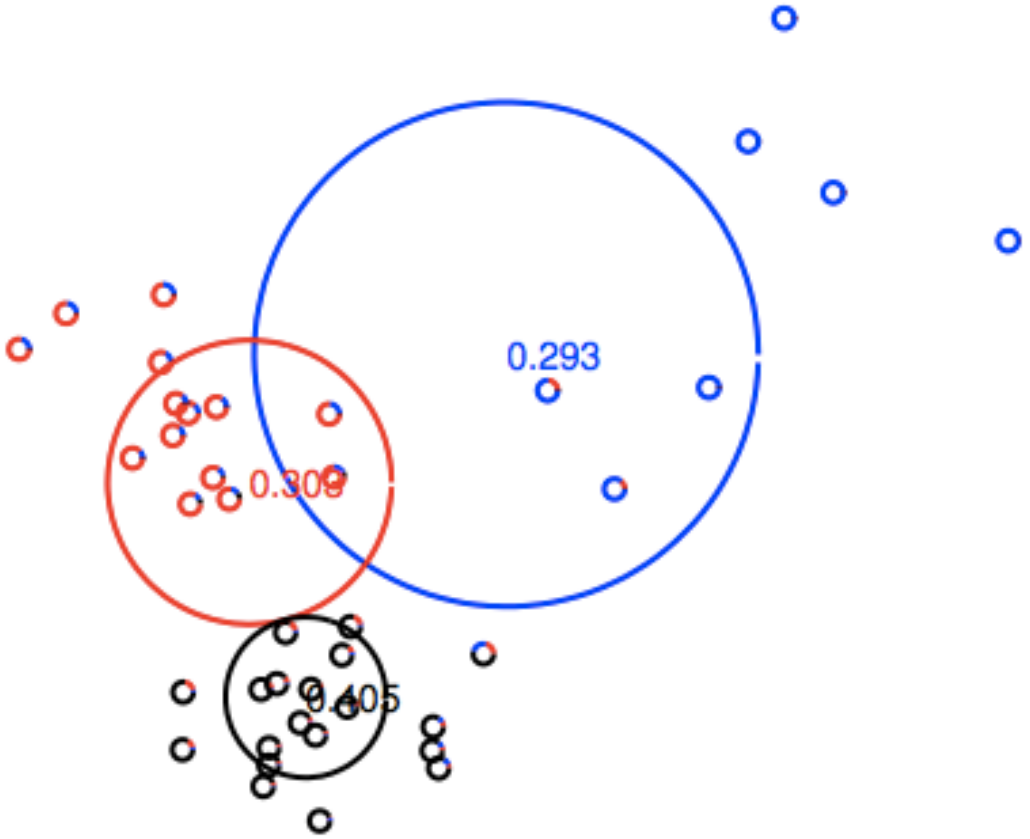
Mixture of Gaussians example



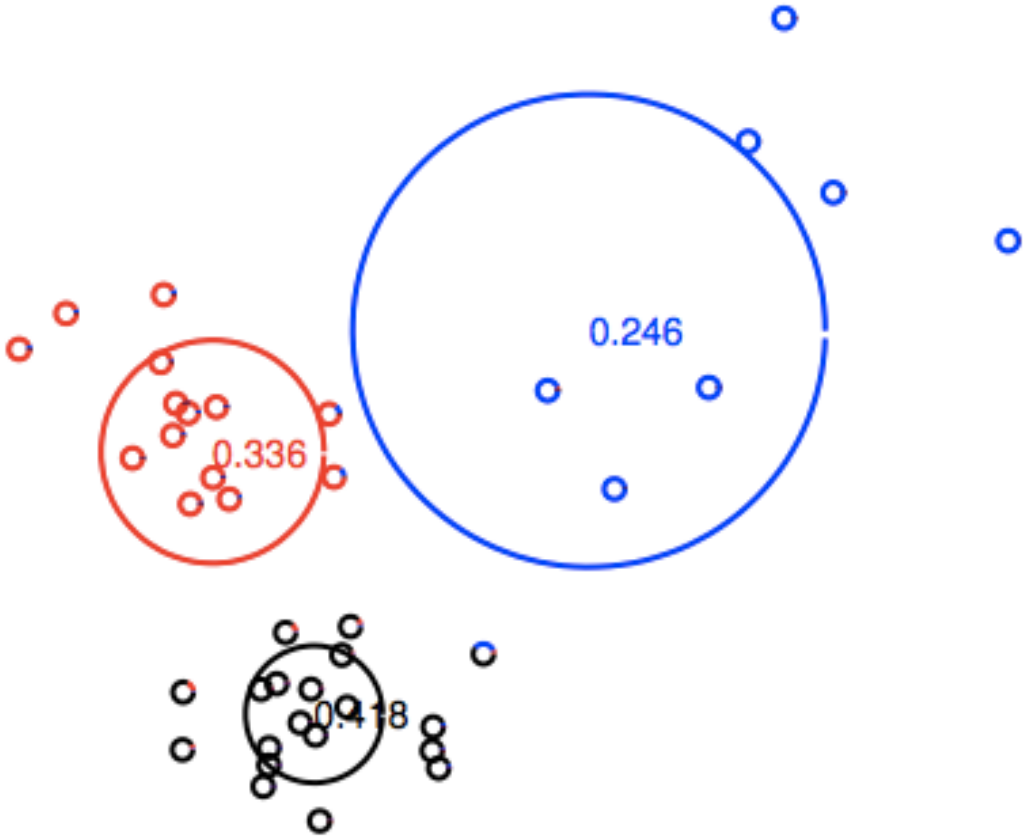
Mixture of Gaussians example



