Classification: Probabilistic Generative Model

Classification

x → Function → Class n

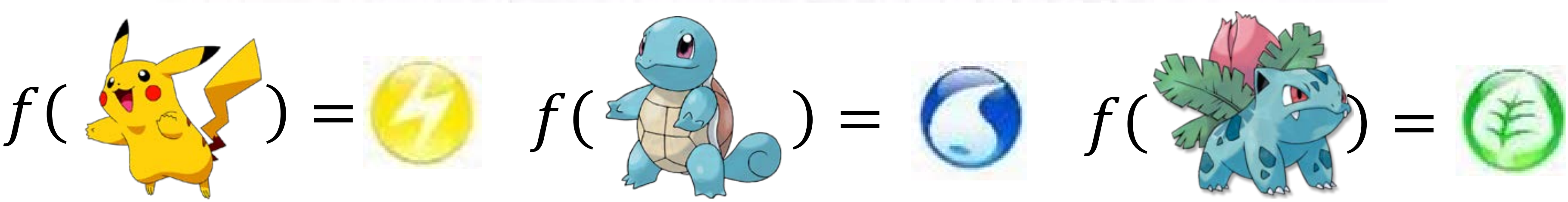
- Credit Scoring
  - Input: income, savings, profession, age, past financial history ……
  - Output: accept or refuse
- Medical Diagnosis
  - Input: current symptoms, age, gender, past medical history ……
  - Output: which kind of diseases
- Handwritten character recognition

Input:  output: 金

- Face recognition
  - Input: image of a face, output: person

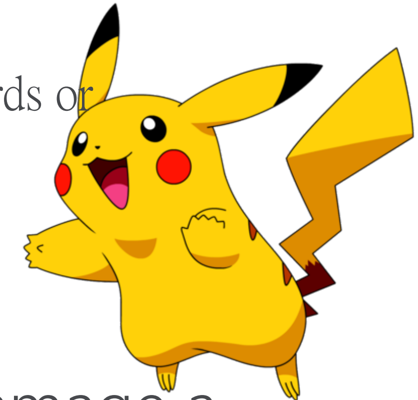
Example Application

Recognize the type of Pokémon.



Example Application

pokemon games
(*NOT* pokemon cards or
Pokemon Go)



x

- **HP**: hit points, or health, defines how much damage a pokemon can withstand before faintir 35
- **Attack**: the base modifier for normal attacks (eg. Scratch, Punch) 55
- **Defense**: the base damage resistance against normal attacks 40
- **SP Atk**: special attack, the b 50 nodifier for special attacks (e.g. fire blast, bubble beam) 50
- **SP Def**: the base damage resistance against special attacks 90
- **Spee** ch rour

y

Can we predict the "type" of pokemon based on the information?

Ability value

The purpose is that if we know the type of the enemy Pokémon based on the ability value, we can choose the counter type of the Pokémon to battle.

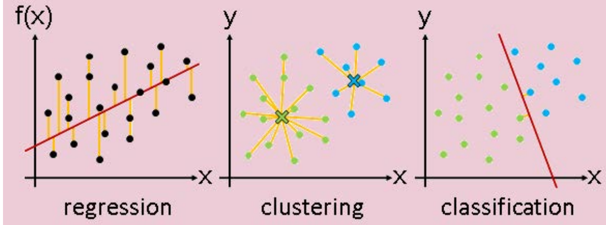
Example Application

× 攻擊方的屬性 \ 防禦方的屬性	一般	格鬥	飛行	毒	地面	岩石	蟲	幽靈	鋼	火	水	草	電	超能力	冰	龍	惡	妖精
一般	1×	1×	1×	1×	1×	½×	1×	0×	½×	1×	1×	1×	1×	1×	1×	1×	1×	1×
格鬥	2×	1×	½×	½×	1×	2×	½×	0×	2×	1×	1×	1×	1×	½×	2×	1×	2×	½×
飛行	1×	2×	1×	1×	1×	½×	2×	1×	½×	1×	1×	2×	½×	1×	1×	1×	1×	1×
毒	1×	1×	1×	½×	½×	½×	1×	½×	0×	1×	1×	2×	1×	1×	1×	1×	1×	2×
地面	1×	1×	0×	2×	1×	2×	½×	1×	2×	2×	1×	½×	2×	1×	1×	1×	1×	1×
岩石	1×	½×	2×	1×	½×	1×	2×	1×	½×	2×	1×	1×	1×	1×	2×	1×	1×	1×
蟲	1×	½×	½×	½×	1×	1×	1×	½×	½×	½×	1×	2×	1×	2×	1×	1×	2×	½×
幽靈	0×	1×	1×	1×	1×	1×	1×	2×	1×	1×	1×	1×	1×	2×	1×	1×	½×	1×
鋼	1×	1×	1×	1×	1×	2×	1×	1×	½×	½×	½×	1×	½×	1×	2×	1×	1×	2×
火	1×	1×	1×	1×	1×	½×	2×	1×	2×	½×	½×	2×	1×	1×	2×	½×	1×	1×
水	1×	1×	1×	1×	2×	2×	1×	1×	1×	2×	½×	½×	1×	1×	1×	½×	1×	1×
草	1×	1×	½×	½×	2×	2×	½×	1×	½×	½×	2×	½×	1×	1×	1×	½×	1×	1×
電	1×	1×	2×	1×	0×	1×	1×	1×	1×	1×	2×	½×	½×	1×	1×	½×	1×	1×
超能力	1×	2×	1×	2×	1×	1×	1×	1×	½×	1×	1×	1×	1×	½×	1×	1×	0×	1×
冰	1×	1×	2×	1×	2×	1×	1×	1×	½×	½×	½×	2×	1×	1×	½×	2×	1×	1×
龍	1×	1×	1×	1×	1×	1×	1×	1×	½×	1×	1×	1×	1×	1×	1×	2×	1×	0×
惡	1×	½×	1×	1×	1×	1×	1×	2×	1×	1×	1×	1×	1×	2×	1×	1×	½×	½×
妖精	1×	2×	1×	½×	1×	1×	1×	1×	½×	½×	1×	1×	1×	1×	1×	2×	2×	1×

這些倍數適用於XY及之後的遊戲。

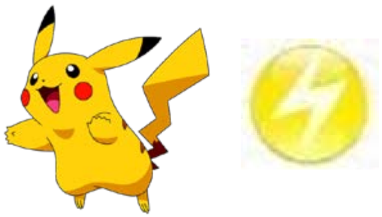
How to do Classification

Regression 讓所有間距平方和盡量小。 Clustering 讓最大間距盡量小。 Classification 讓最小間距盡量大。三者都是針對間距進行 Optimization 。



- Training data for Classification

$(x^1, \hat{y}^1)$ $(x^2, \hat{y}^2)$ …… $(x^N, \hat{y}^N)$



Classification as Regression?

