

# ITCS 6156/8156 Fall 2024

## Machine Learning

# Generative Models for Classification

Instructor: Hongfei Xue

Email: [hongfei.xue@charlotte.edu](mailto:hongfei.xue@charlotte.edu)

Class Meeting: Tue & Thu, 4:00 PM – 5:15 PM, WWH 130



Some content in the slides is based on Dr. Raquel Urtasun's lecture

# Classification

- Given inputs  $x$  and classes  $y$  we can do classification in several ways. How?



(features)

$x$

e.g:

- height
- weight
- color

(class label)

$y$

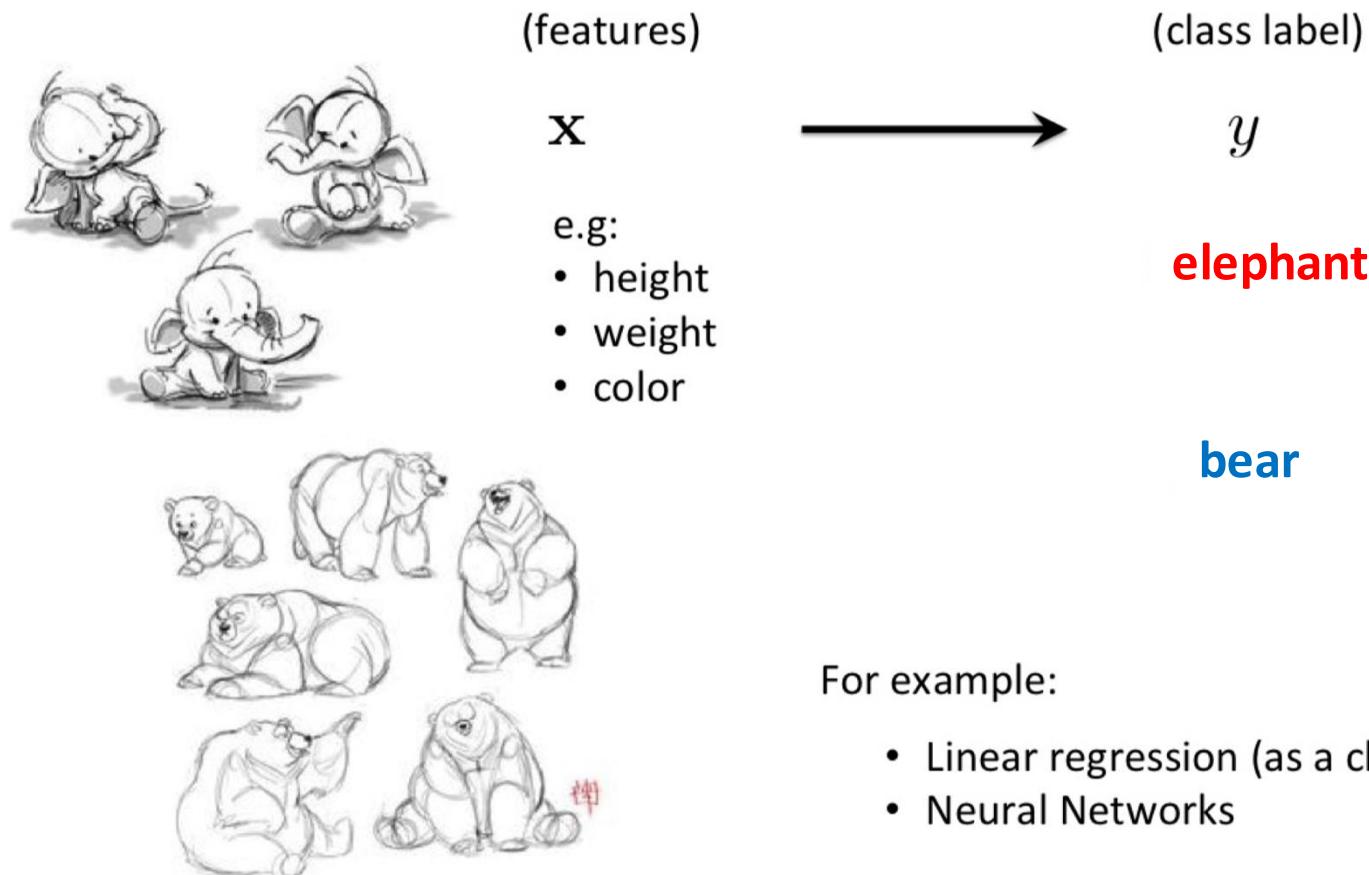
elephant



bear

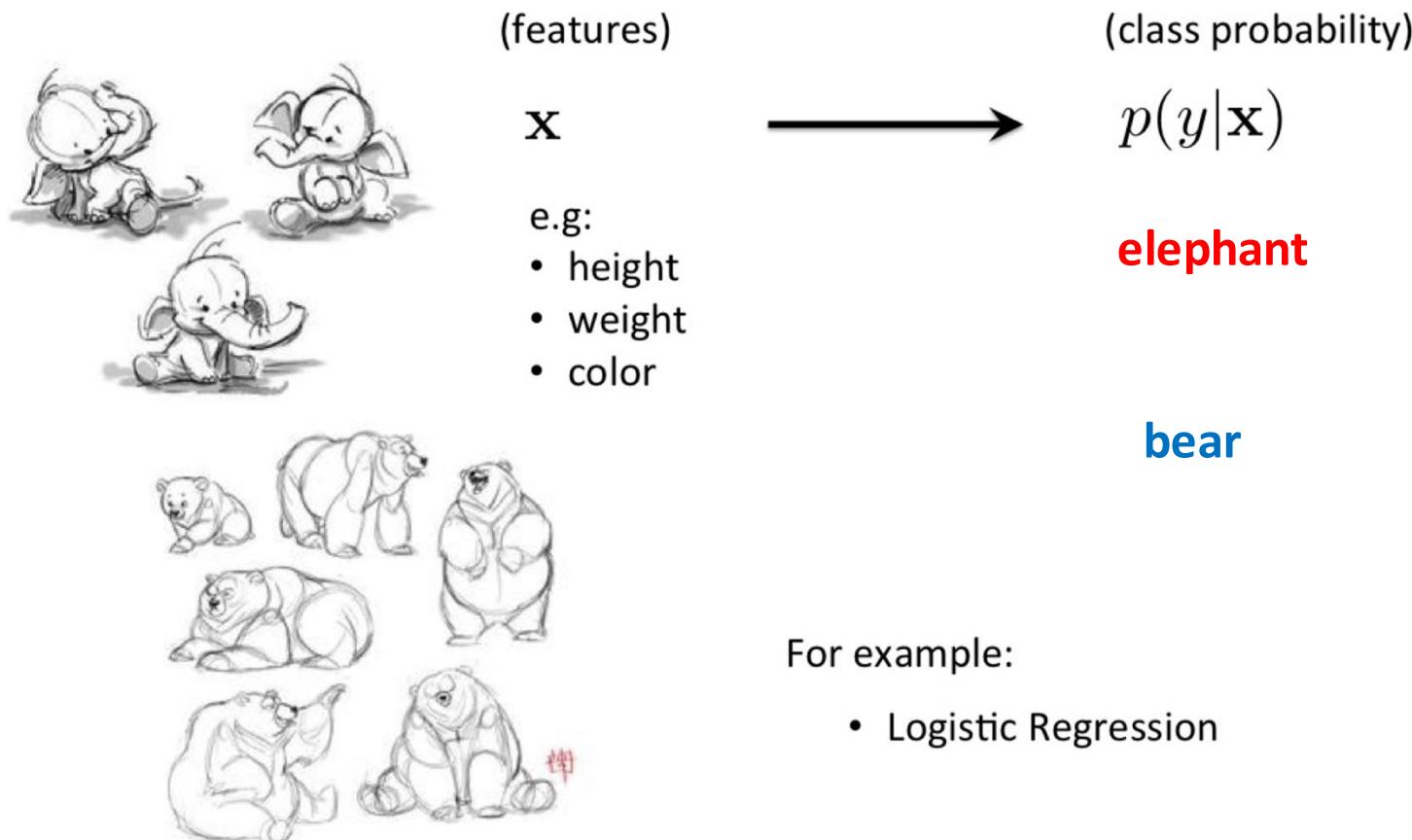
# Discriminative Classifiers

- **Discriminative** classifiers try to either:
  - ▶ learn mappings directly from the space of inputs  $\mathcal{X}$  to class labels  $\{0, 1, 2, \dots, K\}$



