Section Mixture of Gaussians.

- Mixture of Gaussians.
- The EM Algorithm
- Factor Analysis

LDA/ QDA

➢ Gaussian Discriminant Analysis (Review)

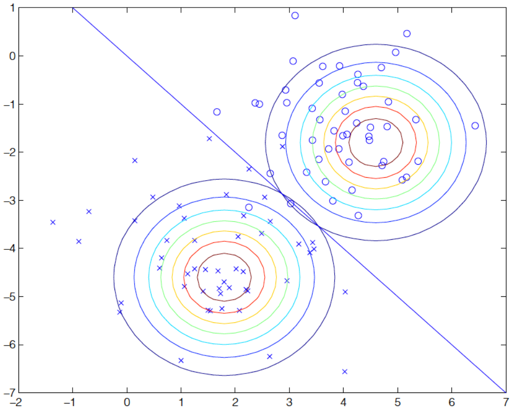
The goal is to define some parametric family of probability distributions and then maximize the likelihood of your data under this distribution by finding the best parameters.

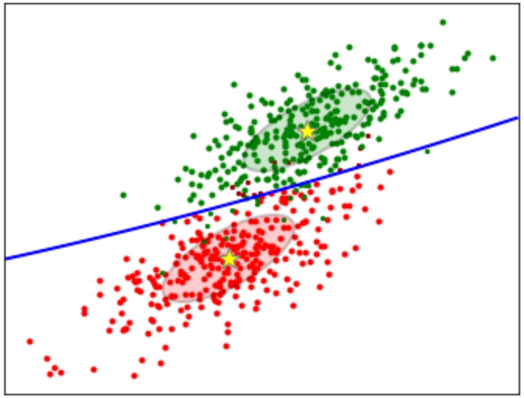
Data: $\left(\vec{x}^{(i)}, y^{(i)}\right)$ for classification

Want probability $P(y = k \mid \vec{x})$

By Bayes' Theorem, we calculate

$P(y)$ and $P(\vec{x} \mid y = k)$





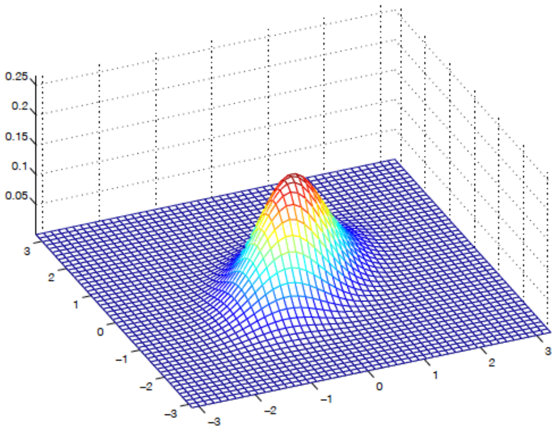
Suppose

$y \sim \text{Categorical}(\phi_1, \ldots, \phi_K)$

$\vec{x} \mid y = k \sim Normal\ (\vec{\mu}_k, \Sigma_k)$

Multivariant Gaussian Distribution $X{\sim}Normal\,(\vec{\mu}\,,\Sigma\,)$

$$f(\mathrm{X}; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathrm{X} - \mu)^T \Sigma^{-1} (\mathrm{X} - \mu)\right)$$



LDA

Suppose $\Sigma_0 = \Sigma_1 = \Sigma$ $\qquad$ $y \in \{0, 1\}$ *and* $y \sim Bernouli(\phi)$

Log likelihood function

$$\log(P(Data)) = \ell(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

$$= \log \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi).$$

Calculate the partial derivatives and **maximize** the (log) likelihood function:

$$\phi = \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$$

➢ QDA

Suppose Σ_0 and Σ_1 may different.