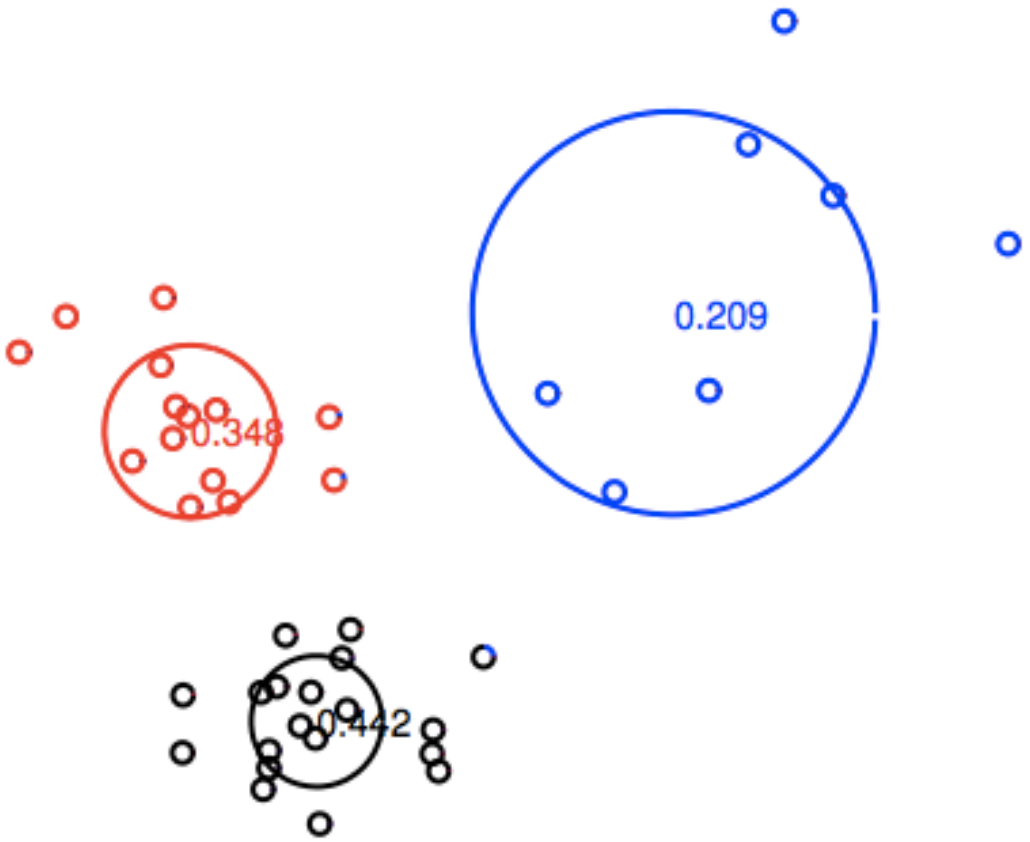
Mixture of Gaussians example



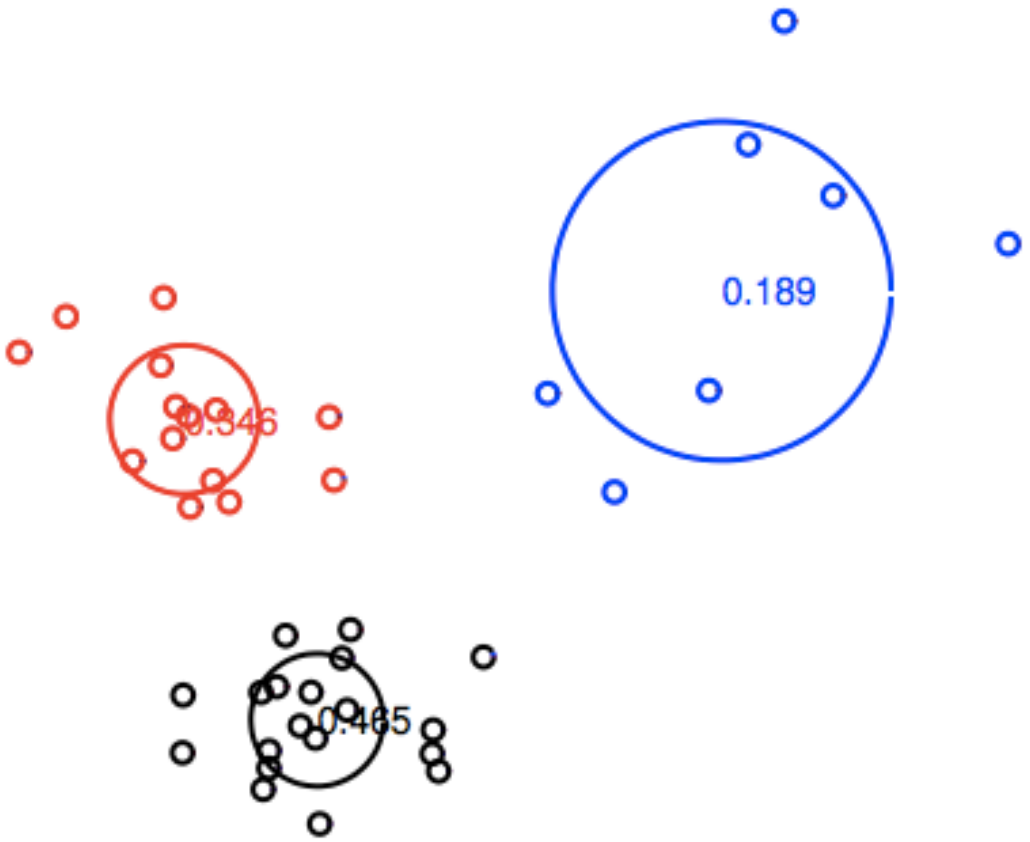
Mixture of Gaussians example



Mixture of Gaussians example

