# Probabilistic Models for Supervised Learning (contd)

CS771: Introduction to Machine Learning

Piyush Rai

# Recap: Prob. Models for Supervised Learning

- Goal: Learn the conditional distribution $p(y|x)$. Broadly, two approaches

## Discriminative Approach

$$p(y|x) = p(y|f(x, w))$$

$f$ can be any function which uses inputs and weights $w$ to defines parameters of distr. $p$

Some examples

$$p(y|x) = \mathcal{N}(y|w^\top x, \beta^{-1})$$

$$p(y|x) = \text{Bernoulli}(y|\sigma(w^\top x))$$

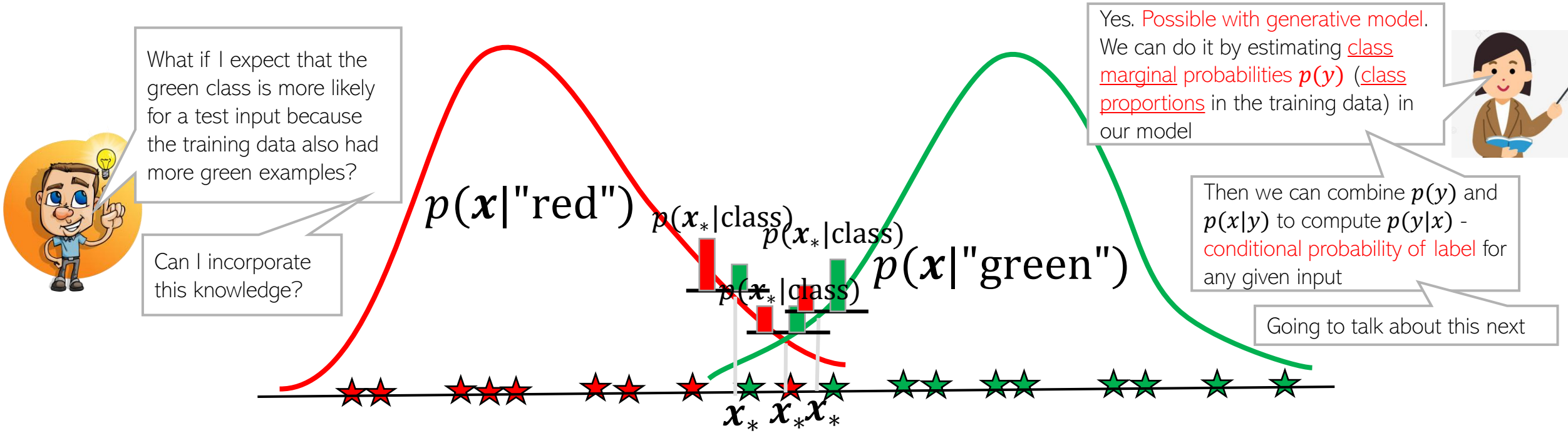
## Generative Approach

$$p(y|x) = \frac{p(y, x)}{p(x)}$$

Requires estimating the joint distribution of inputs and outputs to get the conditional $p(y|x)$ (unlike the discriminative approach which directly estimates the conditional $p(y|x)$ and does not model the distribution of $x$)

# Generative Classification: A Basic Idea

- Learn the probability distribution $p(x|y=k)$ of inputs from each class $k$




- We usually assume some form for $p(x|y=k)$ (e.g., Gaussian) and estimate the parameters of that distribution (MLE/MAP/fully posterior)
- We then predict label of test input $x_*$ by comparing probabilities under each class
  - Or can report the probability of belonging to each class (soft prediction)

# Generative Classification

- Suppose we have training data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ from $K$ classes