Want maximize posterior probability: $P(y = k \mid \vec{x}) = \dfrac{P(\vec{x} \mid y = k)\, P(y = k)}{P(\vec{x})}$

$$\log P(y = k \mid \vec{x}) = \log P(\vec{x} \mid y = k) + \log P(y = k) - \log P(\vec{x})$$

$$= -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\vec{x} - \mu_k)^T \Sigma_k^{-1} (\vec{x} - \mu_k) \ + \log \phi_k \ + \text{constant}$$

If $\Sigma_0 = \Sigma_1 = \cdots = \Sigma_\mathrm{k}$

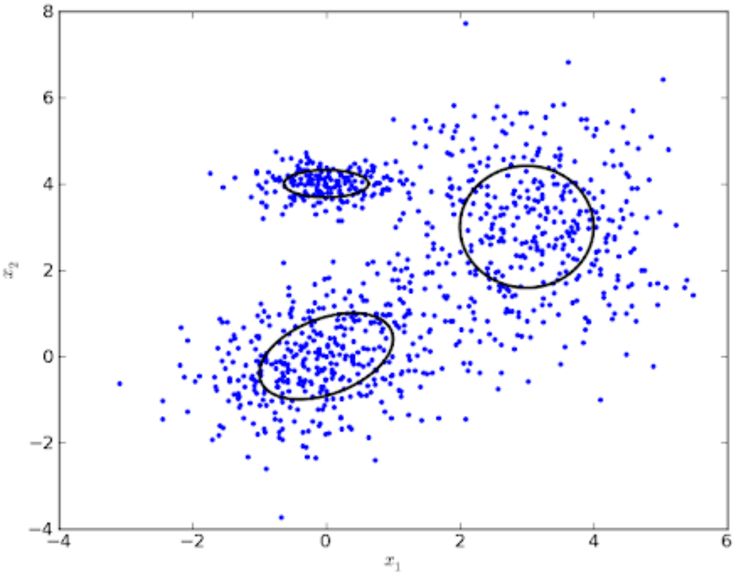
$$= \vec{x}^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \phi_k \ - \frac{1}{2}\log|\Sigma| + \text{constant}$$

➢ Mixture of Gaussians.

Data: $\underline{(\vec{x}^{(i)})\ z^{(i)})}$

No label on points.
Unsupervised learning.

We wish to model the data by specifying a joint distribution



$$P(\vec{x}^{(i)}, z^{(i)}) = P(\vec{x}^{(i)}|z^{(i)})P(z^{(i)})$$

The parameters $\phi_j = P(z^{(i)} = j)$

Here $z^{(i)} \sim \text{Categorical}(\phi_1, \ldots, \phi_K)$

$$\phi_i \geq 0; \ \sum_{j=1}^{K} \phi_j = 1$$

$$\vec{x}^{(i)}|z^{(i)} = j \sim Normal(\mu_j, \Sigma_j)$$

This is called the **mixture of Gaussians** (or Gaussian mixture) model.
A *Gaussian Mixture* is a function that is comprised of several Gaussians

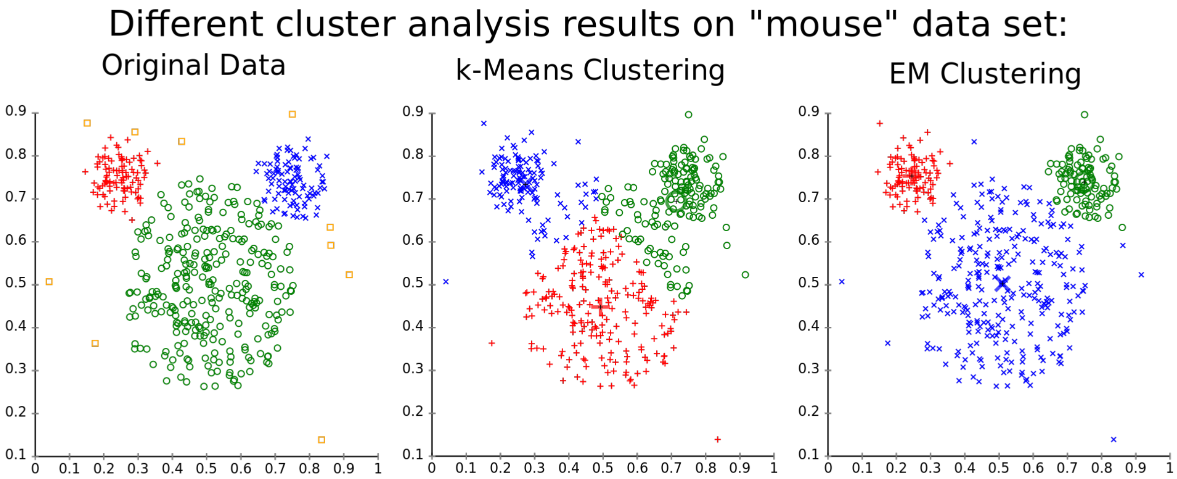
Here: $z^{(i)} \in \{1, \dots, K\}$ are **latent** random variables. They're hidden/unobserved.

