Introduction to Generative Models

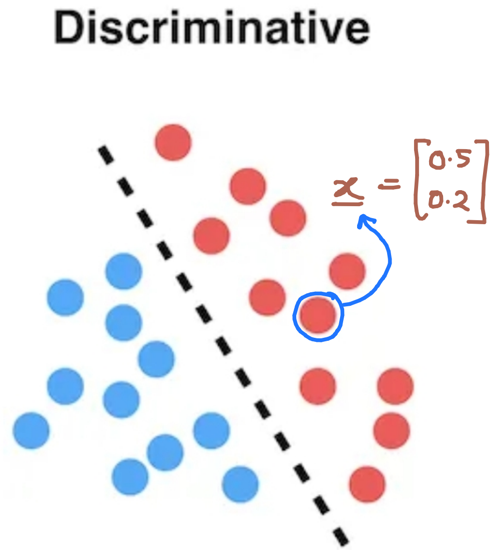
The models introduced in this course so far are so-called discriminative models

- e.g. Logistic regression, SVM, Decision trees, Random Forests

- They are designed to learn from data how to predict the output conditionally given the input

  * Say $p(y=1 \mid \underline{x} = [0.5, 0.2]^T) = 0.7$

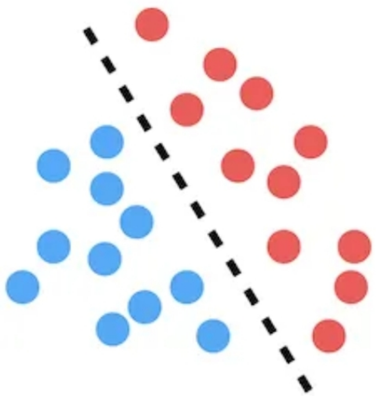
    $p(y=-1 \mid \underline{x} = [0.5, 0.2]^T) = 0.3$



- They are also called conditional models

- They aim to model $p(y \mid \underline{x})$

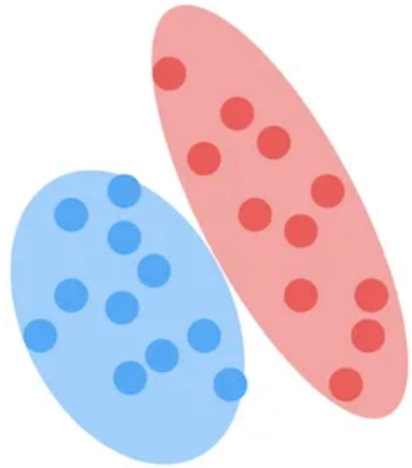
Discriminative



- Only describe the conditional distribution of the output for a given input $p(y \mid \underline{x})$

- Has limited understanding
  - Cannot be used to simulate more data
  - Cannot find patterns with only input variables

Generative



- Describes the joint distribution of both inputs and outputs

$$p(\underline{x}, y)$$

- Has deeper understanding of the data
  - Can simulate more data
  - Can find patterns among inputs in the absence of output values

- Probabilistic notations for generative models : $p(\underline{x}, y \mid \underline{\theta})$, $p_{\theta}(\underline{x}, y)$
  - The models depend upon some learnable parameter $\underline{\theta}$

- Can generative models also predict the output y given an input $\underline{x}$?
  - Yes, we will need to obtain the conditional distribution $p(y \mid \underline{x})$ from $p(\underline{x}, y)$ using probability theory

- We will demonstrate this idea using generative Gaussian mixture model (GMM) $\longrightarrow$ applicable to both
  - Supervised setting $(y, \underline{x})$
  - Un-supervised setting $(\underline{x})$

Gaussian Mixture Model (for classification)

- Consider a classification problem
  - $\underline{x}$ is numerical and y is a categorical variable
- GMM attempts to model $p(\underline{x}, y) \leftrightarrow$ joint distribution of $\underline{x}$ and y
- It makes use of the factorization