# Discriminative Classifiers

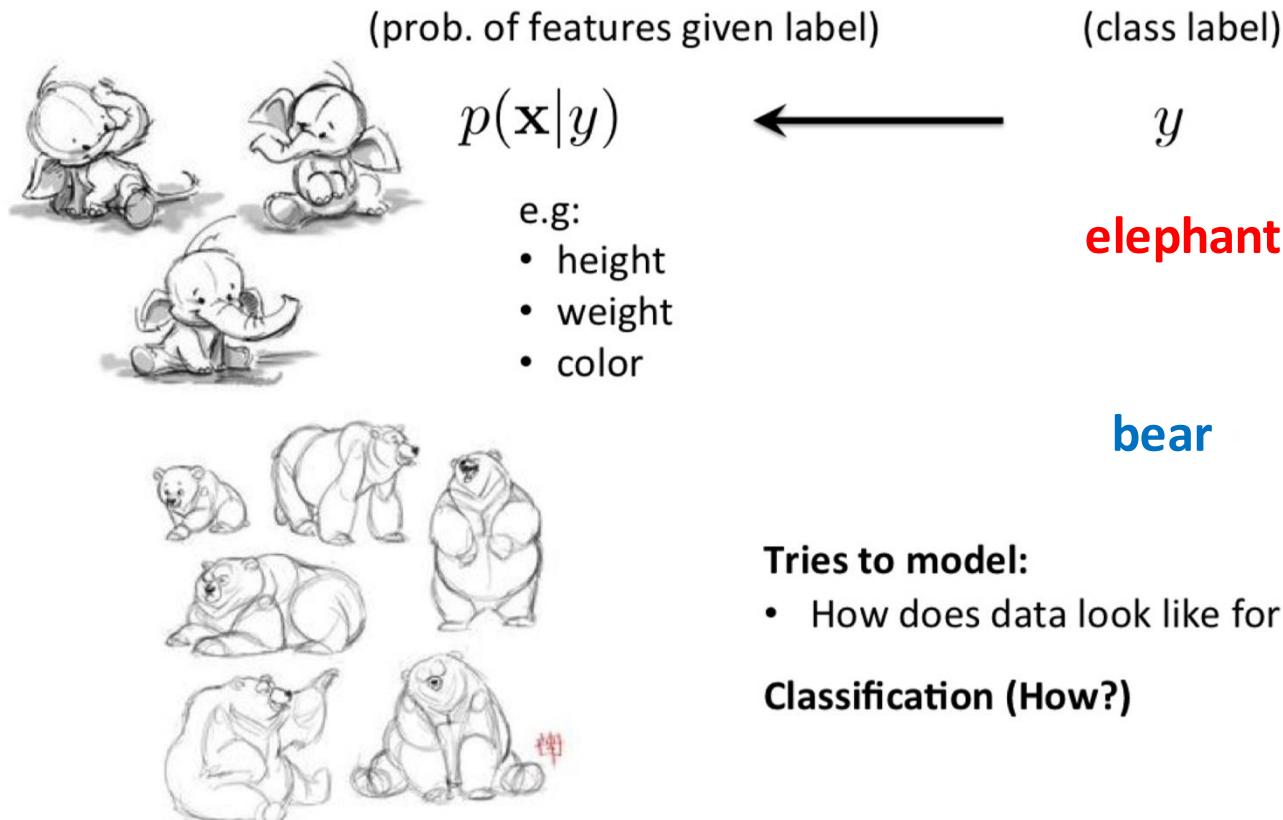
- **Discriminative** classifiers try to either:
  - ▶ or try to learn  $p(y|\mathbf{x})$  directly



# Generative Classifiers

How about this approach: build a model of “how data for a class looks like”

- **Generative** classifiers try to model  $p(\mathbf{x}|y)$
- Classification via Bayes rule (thus also called Bayes classifiers)



# Generative vs Discriminative

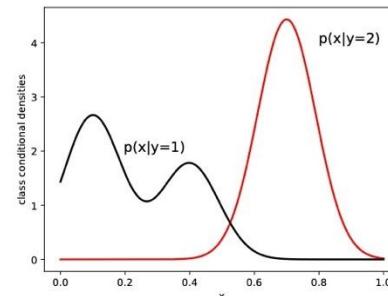
Two approaches to classification:

- **Discriminative** classifiers estimate parameters of decision boundary/class separator directly from labeled examples
  - ▶ learn  $p(y|x)$  directly (logistic regression models)
  - ▶ learn mappings from inputs to classes (least-squares, neural nets)
- **Generative approach:** model the distribution of inputs characteristic of the class (Bayes classifier)
  - ▶ Build a model of  $p(x|y)$
  - ▶ Apply Bayes Rule

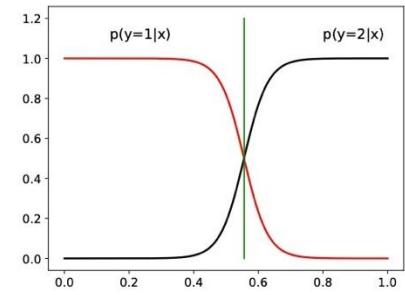
# Generative vs Discriminative

- **Discriminative Classifiers:**

- Often have better accuracy (Probability estimates for posteriors  $P(Y | X)$  often more accurate. Generative models make often unrealistic assumptions on the  $P(x|y)$ )
- Can handle feature preprocessing (E.g. via nonlinear feature maps  $\phi : R^d \rightarrow R^D$  with  $D >> d$  (explicitly) or kernel trick (implicitly). Modelling distributions of preprocessed features too hard in generative models.)



(a)



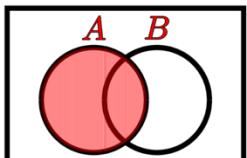
(b)

- **Generative Classifiers:**

- Easy (not expensive) to fit. ("Training" often amounts to averaging / counting things. No optimization algorithms necessary.)
- Can handle missing feature data more easily.
- Builds model for different classes separately. (Good if new classes are added to the problem.)
- Handle unlabeled training data better.

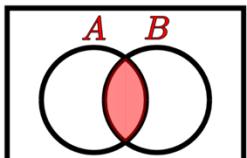
Figure 9.8: The class-conditional densities  $p(x|y = c)$  (left) may be more complex than the class posteriors  $p(y = c|x)$  (right). Adapted from Figure 1.27 of [Bis06]. Generated by [code.probml.ai/book1/9.8](http://code.probml.ai/book1/9.8).