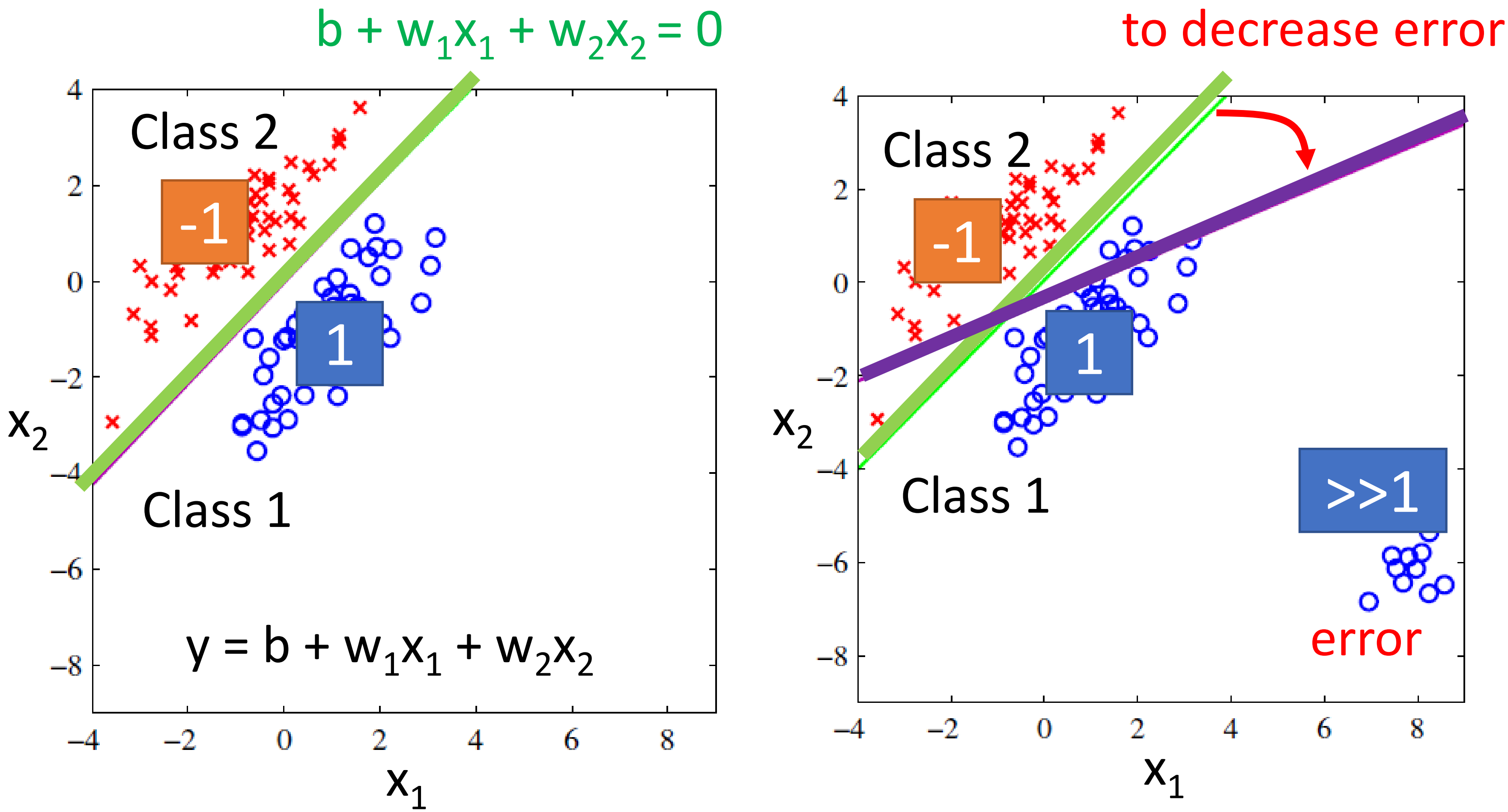
Binary classification as example

Use regression to train the classification problem.

⇒ Problematic (Next page)

Training: Class 1 means the target is 1; Class 2 means the target is -1

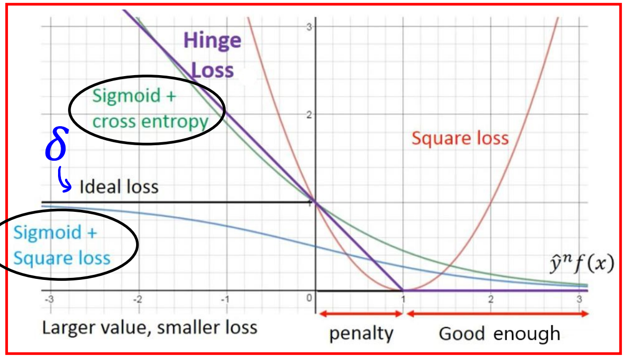
Testing: closer to 1 → class 1; closer to -1 → class 2



1° Penalize to the examples that are "too correct" … (Bishop, P186)

2° • Multiple class: Class 1 means the target is 1; Class 2 means the target is 2; Class 3 means the target is 3 …… problematic

With square loss or cross entropy?
Discuss in the next chapter.



Ideal Alternatives

Change regression to classification.

Ideal: Add δ
Approximation: Add sigmoid

At the end of this chapter

- Function (Model):

$f(x)$

$x \Rightarrow$

$g(x) > 0$	Output = class 1
$else$	Output = class 2

- Loss function:

$$L(f) = \sum_n \delta(f(x^n) \neq \hat{y}^n)$$

The number of times f get incorrect results on training data.

We can't differentiate this formula. $\Rightarrow$ We can't use gradient descent.

- Find the best function: $\Rightarrow$ Use sigmoid instead of δ
  - Example: Perceptron, SVM

Not Today

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

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$$

Two Boxes

We can regard a box as a class label, and regard a colorful ball as an object.

y x

(B$_1$) Box 1

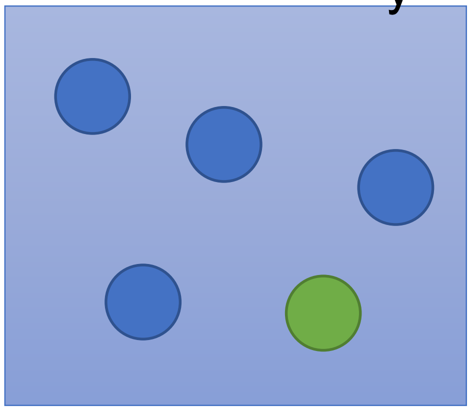
$P(B_1) = 2/3$

+

$P(B_2)$

‖

1

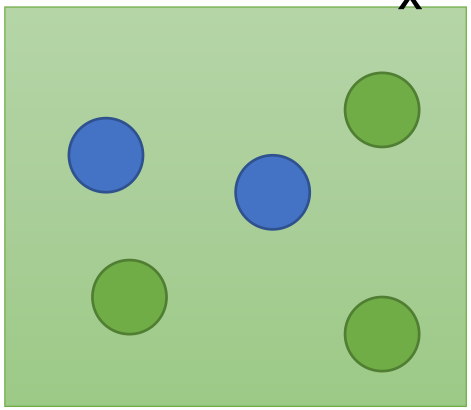


$P(\text{Blue}|B_1) = 4/5$

$P(\text{Green}|B_1) = 1/5$

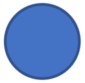
(B$_2$) Box 2

$P(B_2) = 1/3$



$P(\text{Blue}|B_1) = 2/5$

$P(\text{Green}|B_1) = 3/5$

 from one of the boxes

Where does it come from?

(Distribution)

Likelihood Prior

Posterior

|||

Objective of classification problem

$$\overset{y}{P(B_1} \mid \overset{x}{\text{Blue})} = \frac{P(\text{Blue}|B_1)P(B_1)}{\boxed{P(\text{Blue}|B_1)P(B_1) + P(\text{Blue}|B_2)P(B_2)}} = P(B)$$

belongs to

Prior 、Posterior 和 Likelihood 的理解与几种表达方式

1. 基本定义

Prior - 先验概率：预先知道事件发生的概率。

Posterior – 后验概率：Given evidence/experience/observation，Random event 发生的概率。

Likelihood – 似然：不符合概率的性质，但有 Probability 的含义。

2. 核心概念

$$\text{Posterior probability} \propto \text{Likelihood} \times \text{Prior probability}$$

后验概率 ← 先验概率

3. 从两个事件的角度

$$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

4. 从假说和数据的角度

$$P(h \mid Data) = \frac{P(data \mid h) P(h)}{P(data)}$$

5. 从概率分布的参数估计的角度

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

x： Observation

θ：Probability 的参数

p(x|θ)：Probability of x given θ。

Likelihood of θ given x observed。

Likelihood is a Function of θ。

6. 从估计 Sample 权重的角度

Predictions Matrix 表达 $\mathbf{y} = \mathbf{X}\boldsymbol{\omega}$

X： matrix。每一行是一个Sample，每一行的每一个元素是一个 Feature。

Ω：weights

y：vector。是X每一行与weights的乘积

Bayes 公式
求解 Posterior $p(\boldsymbol{w} \mid \boldsymbol{X}, \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w})$

7. Likelihood的详细探讨

Likelihood：Given outcomes x，the likelihood of parameter θ。
Equal to the probability of observed x, given parameter θ。

$$\mathcal{L}(\theta|x) = P(x|\theta)$$ L(θ|x) 这个标记有时候容易引起误解

Likelihood：Observed values x1, x2, ⋯, xn fixed，θ is variable。

$$\mathcal{L}(\theta; x_1, \ldots, x_n) = f(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta)$$

用分号分割避免误解

$$P(\theta \mid x_1, x_2, \ldots, x_n) = \frac{f(x_1, x_2, \ldots, x_n \mid \theta) P(\theta)}{P(x_1, x_2, \ldots, x_n)}$$

P(θ) 是参数 θ 的先验分布。

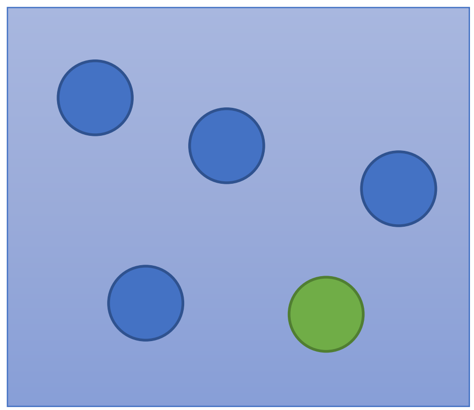
In this chapter, we only assume there are