- The conditional probability of label $y_n$ given the input $\boldsymbol{x}_n$

$$p(y_n = k|x_n) = \frac{p(\boldsymbol{x}_n, y_n = k)}{p(x_n)}$$

Known as the "class-conditional" distribution

Probability distribution of the inputs from class $k$

Known as "class-marginal" or "class-prior" distribution

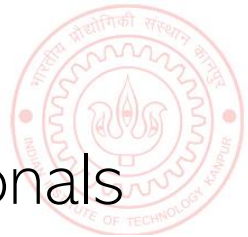
$$= \frac{p(y_n = k) \times p(\boldsymbol{x}_n|y_n = k)}{p(\boldsymbol{x}_n)}$$

The numerator (joint distribution of $x_n$ and $y_n$) summed over all $K$ values of $y_n$

Marginal distribution of just the labels (not looking at the inputs) – Bernoulli/multinoulli

Marginal distribution of the input $\boldsymbol{x}_n$

- We use the training data to estimate the class-marginal and class-conditionals



# Estimating Class Marginals

- Estimating class marginals $p(y = k)$ is usually straightforward
- Since labels are discrete, we assume class marginal $p(y)$ to be a multinoulli

If only two classes, assume Bernoulli

$\pi_k = p(y = k)$

These probabilities sum to 1: $\sum_{k=1}^{K} \pi_k = 1$

$$p(y|\boldsymbol{\pi}) = \text{multinoulli}(y|\pi_1, \pi_2, \ldots, \pi_K) = \prod_{k=1}^{K} \pi_k^{\mathbb{I}[y=k]}$$

- Given $N$ i.i.d. labelled examples $\{(x_n, y_n)\}_{n=1}^{N}$, $y_n \in \{1,2,\ldots,K\}$ the MLE soln

$$\boldsymbol{\pi}_{MLE} = \underset{\boldsymbol{\pi}}{\text{argmax}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{\pi})$$

Subject to constraint $\sum_{k=1}^{K} \pi_k = 1$

- MLE solution is $p(y = k) = \pi_k = N_k/N$ where $N_k = \sum_{n=1}^{N} \mathbb{I}[y = k]$
  - Thus $p(y = k) = \pi_k$ is simply the fraction of inputs from class $k$
- Can also compute MAP estimate or full posterior of $\boldsymbol{\pi}$ using a Dirichlet prior

# Estimating Class-Conditionals

To be estimated using the $N_k$ training inputs $\{\boldsymbol{x}_n : y_n = k\}$ from class $k$

- Can assume a distribution $p(\boldsymbol{x}|y = k) = p(\boldsymbol{x}|\theta_k)$ for inputs of each class $k$
- If $\boldsymbol{x}$ is $D$-dimensional, $p(\boldsymbol{x}|\theta_k)$ will be a $D$-dimensional distribution
- Can compute MLE/MAP estimate or full posterior of $\theta_k$
  - This essentially is a density estimation problem for the class-cond.
  - In principle, can use any density estimation method
- Choice of the form of $p(\boldsymbol{x}|\theta_k)$ depends on various factors
  - Nature of input features, e.g.,
    - If $\boldsymbol{x} \in \mathbb{R}^D$, can use a $D$-dim Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
    - If $\boldsymbol{x} \in \{0,1\}^D$, can use $D$ Bernoullis (one for each feature)
    - Can also choose other more sophisticated distributions
  - Amount of training data available (important)
    - If $D$ large and $N_k$ small, it will be difficult to get a good estimate $\theta_k$

E.g., if $p(\boldsymbol{x}|\theta_k)$ is multivariate Gaussian then assume it to have a diagonal covariance matrix instead of full covariance matrix

Such assumptions greatly reduce the number of parameters to be estimated

In such cases, we may need to regularize $\theta_k$ or make some simplifying assumptions on $p(\boldsymbol{x}|\theta_k)$, such as features being conditionally independent given class e.g., $p(\boldsymbol{x}|\theta_k) = \prod_{d=1}^{D} p(x_d|\theta_{kd})$ - naïve Bayes