# Probability Formulas



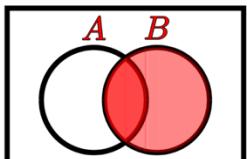
$$P(A)$$

Intersection

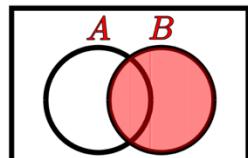


$$P(A \cap B)$$

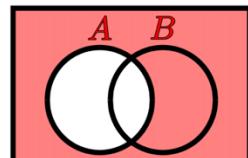
Conditional



$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

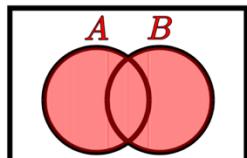


$$P(B)$$

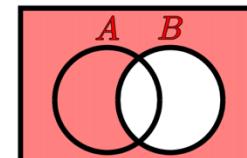


$$P(A)^c$$

Union

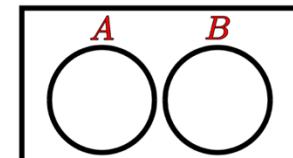


$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$



$$P(B)^c$$

Mutually Exclusive

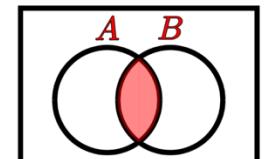


$$P(A \cap B) = 0$$

Bayes' Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$



$$P(A \cap B) = P(A) \cdot P(B)$$

$$P(A|B) = P(A)$$

# Bayes Classifier

- Aim to diagnose whether patient has diabetes: classify into one of two classes (yes  $C=1$ ; no  $C=0$ )
- Run battery of tests on the patients, get  $\mathbf{x}$  for each patient
- Given patient's results:  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$  we want to compute class probabilities using Bayes Rule:

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)p(C)}{p(\mathbf{x})}$$

- More formally

$$\text{posterior} = \frac{\text{Class likelihood} \times \text{prior}}{\text{Evidence}}$$

- How can we compute  $p(\mathbf{x})$  for the two class case?

$$p(\mathbf{x}) = p(\mathbf{x}|C=0)p(C=0) + p(\mathbf{x}|C=1)p(C=1)$$

- To compute  $p(C|\mathbf{x})$  we need:  $p(\mathbf{x}|C)$  and  $p(C)$

# A Discrete Example

- For a genetic defect test we have the following information:
  - 1% of people have a certain genetic defect;
  - For people who have genetic defects, 90% of tests show positive results;
  - For people who have no genetic defects, 90% tests show negative results.
- If a person gets a positive test result, what is the probability he/she actually have the genetic defect?
- Solution:

$$\begin{aligned}P(G = 1|T = 1) &= \frac{P(T = 1|G = 1)P(G = 1)}{P(T = 1)} \\&= \frac{P(T = 1|G = 1)P(G = 1)}{P(T = 1|G = 1)P(G = 1) + P(T = 1|G = 0)P(G = 0)} \\&= \frac{0.9 * 0.01}{0.9 * 0.01 + 0.1 * 0.99} = 8.33\%\end{aligned}$$

# Classification: Diabetes Example

- Let's start with the simplest case where the input is only 1-dimensional, for example: white blood cell count (this is our  $x$ )
- We need to choose a probability distribution  $p(x|C)$  that makes sense