Two Classes

Estimating the Probabilities From training data

Class 1

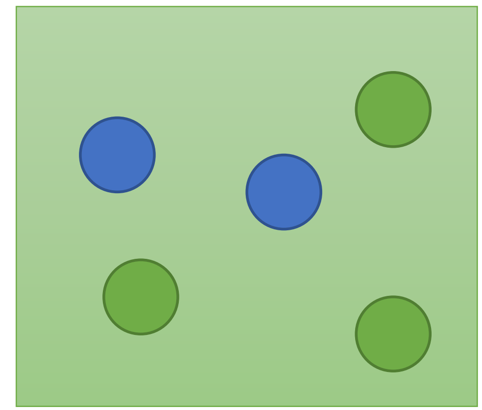
$\boxed{P(C_1)}$



$\boxed{P(x|C_1)}$

Class 2

$\boxed{P(C_2)}$



$\boxed{P(x|C_2)}$

Given an x, which class does it belong to

We can use prior probability and likelihood (distribution) to generate posterior probability or object's probability.

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

Generative Model $P(x) = P(x|C_1)P(C_1) + P(x|C_2)P(C_2)$

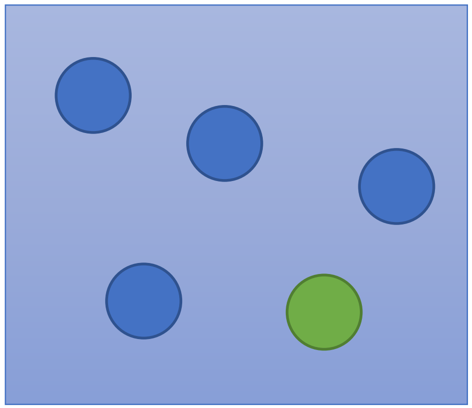
$P(x) = \Sigma\, P(x|C_i)\, P(C_i)$

For Pokémon instead of colorful balls:

Prior

Class 1

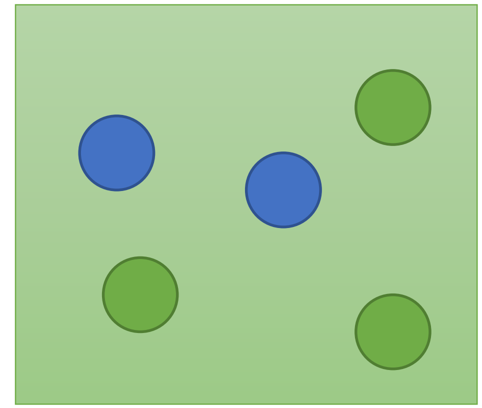
$\boxed{P(C_1)}$



Water

Class 2

$\boxed{P(C_2)}$



Normal

Water and Normal type with ID < 400 for training, rest for testing

Training: 79 Water, 61 Normal

$$P(C_1) = 79 / (79 + 61) = 0.56$$

$$P(C_2) = 61 / (79 + 61) = 0.44$$

Probability from Class

There is no sea turtle in training data, so is the probability of sea turtle belongs to the water type equal to 0? No (Next page)

$P(x|C_1) = ?$ P( |Water) = ?

Likelihood

Each Pokémon is represented as a vector by its attribute. → feature

Water Type

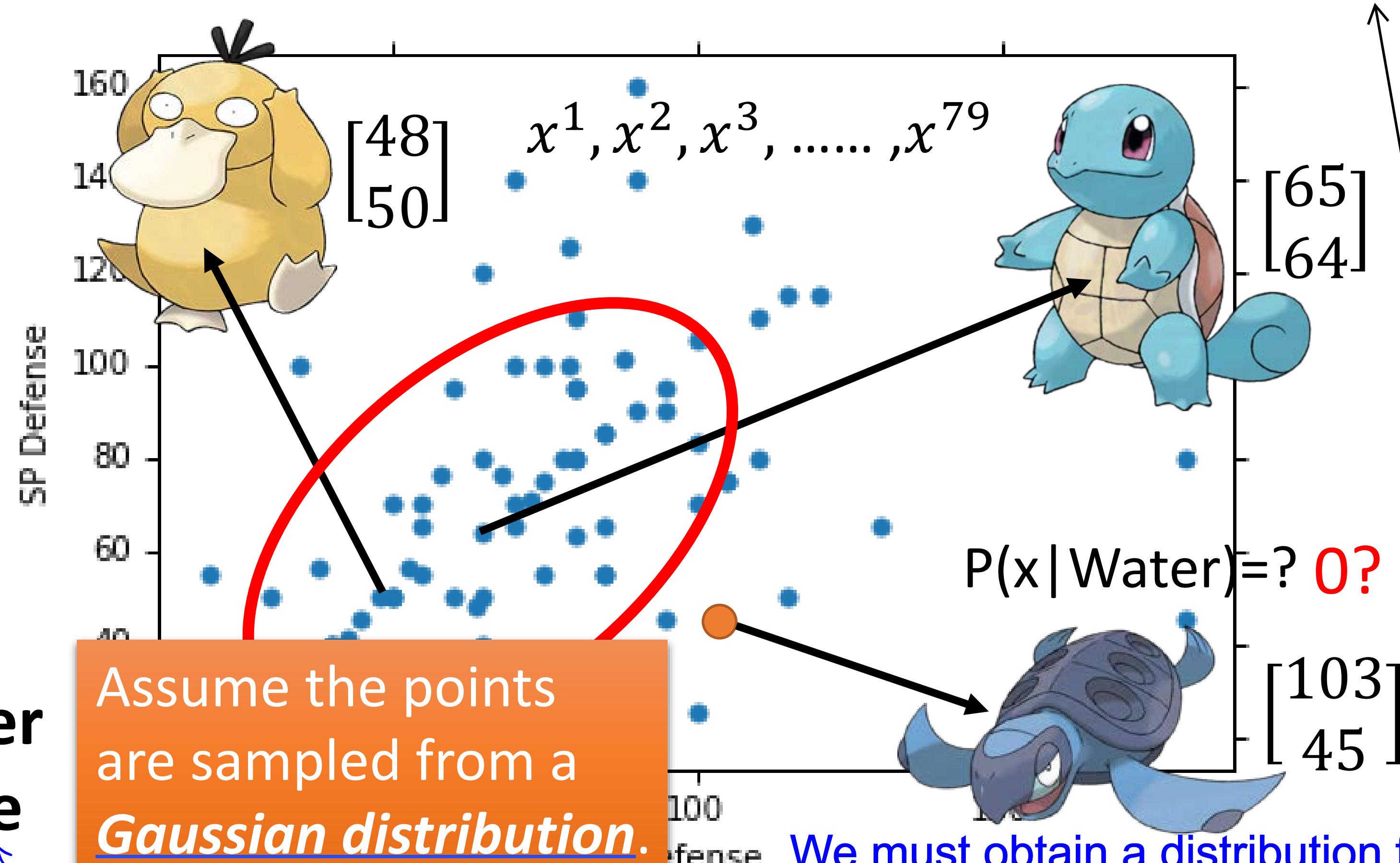
79 in total

Probability from Class - Feature

We form the distribution of the water Pokémon based on their features.
(ex: Defense and SP Defense)

- Considering **Defense** and **SP Defense**

#Gaussian = #class



Water Type

1 to 1

Assume the points are sampled from a ***Gaussian distribution***.