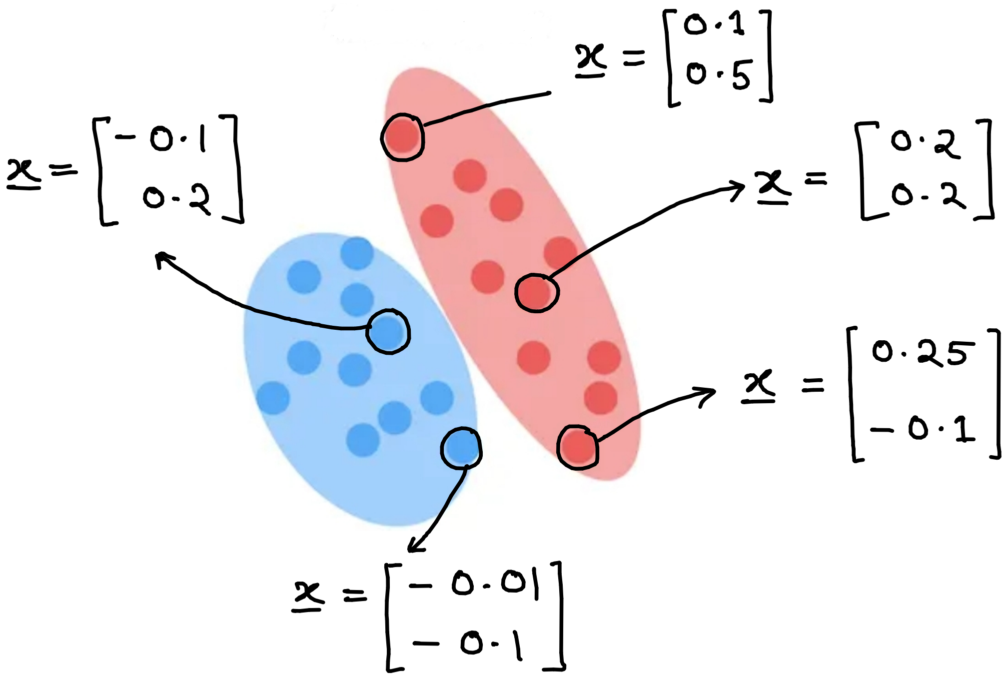
Marginalization
$p(y) = \int p(\underline{x}, y)\, d\underline{x}$

$$p(\underline{x}, y) = \underbrace{p(\underline{x} \mid y)}_{\text{class-conditional distribution of } x \text{ for a certain class } y} \; \underbrace{p(y)}_{\text{marginal distribution of } y}$$

- y is categorical $\Leftrightarrow$ $y \in$ set of classes $\{1, 2, \dots, M\}$

$$\text{Consider} \begin{cases} p(y=1) = \pi_1 \\ p(y=2) = \pi_2 \\ \vdots \\ p(y=M) = \pi_M \end{cases}$$

Unknown parameters



- Intuition:

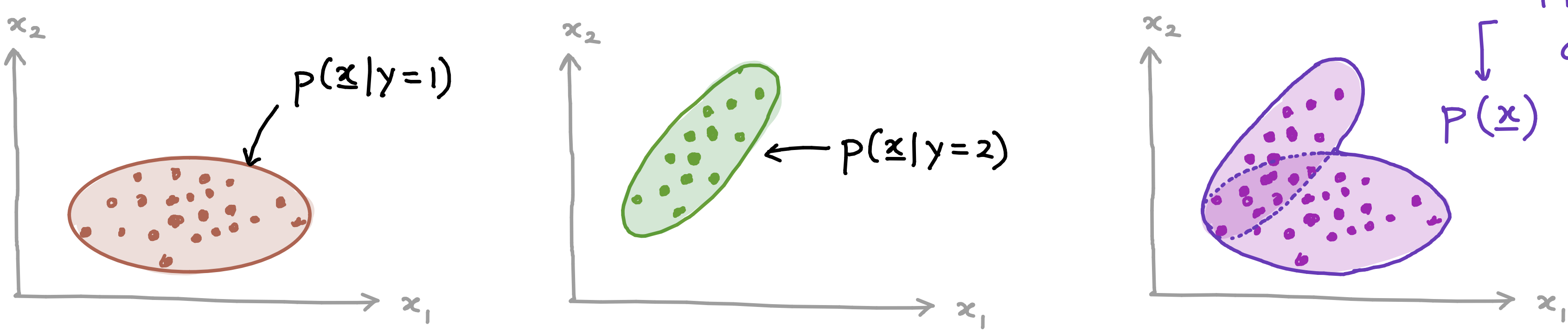
  If it is possible to predict the class y based on $\underline{x}$, then the distribution of $\underline{x}$ may be estimated from y

- The basic assumption for a GMM: is that $p(\underline{x} \mid y)$ is a Gaussian distribution

these values depend on y

$$p(\underline{x} \mid y) = \mathcal{N}\left(\underline{x} \mid \underline{\mu}_y, \underline{\underline{\Sigma}}_y\right)$$

For example:



Marginal distribution of $\underline{x}$ is a mixture of Gaussians (hence GMM)

