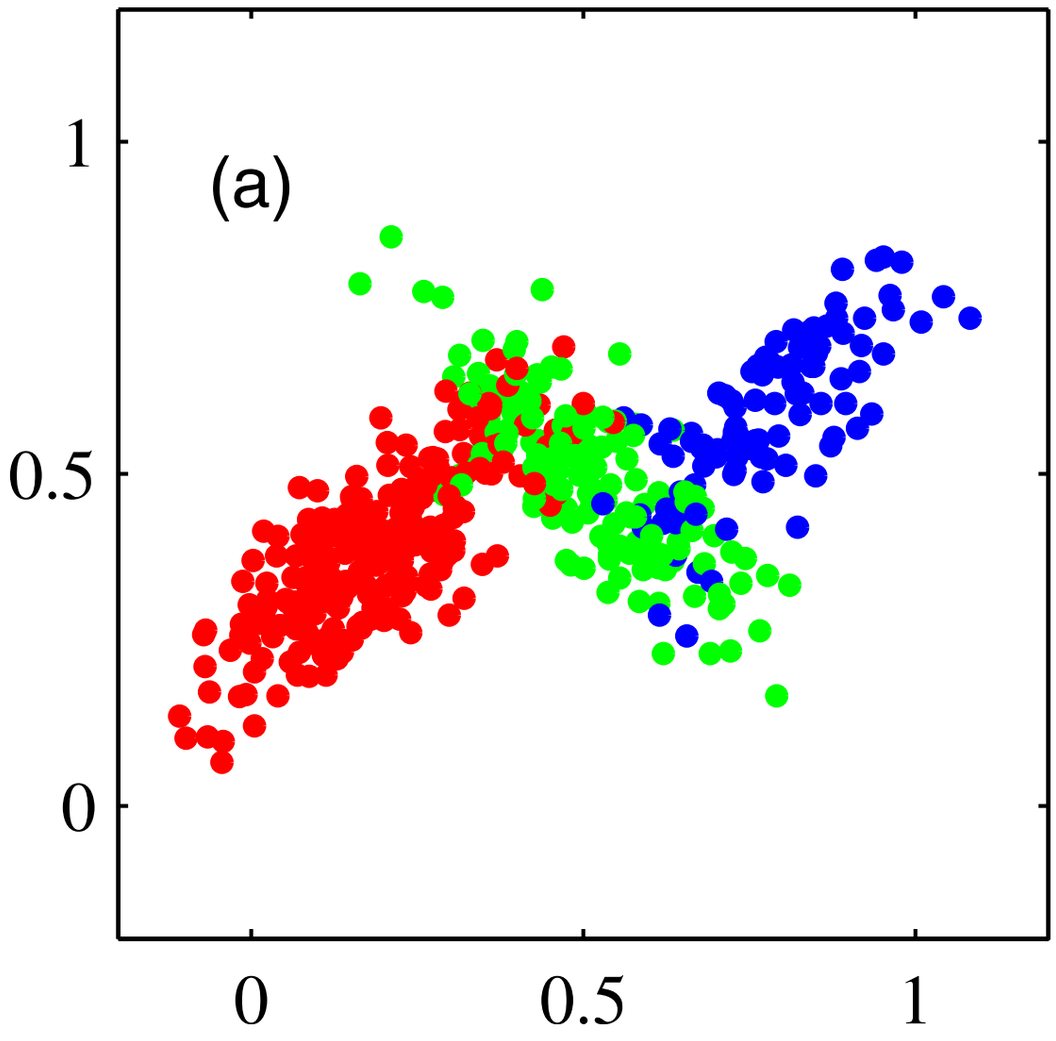
Gaussian Mixture Model

STAT 37710 / CMSC 35300
Rebecca Willett and Yuxin Chen

Recap: Generative Modeling and Gaussian Bayes classifiers

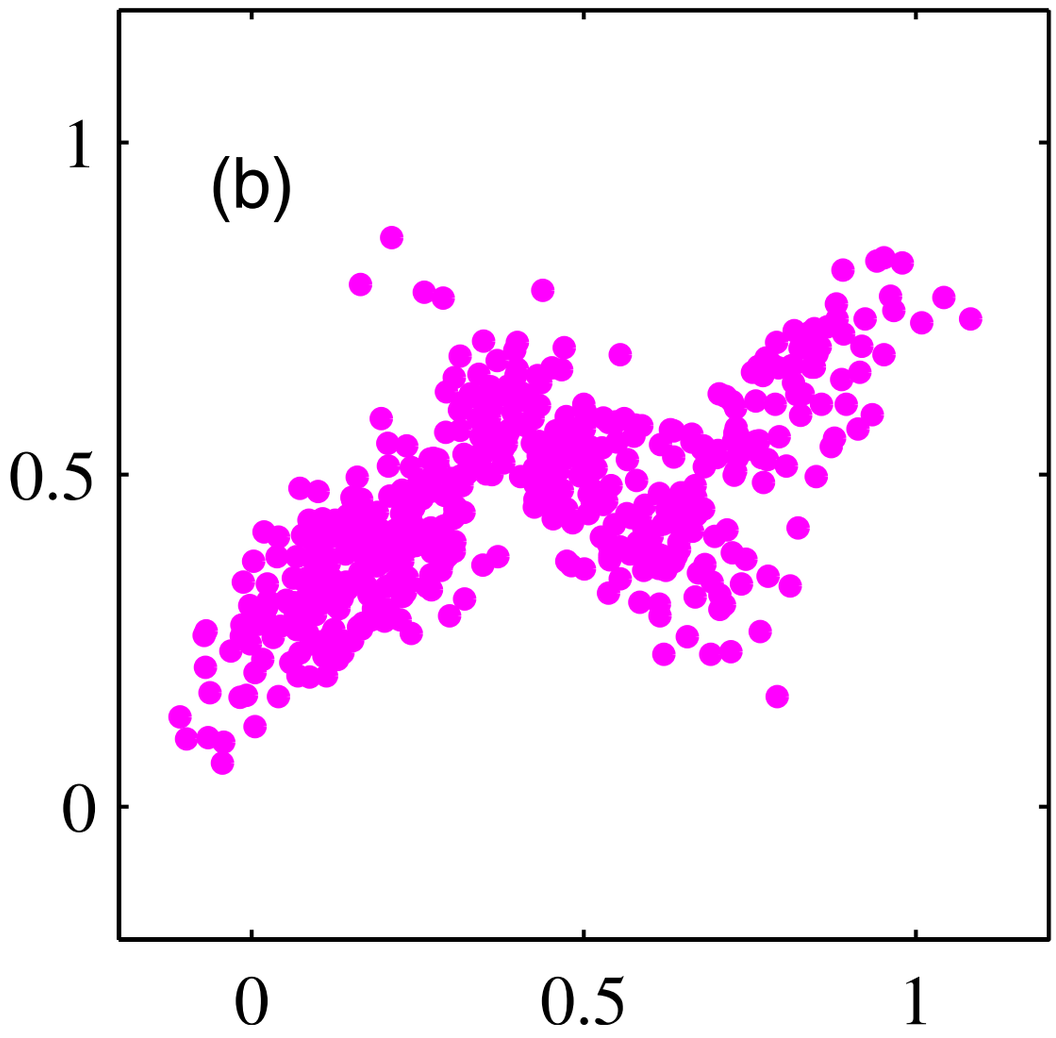
Given data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, model

- class label $P(Y = y) = p_y,\ y \in \mathcal{Y} = \{1, \ldots, k\}$
- features as multivariate Gaussians: $P(x \mid y) = \mathcal{N}(x; \mu_y, \Sigma_y)$



$P(X, Y)$ corresponds to a Gaussian Bayes classifier.