Especially if the number of features ($D$) is very large because large value of $D$ means $k$ consists of a large number of parameters (e.g., in the Gaussian case, $\theta_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $D$ params for $\boldsymbol{\mu}_k$ and $O(D^2)$ params for $\boldsymbol{\Sigma}_k$. Can overfit

# Generative Classification: At Test Time

- Recall the form of the conditional distribution of the label

Class-marginal accounts for the frequency of class $k$ labels in the training data

Class-conditional distribution of inputs accounts for the shape/spread of class $k$

$$p(y_* = k|\boldsymbol{x}_*) = \frac{p(y_* = k) \times p(\boldsymbol{x}_*|y_* = k)}{p(\boldsymbol{x}_*)}$$

$$\propto p(y_* = k) \times p(\boldsymbol{x}_*|y_* = k)$$

$$\propto \hat{\pi}_k \times p(\boldsymbol{x}_*|\hat{\theta}_k)$$

Probability of $\boldsymbol{x}_*$ belonging to class $k$ is proportional to the fraction of training inputs from class $k$ times the probability of $\boldsymbol{x}_*$ under the distribution of inputs from class $k$

- If we assume the class-marginal to be uniform ($p(y_* = k) = 1/K$) then

$$p(y_* = k|\boldsymbol{x}_*) \propto p(\boldsymbol{x}_*|\hat{\theta}_k)$$

- The most likely label is $y_* = \text{argmax}_{k \in \{1,2,\ldots,K\}} p(y_* = k|\boldsymbol{x}_*)$

# Gen. Classifn. using Gaussian Class-conditionals

- The generative classification model $p(y=k|\boldsymbol{x}) = \frac{p(y=k)p(\boldsymbol{x}|y=k)}{p(\boldsymbol{x})}$

A benefit of modeling each class by a distribution (recall that LwP had issues)

- Assume each class-conditional $p(\boldsymbol{x}|y=k)$ to be a Gaussian

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}}\exp[-(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)]$$

Since the Gaussian's covariance models its shape, we can learn the shape of each class ☺

- Class marginal is multinoulli $p(y=k) = \pi_k, \pi_k \in (0,1), \sum_{k=1}^K \pi_k = 1$
- Let's denote the parameters of the model collectively by $\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$
  - Can estimate these using MLE/MAP/Bayesian inference
  - Already saw the MLE solution for $\boldsymbol{\pi}$: $\pi_k = N_k/N$ (can also do MAP)
  - MLE solution for $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{y_n=k}\boldsymbol{x}_n$, $\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{y_n=k}(\boldsymbol{x}_n-\boldsymbol{\mu}_k)(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^\top$

Can also do MAP estimation for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using a Gaussian prior on $\boldsymbol{\mu}_k$ and inverse Wishart prior on $\boldsymbol{\Sigma}_k$

Exercise: Try to derive this. I will provide a separate note containing the derivation

- If using point est (MLE/MAP) for $\theta$, predictive distribution will be

$$p(y_* = k|\boldsymbol{x}_*, \theta) = \frac{\pi_k|\boldsymbol{\Sigma}_k|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_*-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_*-\boldsymbol{\mu}_k)\right]}{\sum_{k=1}^K \pi_k|\boldsymbol{\Sigma}_k|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_*-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_*-\boldsymbol{\mu}_k)\right]}$$

Can predict the most likely class for the test input $\boldsymbol{x}_*$ by comparing these probabilities for all values of $k$