Supervised Learning of GMM

- The unknown parameters of GMM that are to be learned from data are

$$\underline{\Theta} = \{ \underline{\mu}_m, \underline{\underline{\Sigma}}_m, \pi_m \}_{m=1}^{M}$$

or, equivalently,
$$\underline{\Theta} = \begin{bmatrix} \underline{\mu}_1 \\ \vdots \\ \underline{\mu}_M \\ \text{vec}(\underline{\underline{\Sigma}}_1) \\ \vdots \\ \text{vec}(\underline{\underline{\Sigma}}_M) \\ \pi_1 \\ \vdots \\ \pi_M \end{bmatrix}$$

- Training data consists of $\mathcal{T} = \{ \underline{x}^{(i)}, y^{(i)} \}_{i=1}^{N}$

- The parameter vector $\underline{\Theta}$ is learned by maximizing the log-likelihood of data

$$\hat{\underline{\Theta}} = \arg\max_{\underline{\Theta}} \; \ln \underbrace{p\left(\{ \underline{x}^{(i)}, y^{(i)} \}_{i=1}^{N} \mid \underline{\Theta} \right)}_{\text{joint distribution}}$$

It is due to the generative nature of the model that we maximize the joint distribution (and not the conditional distribution $p(y \mid \underline{x})$ as in discriminative models)

- The log-likelihood could be written as:

$$\ln p\left(\{\underline{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \mid \underline{\theta}\right) = \ln\left(p\left(x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, \ldots, x^{(N)}, y^{(N)} \mid \underline{\theta}\right)\right)$$

Assuming independence of data points

$$= \ln\left(p\left(\underline{x}^{(1)}, y^{(1)} \mid \underline{\theta}\right) p\left(\underline{x}^{(2)}, y^{(2)} \mid \underline{\theta}\right), \ldots, p\left(\underline{x}^{(N)}, y^{(N)} \mid \underline{\theta}\right)\right)$$

$$= \ln\left(p\left(\underline{x}^{(1)} \mid y^{(1)}, \underline{\theta}\right) p\left(y^{(1)} \mid \underline{\theta}\right), \ldots, p\left(\underline{x}^{(N)} \mid y^{(N)}, \underline{\theta}\right) p\left(y^{(N)} \mid \underline{\theta}\right)\right)$$

$$= \sum_{i=1}^{N} \left\{\ln p\left(\underline{x}^{(i)} \mid y^{(i)}, \underline{\theta}\right) + \ln p\left(y^{(i)} \mid \underline{\theta}\right)\right\}$$

One could further expand the expression for each class value

$$p\left(y^{(i)} = m \mid \underline{\theta}\right) = \pi_m$$

$$p\left(\underline{x}^{(i)} \mid y^{(i)} = m, \underline{\theta}\right) = \mathcal{N}\left(\underline{x}^{(i)} \mid \underline{\mu}_m, \underline{\underline{\Sigma}}_m\right)$$

$$= \sum_{i=1}^{N} \sum_{m=1}^{M} \left\{\ln p\left(\underline{x}^{(i)} \mid y^{(i)} = m, \underline{\theta}\right) + \ln p\left(y^{(i)} = m \mid \underline{\theta}\right)\right\}$$

$$= \sum_{i=1}^{N} \sum_{m=1}^{M} \mathbb{I}\{y^{(i)} = m\} \left\{\ln \mathcal{N}\left(x^{(i)} \mid \underline{\mu}_m, \underline{\underline{\Sigma}}_m\right) + \ln p\left(y^{(i)} \mid \underline{\theta}\right)\right\}$$

Indicator function

- Optimization problem

$$\hat{\underline{\theta}} = \arg\max_{\underline{\theta}} \sum_{i=1}^{N} \sum_{m=1}^{M} \mathbb{I}\{y^{(i)} = m\} \left\{ \ln \mathcal{N}\left(\underline{x}^{(i)} \mid \underline{\mu}_m, \underline{\underline{\Sigma}}_m\right) + \ln p(y^{(i)} \mid \underline{\theta}) \right\}$$

- It turns out that the above optimization problem has CLOSED-FORM solution
  - Marginal class probabilities, $\{\pi_m\}_{m=1}^{M}$:

    $$\hat{\pi}_m = \frac{n_m}{N}$$

    n_m ← number of training points in class 'm'

    (i.e. proportion of the class in training data)

  - Mean vector of each class, $\underline{\mu}_m$:

    $$\hat{\underline{\mu}}_m = \frac{1}{n_m} \sum_{i:y^{(i)}=m} \underline{x}^{(i)}$$

    empirical mean among all training points of class 'm'

  - Covariance matrix $\underline{\underline{\Sigma}}_m$ for each class :

    $$\hat{\underline{\underline{\Sigma}}}_m = \frac{1}{n_m} \sum_{i:y^{(i)}=m} \left(\underline{x}^{(i)} - \hat{\underline{\mu}}_m\right)\left(\underline{x}^{(i)} - \hat{\underline{\mu}}_m\right)^T$$

Note: We could compute the parameters $\{\hat{\pi}_m, \hat{\underline{\mu}}_m, \hat{\underline{\underline{\Sigma}}}_m\}_{m=1}^{M}$ irrespective of whether the data actually comes from a Gaussian distribution or not!