Missing labels

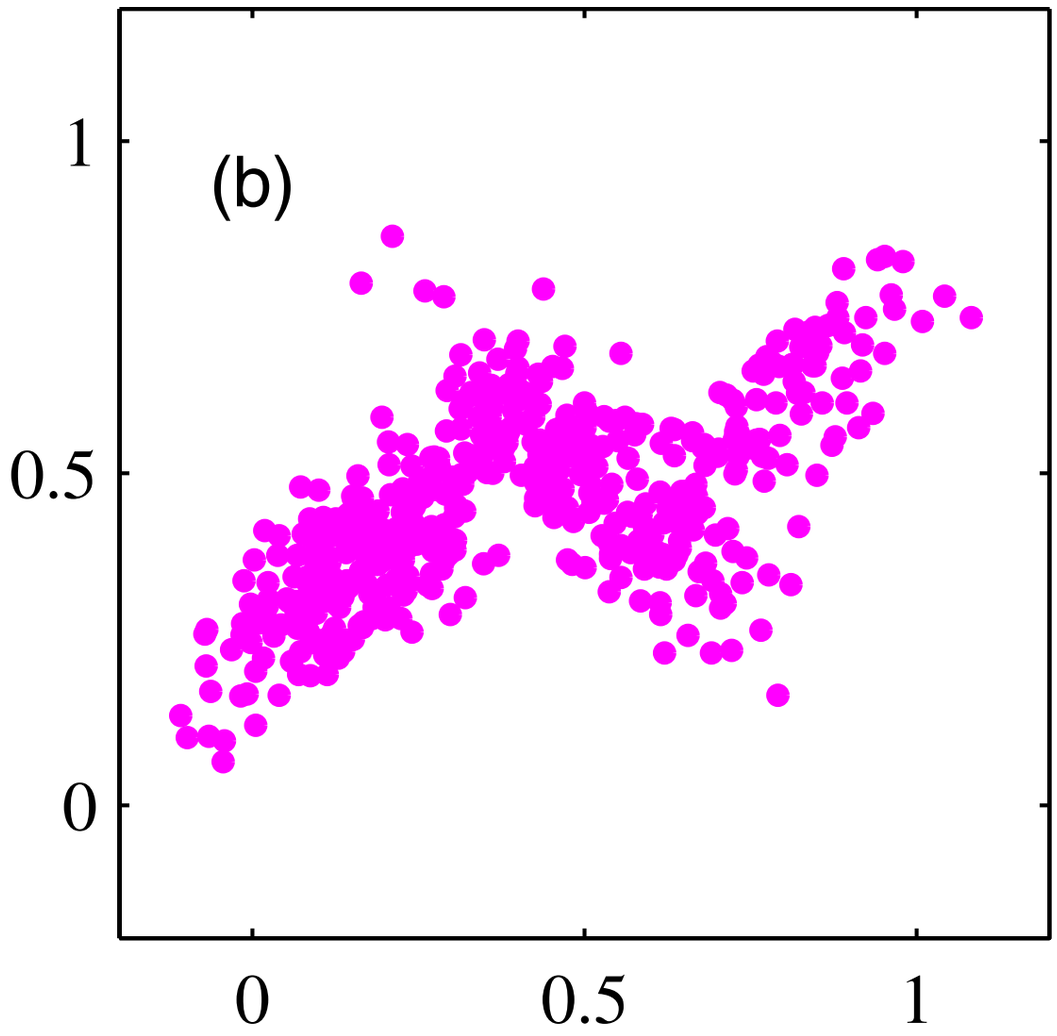


- What can we say about $P(X)$ if we can't access their labels?
- How can we hope to make inferences about labels?

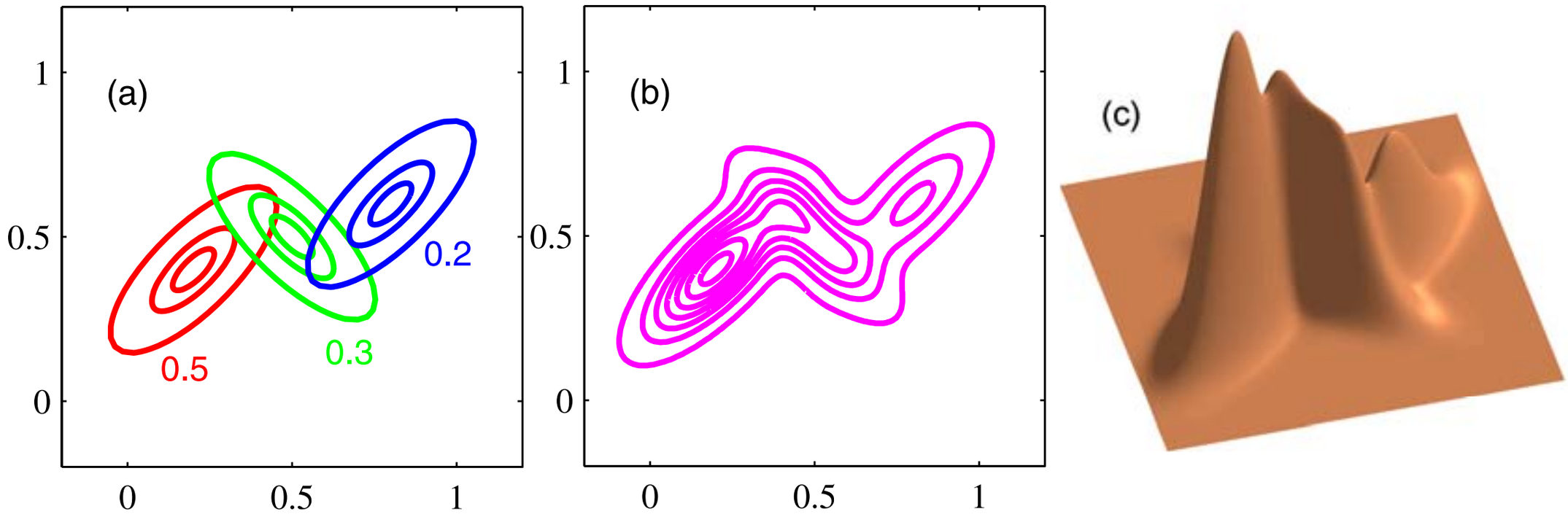
Let's make the same assumption on how the data is generated..

Training a Gaussian Bayes classifier without labels

$P(X, Y)$: Gaussian Bayes classifier, with $P(Y = y) = p_y$ and $P(X = x \mid y) = \mathcal{N}(x; \mu_y, \Sigma_y)$
Given $\{x_1, \ldots, x_n\}$, what can we say about $P(X)$?