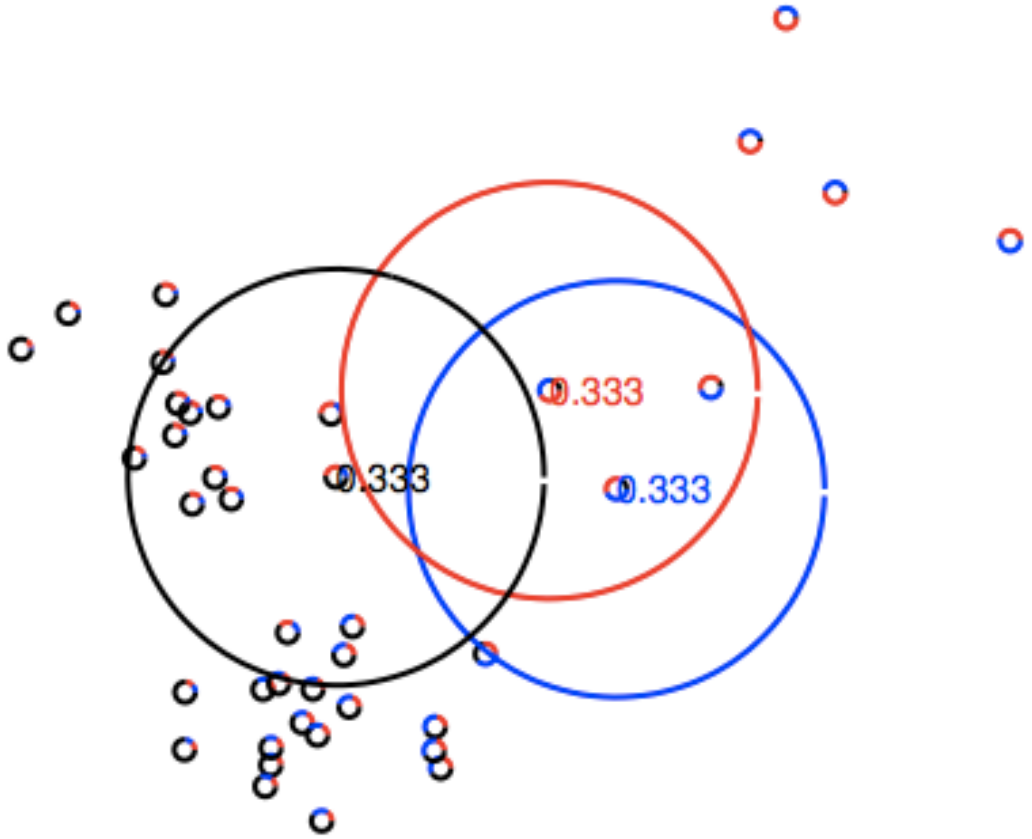


Mixture of Gaussians example

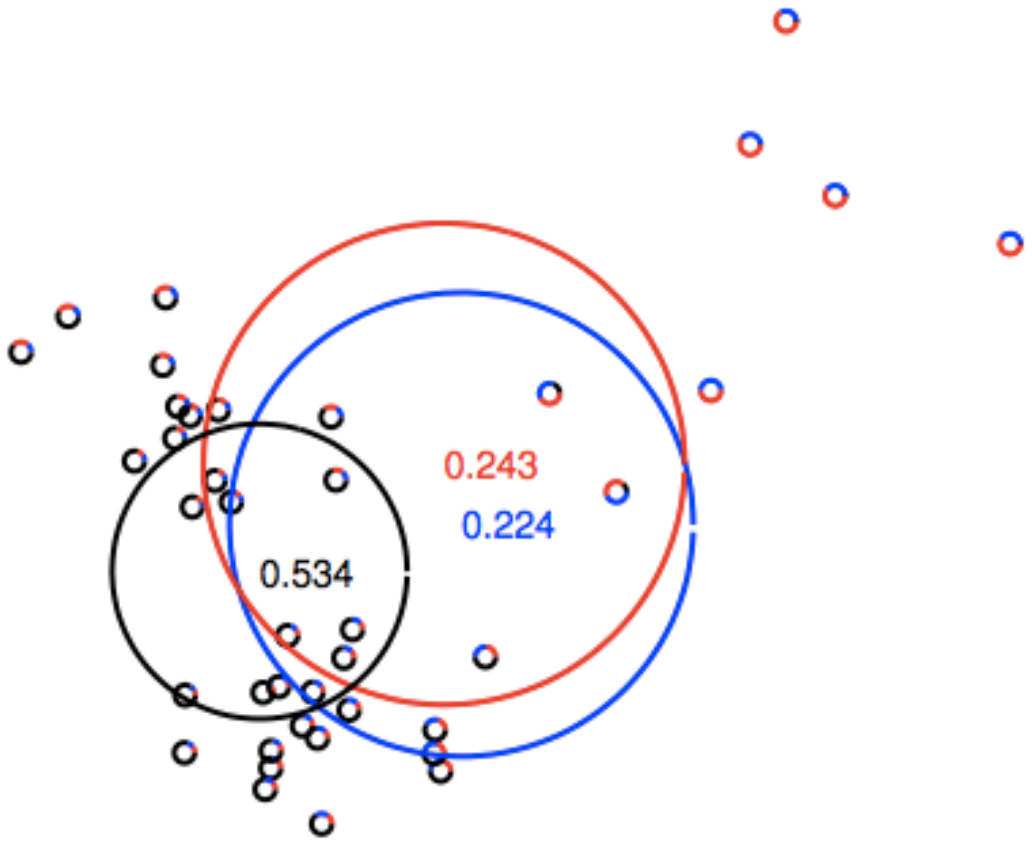


Mixture of Gaussians example