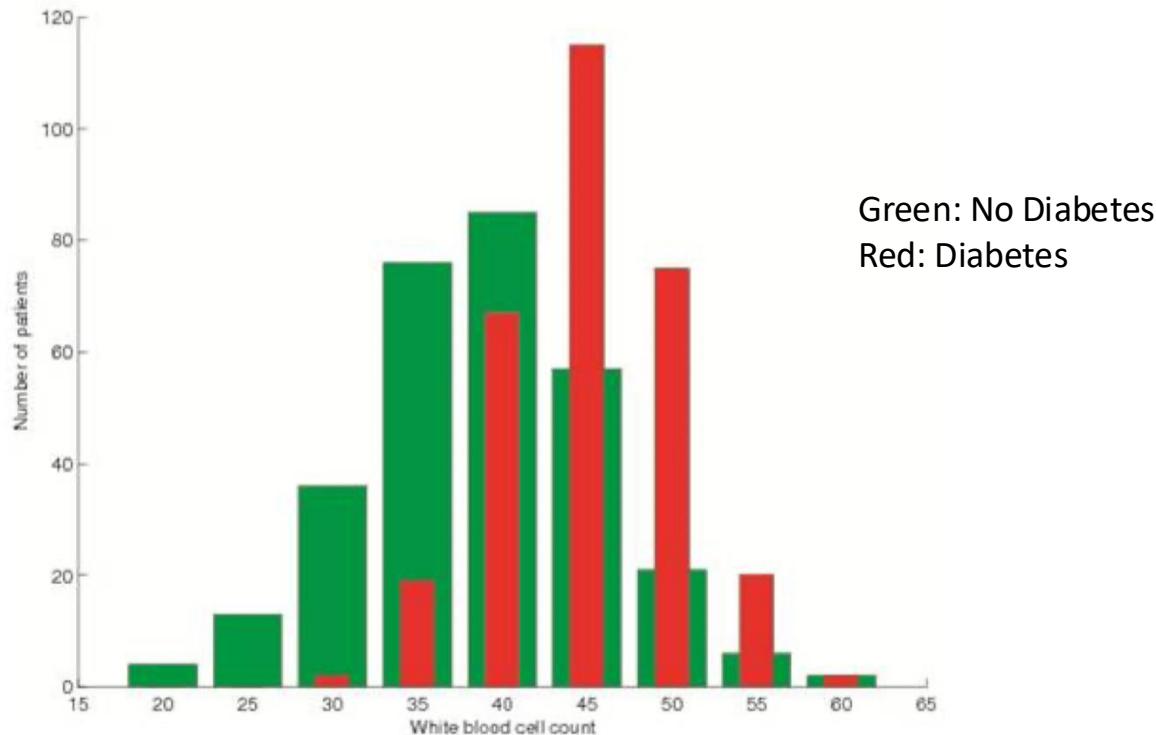


Figure : Our example (showing counts of patients for input value): What distribution to choose?

# Gaussian Bayes Classifier

- Our first generative classifier assumes that  $p(\mathbf{x}|y)$  is distributed according to a multivariate normal (Gaussian) distribution
- This classifier is called Gaussian Discriminant Analysis
- Let's first continue our simple case when inputs are just 1-dim and have a Gaussian distribution:

$$p(x|C) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu_C)^2}{2\sigma_C^2}\right)$$

with  $\mu \in \Re$  and  $\sigma^2 \in \Re^+$

- Notice that we have different parameters for different classes
- How can I fit a Gaussian distribution to my data?

# MLE for Gaussians

- Let's assume that the class-conditional densities are Gaussian

$$p(x|C) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu_C)^2}{2\sigma_C^2}\right)$$

with  $\mu \in \Re$  and  $\sigma^2 \in \Re^+$

- How can I fit a Gaussian distribution to my data?
- We are given a set of training examples  $\{x^{(n)}, t^{(n)}\}_{n=1, \dots, N}$  with  $t^{(n)} \in \{0, 1\}$  and we want to estimate the model parameters  $\{(\mu_0, \sigma_0), (\mu_1, \sigma_1)\}$
- First divide the training examples into two classes according to  $t^{(n)}$ , and for each class take all the examples and fit a Gaussian to model  $p(x|C)$
- Let's try **maximum likelihood estimation** (MLE)

# MLE for Gaussians

(note: we are dropping subscript  $C$  for simplicity of notation)

- We assume that the data points that we have are **independent** and **identically distributed**

$$p(x^{(1)}, \dots, x^{(N)} | C) = \prod_{n=1}^N p(x^{(n)} | C) = \prod_{n=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(n)} - \mu)^2}{2\sigma^2}\right)$$

- Now we want to maximize the likelihood, or minimize its negative (if you think in terms of a loss)

$$\begin{aligned}\ell_{log-loss} &= -\ln p(x^{(1)}, \dots, x^{(N)} | C) = -\ln \left( \prod_{n=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x^{(n)} - \mu)^2}{2\sigma^2}\right) \right) \\ &= \sum_{n=1}^N \ln(\sqrt{2\pi}\sigma) + \sum_{n=1}^N \frac{(x^{(n)} - \mu)^2}{2\sigma^2} = \frac{N}{2} \ln(2\pi\sigma^2) + \sum_{n=1}^N \frac{(x^{(n)} - \mu)^2}{2\sigma^2}\end{aligned}$$

# Computing the Mean

- (let's try to find a) Closed-form solution: Write  $\frac{d\ell_{\text{log-loss}}}{d\mu}$  and  $\frac{d\ell_{\text{log-loss}}}{d\sigma^2}$  and equal it to 0 to find the parameters  $\mu$  and  $\sigma^2$

$$\begin{aligned}\frac{\partial \ell_{\text{log-loss}}}{\partial \mu} &= \frac{\partial \left( \frac{N}{2} \ln (2\pi\sigma^2) + \sum_{n=1}^N \frac{(x^{(n)} - \mu)^2}{2\sigma^2} \right)}{\partial \mu} = \frac{d \left( \sum_{n=1}^N \frac{(x^{(n)} - \mu)^2}{2\sigma^2} \right)}{d\mu} \\ &= \frac{-\sum_{n=1}^N 2(x^{(n)} - \mu)}{2\sigma^2} = -\sum_{n=1}^N \frac{(x^{(n)} - \mu)}{\sigma^2} = \frac{N\mu - \sum_{n=1}^N x^{(n)}}{\sigma^2}\end{aligned}$$

- And equating to zero we have

$$\frac{d\ell_{\text{log-loss}}}{d\mu} = 0 = \frac{N\mu - \sum_{n=1}^N x^{(n)}}{\sigma^2}$$