We must obtain a distribution for each class. ⇒ GMM https://blog.slinuxer.com/tag/pca

Gaussian Distribution

Be used to calculate likelihood

$$\boxed{f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}}$$

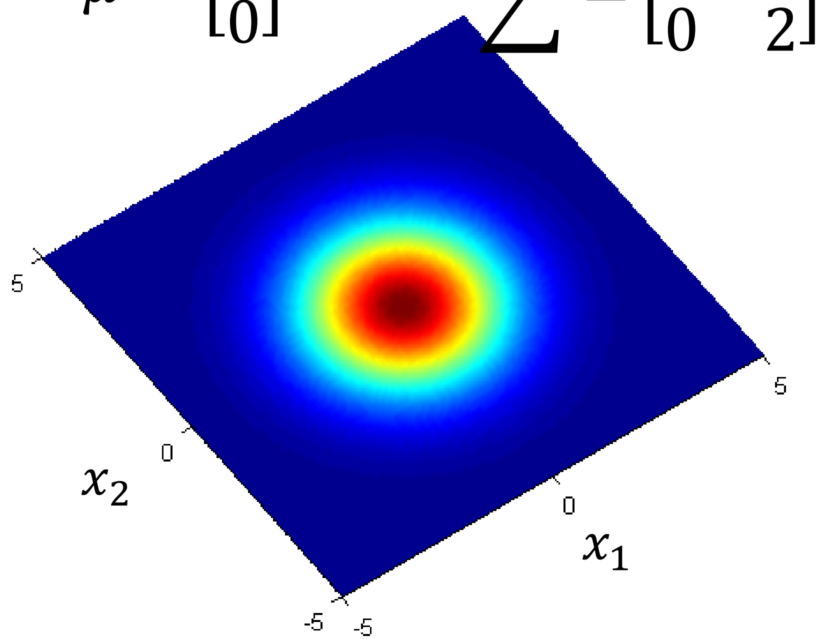
Input: vector x, output: probability of sampling x

The shape of the function determines by **mean $\boldsymbol{\mu}$** and **covariance matrix $\boldsymbol{\Sigma}$**

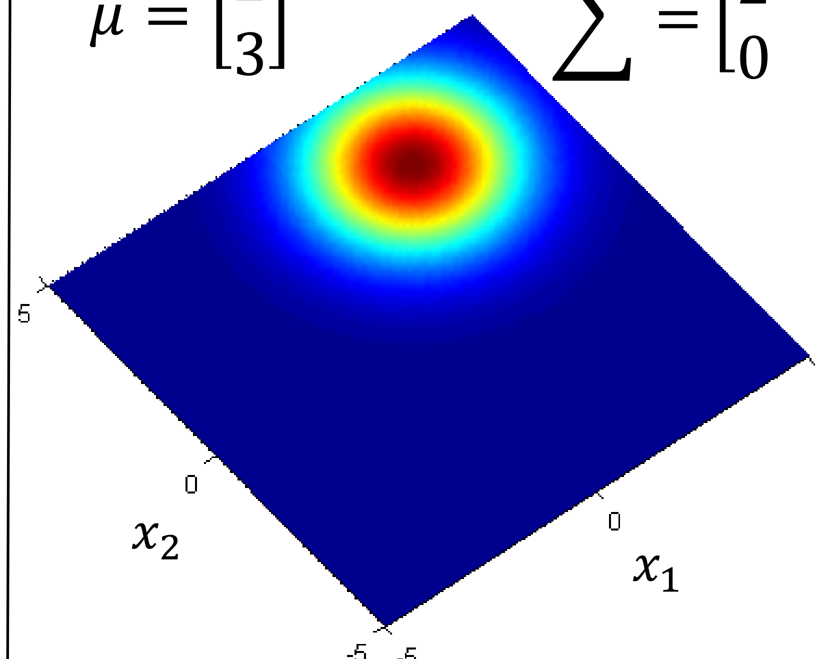
come from C_i

We can calculate $P(x|C_i)$ through the distribution of C_i. ($f_{\mu,\Sigma}(x) = P(x|C_i)$)

$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\sum = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$



$\mu = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $\sum = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$



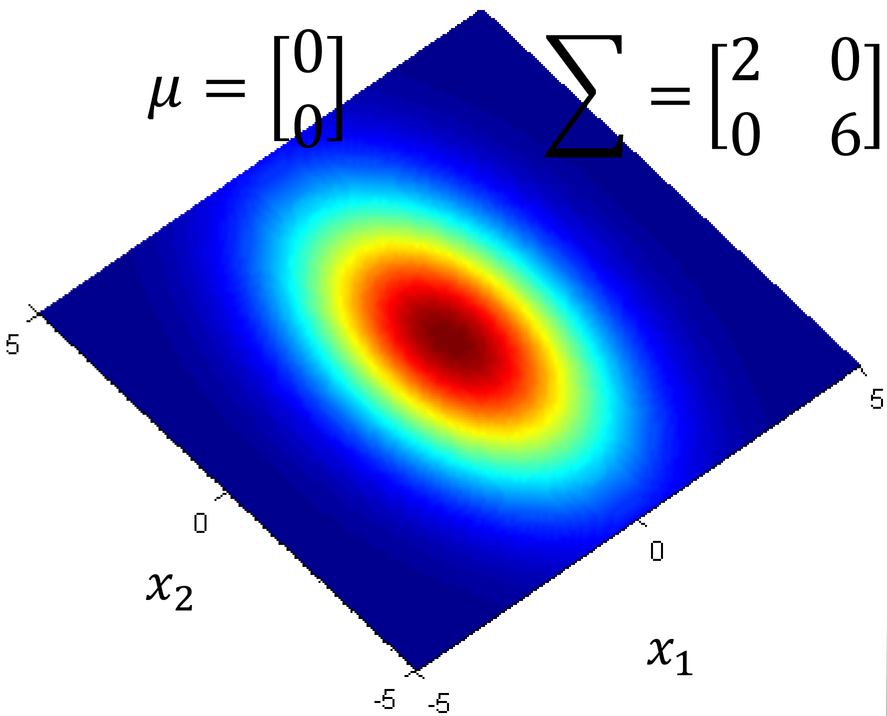
Gaussian Distribution

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

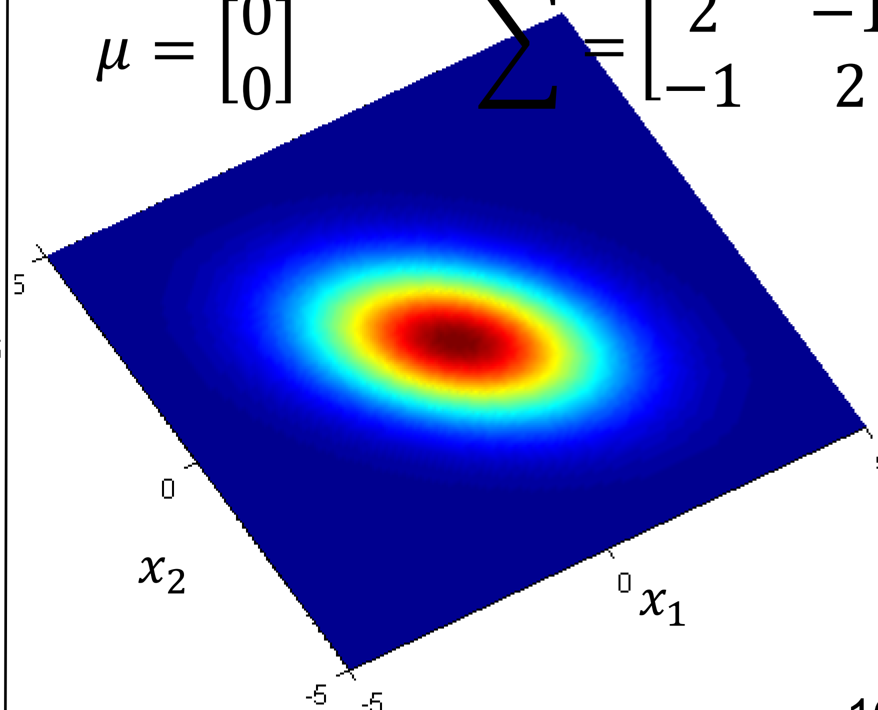
Input: vector x, output: probability of sampling x

The shape of the function determines by **mean $\boldsymbol{\mu}$** and **covariance matrix $\boldsymbol{\Sigma}$**

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \sum = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}$$