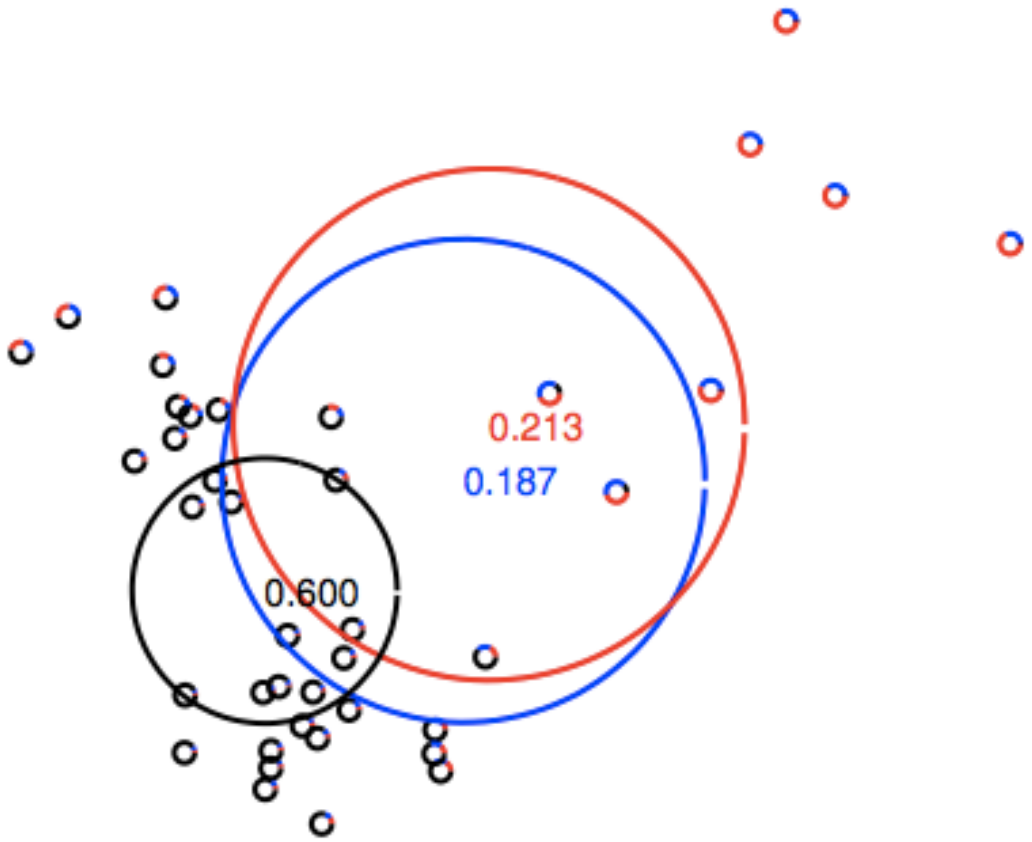
- initial 3-component mixture



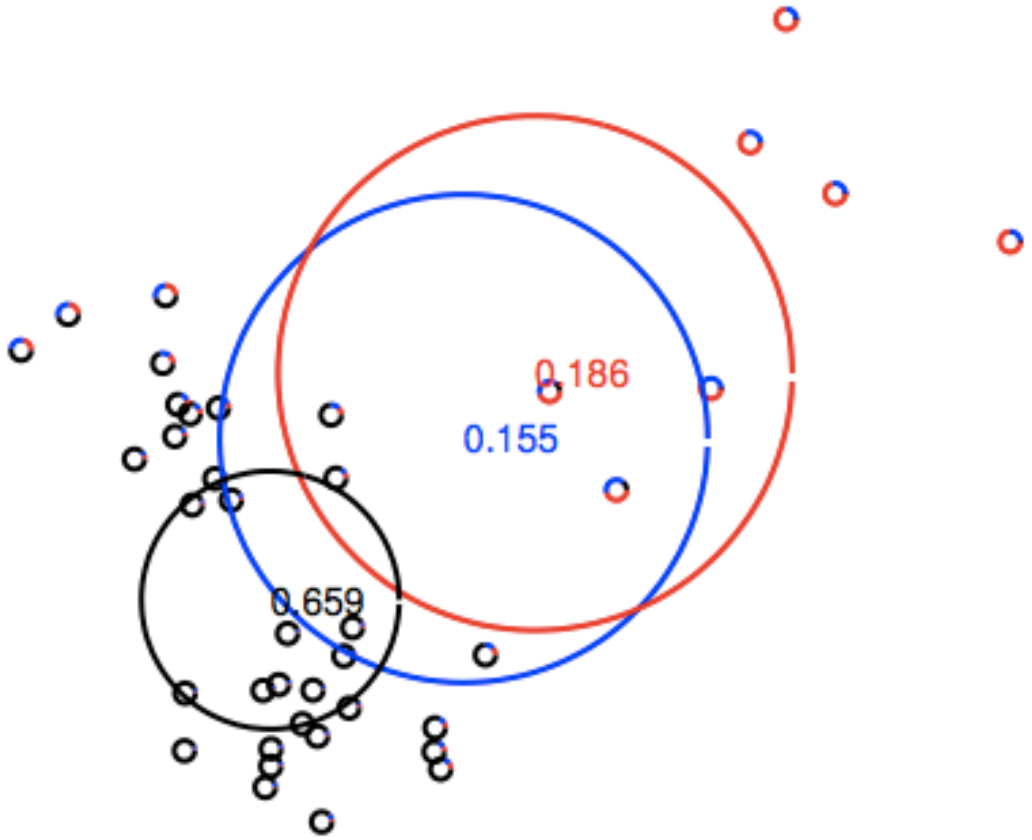
Mixture of Gaussians example



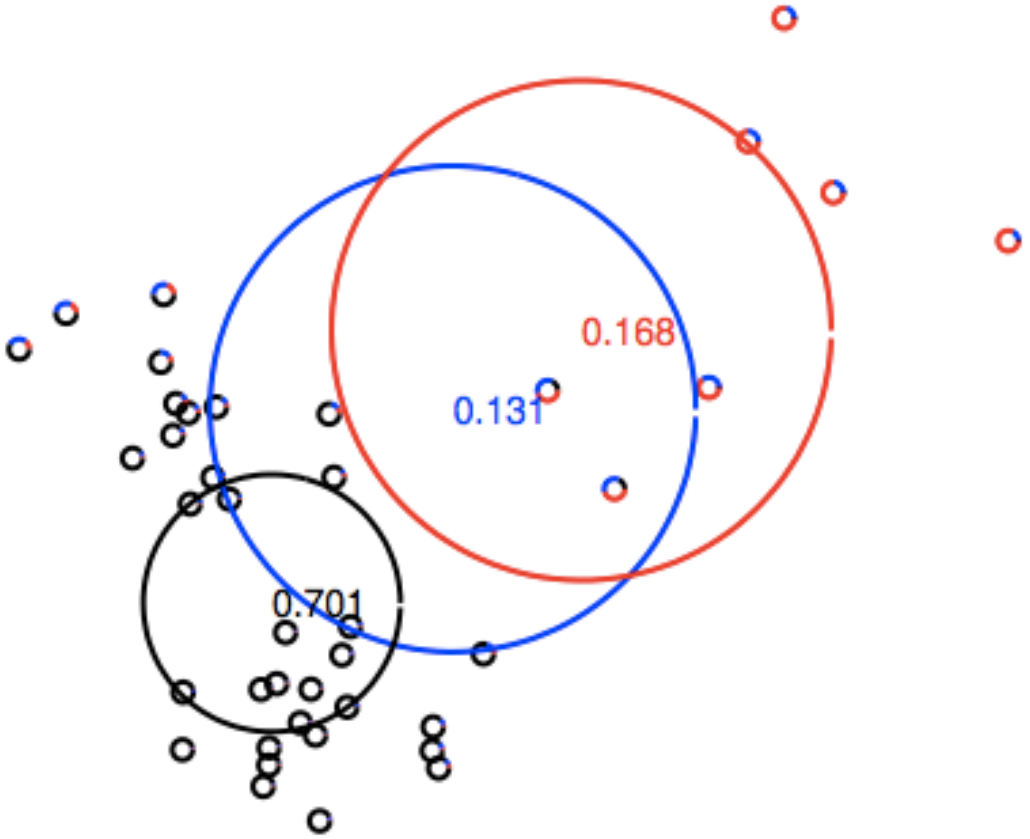