Thus

$$\boxed{\mu = \frac{1}{N} \sum_{n=1}^N x^{(n)}}$$

# Computing the Variance

- And for  $\sigma^2$ :

$$\begin{aligned}\frac{d\ell_{\text{log-loss}}}{d\sigma^2} &= \frac{d \left( \frac{N}{2} \ln (2\pi\sigma^2) + \sum_{n=1}^N \frac{(x^{(n)} - \mu)^2}{2\sigma^2} \right)}{d\sigma^2} \\ &= \frac{N}{2} \frac{1}{2\pi\sigma^2} 2\pi + \frac{\sum_{n=1}^N (x^{(n)} - \mu)^2}{2} \left( \frac{-1}{\sigma^4} \right) \\ &= \frac{N}{2\sigma^2} - \frac{\sum_{n=1}^N (x^{(n)} - \mu)^2}{2\sigma^4}\end{aligned}$$

- And equating to zero we have

$$\frac{d\ell_{\text{log-loss}}}{d\sigma^2} = 0 = \frac{N}{2\sigma^2} - \frac{\sum_{n=1}^N (x^{(n)} - \mu)^2}{2\sigma^4} = \frac{N\sigma^2 - \sum_{n=1}^N (x^{(n)} - \mu)^2}{2\sigma^4}$$

- Thus:

$$\boxed{\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x^{(n)} - \mu)^2}$$

# MLE of a Gaussian

- In summary, we can compute the parameters of a Gaussian distribution in closed form for each class by taking the training points that belong to that class

**MLE estimates of parameters for a Gaussian distribution:**

$$\mu = \frac{1}{N} \sum_{n=1}^N x^{(n)}$$

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x^{(n)} - \mu)^2$$

# Posterior Probability

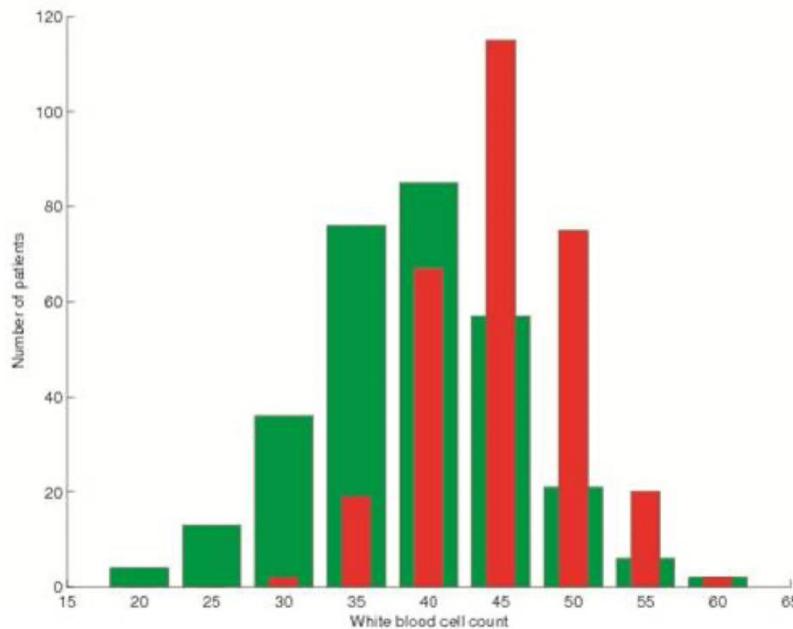
- We now have  $p(x|C)$
- In order to compute the **posterior probability**:

$$\begin{aligned} p(C|x) &= \frac{p(x|C)p(C)}{p(x)} \\ &= \frac{p(x|C)p(C)}{p(x|C=0)p(C=0) + p(x|C=1)p(C=1)} \end{aligned}$$

given a new observation, we still need to compute the **prior**

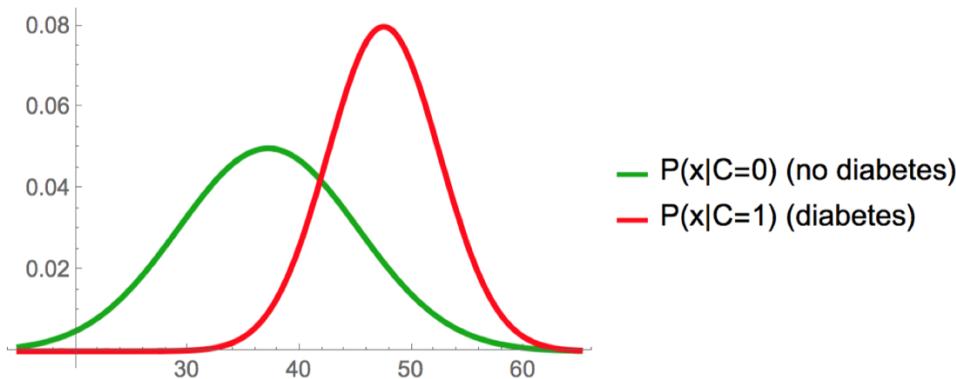
- **Prior:** In the absence of any observation, what do I know about the problem?

# Diabetes Example



- Doctor has a prior  $p(C = 0) = 0.8$ , how?
- A new patient comes in, the doctor measures  $x = 48$
- Does the patient have diabetes?