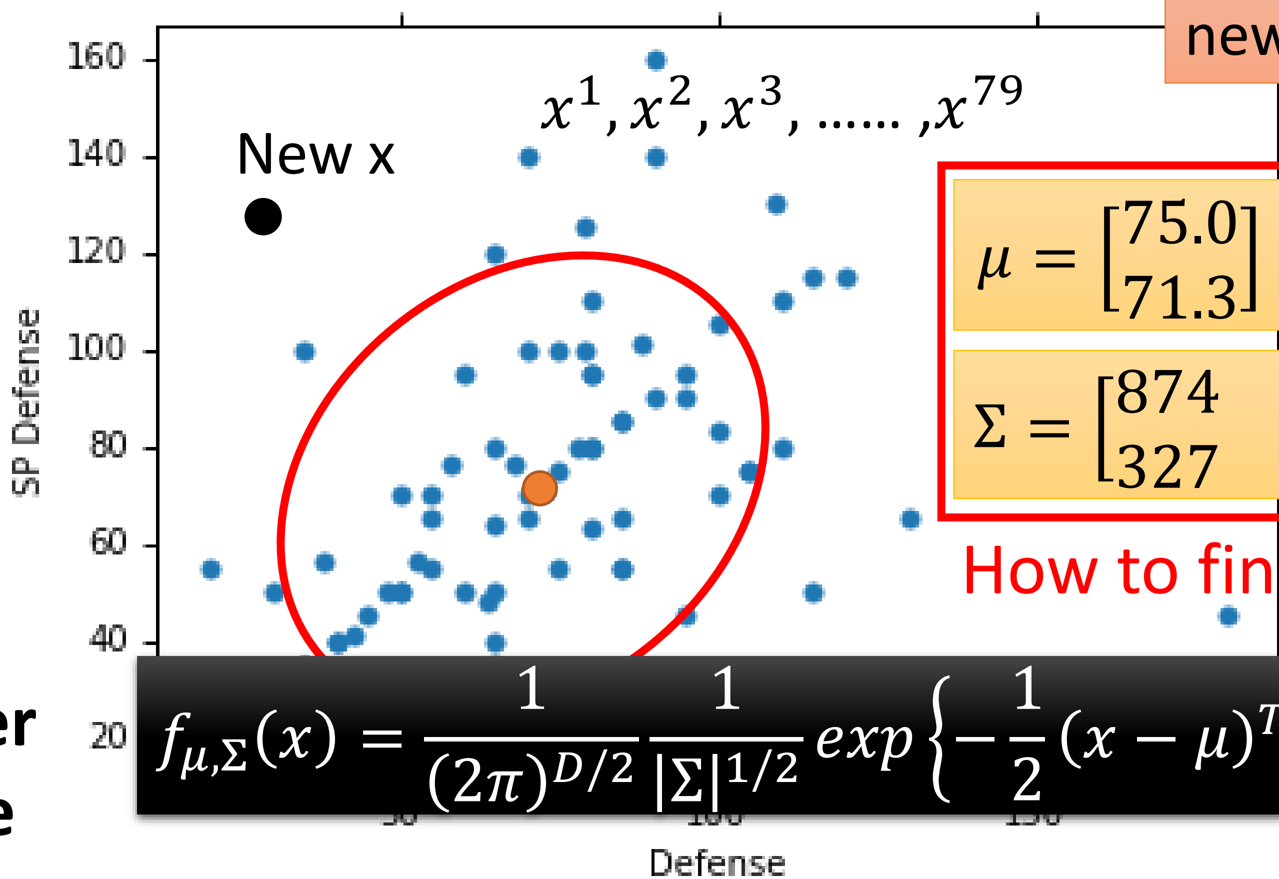
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \sum = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$



Probability from Class

Assume the points are sampled from a Gaussian distribution

Find the Gaussian distribution behind them ➡ Probability for new points



$$\mu = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix}$$

How to find them? MLE

Water Type

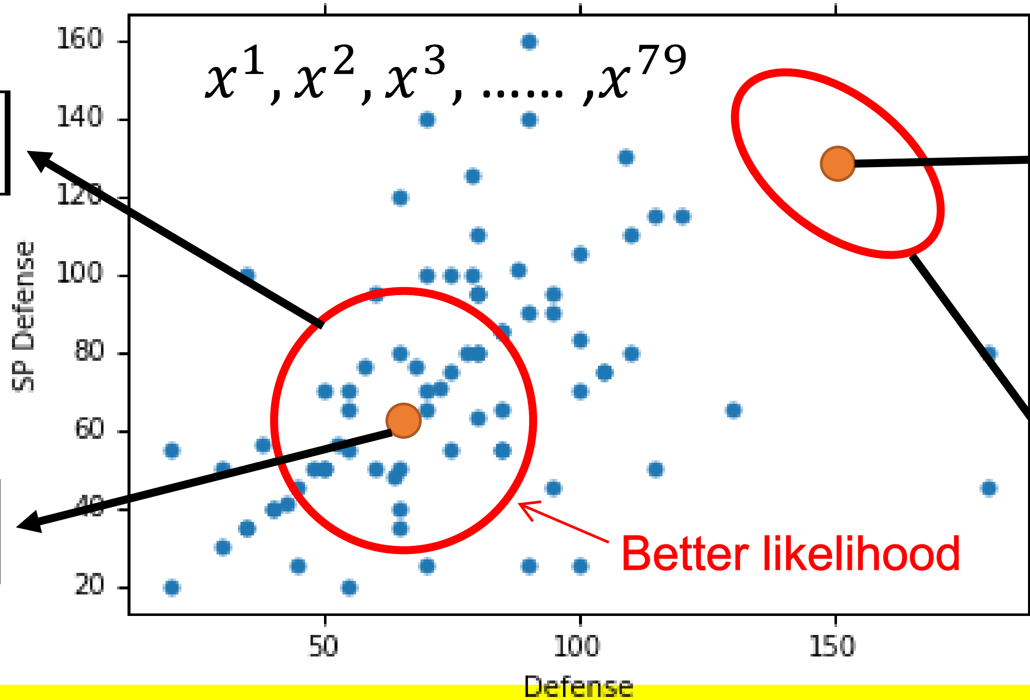
$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right\}$$

Likelihood is kind of like the objective function of Gaussian distribution.

Maximum Likelihood

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

$$\sum = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \qquad \mu = \begin{bmatrix} 75 \\ 75 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 150 \\ 140 \end{bmatrix} \qquad \sum = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

The Gaussian with any mean μ and covariance matrix Σ can generate these points. → Different Likelihood

Likelihood of a Gaussian with mean μ and covariance matrix Σ

The bigger the better

= the probability of the Gaussian samples $x^1, x^2, x^3, \ldots\ldots, x^{79}$

L stands for likelihood instead of loss function.

$$L(\mu, \Sigma) = f_{\mu,\Sigma}(x^1) f_{\mu,\Sigma}(x^2) f_{\mu,\Sigma}(x^3) \ldots\ldots f_{\mu,\Sigma}(x^{79})$$

It looks like we don't need y to calculate likelihood, but we need to remember that these X come from the same y. So it's still a supervised learning.

Maximum Likelihood

Supervised GMM: classification
Unsupervised GMM: clustering

We have the "Water" type Pokémons: $x^1, x^2, x^3, \ldots\ldots, x^{79}$

We assume $x^1, x^2, x^3, \ldots\ldots, x^{79}$ generate from the Gaussian (μ^*, Σ^*) with the ***maximum likelihood***

$$L(\mu, \Sigma) = f_{\mu,\Sigma}(x^1) f_{\mu,\Sigma}(x^2) f_{\mu,\Sigma}(x^3) \ldots\ldots f_{\mu,\Sigma}(x^{79})$$

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

$\frac{dL}{d\mu} = 0$

$\frac{dL}{d\Sigma} = 0$

$$\mu^*, \Sigma^* = arg \max_{\mu,\Sigma} L(\mu, \Sigma)$$

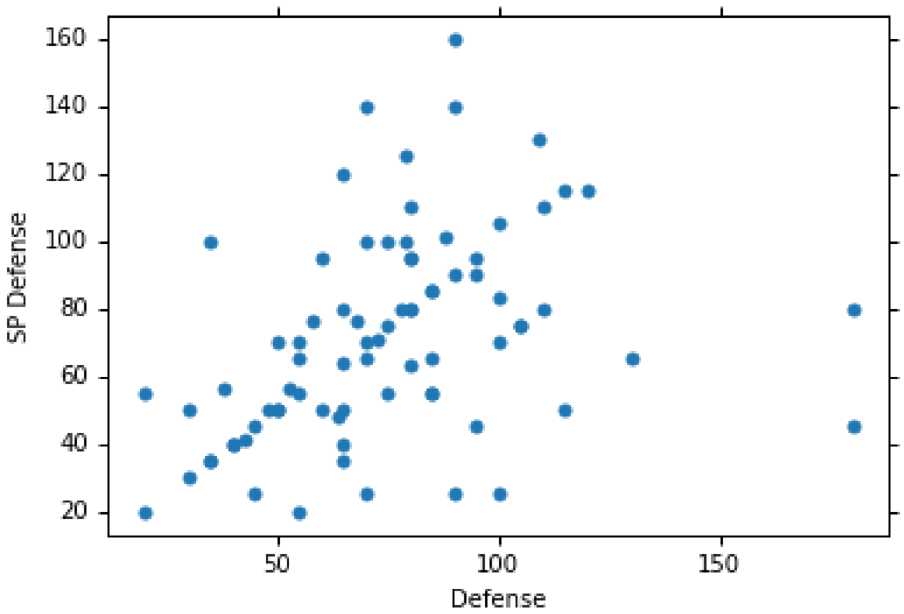
Closed-form solution for each class

$$\mu^* = \frac{1}{79} \sum_{n=1}^{79} x^n \qquad \Sigma^* = \frac{1}{79} \sum_{n=1}^{79} (x^n - \mu^*)(x^n - \mu^*)^T$$