Mixture of Gaussians example



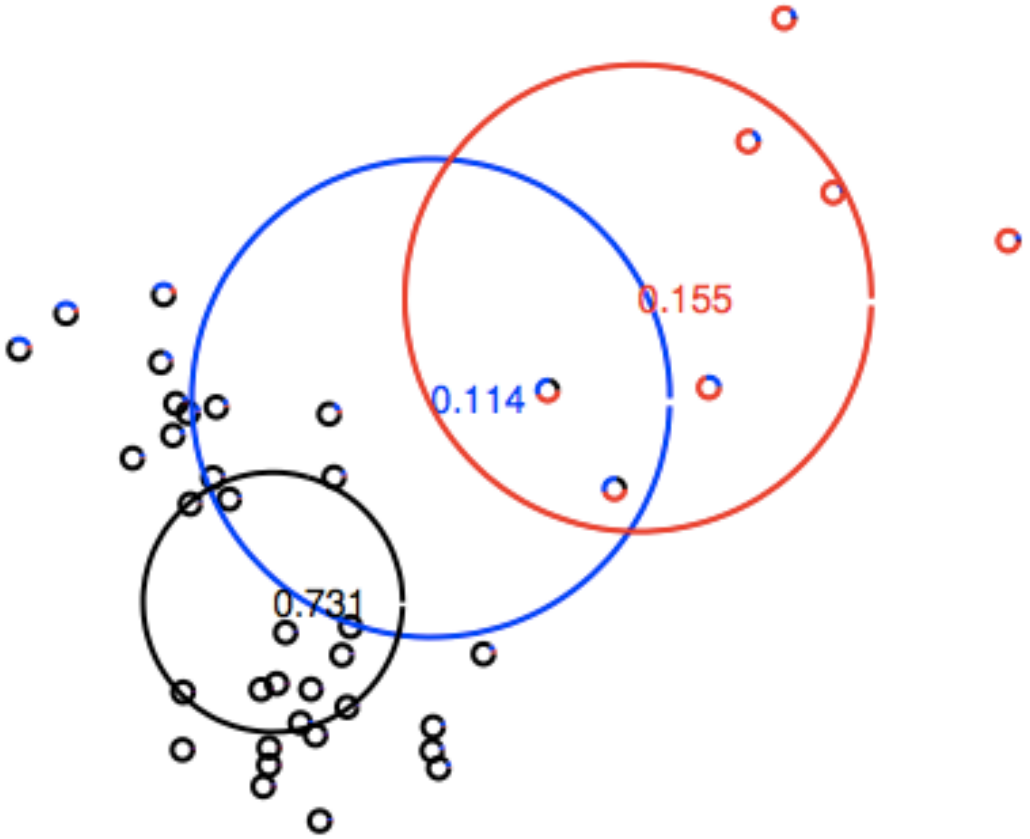
Mixture of Gaussians example



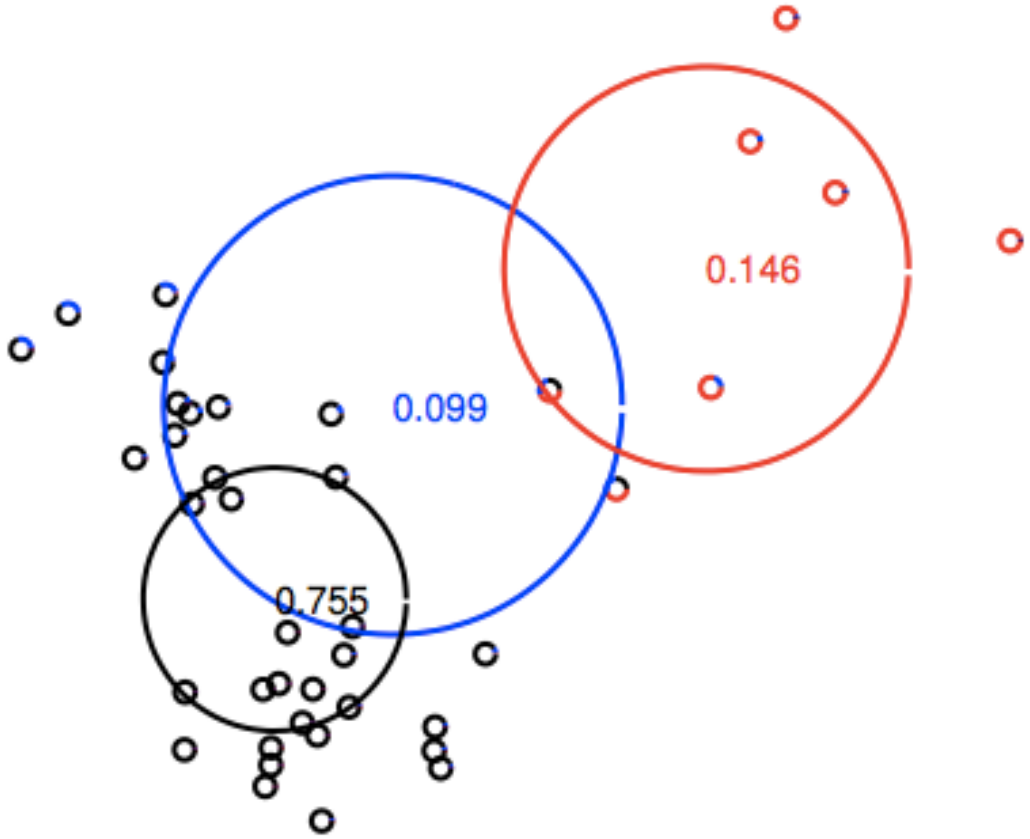