# Diabetes Example



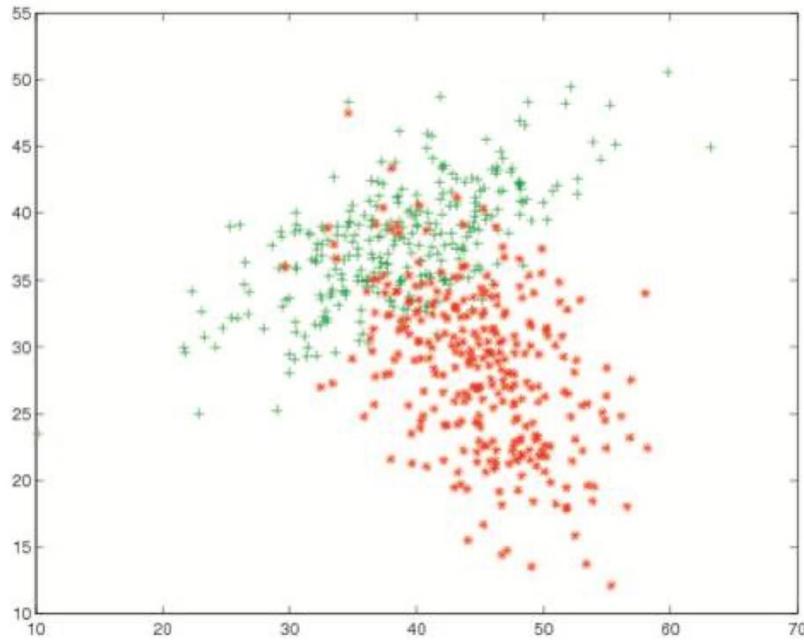
- Compute  $p(x = 48|C = 0)$  and  $p(x = 48|C = 1)$  via our estimated Gaussian distributions
- Compute posterior  $p(C = 0|x = 48)$  via Bayes rule using the prior (how can we get  $p(C = 1|x = 48)$ ?)
- How can we decide on diabetes/non-diabetes?

# Bayes Classifier

- Use Bayes classifier to classify new patients (unseen test examples)
- Simple Bayes classifier: estimate posterior probability of each class
- What should the decision criterion be?
- The optimal decision is the one that minimizes the expected number of mistakes

# Multiple-dimensional Inputs

- Add second observation: Plasma glucose value
- Now our input  $\mathbf{x}$  is 2-dimensional



# Gaussian Bayes Classifier

- Gaussian Discriminant Analysis in its general form assumes that  $p(\mathbf{x}|t)$  is distributed according to a multivariate normal (Gaussian) distribution
- Multivariate Gaussian distribution:

$$p(\mathbf{x}|t = k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp [-(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)]$$

where  $|\Sigma_k|$  denotes the determinant of the matrix, and  $d$  is dimension of  $\mathbf{x}$

- Each class  $k$  has associated mean vector  $\boldsymbol{\mu}_k$  and covariance matrix  $\boldsymbol{\Sigma}_k$
- Typically the classes share a single covariance matrix  $\boldsymbol{\Sigma}$  (“share” means that they have the same parameters; the covariance matrix in this case):  
$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_k$$

# Multivariate Data

- Multiple measurements (sensors)
- $d$  inputs/features/attributes
- $N$  instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

Suppose  $X$  has mean  $\mu_X$  and  $Y$  has mean  $\mu_Y$ .

- Covariance

$$\text{cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[XY] - \mu_X\mu_Y$$

Maximized when  $X = Y$ , in which case it is  $\text{var}(X)$ .

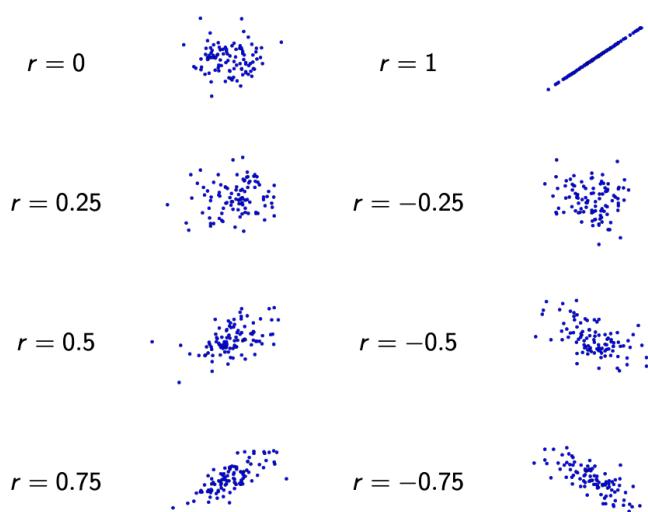
In general, it is at most  $\text{std}(X)\text{std}(Y)$ .

- Correlation

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\text{std}(X)\text{std}(Y)}$$

This is always in the range  $[-1, 1]$ .