Gaussian mixture



Definition: Gaussian mixture

Convex-combination of Gaussian distributions

$$P(x \mid \theta) = P(x \mid \mu, \Sigma, w) = \sum_{i=1}^{k} w_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

where $w_i > 0$ and $\sum_i w_i = 1$

Mixture modeling

- Model each cluster $j \in \{1, \ldots, k\}$ as a probability distribution

$$P(x \mid \theta_j)$$

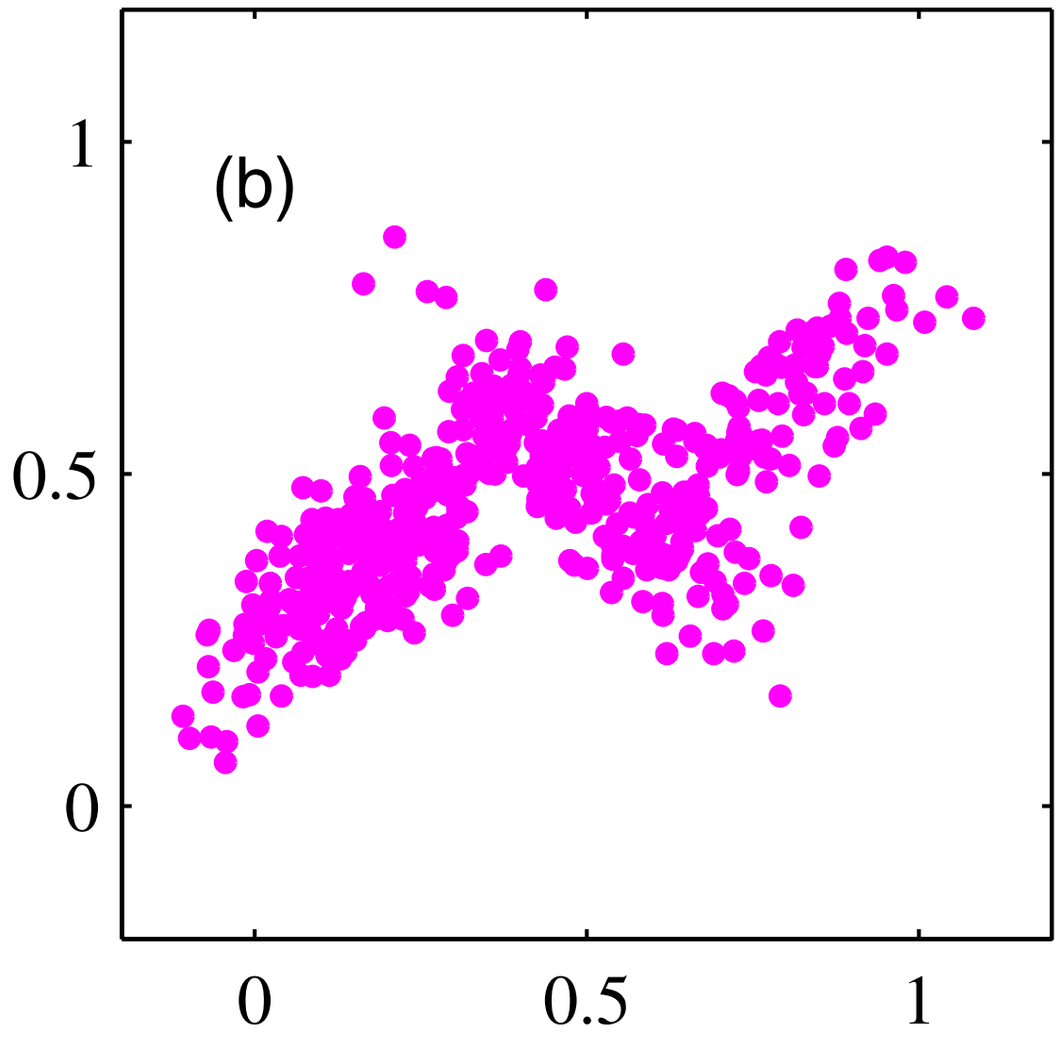
- Assume data $D = \{x_1, \ldots, x_n\}$ is sampled i.i.d. with likelihood

$$P(D \mid \theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} w_j P(x_i \mid \theta_j)$$

- Choose parameters to minimize negative log likelihood

$$L(D; \theta) = -\sum_{i=1}^{n} \log \sum_{j=1}^{k} w_j P(x_i \mid \theta_j)$$

Fitting a Gaussian mixture model



$$(\hat{\mu}, \hat{\Sigma}, \hat{w}) = \arg\min - \sum_{i=1}^{n} \log P(x_i \mid \mu, \Sigma, w)$$

$$= \arg\min - \sum_{i=1}^{n} \log \sum_{j=1}^{k} w_j \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

MLE for Gaussian mixture models

$$L(\mu_{1:k}, \Sigma_{1:k}, w_{1:k}) = -\sum_{i=1}^{n} \log \sum_{j=1}^{k} w_j \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

- Non-convex
  - Gradient descent?
- Contrained optimization on weights and covariance matrices
  - weights must sum to 1
  - covariance matrices must remain symmetric positive definite
- Gradient-based approach not well suited for this problem

GMMs vs Gaussian Bayes Classifiers

Let z be cluster index, GMM: $P(z, x) = w_z \mathcal{N}(x; \mu_z, \Sigma_z)$

In contrast to GBCs, in GMMs the (label) variable z is unobserved

- Fitting a GMM $=$ Training a GBC without labels
- Clustering $=$ latent variable modeling

Example: MLE solution for GBC

If we could get the labels $z := y$, can compute MLE in closed form!

$$\hat{p}_y = \frac{\text{Count}(Y = y)}{n}$$

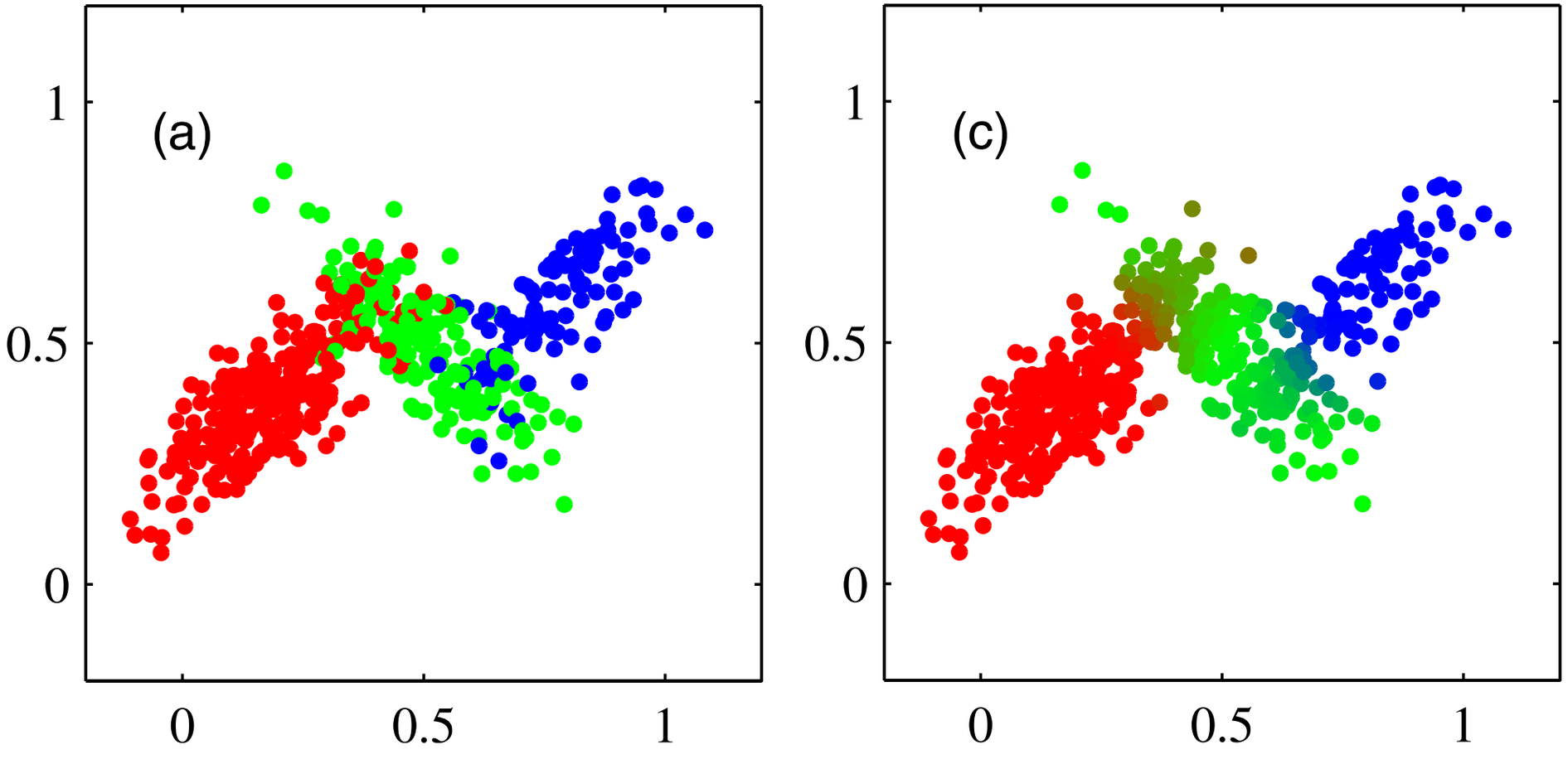
$$\hat{\mu}_y = \frac{1}{\text{Count}(Y = y)} \sum_{i:y_i=y} x_i$$