Mixture of Gaussians example



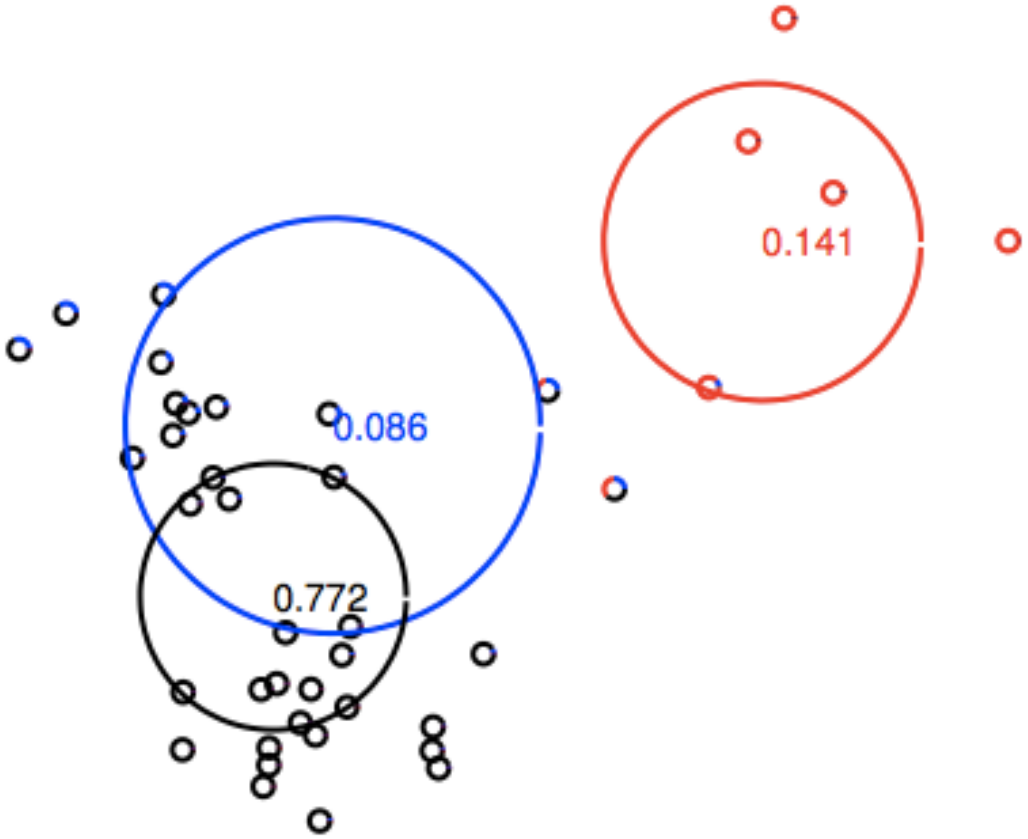
Mixture of Gaussians example



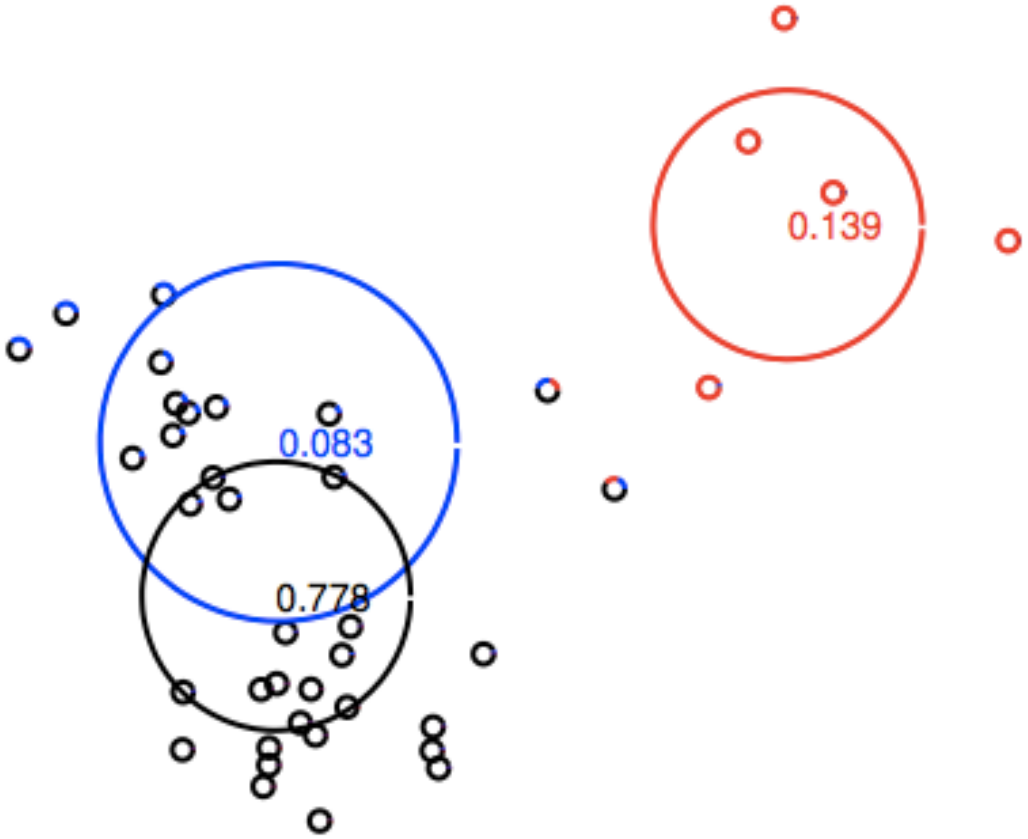
Mixture of Gaussians example