$\vec{x}^{(i)}$ was drawn from one of K Gaussians depending on $z^{(i)}$.

Latent random variables make the estimation problem difficult.

ϕ_i defines how big or small the Gaussian function will be.

➢ Mixture of Gaussians v.s. K-means



K-means is the same as the mixture of Gaussian, where each Gaussian is spherical (zero mean, Identity covariance matrix)

The parameters of Gaussian Mixture model are ϕ_j, μ_j and Σ_j. To estimate them, we can write down the likelihood of our data:

$$\begin{aligned} \ell(\phi, \mu, \Sigma) &= \sum_{i=1}^{m} \log p(x^{(i)}; \phi, \mu, \Sigma) \\ &= \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi) \end{aligned}$$

- We have a hidden/latent variable $z^{(i)}$ for every observation.
- General problem: sum inside the log.
- How can we optimize this?

The random variables $z^{(i)}$ indicate which of the K Gaussians each $\vec{x}^{(i)}$ comes from.

If we knew what the $z^{(i)}$ were, the maximum likelihood problem will be easy.

we could then write down the likelihood as

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p(x^{(i)}|z^{(i)}; \mu, \Sigma) + \log p(z^{(i)}; \phi).$$

Maximizing this with respect to ϕ_j, μ_j and Σ_j gives the parameters:

$$\begin{aligned}
\phi_j &= \frac{1}{m}\sum_{i=1}^{m} 1\{z^{(i)} = j\}, \\
\mu_j &= \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\}x^{(i)}}{\sum_{i=1}^{m} 1\{z^{(i)} = j\}}, \\
\Sigma_j &= \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} 1\{z^{(i)} = j\}}.
\end{aligned}$$

So, these estimate formulas are the same as in LDA/QDA.

However, in our data here, we don't know the information of $z^{(i)}$.

➢ EM algorithm

EM (Expectation-Maximization) for density estimation.

EM is an elegant and powerful method for finding maximum likelihood solutions for models with latent variables.

1. E-step:

- In order to adjust the parameters, we must first solve the inference problem: Which Gaussian generated each datapoint?
- It tries to “guess” the value of the $z^{(i)}$ for each $\vec{x}^{(i)}$.
- We cannot be sure, so it’s a distribution over all possibilities.

- For each i, j, set

$$w_j^{(i)} = P\left(z^{(i)} = j \mid \vec{x}^{(i)};\ \phi, \mu \text{ and } \Sigma\right)$$

By Bayes’ rule, (parameters from the previous step)

$$p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{l=1}^{k} p(x^{(i)} | z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \phi)}$$

2. M-step:

- Each Gaussian gets a certain amount of posterior probability for each datapoint.
- We fit each Gaussian to the weighted datapoints.
- We can derive closed form updates for all parameters.
- Maximize the expectation of the complete data log-likelihood function.

$$
\begin{aligned}
\phi_j &:= \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}, \\
\mu_j &:= \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}, \\
\Sigma_j &:= \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}
\end{aligned}
$$