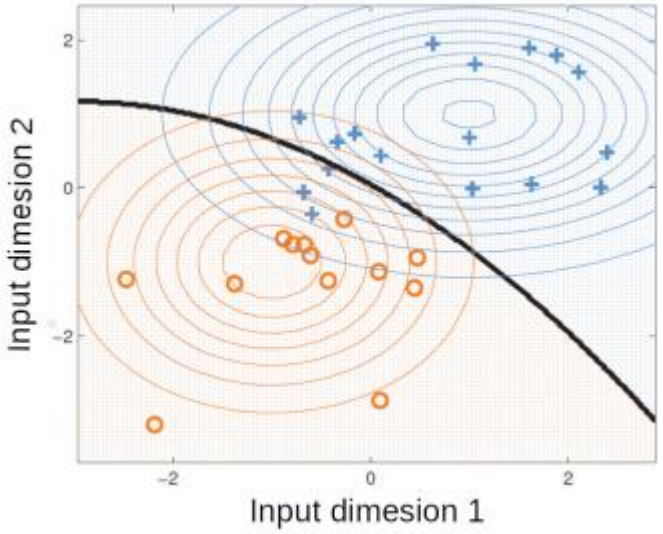
Note that the exponent has a Mahalanobis distance like term. Also, accounts for the fraction of training examples in class k

# Decision Boundary with Gaussian Class-Conditional

- As we saw, the prediction rule when using Gaussian class-conditional

$$p(y=k|\boldsymbol{x},\theta)=\frac{\pi_k|\boldsymbol{\Sigma_k}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]}{\sum_{k=1}^{K}\pi_k|\boldsymbol{\Sigma_k}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]}$$

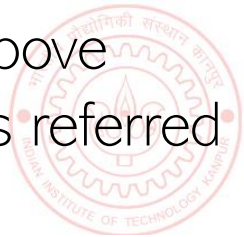
- The decision boundary between any pair of classes will be a quadratic curve




Reason: For any two classes $k$ and $k'$ at the decision boundary, we will have $p(y=k|x,\theta)=p(y=k'|x,\theta)$. Comparing their logs and ignoring terms that don't contain $\boldsymbol{x}$, can easily see that

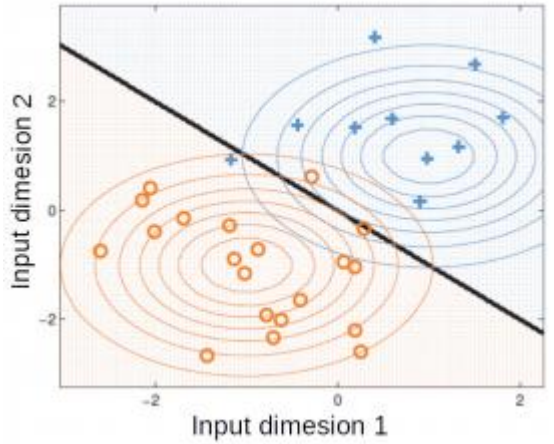
$$(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)-(\mathbf{x}-\boldsymbol{\mu}_{k'})^\top\boldsymbol{\Sigma}_{k'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{k'})=0$$

Decision boundary contains all inputs $\boldsymbol{x}$ that satisfy the above. This is a quadratic function of $\boldsymbol{x}$ (this model is sometimes referred to Quadratic Discriminant Analysis)



# Decision Boundary with Gaussian Class-Conditional

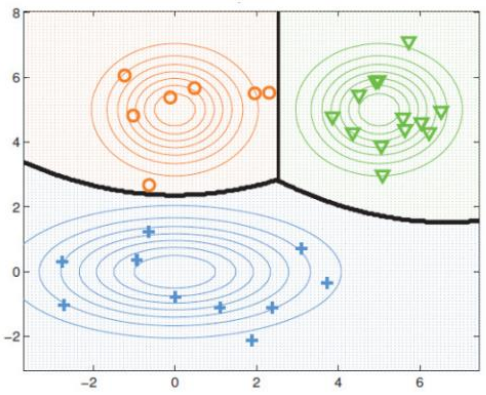
- Assume all classes are modeled using the same covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$
- In this case, the decision boundary b/w any pair of classes will be linear