## Correlation pictures



# Multivariate Parameters

- Mean

$$\mathbb{E}[\mathbf{x}] = [\mu_1, \dots, \mu_d]^T$$

- Covariance

$$\Sigma = \text{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mu)^T (\mathbf{x} - \mu)] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

- Correlation =  $\text{Corr}(\mathbf{x})$  is the covariance divided by the product of standard deviation

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

A distribution over  $(x_1, x_2) \in \mathbb{R}^2$ , parametrized by:

- **Mean**  $(\mu_1, \mu_2) \in \mathbb{R}^2$ , where  $\mu_1 = \mathbb{E}(X_1)$  and  $\mu_2 = \mathbb{E}(X_2)$

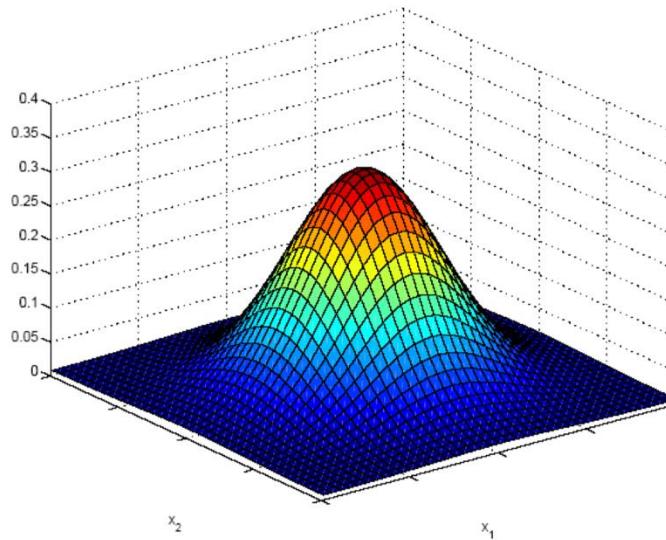
- **Covariance matrix**  $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$  where

$$\left\{ \begin{array}{l} \Sigma_{11} = \text{var}(X_1) \\ \Sigma_{22} = \text{var}(X_2) \\ \Sigma_{12} = \Sigma_{21} = \text{cov}(X_1, X_2) \end{array} \right\}$$

# Multivariate Gaussian Distribution

- $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ , a Gaussian (or normal) distribution defined as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp [-(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)]$$



- Mahalanobis distance  $(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k)$  measures the distance from  $\mathbf{x}$  to  $\mu$  in terms of  $\Sigma$
- It normalizes for difference in variances and correlations

# Simplifying the Model

What if  $\mathbf{x}$  is high-dimensional?

- For Gaussian Bayes Classifier, if input  $\mathbf{x}$  is high-dimensional, then covariance matrix has many parameters
- Save some parameters by using a shared covariance for the classes
- Any other idea you can think of?

# Naive Bayes

- Given patient's results:  $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$  we want to update class probabilities using Bayes Rule:

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)p(C)}{p(\mathbf{x})}$$

- More formally

$$\text{posterior} = \frac{\text{Class likelihood} \times \text{prior}}{\text{Evidence}}$$

- Naive Bayes** is an alternative generative model: Assumes features independent given the class

$$p(\mathbf{x}|t = k) = \prod_{i=1}^d p(x_i|t = k)$$

# Naive Bayes Classifier

Given

- prior  $p(t = k)$
- assuming features are conditionally independent given the class
- likelihood  $p(x_i|t = k)$  for each  $x_i$

The decision rule

$$y = \arg \max_k p(t = k) \prod_{i=1}^d p(x_i|t = k)$$

- If the assumption of conditional independence holds, NB is the optimal classifier
- If not, a heavily regularized version of generative classifier
- Note: NB's assumptions (cond. independence) typically do not hold in practice. However, the resulting algorithm still works well on many problems, and it typically serves as a decent baseline for more sophisticated models

# A Discrete Example

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x (1)	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
x (2)	S	M	M	S	S	S	M	M	L	L	L	M	M	L	L
y	-1	-1	1	1	-1	-1	1	1	1	1	1	1	1	1	-1

$x(1)$ : Humidity (level 1, 2, 3);

$x(2)$ : Wind intensity (Low, Mediate, Strong);

$y$ : Rain (1) or not (-1).

- Will it rain when humidity is at level 2 and wind intensity is strong?

$$\begin{aligned}
 & P(y = 1|H = 2, W = S) \\
 &= \frac{P(y = 1)P(H = 2, W = S|y = 1)}{P(H = 2, W = s)} \\
 &= \frac{P(y = 1)P(H = 2|y = 1)P(W = S|y = 1)}{P(H = 2, W = s)} \\
 &= \frac{\frac{10}{15} * \frac{4}{10} * \frac{1}{10}}{P(H = 2, W = s)} = \frac{\frac{2}{75}}{P(H = 2, W = s)}
 \end{aligned}$$

$$\begin{aligned}
 & P(y = 0|H = 2, W = S) \\
 &= \frac{P(y = 0)P(H = 2, W = S|y = 0)}{P(H = 2, W = s)} \\
 &= \frac{P(y = 0)P(H = 2|y = 0)P(W = S|y = 0)}{P(H = 2, W = s)} \\
 &= \frac{\frac{5}{15} * \frac{1}{5} * \frac{3}{5}}{P(H = 2, W = s)} = \frac{\frac{3}{75}}{P(H = 2, W = s)}
 \end{aligned}$$

- It will probably not rain.

# Questions?