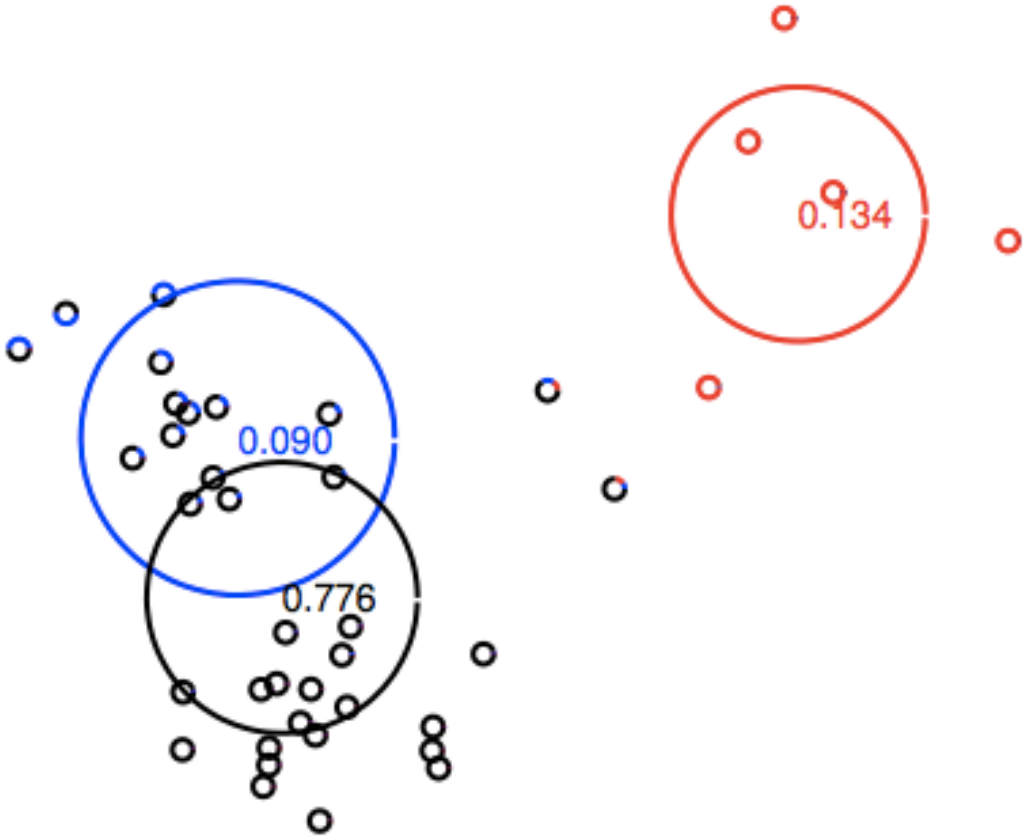
Mixture of Gaussians example



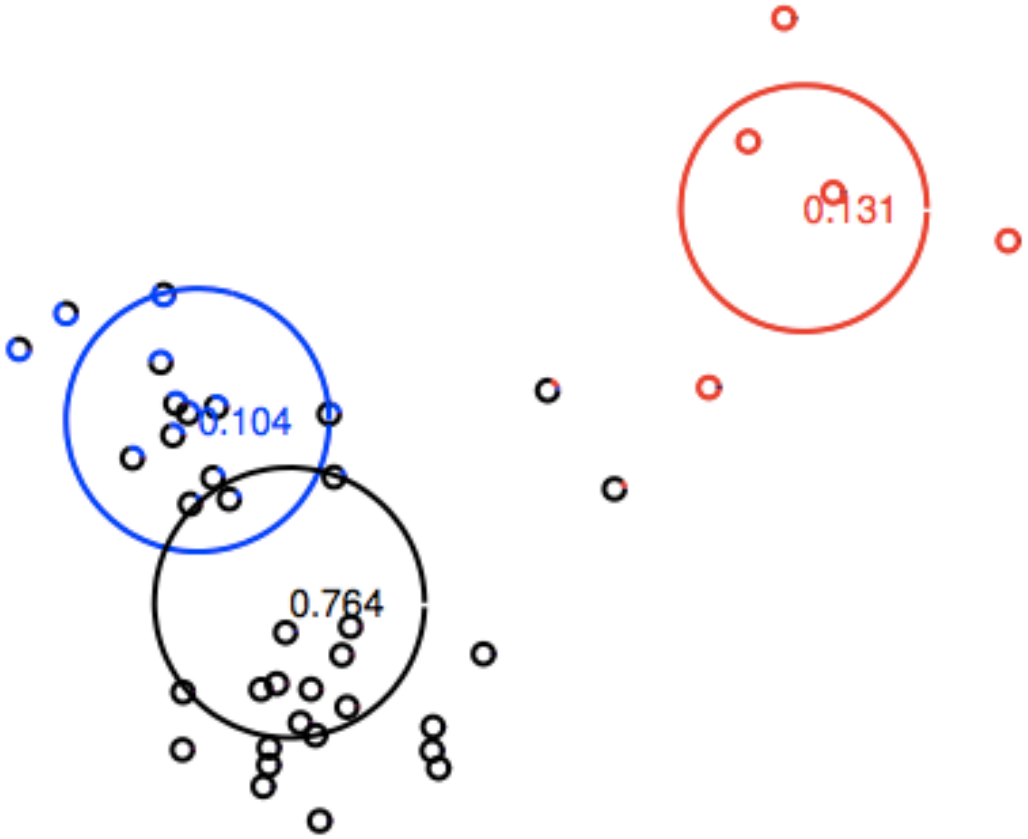
Mixture of Gaussians example