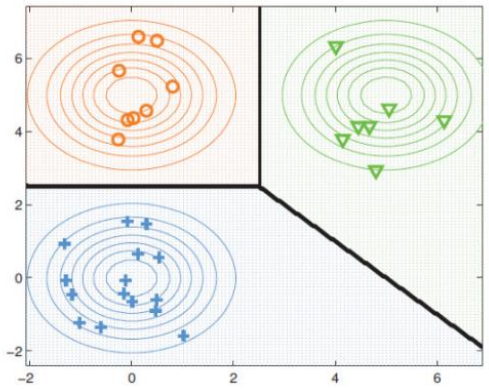



Reason: Again using $p(y = k|x, \theta) = p(y = k'|x, \theta)$, comparing their logs and ignoring terms that don't contain $\boldsymbol{x}$, we have

$$(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) - (\mathbf{x} - \boldsymbol{\mu}_{k'})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{k'}) = 0$$

Quadratic terms of $\boldsymbol{x}$ will cancel out; only linear terms will remain; hence decision boundary will be a linear function of $\boldsymbol{x}$ (Exercise: Verify that we can indeed write the decision boundary between this pair of classes as $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ where $\boldsymbol{w}$ and $b$ depend on $\boldsymbol{\mu}_k$, $\boldsymbol{\mu}_{k'}$ and $\boldsymbol{\Sigma}$)





If we assume the covariance matrices of the assumed Gaussian class-conditionals for any pair of classes to be equal, then the learned separation boundary b/w this pair of classes will be linear; otherwise, quadratic as shown in the figure on left

# A Closer Look at the Linear Case

- For the linear case (when $\boldsymbol{\Sigma_k = \Sigma, \forall k}$), the class conditional probability

$$p(y = k|\boldsymbol{x}, \theta) \propto \pi_k \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

- Expanding further, we can write the above as

$$p(y = k|\boldsymbol{x}, \theta) \propto \exp\left[\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k\right] \exp\left[\boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{x}\right]$$

- Therefore, the above class posterior probability can be written as

$$p(y = k|\boldsymbol{x}, \theta) = \frac{\exp\left[\boldsymbol{w}_k^\top \boldsymbol{x} + b_k\right]}{\sum_{k=1}^{K} \exp\left[\boldsymbol{w}_k^\top \boldsymbol{x} + b_k\right]}$$

$$\boldsymbol{w}_k = \Sigma^{-1}\boldsymbol{\mu}_k \qquad b_k = -\tfrac{1}{2}\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

If all Gaussians class-cond have the same covariance matrix (basically, of all classes are assumed to have the same shape)

- The above has *exactly* the same form as softmax classification (thus softmax is a special case of a generative classification model with Gaussian class-conditionals)

# A Very Special Case: LwP Revisited

- Note the prediction rule when $\boldsymbol{\Sigma_k = \Sigma, \forall k}$

$$\hat{y} = \arg\max_k p(y = k|\boldsymbol{x}) = \arg\max_k \pi_k \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

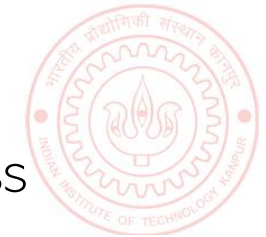
$$= \arg\max_k \log \pi_k - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)$$

- Also assume all classes to have equal no. of training examples, i.e., $\boldsymbol{\pi_k = 1/K}$. Then

$$\hat{y} = \arg\min_k (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)$$

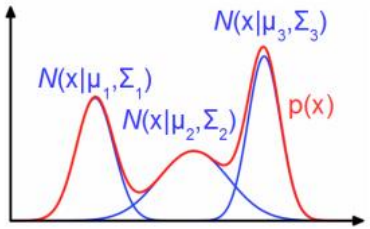
The Mahalanobis distance matrix = $\boldsymbol{\Sigma^{-1}}$

- Equivalent to assigning $\boldsymbol{x}$ to the "closest" class in terms of a Mahalanobis distance
- If we further assume $\boldsymbol{\Sigma = I_D}$ then the above is exactly the LwP rule
  - Thus LwP assumes spherical classes with roughly equal number of inputs from each class



# Unsupervised Generative Classification

- In generative classification, we estimate
  - Class marginal distribution $p(y) = \textbf{multinoulli}(y|\boldsymbol{\pi})$
  - $K$ class-conditional distributions $p(x|\theta_k), k = 1,2,\ldots,K$




- Can we estimate $\{\pi, \theta_1, \theta_2, \ldots, \theta_K\}$ if the training labels are not known?