average

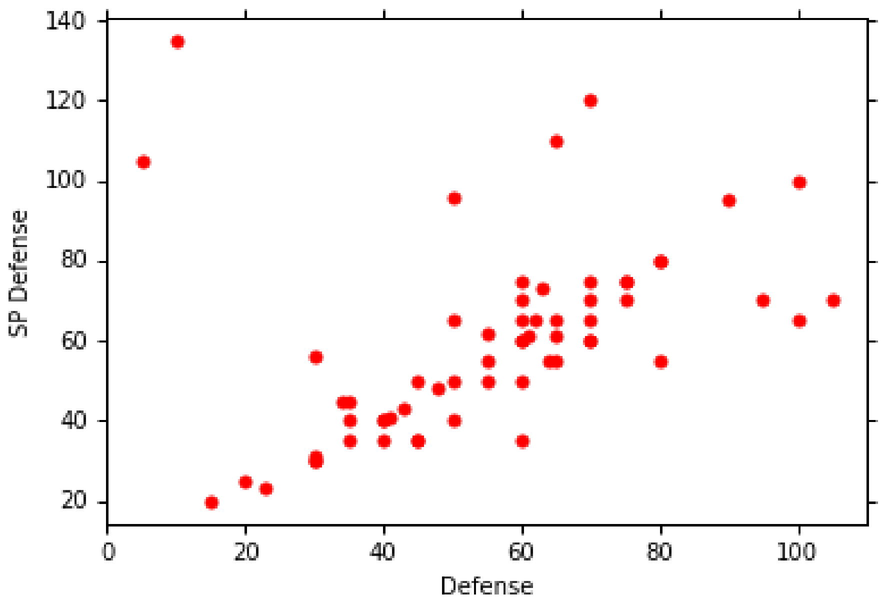
Maximum Likelihood

Class 1: Water



$$\mu^1 = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix} \quad \Sigma^1 = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix}$$

Class 2: Normal



$$\mu^2 = \begin{bmatrix} 55.6 \\ 59.8 \end{bmatrix} \quad \Sigma^2 = \begin{bmatrix} 847 & 422 \\ 422 & 685 \end{bmatrix}$$

Now we can do classification ☺

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

$$\mu^1 = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix} \quad \Sigma^1 = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix}$$

P(C1)
= 79 / (79 + 61) =0.56

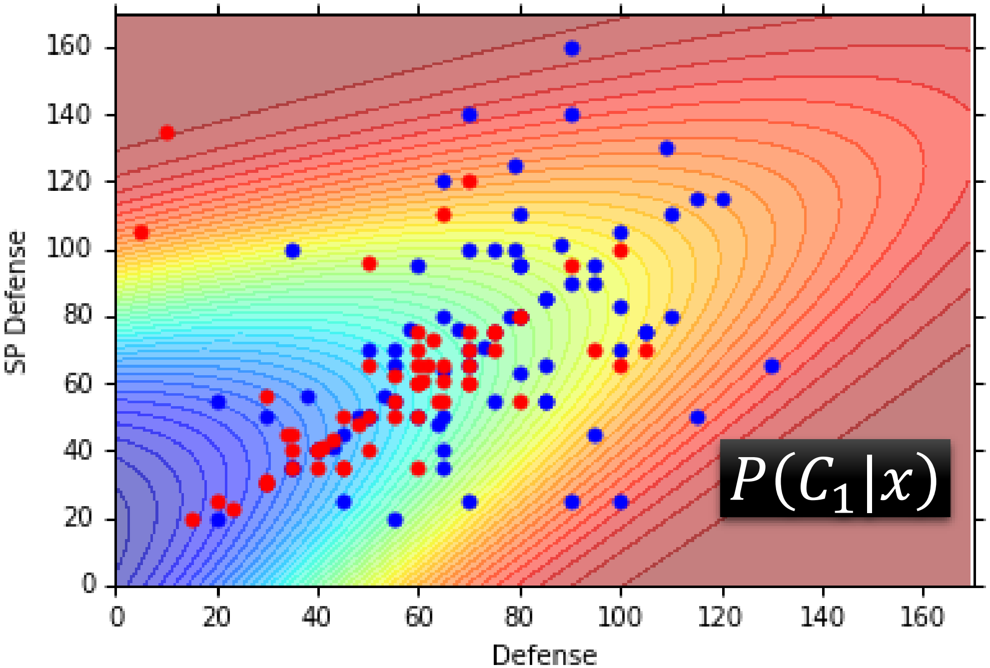
$$\boxed{P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}}$$

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

$$\mu^2 = \begin{bmatrix} 55.6 \\ 59.8 \end{bmatrix} \quad \Sigma^2 = \begin{bmatrix} 847 & 422 \\ 422 & 685 \end{bmatrix}$$

P(C2)
= 61 / (79 + 61)
=0.44

If $P(C_1|x) > 0.5$ ➡ x belongs to class 1 (Water)



Blue points: C_1 (Water), Red points: C_2 (Normal)

How's the results?

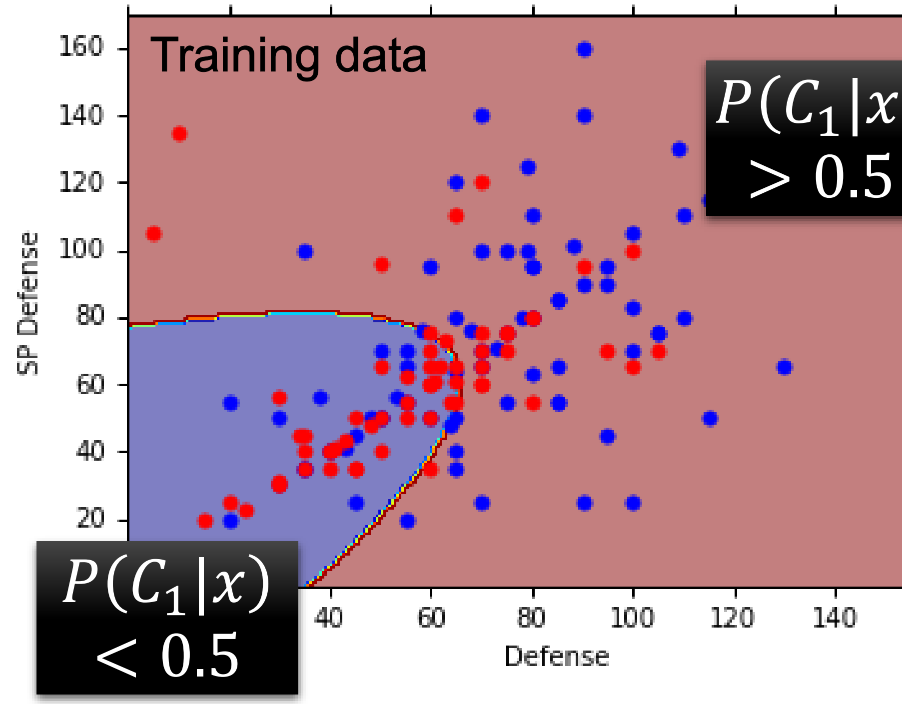
Testing data: 47% accuracy ☹

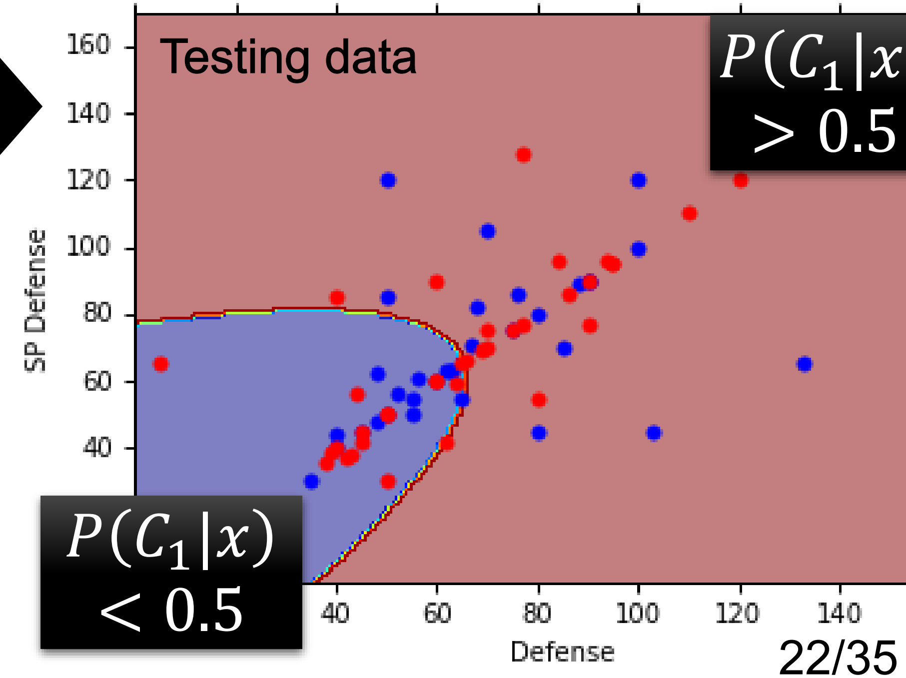
All: hp, att, sp att,
de, sp de, speed (6 features)