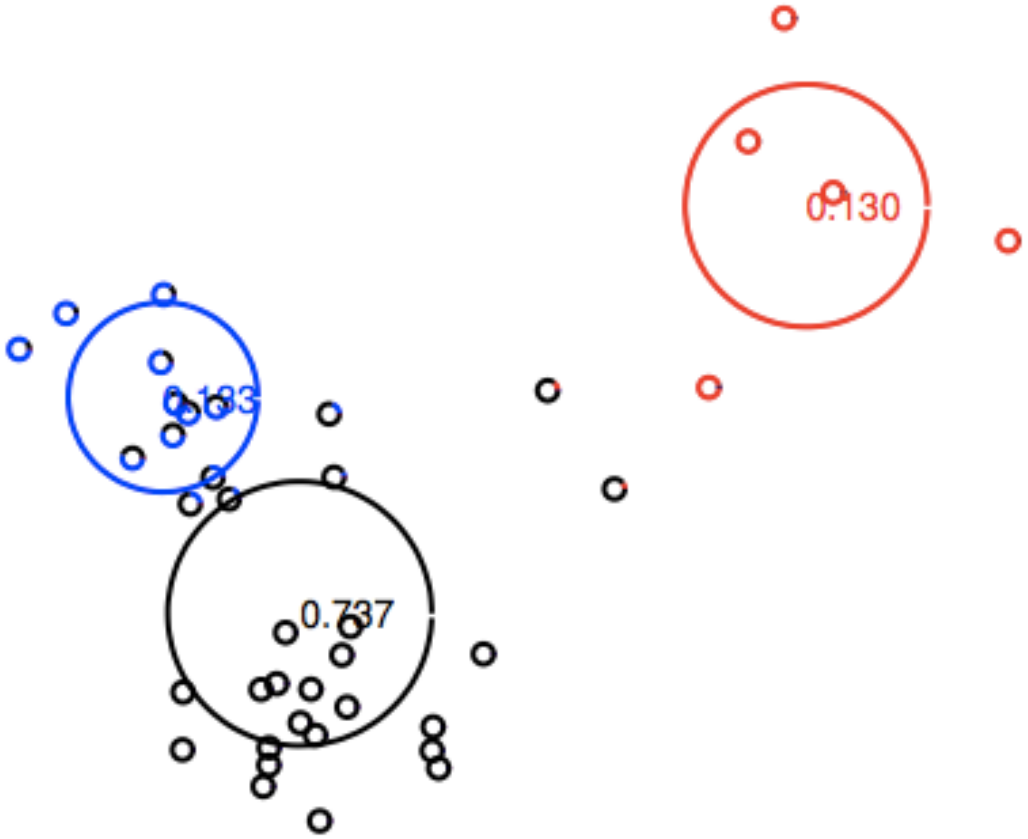
Mixture of Gaussians example



Mixture of Gaussians example



Mixture of Gaussians example



Mixture of Gaussians example