- **Initial Value: (step 0)**

  1. Random initialization.
  2. By K-means.

- **Scores:** Log-likelihood of our data:

$$\begin{aligned}\ell(\phi,\mu,\Sigma) &= \sum_{i=1}^{m} \log p(x^{(i)};\phi,\mu,\Sigma) \\ &= \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)};\mu,\Sigma)p(z^{(i)};\phi)\end{aligned}$$

sklearn

```
from sklearn.mixture import GaussianMixture

gmm = GaussianMixture(n_components=n_clusters, max_iter=50).fit(X)
gmm_scores = gmm.score_samples(X)
```

➢ Factor Analysis

- In EM algorithm for Gaussian Mixture, we usually suppose there are sufficient data to be able to discern the multiple-Gaussian structure in the data.

- This would be the case if our training set size m was significantly larger than the dimension n of the data. ($m \gg n$)

- Now, if $m \ll n$, it might be difficult to model the data even with a single Gaussian, much less a mixture of Gaussian. (pixels of pictures.)

- If we model the data as Gaussian, and estimate the mean and covariance using the usual maximum likelihood estimators,

$$\begin{aligned} \mu &= \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \\ \Sigma &= \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T \end{aligned}$$

- The matrix Σ is not invertible and det(Σ)=0!

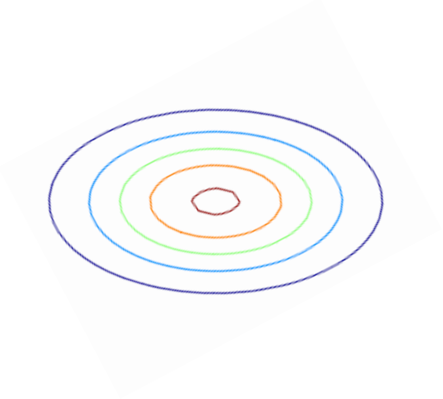
- More generally, unless m exceeds n by some reasonable amount, the maximum likelihood estimates of the mean and covariance may be quite poor.
- Nonetheless, we would still like to be able to fit a reasonable Gaussian model to the data, and perhaps capture some interesting covariance structure in the data.

Possible Solutions: Put restrictions on Σ.

For example, choose to fit a covariance matrix Σ that is diagonal.

$$\Sigma_{jj} = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$$

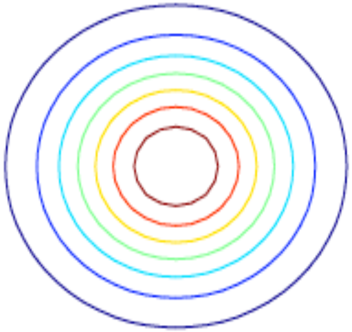
Recall that the contours of a Gaussian density are ellipses. A diagonal Σ corresponds to a Gaussian where the major axes of these ellipses are axis-aligned.



We may place a further restriction on the covariance matrix that not only must it be diagonal, but its diagonal entries must all be equal.
In this setting, we have $\Sigma = \sigma^2 I$, where σ^2 is the parameter under our control.
The maximum likelihood estimate of σ^2 can be found to be:

$$\sigma^2 = \frac{1}{mn} \sum_{j=1}^{n} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$$

This model corresponds to using Gaussians whose densities have contours that are circles (in 2 dimensions; or spheres/hyperspheres in higher dimensions)



If we were fitting a full, unconstrained, covariance matrix Σ to data, it was necessary that $m \geq n + 1$ in order for the maximum likelihood estimate Σ of not to be singular. Under either of the two restrictions above, we may obtain non-singular Σ when $m \geq 2$.

However, restricting Σ to be diagonal also means modeling the different coordinates $x_i \; and \; x_j$ of the data as being uncorrelated and independent. This will fail to capture interesting correlation structure in the data.

we will describe the factor analysis model, which uses more parameters than the diagonal Σ and captures some correlations in the data, but also without having to fit a full covariance matrix.

$$\begin{aligned} z &\sim \mathcal{N}(0, I) \\ x|z &\sim \mathcal{N}(\mu + \Lambda z, \Psi) \end{aligned}$$