Discriminant Analysis

- We have now learned the GMM $p(\underline{x}, y)$ generative model, where $\underline{x}$ is numerical and y is categorical

- How to predict the output label given new inputs using GMM?
  - By using conditional distribution $p(y|\underline{x})$

- From probability theory, we have

$$\underbrace{p(y|\underline{x})}_{\text{called the predictive distribution}} = \frac{p(\underline{x}, y)}{p(\underline{x})} = \frac{p(\underline{x}, y)}{\sum_{j=1}^{M} p(\underline{x}, y=j)} = \frac{p(\underline{x}|y)\, p(y)}{\sum_{j=1}^{M} p(\underline{x}|y=j)\, p(y=j)}$$

- Therefore, we get a GMM classifier (acting now as a discriminative model)

$$\boxed{p(y=m|\underline{x}^*) = \frac{\hat{\pi}_m \, \mathcal{N}(\underline{x}^* | \hat{\underline{\mu}}_m, \hat{\underline{\underline{\Sigma}}}_m)}{\sum_{j=1}^{M} \hat{\pi}_j \, \mathcal{N}(\underline{x}^* | \hat{\underline{\mu}}_j, \hat{\underline{\underline{\Sigma}}}_j)}}$$

$\underline{x} \in \mathbb{R}^p$

$$\mathcal{N}(\underline{x} | \underline{\mu}_m, \underline{\underline{\Sigma}}_m) = \frac{1}{(2\pi)^{p/2} |\underline{\underline{\Sigma}}|^{1/2}} \exp\left(-\frac{1}{2} (\underline{x} - \underline{\mu}_m)^T \underline{\underline{\Sigma}}_m^{-1} (\underline{x} - \underline{\mu}_m)\right)$$

- GMM classifier class probability prediction

$$p(y=m \mid \underline{x}^*) = \frac{\hat{\pi}_m \, \mathcal{N}(\underline{x}^* \mid \hat{\underline{\mu}}_m, \hat{\underline{\underline{\Sigma}}}_m)}{\sum_{j=1}^{M} \hat{\pi}_j \, \mathcal{N}(\underline{x}^* \mid \hat{\underline{\mu}}_j, \hat{\underline{\underline{\Sigma}}}_j)}$$

- We can obtain hard predictions $\hat{y}^*$ by selecting the class which is most probable

$$\hat{y}^* = \arg\max_{m} \; p(y=m \mid \underline{x}^*)$$

$$p(y=m \mid \underline{x}^*) = \frac{\hat{\pi}_m \, \mathcal{N}(\underline{x}^* \mid \hat{\underline{\mu}}_m, \hat{\underline{\underline{\Sigma}}}_m)}{\sum_{j=1}^{M} \hat{\pi}_j \, \mathcal{N}(\underline{x}^* \mid \hat{\underline{\mu}}_j, \hat{\underline{\underline{\Sigma}}}_j)}$$

only the numerator depends on 'm'