$$\hat{\Sigma} y = \frac{1}{\text{Count}(Y = y)} \sum_{i:y_i=y} (x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^\top$$

Responsibility of cluster z for data point x

Given model $P(z \mid \theta), P(x \mid z, \theta)$. For each x, compute posterior distribution over cluster membership:

$$\gamma_j(x) := \underbrace{P(Z = j \mid x, \mu, \Sigma, w) = \frac{w_j P(x \mid \mu_j, \Sigma_j)}{\sum_\ell w_\ell P(x \mid \mu_\ell, \Sigma_\ell)}}_{P(z|x,\theta)=\frac{P(z|\theta)P(x|z,\theta)}{\sum_z P(z|\theta)P(x|z,\theta)}}$$



MLE for Gaussian mixture

At MLE:

$$(\hat{\mu}, \hat{\Sigma}, \hat{w}) = \arg\min - \sum_i \log \sum_{j=1}^{k} w_j \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

it holds that

$$\hat{w}_j = \frac{1}{n} \sum_{i=1}^{n} \gamma_j(x_i)$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} \gamma_j(x_i) x_i}{\sum_{i=1}^{n} \gamma_j(x_i)}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n} \gamma_j(x_i)(x_i - \hat{\mu}_i)(x_i - \hat{\mu}_j)^\top}{\sum_{i=1}^{n} \gamma_j(x_i)}$$

MLE for Gaussian mixture

At MLE:

$$(\hat{\mu}, \hat{\Sigma}, \hat{w}) = \arg\min - \sum_i \log \sum_{j=1}^{k} w_j \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

it holds that

$$\hat{w}_j = \frac{1}{n} \sum_{i=1}^{n} \gamma_j(x_i)$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} \gamma_j(x_i) x_i}{\sum_{i=1}^{n} \gamma_j(x_i)}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n} \gamma_j(x_i)(x_i - \hat{\mu}_i)(x_i - \hat{\mu}_j)^\top}{\sum_{i=1}^{n} \gamma_j(x_i)}$$

Equations are coupled – Difficult to solve jointly

Algorithm: Expectation-Maximization for GMMs

Initialize parameters $w_{1:k}^{(0)}, \mu_{1:k}^{(0)}, \Sigma_{1:k}^{(0)}$
While not converged:

- **E-step**: calculate cluster membership weights for each point

$$\gamma_j^{(t)}(x_i) \leftarrow \frac{w_j P(x_i \mid \mu_j^{(t-1)}, \Sigma_j^{(t-1)})}{\sum_\ell w_\ell P(x_i \mid \mu_\ell^{(t-1)}, \Sigma_\ell^{(t-1)})}$$

- **M-step**: Fit clusters to weighted data points (closed-form)

$$w_j^{(t)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \gamma_j^{(t)}(x_i)$$

$$\mu_j^{(t)} \leftarrow \frac{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i) x_i}{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)}$$

$$\Sigma_j^{(t)} \leftarrow \frac{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)(x_i - \mu_i^{(t)})(x_i - \mu_j^{(t)})^\top}{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)}$$

Special case (constrained EM)

Example: EM for GMM with uniform weightts and ≈ 0 covariance

Suppose $w_1 = \cdots = w_k = \frac{1}{k}$; $\Sigma_1 = \cdots = \Sigma_k = \sigma^2 I$, and $\sigma \to 0$.

- Initialize parameters $\mu_{1:k}^{(0)}$
- While not converged:
  - **E-step**: calculate cluster membership weight for each point

$$\gamma_j^{(t)}(x_i) \leftarrow \frac{\frac{1}{k} P(x_i \mid \mu_j^{(t-1)}, \sigma^2 I)}{\frac{1}{k} \sum_{\ell=1}^{k} P(x_i \mid \mu_\ell^{(t-1)}, \sigma^2 I)} = \begin{cases} 1 & \text{if } j \overset{*}{=} \arg\min_\ell \left\| x_i - \mu_\ell^{(t-1)} \right\| \\ 0 & \text{o.w.} \end{cases}$$

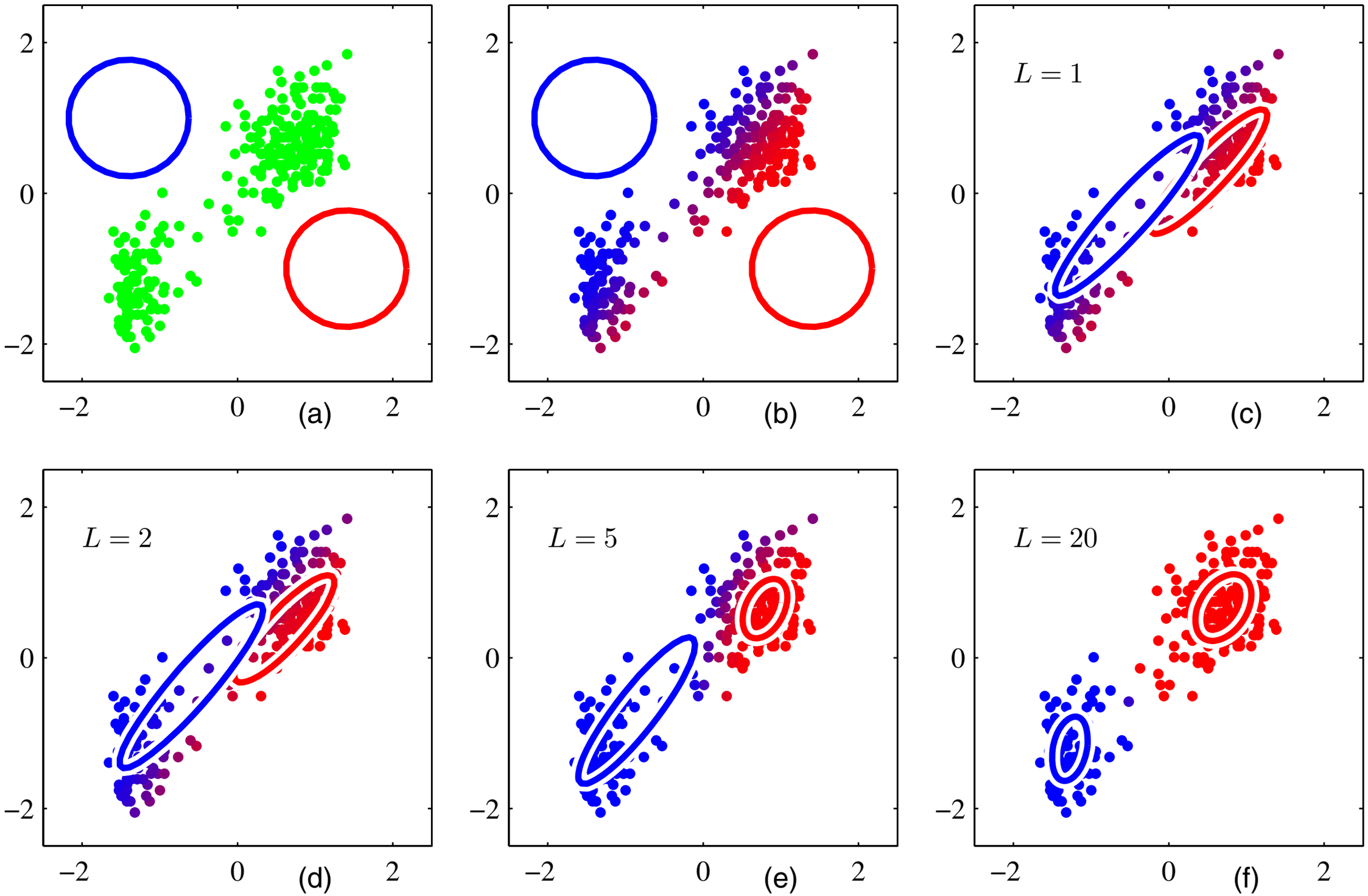
* assume no ties

  - **M-step**: Fit clusters to data points

$$\mu_j^{(t)} \leftarrow \frac{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i) x_i}{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)} = \frac{1}{\left| C_j^{(t)} \right|} \sum_{i \in C_j^{(t)}} x_i, \quad \text{where } C_j^{(t)} := \{i : \gamma_j^{(t)}(x_i) = 1\}$$

EM reduces to k-**means** algorithm under the above assumptions

EM example



The general EM algorithm

EM algorithm is equivalent to the following procedure:

- **E-step**: Calculate the **e**xpected complete-data log likelihood (= function of θ)

$$Q(\theta; \theta^{(t-1)}) = \mathbb{E}_{Z_{1:n}} \left[\log P(x_{1:n}, Z_{1:n} \mid \theta) \,\middle|\, x_{1:n}, \theta^{(t-1)} \right]$$

- **M-step**: **M**aximize

$$\theta^t = \arg\max_{\theta} Q(\theta; \theta^{t-1})$$

EM objective function (for GMMs)

Notation: RV Z; "realization" z.

$$Q(\theta;\theta^{(t-1)}) = \mathbb{E}_{Z_{1:n}}\left[\log P(x_{1:n}, Z_{1:n} \mid \theta) \mid x_{1:n}, \theta^{(t-1)}\right]$$

$$\overset{\text{i.i.d.}}{=} \mathbb{E}_{Z_{1:n}}\left[\sum_{i=1}^{n} \log P(x_i, Z_i \mid \theta) \mid x_{1:n}, \theta^{(t-1)}\right]$$

$$= \sum_{i=1}^{n} \mathbb{E}_{Z_i}\left[\log P(x_i, Z_i \mid \theta) \mid x_i, \theta^{(t-1)}\right]$$

$$= \sum_{i=1}^{n}\sum_{z=1}^{k} \underbrace{P(Z_i = z \mid x_i, \theta^{(t-1)})}_{\gamma_z(x_i)} \log \underbrace{P(x_i, Z_i = z \mid \theta)}_{w_z \mathcal{N}(x_i;\mu_z,\Sigma_z)}$$

$$= \sum_{i=1}^{n}\sum_{z_i=1}^{k} \gamma_{z_i}(x_i) \log P(x_i, z_i \mid \theta)$$

EM algorithm: E-step and M-step

Objective:

$$Q(\theta; \theta^{(t-1)}) = \sum_{i=1}^{n} \sum_{z_i=1}^{k} \gamma_{z_i}(x_i) \log P(x_i, z_i \mid \theta)$$

where $\gamma_z(x_i) = P(z \mid x_i, \theta^{(t-1)})$

E-step: compute $\gamma_z(x_i)$ (*expected sufficient statistics*)

M-step: compute $\theta^{(t)} = \arg\max_\theta Q(\theta; \theta^{(t-1)})$ (*MLE*)

Recall MLE in Gaussian Bayes Classifiers:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log P(x_i, z_i \mid \theta)$$

M-step is equivalent to training a GBC with weighted data

Convergence of the EM algorithm

EM Algorithm monotonically increases the likelihood

$$\log P(x_{1:n} \mid \theta^{(t)}) \geq \log P(x_{1:n} \mid \theta^{(t-1)})$$

- For Gaussian mixtures, EM is guaranteed to converge to a local maximum.
- Quality of solution highly depends on initialization

Proof of the monotonic behavior (sketch)

Goal: find $\hat{\theta} = \arg\max_\theta P(x_{1:n} \mid \theta)$

- Take logs, and expectation w.r.t. $P(Z_{1:n} \mid X_{1:n}, \theta^{(t-1)})$:

$$\begin{aligned}
\log P(x \mid \theta) &= \mathbb{E}_Z\left[\log P(x \mid \theta) \mid x, \theta^{(t-1)}\right] \\
&= \mathbb{E}_Z\left[\log \frac{P(x, Z \mid \theta)}{P(Z \mid x, \theta)} \;\middle|\; x, \theta^{(t-1)}\right] \\
&= \underbrace{\mathbb{E}_Z\left[\log P(x, Z \mid \theta) \mid x, \theta^{(t-1)}\right]}_{Q(\theta;\theta^{(t-1)})} - \mathbb{E}_Z\left[\log P(Z \mid x, \theta) \mid x, \theta^{(t-1)}\right]
\end{aligned}$$

Want to show: $\log P(x \mid \theta^{(t)}) \geq \log P(x \mid \theta^{(t-1)})$

- It suffices to show:

$$Q(\theta^{(t)}; \theta^{(t-1)}) \geq Q(\theta^{(t-1)}; \theta^{(t-1)}) \tag{1}$$

$$\mathbb{E}_Z\left[\log P(Z \mid x, \theta^{(t-1)}) \mid x, \theta^{(t-1)}\right] \geq \mathbb{E}_Z\left[\log P(Z \mid x, \theta^{(t)}) \mid x, \theta^{(t-1)}\right] \tag{2}$$

Proof of the monotonic behavior (cont.)

Want to show

$$\mathbb{E}_Z\Big[\log P(Z \mid x, \theta^{(t-1)}) \mid x, \theta^{(t-1)}\Big] \geq \mathbb{E}_Z\Big[\log P(Z \mid x, \theta^{(t)}) \mid x, \theta^{(t-1)}\Big]$$

Equivalently,

$$\mathbb{E}_Z\left[\log \frac{P(Z \mid x, \theta^{(t-1)})}{P(Z \mid x, \theta^{(t)})} \;\middle|\; x, \theta^{(t-1)}\right] \overset{(*)}{\geq} 0$$

$(*)$ holds due to the fact that the Kullback-Leibler divergence

$$D_{\mathsf{KL}}(P \parallel Q) := \mathbb{E}_{X \sim P}\left[\log \frac{P(X)}{Q(X)}\right] \geq 0$$

Can we claim that EM will converge?

Proof of the monotonic behavior (cont.)

Want to show

$$\mathbb{E}_Z\left[\log P(Z \mid x, \theta^{(t-1)}) \mid x, \theta^{(t-1)}\right] \geq \mathbb{E}_Z\left[\log P(Z \mid x, \theta^{(t)}) \mid x, \theta^{(t-1)}\right]$$

Equivalently,

$$\mathbb{E}_Z\left[\log \frac{P(Z \mid x, \theta^{(t-1)})}{P(Z \mid x, \theta^{(t)})} \;\middle|\; x, \theta^{(t-1)}\right] \overset{(*)}{\geq} 0$$

$(*)$ holds due to the fact that the Kullback-Leibler divergence

$$D_{\mathsf{KL}}(P \parallel Q) := \mathbb{E}_{X \sim P}\left[\log \frac{P(X)}{Q(X)}\right] \geq 0$$

Can we claim that EM will converge?

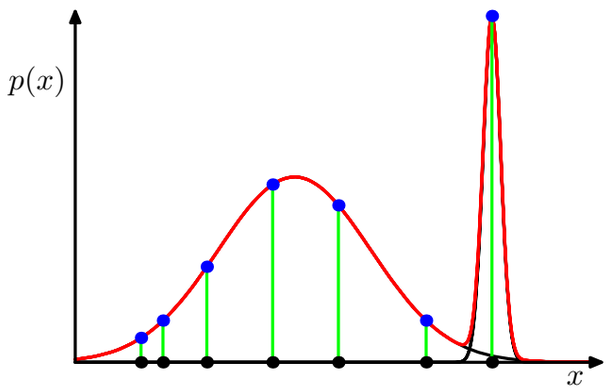
- Not in the degenerate case when log likelihood is unbounded!

Degeneracy of GMMs

Example: Fitting a GMM to single data point

Suppose we are given a single data point. What is the optimal log-likelihood that can be achieved? (1D)

$$-\log P(x \mid \mu, \sigma) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(x-\mu)^2$$



- Loss converges to $-\infty$ as $\mu = x, \sigma \to 0$
- Optimal GMM chooses $k = n$, and puts one Gaussian around each data point with variance tending to 0 – overfitting!
- For $k < n$, MLE for GMM could also lead to singularity issues, i.e. spiky Gaussian components that collapse to single points

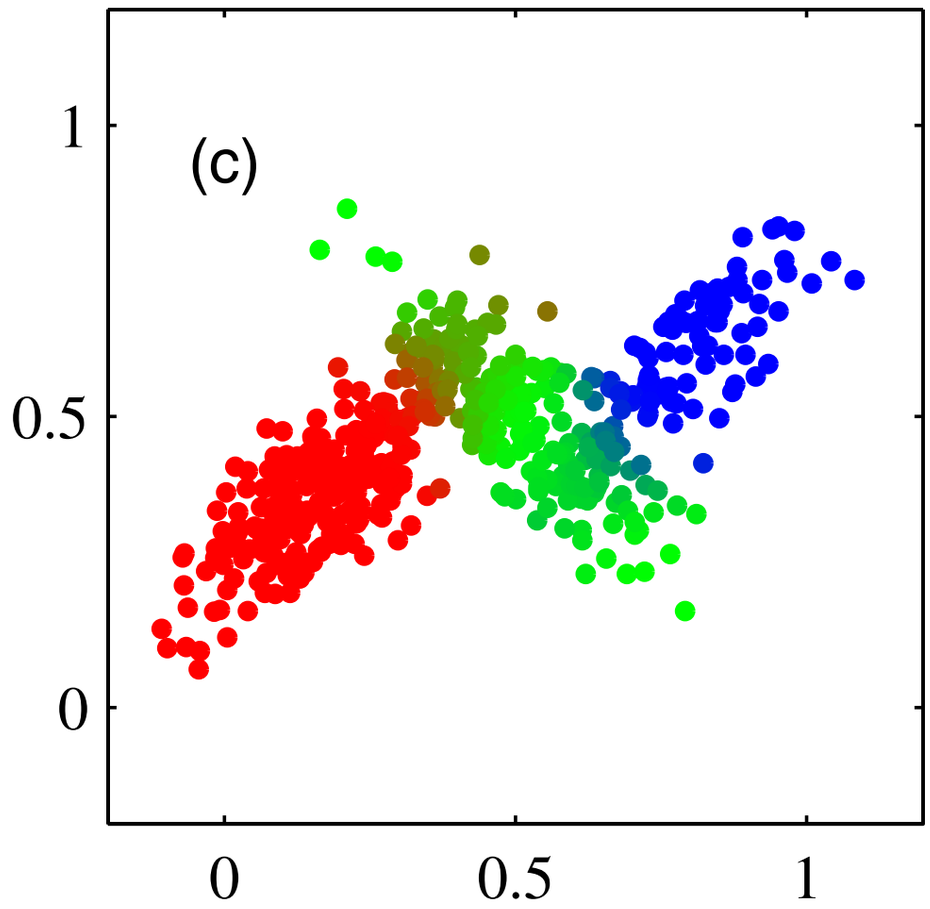
Avoiding degeneracy in GMMs

- Can avoid variances tending to 0 by simply adding a small term to the diagonal of the MLE:

$$\Sigma_j^{(t)} \leftarrow \frac{\sum_{i=1}^n \gamma_j^{(t)}(x_i)(x_i - \mu_j^{(t)})(x_i - \mu_j^{(t)})^\top}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)} + \nu^2 I$$

- Bayesian perspective: This is equivalent to placing a (conjugate) Inverse Wishart prior on the covariance matrix, and computing the MAP instead of MLE to regularize.

Use cases of mixture models



- Can be used for clustering/unsupervised learning
- Can be used as part of more complex statistical models, e.g., classifiers (or more general probabilistic models)
- Can output likelihood $P(x)$ of a point x, which is useful for anomaly/outlier detection

Gaussian-mixture Bayes classifiers

Given labeled data set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- Label $y_i \in \{1, \ldots, c\}$
- Estimate class prior $P(y)$
- Estimate conditional distribution for each class

$$P(x \mid y) = \sum_{j=1}^{k_y} w_j^{(y)} \mathcal{N}(x; \mu_j^{(y)}, \Sigma_j^{(y)})$$

as Gaussian mixture model.

How do we use this model for classification?

Gaussian-mixture Bayes classifiers

Given labeled data set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- Label $y_i \in \{1, \ldots, c\}$
- Estimate class prior $P(y)$
- Estimate conditional distribution for each class

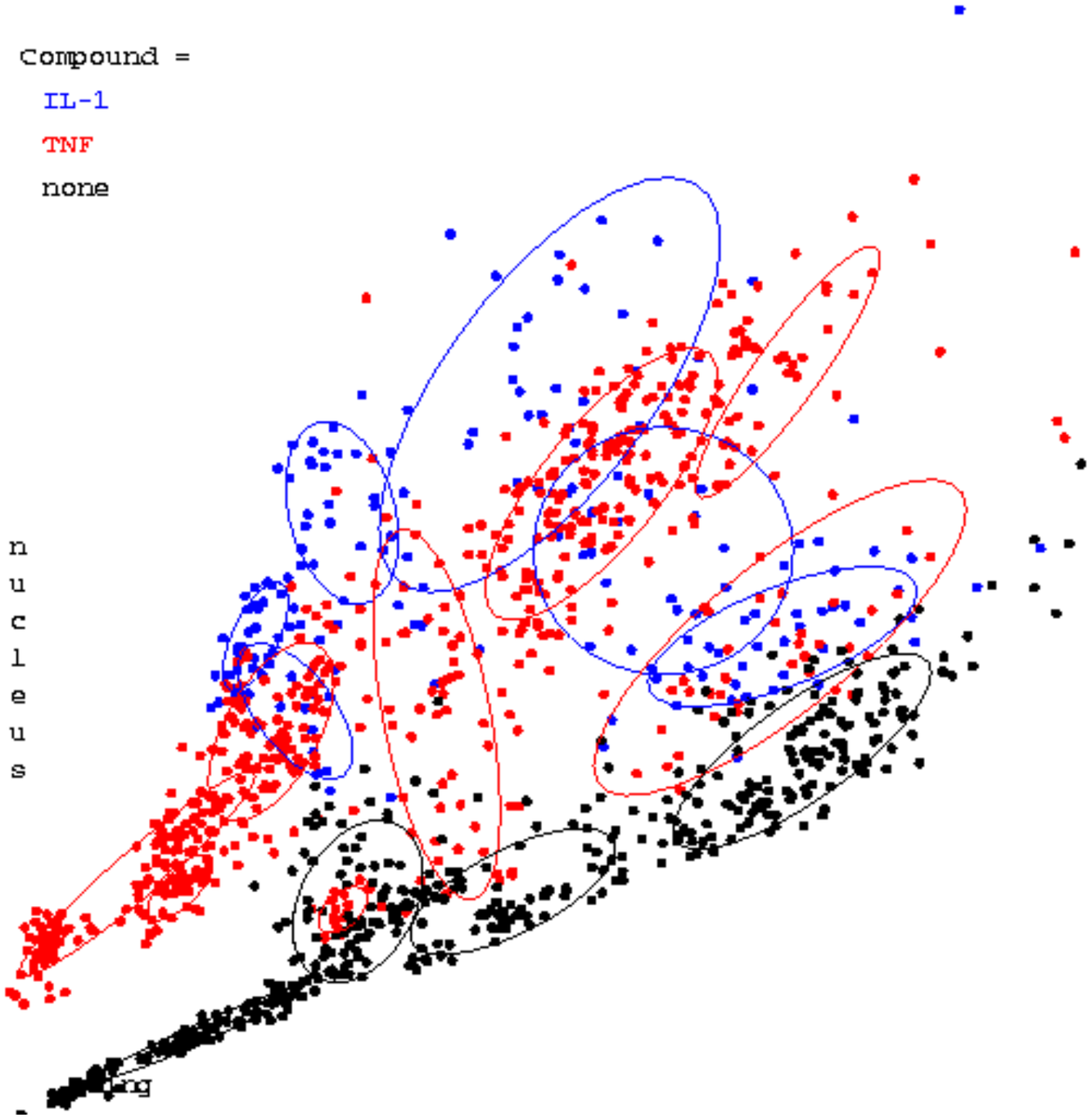
$$P(x \mid y) = \sum_{j=1}^{k_y} w_j^{(y)} \mathcal{N}(x; \mu_j^{(y)}, \Sigma_j^{(y)})$$

as Gaussian mixture model.

How do we use this model for classification?

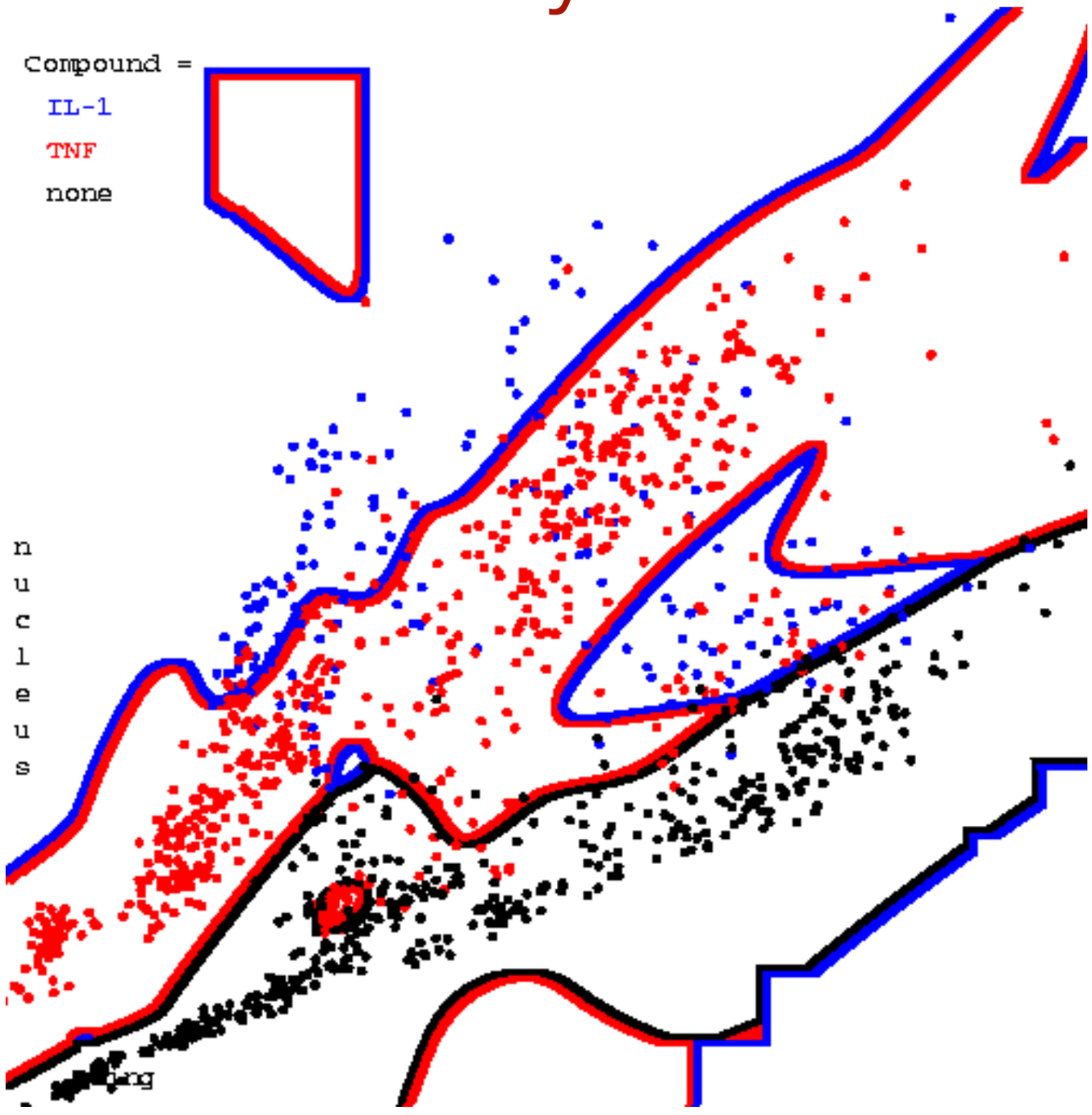
$$P(y \mid x) = \frac{1}{Z} p(y) \sum_{j=1}^{k_y} w_j^{(y)} \mathcal{N}(x; \mu_j^{(y)}, \Sigma_j^{(y)})$$

Gaussian-mixture Bayes classifiers: example



[A. Moore]