μ^1, μ^2: 6-dim vector

Σ^1, Σ^2: 6 x 6 matrices

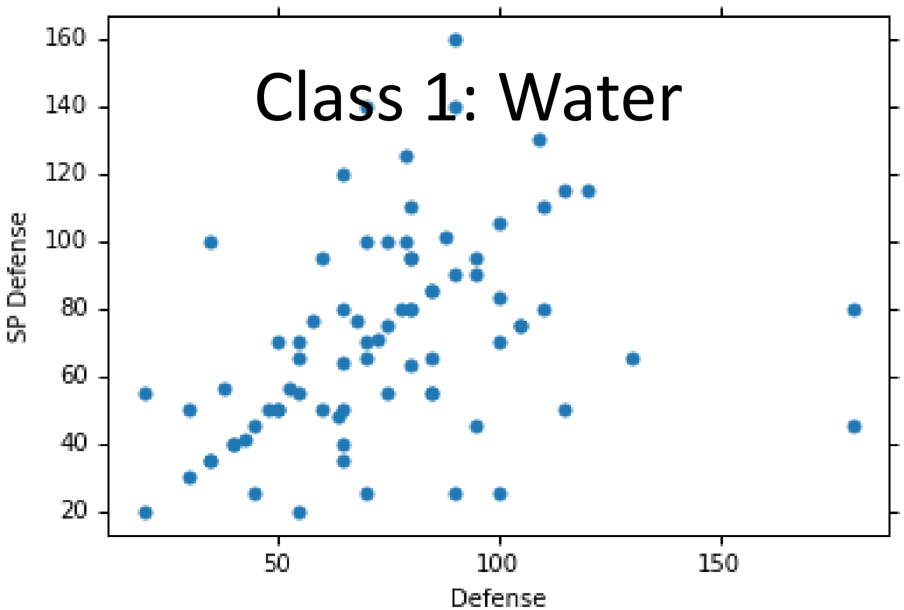
64% accuracy …

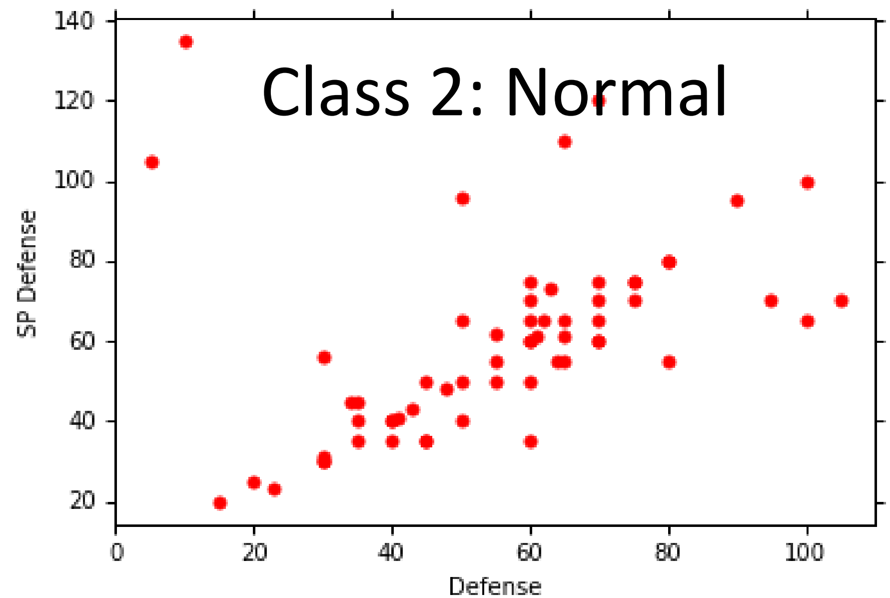




Modifying Model

There are too many parameters in the model. ⇒ Overfitting





$$\mu^1 = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix} \quad \Sigma^1 = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix} \qquad \mu^2 = \begin{bmatrix} 55.6 \\ 59.8 \end{bmatrix} \quad \Sigma^2 = \begin{bmatrix} 847 & 422 \\ 422 & 685 \end{bmatrix}$$

#parameters: n

n²

⇓

The same Σ

We can share the covariance matrix. Less parameters

Modifying Model

Ref: Bishop, chapter 4.2.2

- Maximum likelihood

L(μ¹, Σ¹)

"Water" type Pokémons:

$x^1, x^2, x^3, \ldots\ldots, x^{79}$

L(μ², Σ²)

"Normal" type Pokémons:

$x^{80}, x^{81}, x^{82}, \ldots\ldots, x^{140}$

μ^1 $\quad \Sigma \quad$ μ^2

Find μ^1, μ^2, Σ maximizing the likelihood $L(\mu^1, \mu^2, \Sigma)$

$$L(\mu^1, \mu^2, \Sigma) = f_{\mu^1,\Sigma}(x^1) f_{\mu^1,\Sigma}(x^2) \cdots f_{\mu^1,\Sigma}(x^{79})$$
$$\times f_{\mu^2,\Sigma}(x^{80}) f_{\mu^2,\Sigma}(x^{81}) \cdots f_{\mu^2,\Sigma}(x^{140})$$

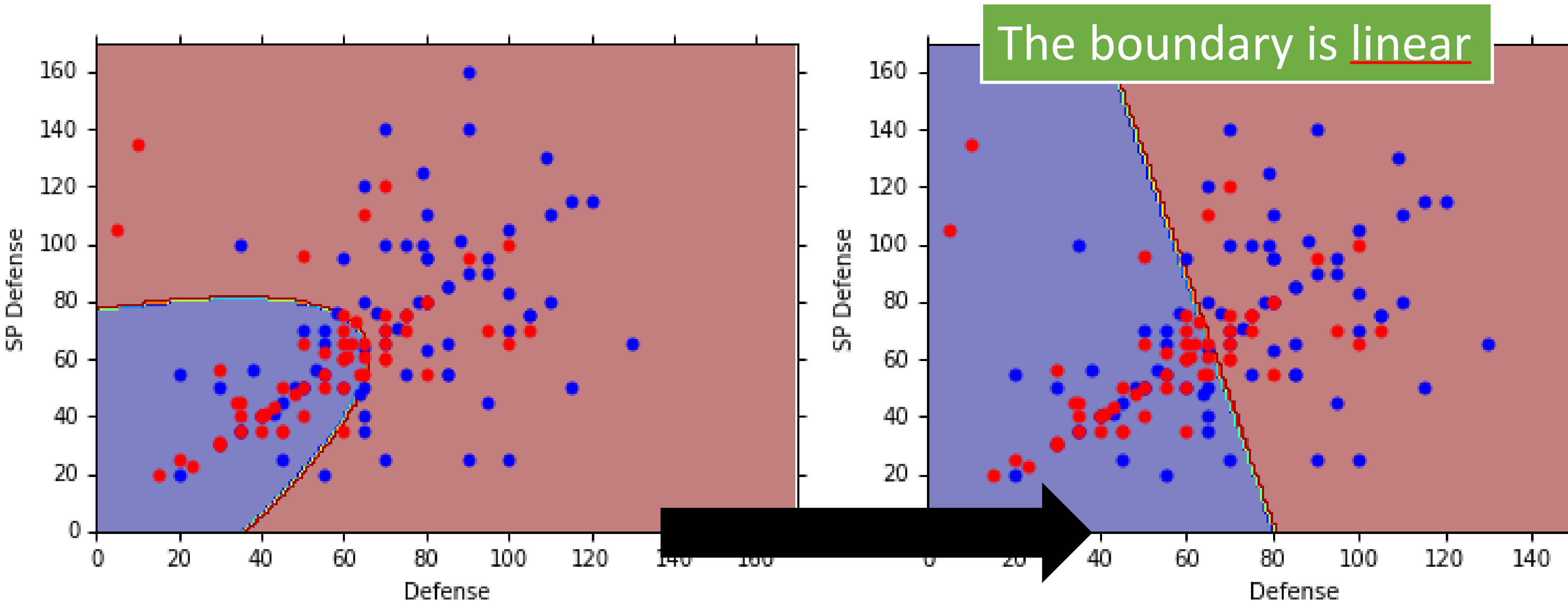
Compare to the original μ, Σ.