denominator only depends on $\underline{x}^*$

- Hard predictions

$$\hat{y}^* = \arg\max_m \; p(y=m \mid \underline{x}^*)$$

- One can also obtain the decision boundaries of the GMM classifier

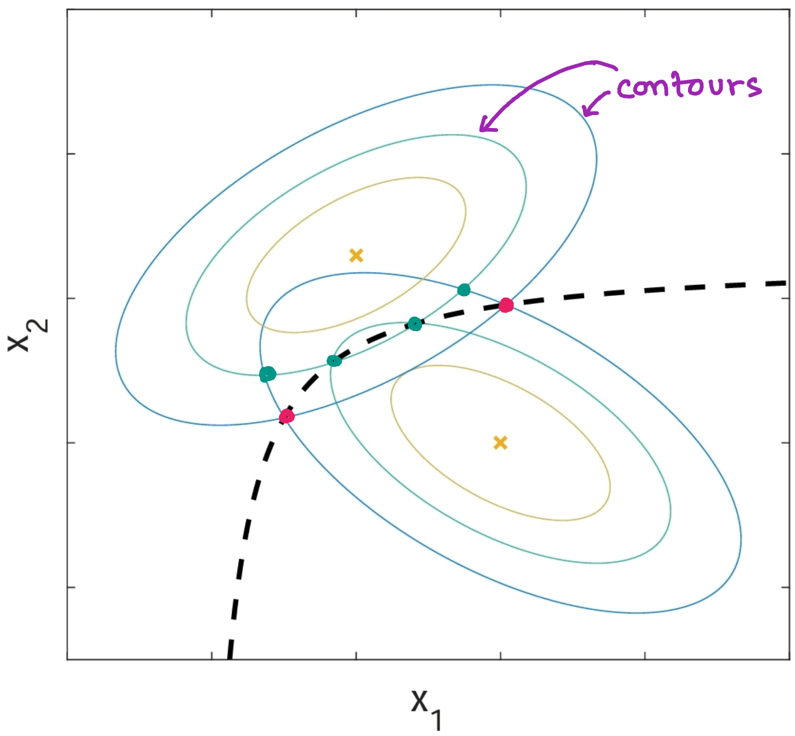
$$\hat{y}^* = \arg\max_m \left\{ \ln \hat{\pi}_m + \ln \mathcal{N}(\underline{x}^* \mid \hat{\underline{\mu}}_m, \hat{\underline{\underline{\Sigma}}}_m) \right\}$$

- Logarithm of Gaussian distribution $\xrightarrow{\text{leads to}}$ Quadratic decision boundaries

$$\propto \quad \underbrace{(\underline{x} - \underline{\mu}_m)^T \underline{\underline{\Sigma}}_m^{-1} (\underline{x} - \underline{\mu}_m)}_{\text{Quadratic in nature}}$$

Therefore, a GMM classification is called Quadratic Discriminant Analysis (QDA)

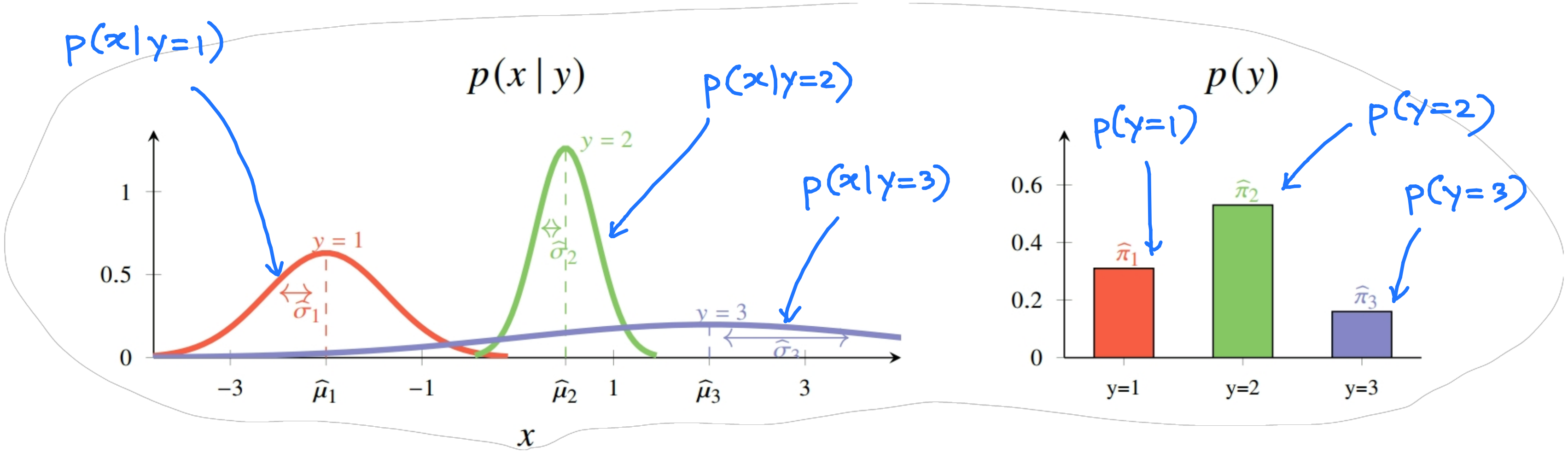
GMM classifier decision boundary (QDA decision boundary)



Two Gaussian PDFs with different covariance matrices intersect along a quadratic line

Illustration of QDA (GMM classifier) for $M=3$ classes

Input dimension, $p=1$



The parameters $\rightarrow$ $\hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2, \hat{\mu}_3, \hat{\sigma}_3, \hat{\pi}_1, \hat{\pi}_2, \hat{\pi}_3$ are learned

The predictive distribution $p(y=m \mid x)$ is shown below: