GENERATIVE LEARNING ALGORITHMS

Machine Learning HS18

DISCRIMINATIVE LEARNING ALGORITHMS

So far we discussed algorithms that directly try to predict p(y|x). For example logistic regression models $p(y|x;\theta)$ as :

$$h_\theta(x) = g(\theta^T x)$$

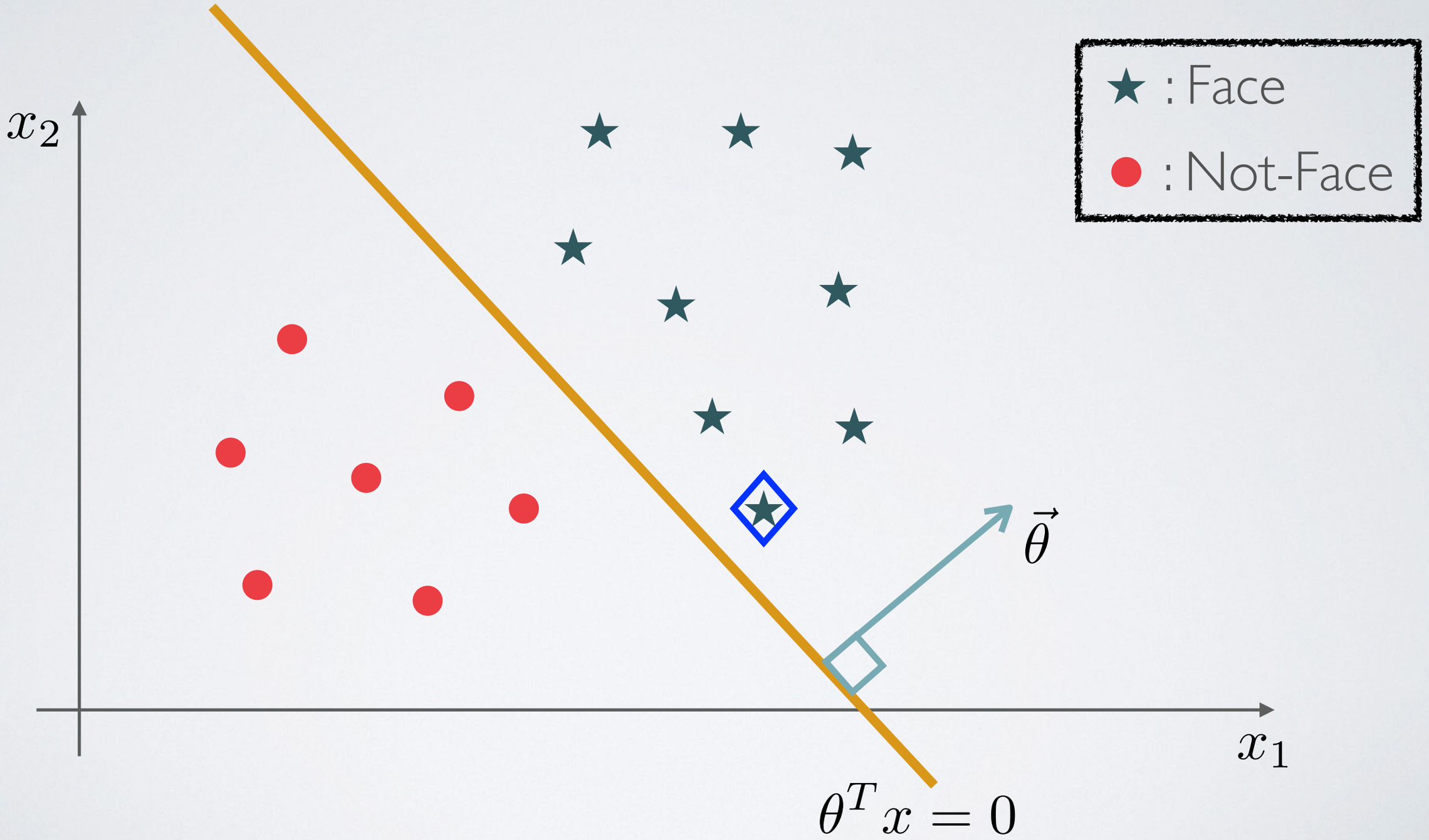
$$h_\theta(x) = p(y = 1|x;\theta)$$

Decision Function:

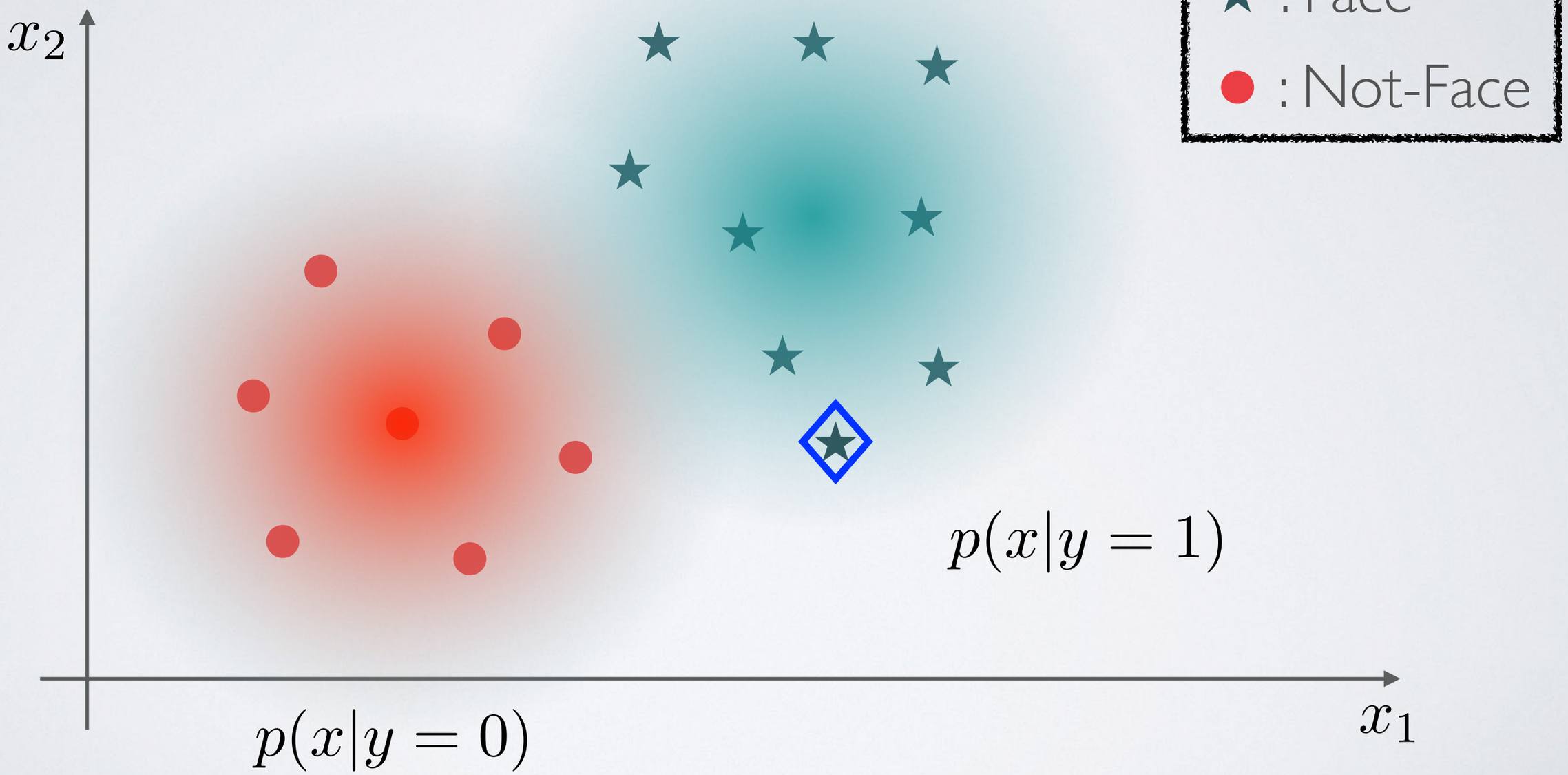
$$\theta^T x \geq 0 \quad \rightarrow \quad y = 1$$

$$\theta^T x < 0 \quad \rightarrow \quad y = 0$$

DISCRIMINATIVE VS. GENERATIVE LEARNING



DISCRIMINATIVE VS. **GENERATIVE** LEARNING



DISCRIMINATIVE VS. GENERATIVE LEARNING

Discriminative Learning:

Directly model $p(y|x)$

Generative Learning:

Model $p(x|y)$ and $p(y)$

Example:

$p(x|y=0)$: model of dog features

$p(x|y=1)$: model of cat features

$p(y)$: class prior

FROM P(X|Y) AND P(Y) TO P(Y|X)

Generative Learning:

If we have a model for $p(x|y)$ and $p(y)$ how do we get $p(y|x)$?

Bayes Rule:

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

$$\frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{p(x|y=0)p(y=0) + p(x|y=1)p(y=1)}$$

FROM P(X|Y) AND P(Y) TO P(Y|X)

Generative Learning:

If we have a model for $p(x|y)$ and $p(y)$ how do we get $p(y|x)$?

Making Predictions:

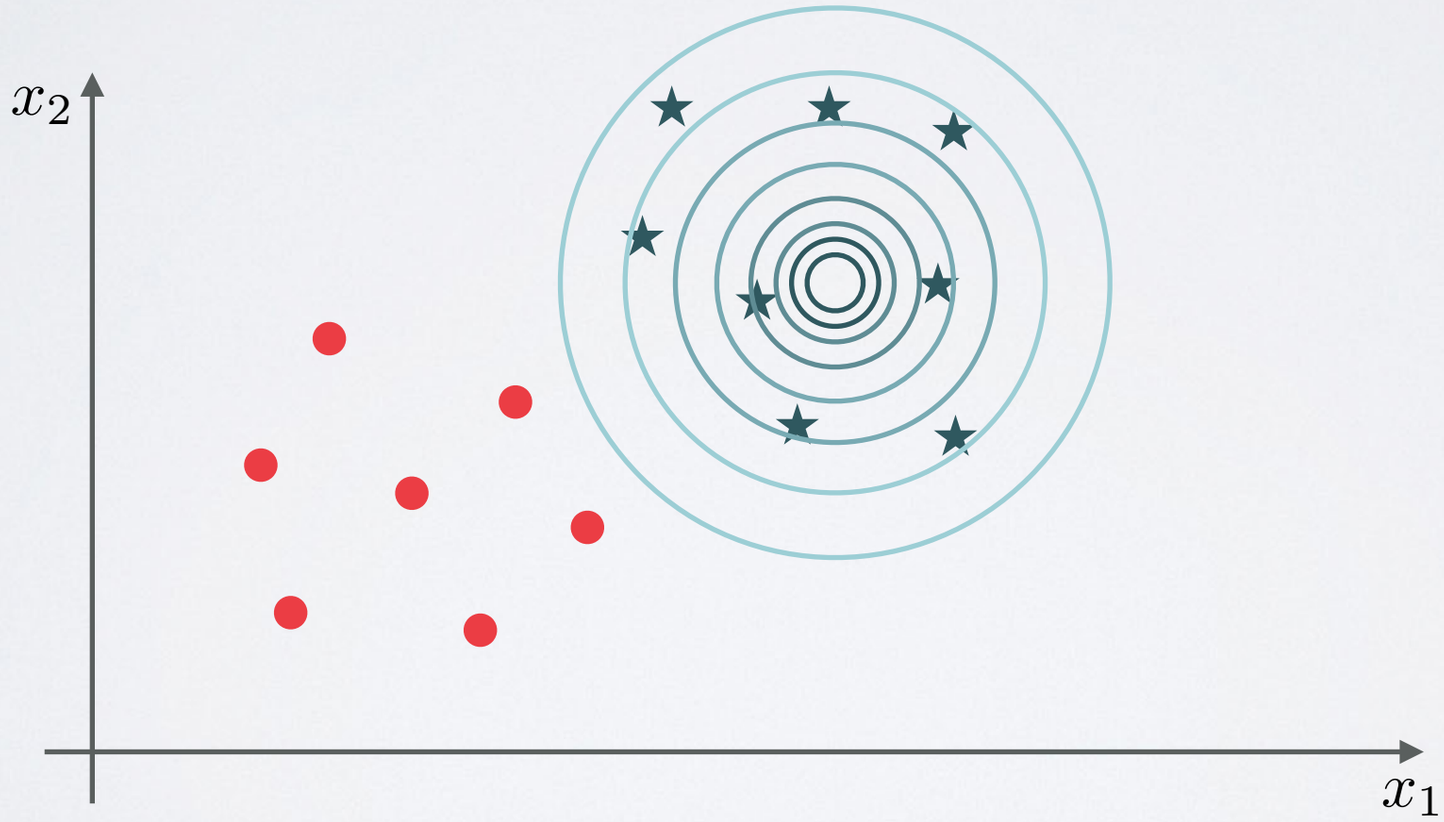
$$\arg\max_y p(y|x) = \arg\max_y \frac{p(x|y)p(y)}{p(x)}$$

$$= \arg\max_y p(x|y)p(y)$$

GAUSSIAN DISCRIMINANT ANALYSIS (GDA)

Key Idea:

Model $p(x|y)$ as **multivariate Gaussian!**



→ First review multivariate Gaussian distributions…

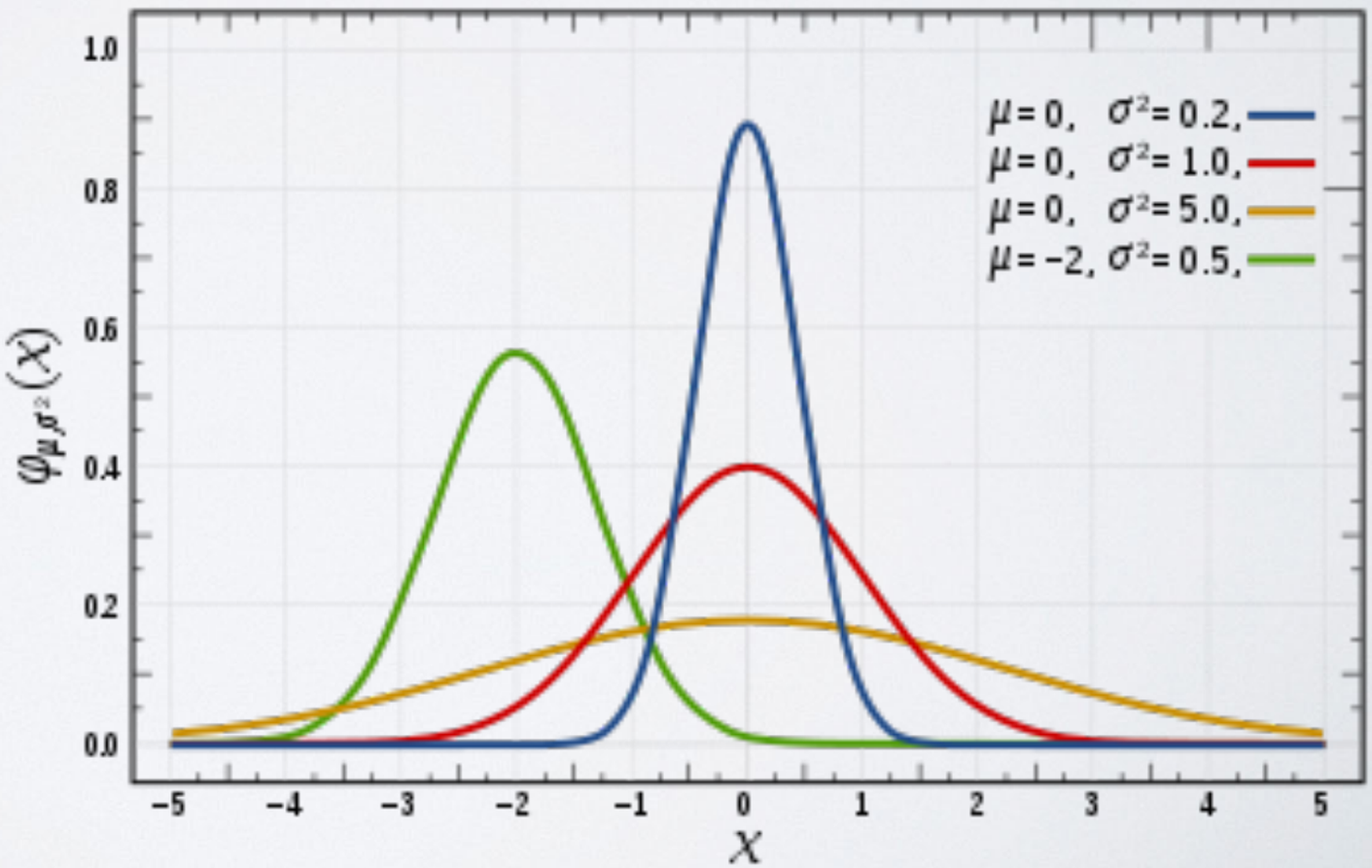
GAUSSIAN DISCRIMINANT ANALYSIS (GDA)

Reminder of univariate Normal distribution:

$X \sim \mathcal{N}\left(\mu, \sigma^2\right)$ pdf: $f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

mean: μ

variance: σ^2



MULTIVARIATE NORMAL DISTRIBUTION

Univariate Normal distribution:

$X \sim \mathcal{N}\left(\mu, \sigma^2\right)$ mean: μ variance: σ^2

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

Multivariate Normal distribution:

$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ mean: $\boldsymbol{\mu} \in \mathbb{R}^n$ covariance: $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$

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

MULTIVARIATE NORMAL DISTRIBUTION

Multivariate Normal distribution $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$**:**

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

Properties:

- $|\Sigma|$ is the determinant of Σ

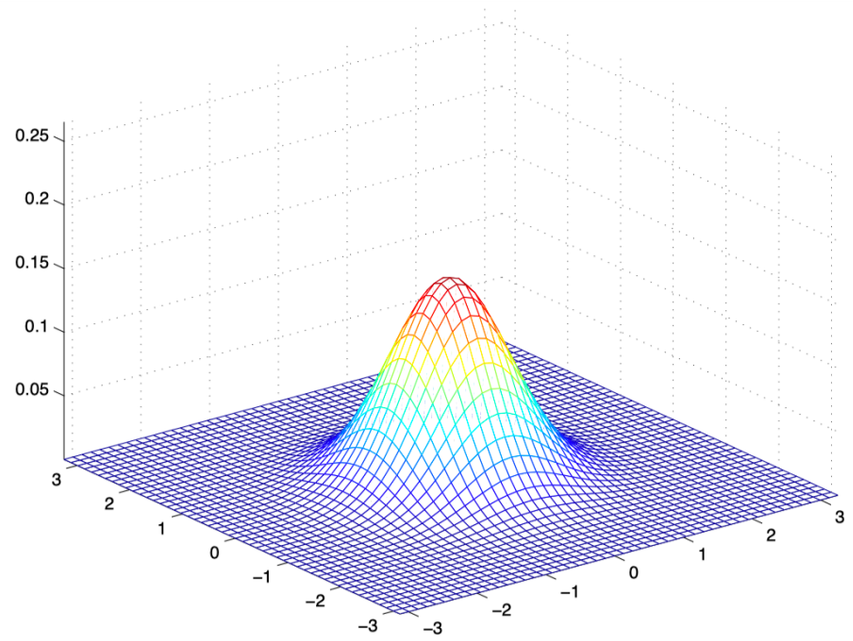
- $\Sigma \geq 0$ symmetric & positive semi-definite (so is Σ^{-1})

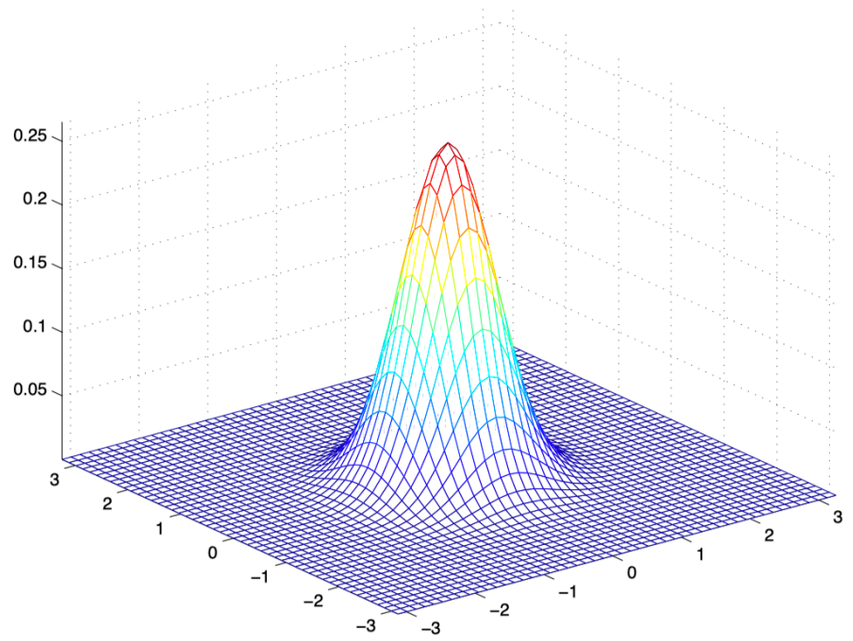
- $\mathrm{E}[X] = \int_x xp(x; \mu, \Sigma)dx = \mu$

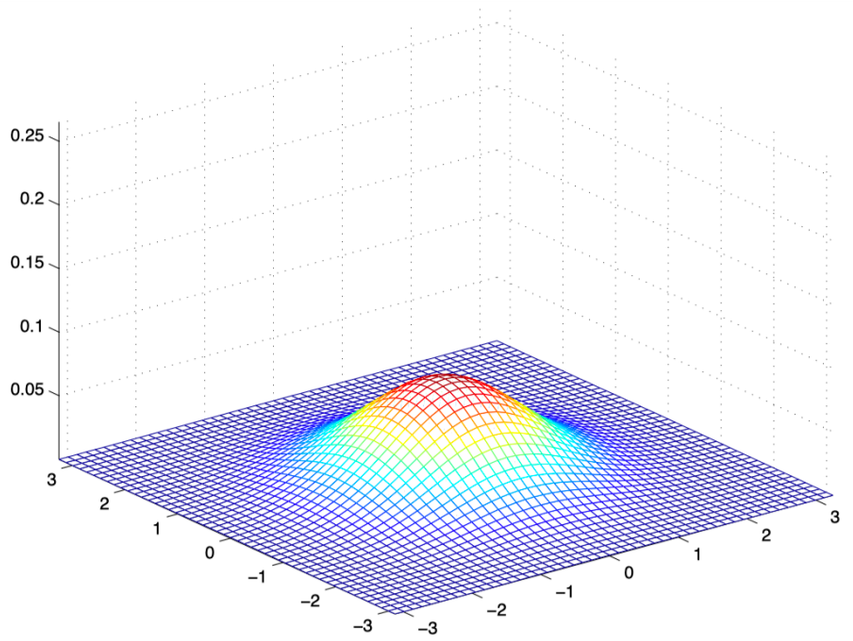
- $\mathrm{Cov}(X) = \mathrm{E}\left[(X - \mathrm{E}[X])(X - \mathrm{E}[X])^T\right] = \Sigma$

MULTIVARIATE NORMAL DISTRIBUTION

Examples:



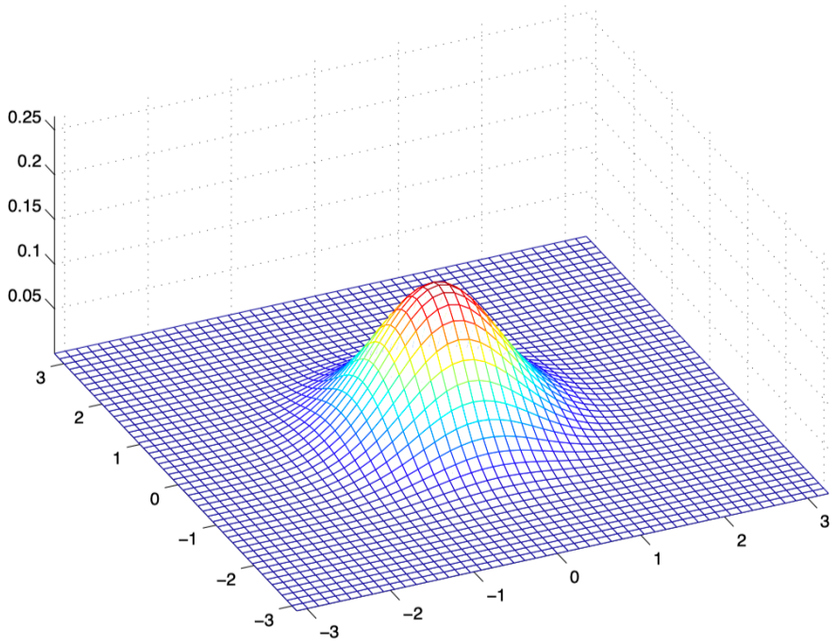


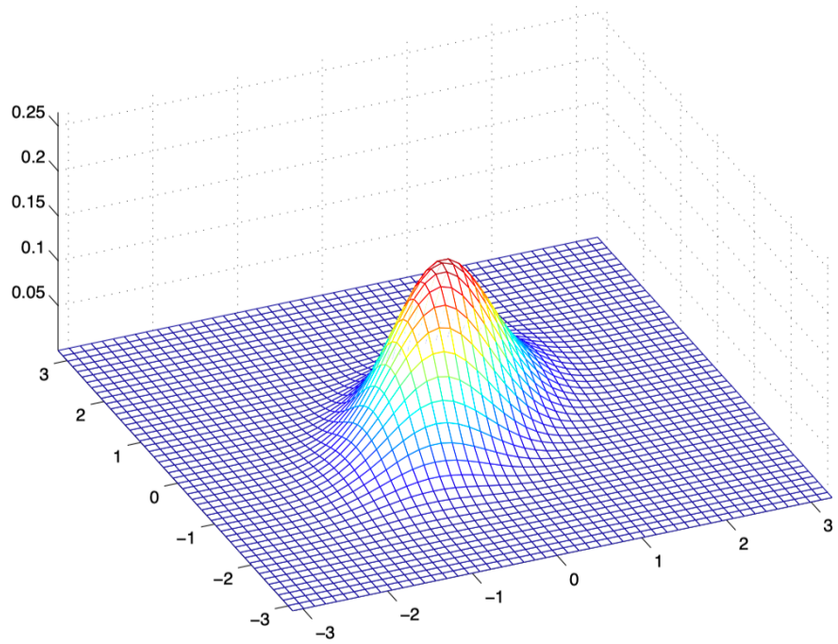


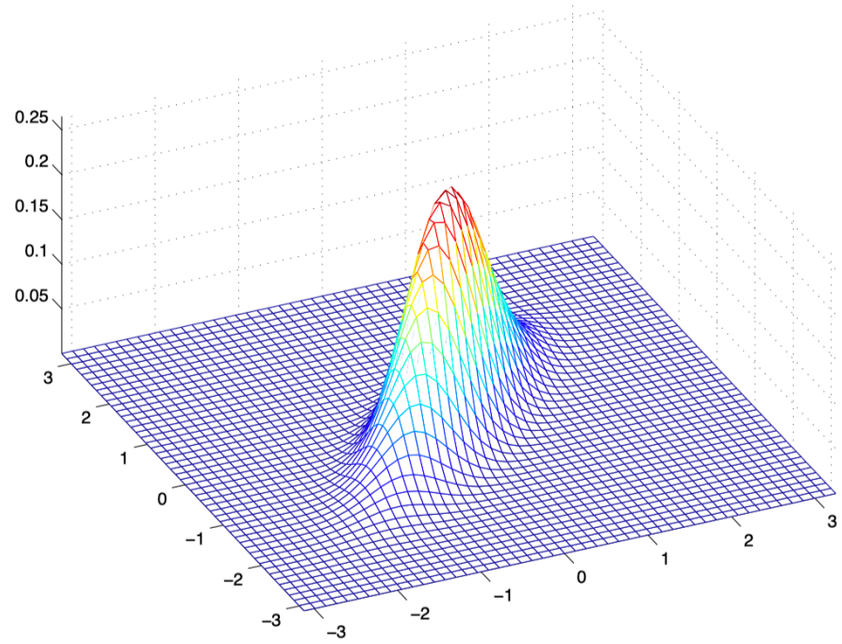
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

MULTIVARIATE NORMAL DISTRIBUTION

Examples:



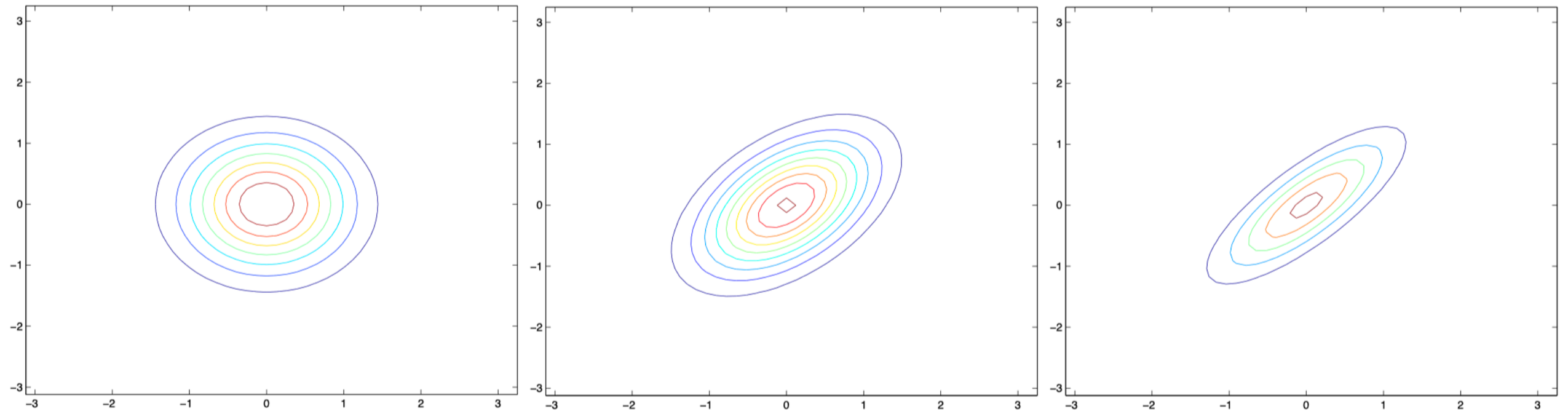




$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

MULTIVARIATE NORMAL DISTRIBUTION

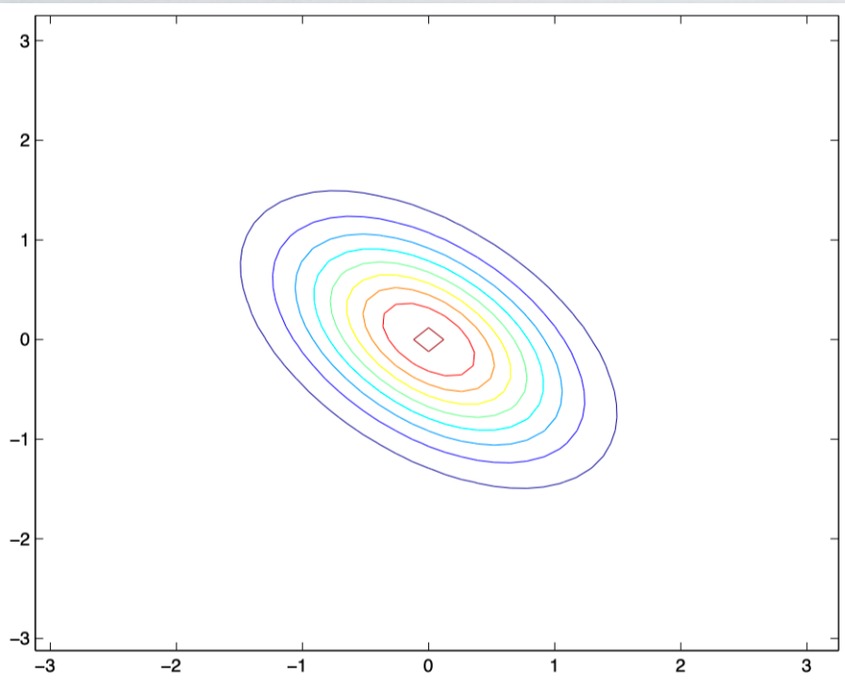
Examples:

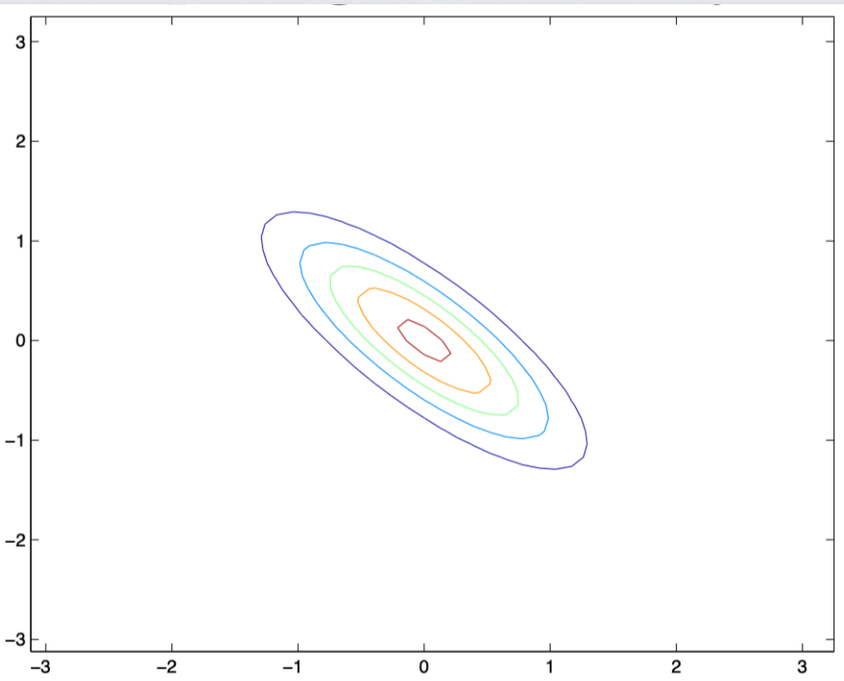


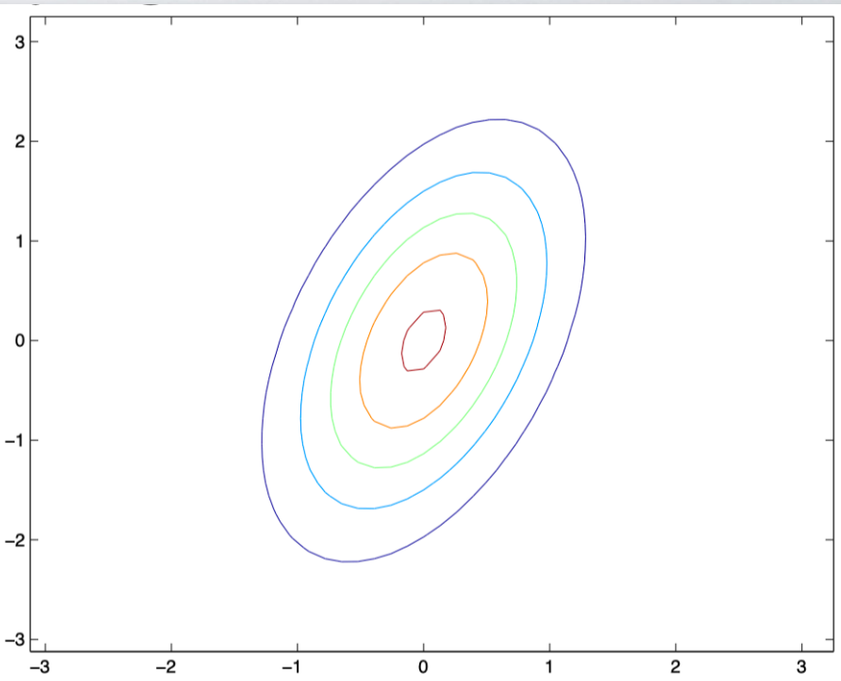
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

MULTIVARIATE NORMAL DISTRIBUTION

Examples:



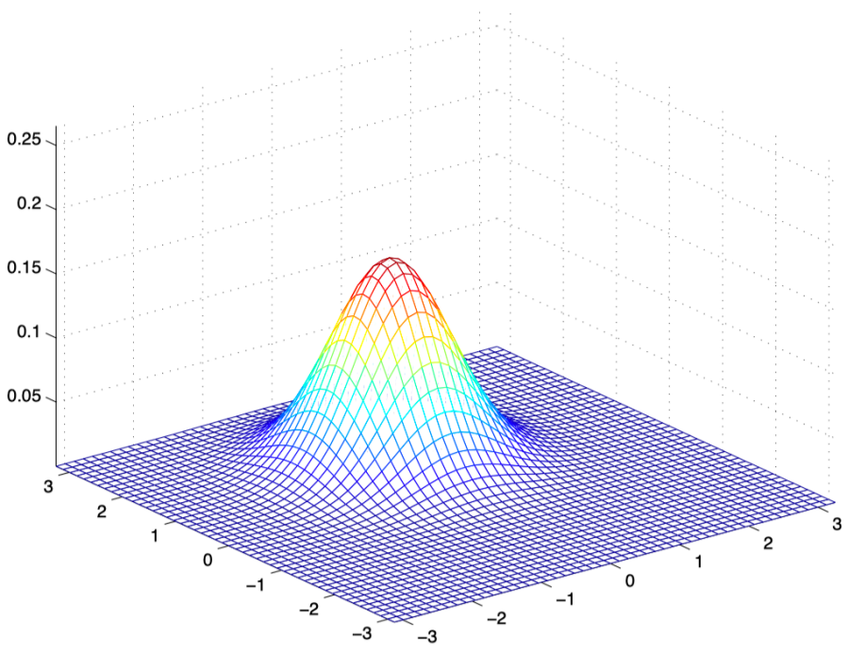




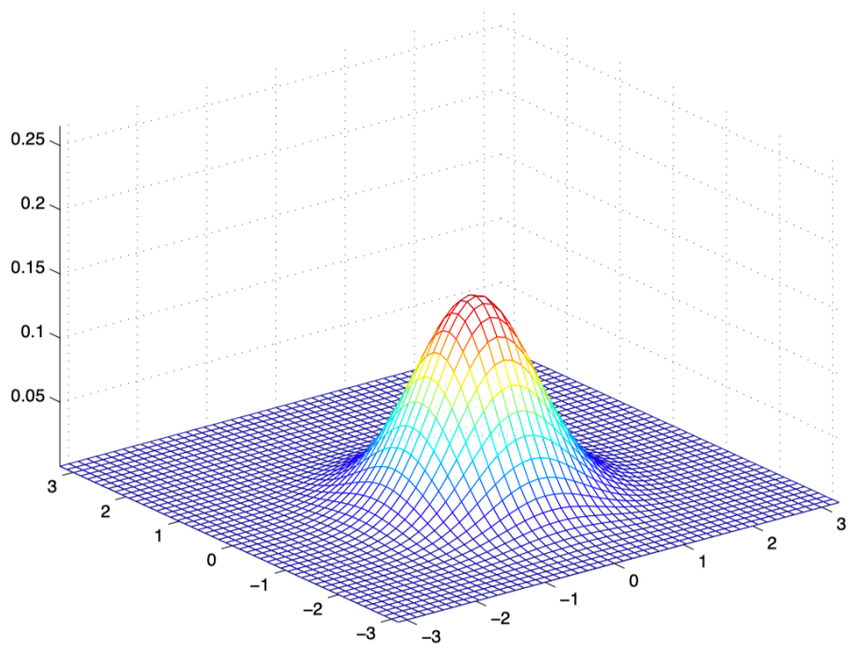
$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 3 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

MULTIVARIATE NORMAL DISTRIBUTION

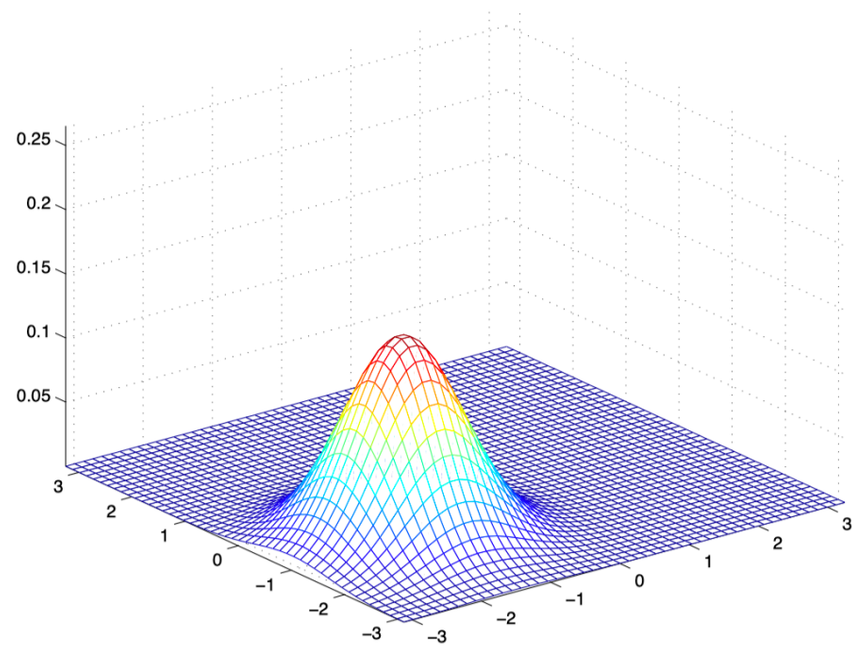
Examples:



$$\mu = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



$$\mu = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}$$



$$\mu = \begin{bmatrix} -1 \\ -1.5 \end{bmatrix}$$

GAUSSIAN DISCRIMINANT ANALYSIS (GDA)

The GDA Model

$$
\begin{aligned}
y &\sim \text{Bernoulli}(\phi) \\
x|y=0 &\sim \mathcal{N}(\mu_0, \Sigma) \\
x|y=1 &\sim \mathcal{N}(\mu_1, \Sigma)
\end{aligned}
$$

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)$$

GAUSSIAN DISCRIMINANT ANALYSIS (GDA)

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)$$

The parameters of the model are: $\phi, \Sigma, \mu_0, \mu_1$

Note:

The model with shared Σ is also called Linear Discriminant Analysis (LDA)

Separate Σ_0, Σ_1 lead to Quadratic Discriminant Analysis (QDA)

ESTIMATING THE PARAMETERS OF GDA

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)$$

The parameters of the model are learnt by maximizing the **log-likelihood** of the training data:

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

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

ESTIMATING THE PARAMETERS OF GDA

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

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

$$\ell(\cdot) = \log \prod_{i=1}^{m} \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(x - \mu_{y^{(i)}}\right)^T \Sigma^{-1}\left(x - \mu_{y^{(i)}}\right)\right) \cdot \phi^{y^{(i)}}(1-\phi)^{(1-y^{(i)})}$$

$$= \sum_{i=1}^{m}\left[-\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x - \mu_{y^{(i)}})^T\Sigma^{-1}(x - \mu_{y^{(i)}}) + y^{(i)}\log\phi + (1 - y^{(i)})\log(1-\phi)\right]$$

ESTIMATING THE PARAMETERS OF GDA

$$\ell(\cdot) = \sum_{i=1}^{m} \left[-\frac{1}{2} \log |\Sigma| - \frac{1}{2} \left(x - \mu_{y(i)} \right)^T \Sigma^{-1} \left(x - \mu_{y^{(i)}} \right) + y^{(i)} \log \phi + \left(1 - y^{(i)} \right) \log(1 - \phi) \right]$$

Example of learning θ :

$$\frac{\partial}{\partial \phi} \ell \left(\phi, \mu_0, \mu_1, \Sigma \right) = \sum_{i=1}^{m} y^{(i)} \frac{1}{\theta} - (1 - y^{(i)}) \frac{1}{1 - \theta} \stackrel{!}{=} 0$$

$$\sum_{i=1}^{m} y^{(i)} (1 - \theta) - (1 - y^{(i)}) \theta = 0$$

$$\sum_{i=1}^{m} y^{(i)} - \theta = 0$$

ESTIMATING THE PARAMETERS OF GDA

$$\ell(\cdot) = \sum_{i=1}^{m} \left[-\frac{1}{2} \log |\Sigma| - \frac{1}{2} \left(x - \mu_{y(i)} \right)^T \Sigma^{-1} \left(x - \mu_{y^{(i)}} \right) + y^{(i)} \log \phi + \left(1 - y^{(i)} \right) \log(1 - \phi) \right]$$