- It then becomes an unsupervised learning problem
  - Mixture modeling or clustering

- We will look at it later but the general idea is based on ALT-OPT
  1. Guess the label of each input given current estimate of $\{\pi, \theta_1, \theta_2, \ldots, \theta_K\}$
  2. Re-estimate $\{\pi, \theta_1, \theta_2, \ldots, \theta_K\}$ given the label guesses
  3. Alternate between steps 1 and 2 till convergence

# Generative Models for Regression

- Yes, we can even model regression problems using a generative approach
- Note that the output y is not longer discrete (so no notion of a class-conditional)
- However, the basic rule of recovering a conditional from joint would still apply

$$p(y|\boldsymbol{x},\theta) = \frac{p(\boldsymbol{x},y|\theta)}{p(\boldsymbol{x}|\theta)}$$

- Thus we can model the joint distribution $p(\boldsymbol{x},y|\theta)$ of features $\boldsymbol{x}$ and outputs $y \in \mathbb{R}$
  - If features are real-valued the we can model $p(\boldsymbol{x},y|\theta)$ using a $(D+1)$-dim Gaussian
  - From this $(D+1)$-dim Gaussian, we can get $p(y|\boldsymbol{x},\theta)$ using Gaussian conditioning formula
  - If joint is Gaussian, any subset of variables ($y$ here), given the rest ($\boldsymbol{x}$ here) is also a Gaussian!
  - Refer to the Gaussian results from maths refresher slides for the result

# Discriminative vs Generative

Proponents of discriminative models: Why bother modeling $\boldsymbol{x}$ if $\boldsymbol{y}$ is what you care about? Just model $\boldsymbol{y}$ directly instead of working hard to model $\boldsymbol{x}$ by learning the class-conditional

- Recall that discriminative approaches model $\boldsymbol{p(y|x)}$ directly
- Generative approaches model $\boldsymbol{p(y|x)}$ via $\boldsymbol{p(x,y)}$
- Number of parameters: Discriminative models have fewer parameters to be learned
  - Just the weight vector/matrix $\boldsymbol{w/W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier
  - For "simple" class-conditionals, easier for gen. classifn model (often closed-form solution)
  - Parameter estimation for discriminative models (logistic/softmax) usually requires iterative methods (although objective functions usually have global optima)
- Dealing with missing features: Generative models can handle this easily
  - E.g., by integrating out the missing features while estimating the parameters (will see later)
- Inputs with features having mixed types: Generative model can handle this
  - Appropriate $\boldsymbol{p(x_d|y)}$ for each type of feature in the input. Difficult for discriminative models

# Discriminative vs Generative (Contd)

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and are ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained on all classes all over again. Generative model will just require estimating the class-cond of newly added classes
- Have lots of labeled training data? Discriminative models usually work very well
- Final Verdict? Despite generative classification having some clear advantages, both methods can be quite powerful (the actual choice may be dictated by the problem)
  - Important to be aware of their strengths/weaknesses, and also the connections between these
- Possibility of a Hybrid Design? Yes, Generative and Disc. models can be combined, e.g.,
  - "Principled Hybrids of Generative and Discriminative Models" (Lassere et al, 2006)
  - "Deep Hybrid Models: Bridging Discriminative & Generative Approaches" (Kuleshov & Ermon, 2017)