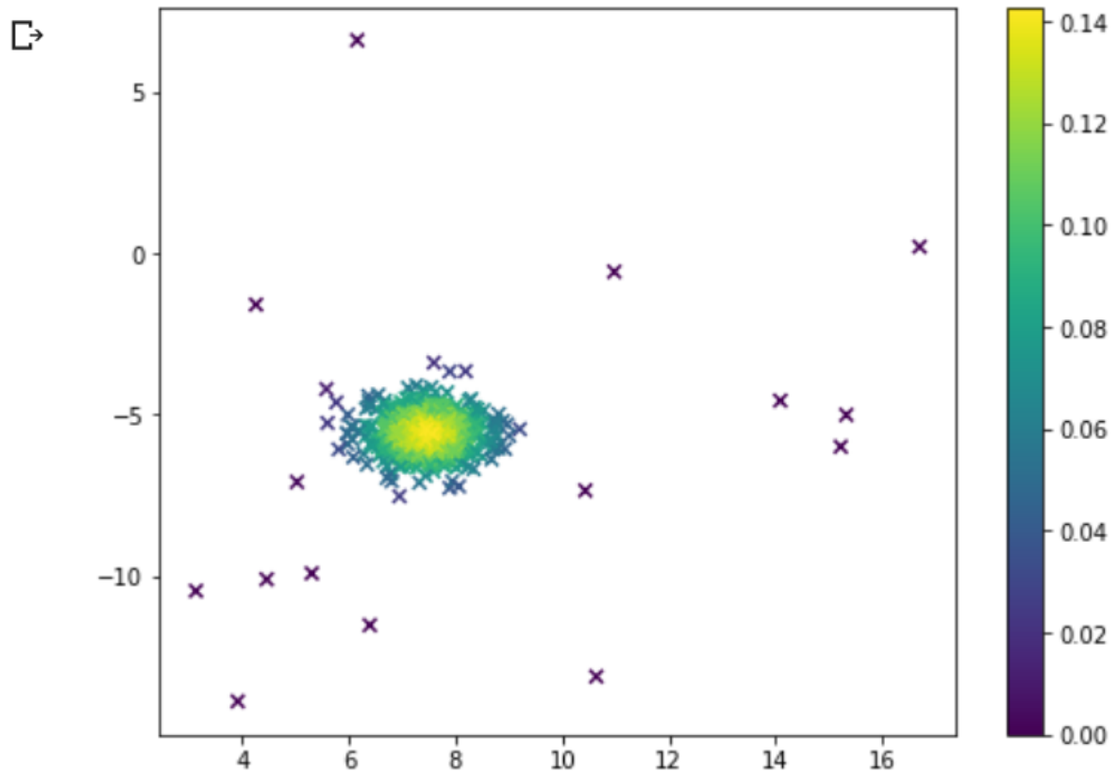
Gaussian-mixture Bayes classifiers: result



[A. Moore]

GMMs for density estimation

- We may be interested in fitting a Gaussian Mixture Model not for clustering but for density estimation

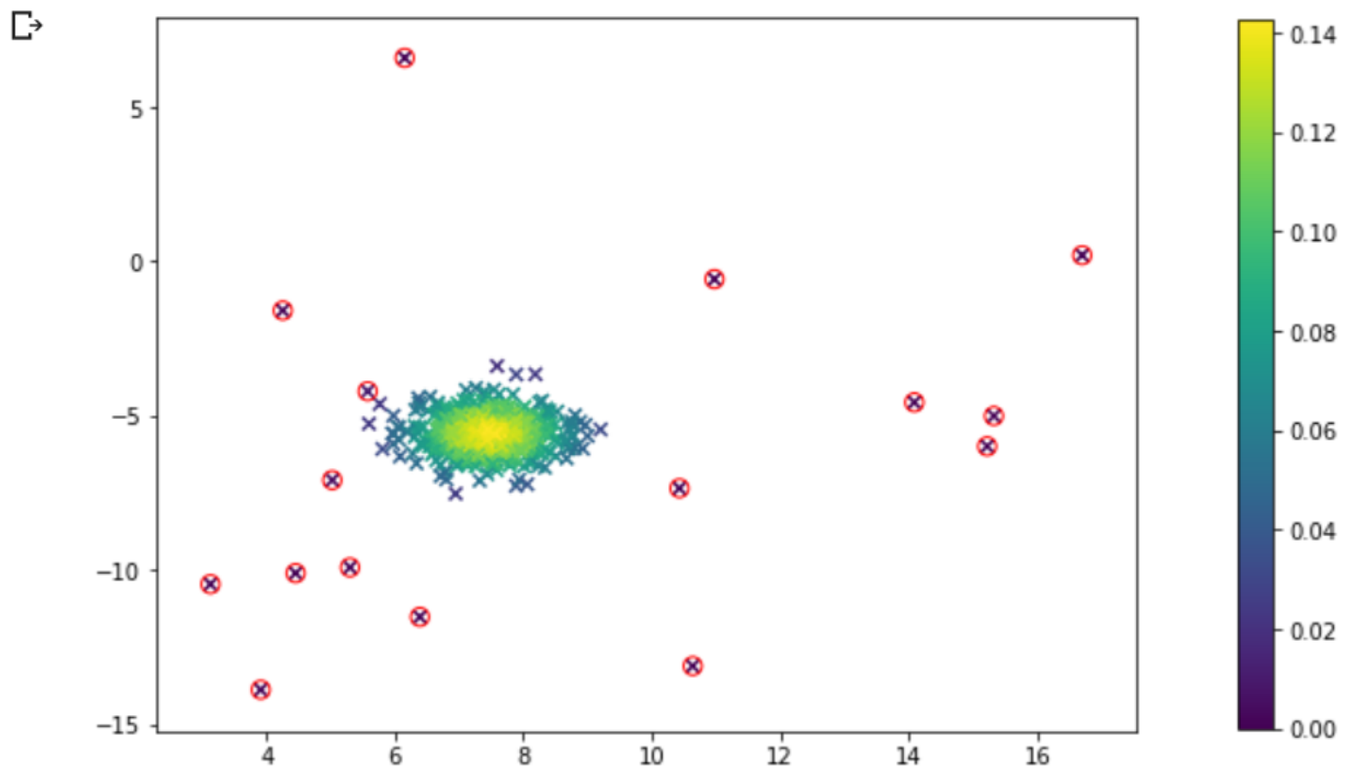


[Dukare] `https://bit.ly/2zjUoMq`

- Generative modeling of $P(Y, X)$
  - Model $P(X)$ as Gaussian mixture
  - Model $P(Y \mid X)$ using logistic regression, neural network, etc.
  - Combines the advantage of accurate predictions/robustness from discriminative model, with the ability to detect outliers!

Anomaly detection with mixture models

- Can determine outliers by comparing the estimated density of a data point against a threshold



[Dukare] `https://bit.ly/2zjUoMq`

- Picking threshold
  - Varying the threshold trades false-positives and false-negatives
  - Can use precision-recall/ ROC curves as evaluation criterion
  - This allows to optimize the threshold

Summary

Gaussian mixture models

- connection to other (un)/supervised models (GBC, k-means)
- use cases (density estimation, classification, generative modeling, anomaly detection ...)

EM algorithm

- a general algorithmic framework for latent variale models
- known to converge to a local optimal
- can be used whenever the E and M steps are tractable.

Reading materials

- Ch 2.3.9 & Ch 9.2 (mixture models): C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006
- Lecture 17 (clustering and k-means): CMSC 25300/35300 & STAT 27700: Mathematical Foundations of Machine Learning, 2019, `https://voices.uchicago.edu/willett/teaching/fall-2019-mathematical-foundations-of-machine-learning/`
- More demos and examples (probabilistic modeling): A. Krause, Introduction to Machine Learning, `https://las.inf.ethz.ch/teaching/introml-s20`
- GMM use cases: A. Moore, Clustering with Gaussian Mixtures, `https://www.cs.cmu.edu/~./awm/tutorials/gmm14.pdf`