μ^1 and μ^2 is the same $\qquad \Sigma = \frac{79}{140}\Sigma^1 + \frac{61}{140}\Sigma^2$

Modifying Model

Only use Defense and SP Defense:

∵ ≡ $\sigma(wx+b)$



The same covariance matrix

All: hp, att, sp att, de, sp de, speed

54% accuracy → 73% accuracy

Three Steps

- Function Set (Model):

$x \rightarrow$

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

If $P(C_1|x) > 0.5$, output: class 1

Otherwise, output: class 2

- Goodness of a function: Distribution's likelihood
  - The mean μ and covariance Σ that maximizing the likelihood (the probability of generating data)
- Find the best function: easy MLE

Probability Distribution

instead of Gaussian distribution

- You can always use the distribution you like ☺

We can assume that the distribution of each feature is independent.

$$P(x|C_1) = P(x_1|C_1)\ P(x_2|C_1)\ \cdots\cdots\ P(x_k|C_1)\ \cdots\cdots$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_K \end{bmatrix}$$

1-D Gaussian

The covariance matrix will be diagonal.

ex: Is it a legendary Pokémon?

For binary features, you may assume they are from Bernoulli distributions.

It's very trivial that we can't choose Gaussian distribution for binary features.

If you assume all the dimensions are independent, then you are using *Naive Bayes Classifier*.

Posterior Probability

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

After some mathematical derivations, we can find that we can implement this in discriminative fashion. (We can then use gradient descent.)

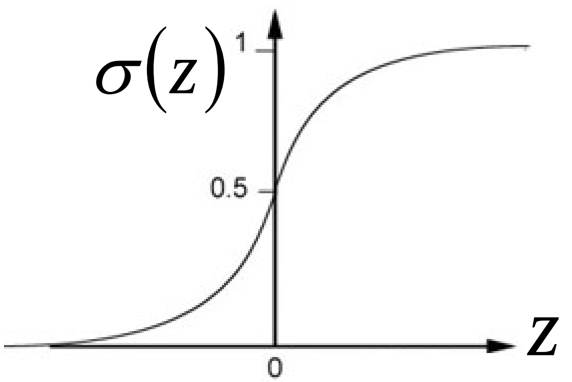
GMM with shared covariance matrix

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

$$= \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}} = \frac{1}{1 + exp(-z)} = \sigma(\overset{wx+b}{\overset{\|}{z}})$$

Sigmoid function

$$\boxed{z = ln\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}}$$



Warning of Math

Prove that $z = wx + b$

Posterior Probability

$$P(C_1|x) = \sigma(z)$$ sigmoid $$z = ln\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$

$$z = ln\frac{P(x|C_1)}{P(x|C_2)} + ln\boxed{\frac{P(C_1)}{P(C_2)}} \Rightarrow \frac{\frac{N_1}{N_1+N_2}}{\frac{N_2}{N_1+N_2}} = \frac{N_1}{N_2}$$

#class 1's occurrences

#class 2's occurrences

$$P(x|C_1) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^1|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}$$

$$P(x|C_2) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^2|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}$$

$$z = \boxed{ln\frac{P(x|C_1)}{P(x|C_2)}} + ln\boxed{\boxed{\frac{P(C_1)}{P(C_2)}} = \frac{N_1}{N_2}}$$

$$P(x|C_1) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^1|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}$$

$$P(x|C_2) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^2|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}$$

$$ln\frac{\cancel{\frac{1}{(2\pi)^{D/2}}}\boxed{\frac{1}{|\Sigma^1|^{1/2}}}\boxed{exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}}}{\cancel{\frac{1}{(2\pi)^{D/2}}}\boxed{\frac{1}{|\Sigma^2|^{1/2}}}\boxed{exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}}}$$

$$= \boxed{ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}}}\boxed{exp\left\{-\frac{1}{2}[(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1) - (x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)]\right\}}$$

$$= ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}} - \frac{1}{2}[(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1) - (x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)]$$

$$z = \boxed{ln\frac{P(x|C_1)}{P(x|C_2)}} + ln\boxed{\frac{P(C_1)}{P(C_2)}} = \frac{N_1}{N_2}$$

$$= ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}} - \frac{1}{2}[\underbrace{(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)}_{①} - \underbrace{(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)}_{②}]$$

①: $(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)$

$$= x^T(\Sigma^1)^{-1}x - x^T(\Sigma^1)^{-1}\mu^1 - (\mu^1)^T(\Sigma^1)^{-1}x + (\mu^1)^T(\Sigma^1)^{-1}\mu^1$$

$$= x^T(\Sigma^1)^{-1}x - 2(\mu^1)^T(\Sigma^1)^{-1}x + (\mu^1)^T(\Sigma^1)^{-1}\mu^1$$

②: $(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)$

$$= x^T(\Sigma^2)^{-1}x - 2(\mu^2)^T(\Sigma^2)^{-1}x + (\mu^2)^T(\Sigma^2)^{-1}\mu^2$$

$$\boxed{\begin{aligned} z = ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}} &- \frac{1}{2}x^T(\Sigma^1)^{-1}x + (\mu^1)^T(\Sigma^1)^{-1}x - \frac{1}{2}(\mu^1)^T(\Sigma^1)^{-1}\mu^1 \\ &+ \frac{1}{2}x^T(\Sigma^2)^{-1}x - (\mu^2)^T(\Sigma^2)^{-1}x + \frac{1}{2}(\mu^2)^T(\Sigma^2)^{-1}\mu^2 + ln\frac{N_1}{N_2} \end{aligned}}$$

End of Warning

$$P(C_1|x) = \sigma(z)$$

$$\boxed{\begin{aligned} z = {} & \cancel{ln \frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}}} \cancel{- \frac{1}{2} x^T (\Sigma^1)^{-1} x} + (\mu^1)^T (\Sigma^1)^{-1} x - \frac{1}{2} (\mu^1)^T (\Sigma^1)^{-1} \mu^1 \\ & \cancel{+ \frac{1}{2} x^T (\Sigma^2)^{-1} x} - (\mu^2)^T (\Sigma^2)^{-1} x + \frac{1}{2} (\mu^2)^T (\Sigma^2)^{-1} \mu^2 + ln \frac{N_1}{N_2} \end{aligned}}$$

$\Sigma_1 = \Sigma_2 = \Sigma$ Assume that the covariance matrices are the same.

$$z = \underbrace{(\mu^1 - \mu^2)^T \Sigma^{-1}}_{\boldsymbol{w}^T \text{ vector}} x \underbrace{- \frac{1}{2} (\mu^1)^T \Sigma^{-1} \mu^1 + \frac{1}{2} (\mu^2)^T \Sigma^{-1} \mu^2 + ln \frac{N_1}{N_2}}_{\text{b scalar}}$$

Discriminative model
⇒ Use gradient descent

$$\boxed{P(C_1|x) = \sigma(w \cdot x + b)}$$

How about directly find ***w*** and b?

This is why the boundary is linear for shared covariance matrix.

In generative model, we estimate $N_1, N_2, \mu^1, \mu^2, \Sigma$

We may obtain better result with different w, b.

Then we have ***w*** and b

Reference

- Bishop: Chapter 4.1 – 4.2
- Data: https://www.kaggle.com/abcsds/pokemon
- Useful posts:
  - https://www.kaggle.com/nishantbhadauria/d/abcsds/pokemon/pokemon-speed-attack-hp-defense-analysis-by-type
  - https://www.kaggle.com/nikos90/d/abcsds/pokemon/mastering-pokebars/discussion
  - https://www.kaggle.com/ndrewgele/d/abcsds/pokemon/visualizing-pok-mon-stats-with-seaborn/discussion