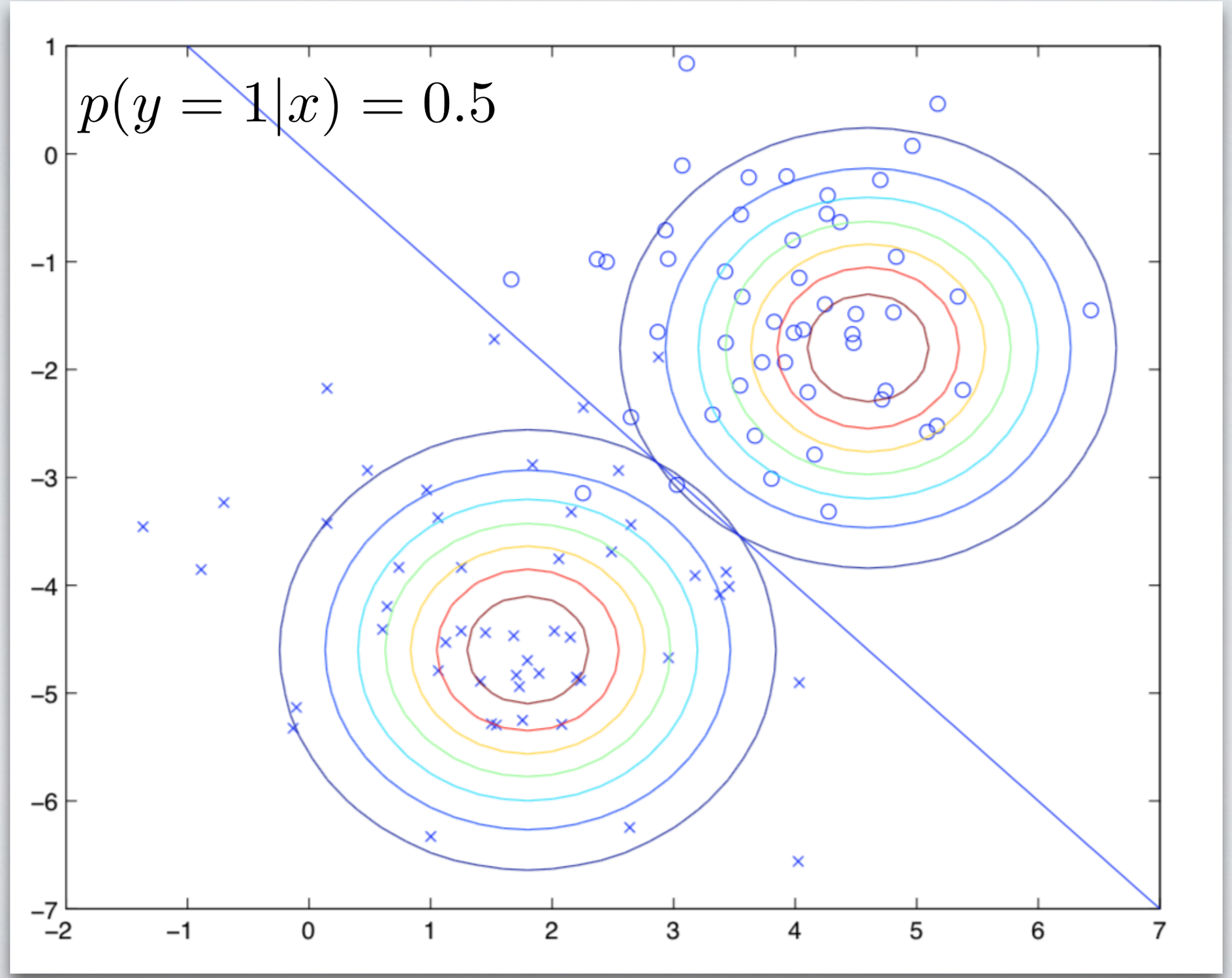
$$\phi = \frac{1}{m} \sum_{i=1}^{m} 1 \left\{ y^{(i)} = 1 \right\}$$

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

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

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

GDA EXAMPLE



GDA PREDICTION FUNCTION

$$p(y) = \phi^y(1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right)$$

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} = \frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0) + p(x|y=1)p(y=1)}$$

$$p(y=1|x;\phi,\Sigma,\mu_0,\mu_1) = \frac{1}{1+\exp(-\theta^T x)}$$

sigmoid with $\theta(\phi,\Sigma,\mu_0,\mu_1)$

GDA VS. LOGISTIC REGRESSION

- GDA and Logistic Regression will give different results in general
- We saw if $p(x|y)$ is gaussian with shared Σ then $p(y|x)$ is logistic. **The converse is not true!**
- GDA makes stronger modelling assumptions
- If the assumptions are true, then you can expect GDA to do better than Logistic Regression
- Logistic Regression will work better when the assumption are incorrect (e.g., non-gaussian data)
- Logistic Regression is used more often in practice

NAIVE BAYES

- So far we considered feature vectors where x_i are real valued
- We now study an algorithm where x_i are discrete
- The example will be e-mail classification into spam or not-spam
- This is a form of text-classification

Features:

- Enumerate a dictionary
- $x_i = 1$ means word i is in the mail
- The set of encoded words is called **vocabulary**
- Dim. x = size of vocabulary

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zygmurgy} \end{matrix}$$

NAIVE BAYES ASSUMPTION

- Given those features we want to build a model of $p(x|y)$ and $p(y)$
- If our vocabulary is of size 5000 and we model $p(x|y)$ with a multinomial distribution over the 2^{5000} possible outcomes we would need a (2^{5000}-1) dimensional parameter vector!
- Therefor make the **Naive Bayes Assumption** of ***conditional independence:*** $p(x_i|y) = p(x_i|y, x_j)$
- ***Note:*** This is different from assuming that x_i are independent (i.e., $p(x_i) = p(x_i|x_j)$)

NAIVE BAYES MODEL

Using the Naive Bayes Assumption our model of $p(x|y)$ is:

$$
\begin{aligned}
p\left(x_1, \ldots, x_n|y\right) &= p\left(x_1|y\right) p\left(x_2|y, x_1\right) p\left(x_3|y, x_1, x_2\right) \cdots p\left(x_n|y, x_1, \ldots, x_{n-1}\right) \\
&= p\left(x_1|y\right) p\left(x_2|y\right) p\left(x_3|y\right) \cdots p\left(x_n|y\right) \\
&= \prod_{i=1}^{n} p\left(x_i|y\right)
\end{aligned}
$$

Model Parameters:

$$
\begin{aligned}
\phi_{i|y=1} &= p\left(x_i = 1|y = 1\right) \\
\phi_{i|y=0} &= p\left(x_i = 1|y = 0\right) \\
\phi_y &= p(y = 1)
\end{aligned}
$$

ESTIMATING THE PARAMETERS

$$p\left(x_1, \ldots, x_n | y\right) = \prod_{i=1}^{n} p\left(x_i | y\right)$$

$$\phi_{i|y=1} = p\left(x_i = 1 | y = 1\right)$$
$$\phi_{i|y=0} = p\left(x_i = 1 | y = 0\right)$$
$$\phi_y = p(y = 1)$$

Again using **maximum likelihood**:

$$\begin{aligned}
\mathcal{L}(\phi_y, \phi_{i|y=0}, \phi_{i|y=1}) &= \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right) \\
&= \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}) p(y^{(i)}) \\
&= \prod_{i=1}^{m} \prod_{j=1}^{n} p\left(x_j^{(i)} | y^{(i)}\right) p\left(y^{(i)}\right)
\end{aligned}$$

ESTIMATING THE PARAMETERS

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}}$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

Simple interpretations:

- ϕ is the fraction of spam e-mails
- $\phi_{j|y=1}$ is the fraction of spam e-mails where word j appears

NAIVE BAYES CLASSIFIER

Making predictions:

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)}$$

$$= \frac{\left(\prod_{i=1}^{n} p\left(x_i|y=1\right)\right) p(y=1)}{\left(\prod_{i=1}^{n} p\left(x_i|y=1\right)\right) p(y=1) + \left(\prod_{i=1}^{n} p\left(x_i|y=0\right)\right) p(y=0)}$$

Classify e-mail as spam if $p(y=1|x) > 0.5$ or some other threshold…

A PROBLEM WITH NAIVE BAYES

Problem setting:

- You successfully trained your spam classifier and use it on your university mail
- After joining a the course ATML on ILIAS you get tons of mails with the word "atml"
- But the word "atml" did **not** appear in your training set!

What happens?

A PROBLEM WITH NAIVE BAYES

Lets look at the parameters for "atml":

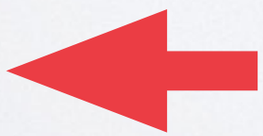
$$\phi_{35000|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_{35000}^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}} = 0$$

$$\phi_{35000|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_{35000}^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\}} = 0$$

The class posterior probability is given by:

$$p(y=1|x) = \frac{\prod_{i=1}^{n} p\left(x_i|y=1\right) p(y=1)}{\prod_{i=1}^{n} p\left(x_i|y=1\right) p(y=1) + \prod_{i=1}^{n} p\left(x_i|y=0\right) p(y=0)}$$

$$= \frac{0}{0}$$



No clue what to do!

SOLUTION: LAPLACE SMOOTHING

Observation:
Its a bad idea to assign 0 probability to an event you haven't seen

Example:
Estimating the mean of multinomial random variable z taking values in $\{1, \ldots, k\}$. Given a set of observations $\left\{z^{(1)}, \ldots, z^{(m)}\right\}$ the ML estimate is given by:

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\}}{m}$$

Laplace Smoothing:

$$\phi_j = \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\} + 1}{m + k}$$

SOLUTION: LAPLACE SMOOTHING

For Naive Bayes:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} + 2}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\} + 1}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 0\right\} + 2}$$

Its like adding one occurrence in spam/not-spam for words not present in the training data

EVENT MODELS FOR TEXT CLASSIFICATION

An alternative to Naive Bayes:

Let the features x_i be the identity of the i-th word in the email. x_i is now an integer in $\{1, \ldots, |V|\}$ and an email is represented by a vector $(x_1, x_2, \ldots, x_n)$

"Hi, how are you?"

(223, 260, 34, 2047)

The model:

$$p(x_1, \ldots, x_n | y) = \prod_{i=1}^{n} p(x_i | y)$$

$x_i|y$ is multinomial!

EVENT MODELS FOR TEXT CLASSIFICATION

Parameters:

$$\phi_y = p(y)$$
$$\phi_{i|y=0} = p\left(x_j = i | y = 0\right)$$
$$\phi_{i|y=1} = p\left(x_j = i | y = 1\right)$$

for any j

Data: $\left\{\left(x^{(i)}, y^{(i)}\right); i = 1, \ldots, m\right\}$ $\quad x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}\right)$

Likelihood:

$$\mathcal{L}(\phi, \phi_{i|y=0}, \phi_{i|y=1}) = \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right)$$
$$= \prod_{i=1}^{m} (\prod_{j=1}^{n_i} p(x_j^{(i)} | y; \phi_{i|y=0}, \phi_{i|y=1})) p\left(y^{(i)}; \phi_y\right)$$

EVENT MODELS FOR TEXT CLASSIFICATION

Maximum Likelihood Estimates:

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\} n_i}$$

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{n_i} 1\left\{y^{(i)} = 0\right\} n_i}$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\left\{y^{(i)} = 1\right\}}{m}$$

SUPPORT VECTOR MACHINES TEASER

Machine Learning HS18

MARGINS INTUITION

Logistic Regression Reminder:

$$h_\theta(x) = g(\theta^T x)$$

$$h_\theta(x) = p(y = 1|x; \theta)$$

Decision Function:

$$\theta^T x \geq 0 \quad \rightarrow \quad y = 1$$

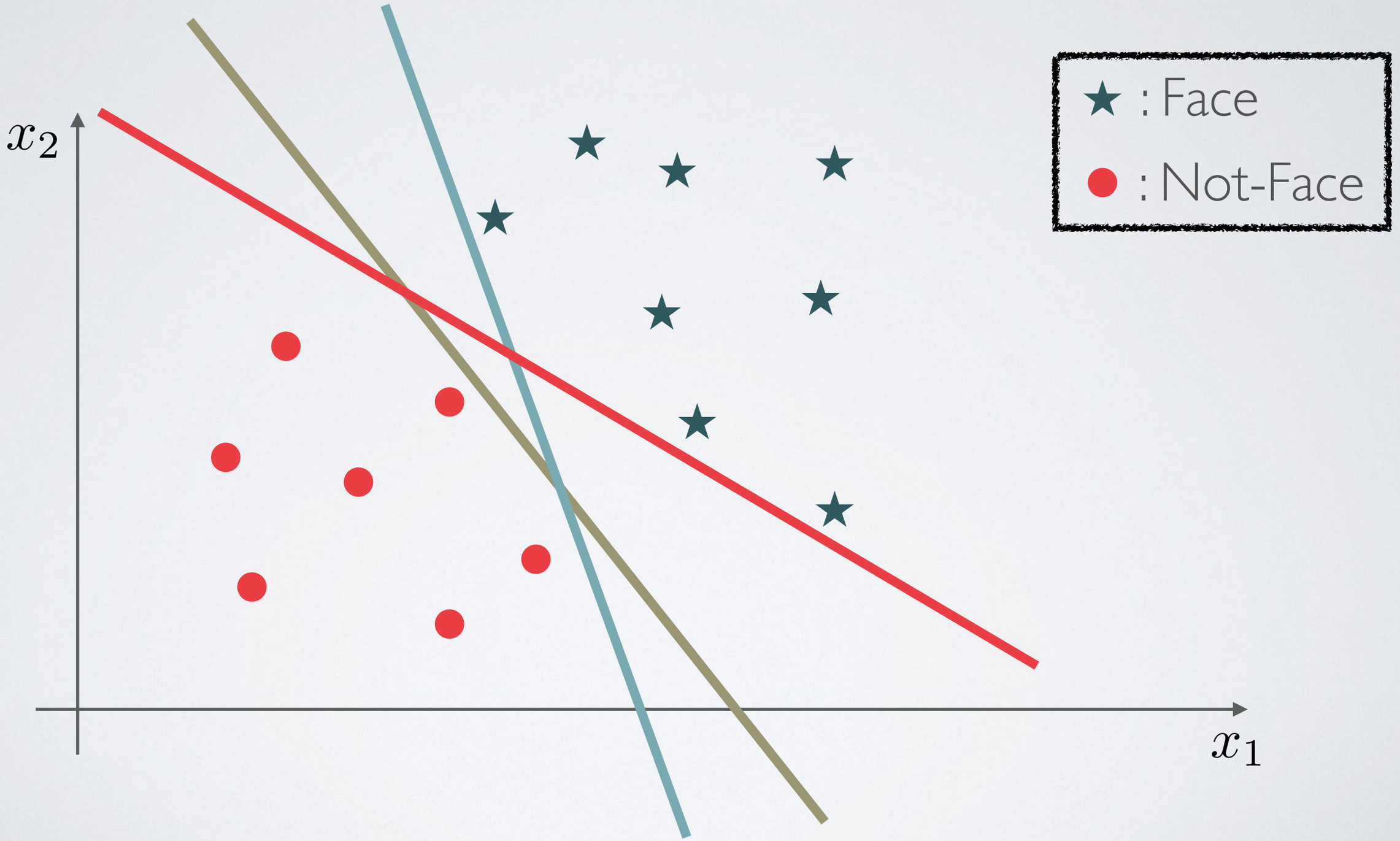
$$\theta^T x < 0 \quad \rightarrow \quad y = 0$$

More confident if:

$$\theta^T x \gg 0$$

$$\theta^T x \ll 0$$

LARGE MARGIN: INTUITION



LARGE MARGIN: INTUITION