Classification

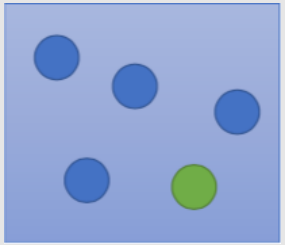
Agenda

- Probabilistic Generative Model

Bayes Methods

Box 1

$P(B_1) = 2/3$

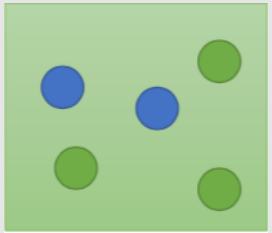


$P(Blue|B_1) = 4/5$

$P(Green|B_1) = 1/5$

Box 2

$P(B_2) = 1/3$



$P(Blue|B_1) = 2/5$

$P(Green|B_1) = 3/5$

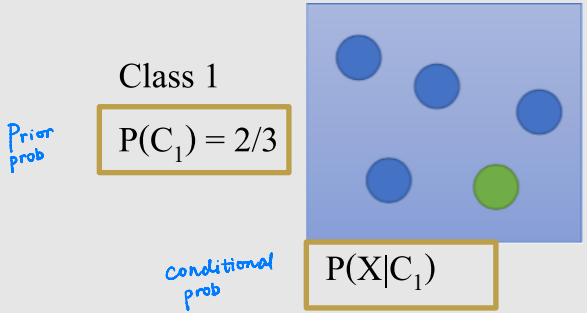
Given a blue ball, where does it come from?

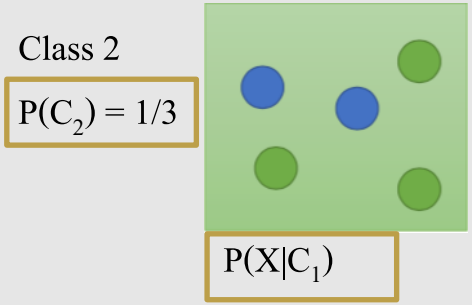
$$P(B_1|Blue) = \frac{P(B_1 \cap Blue)}{P(Blue)}$$

$$= \frac{P(Blue|B_1)P(B_1)}{P(Blue|B_1)P(B_1) + P(Blue|B_2)P(B_2)}$$

Bayes Methods

Estimate the probabilities from the training data





Given a observed x, which class does it belong to?

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

label (class) color

Example:


POKÉMON TYPE SYMBOLS
NORMAL
FIRE
WATER
ELECTRIC
GRASS
ICE
FIGHTING
POISON
GROUND
FLYING
PSYCHIC
BUG
ROCK
GHOST
DRAGON
DARK
STEEL
FAIRY
FALKE
falke2009.deviantart.com

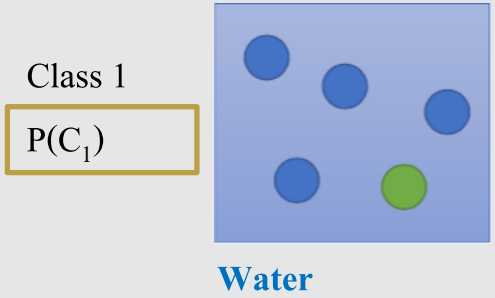
Example

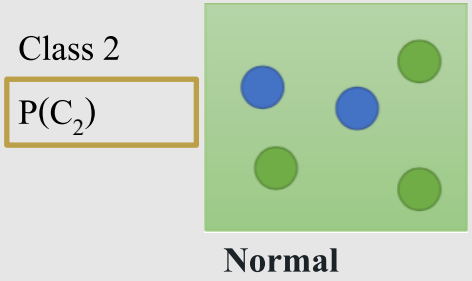
Can we predict the type of a pokemon based on the following features?

- Total: sum of all stats, a general guide to how strong
- HP: hit points/health, how much damage the Pokémon can withstand before fainting
- Attack: the base modifier for normal attacks (eg. Scratch, Punch)
- Defense: the base damage resistance against normal attacks
- SP Atk: special attack, the base modifier for special attacks
- SP Def: the base damage resistance against special attacks
- Speed: determines which Pokémon attacks first each round

Prior

Prair prob





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

Training: 79 Water, 61 Normal

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

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

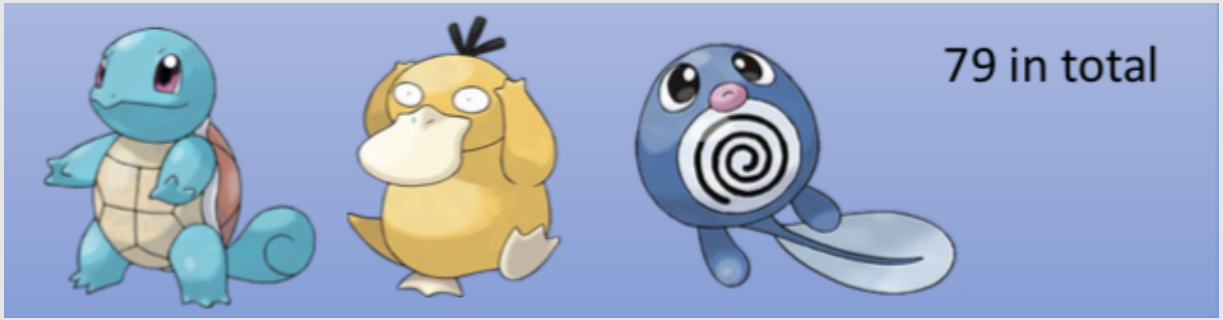
Probability from different classes

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

Conditional prob

How do we estimate $P(x|C_1)$? P( |water)=?

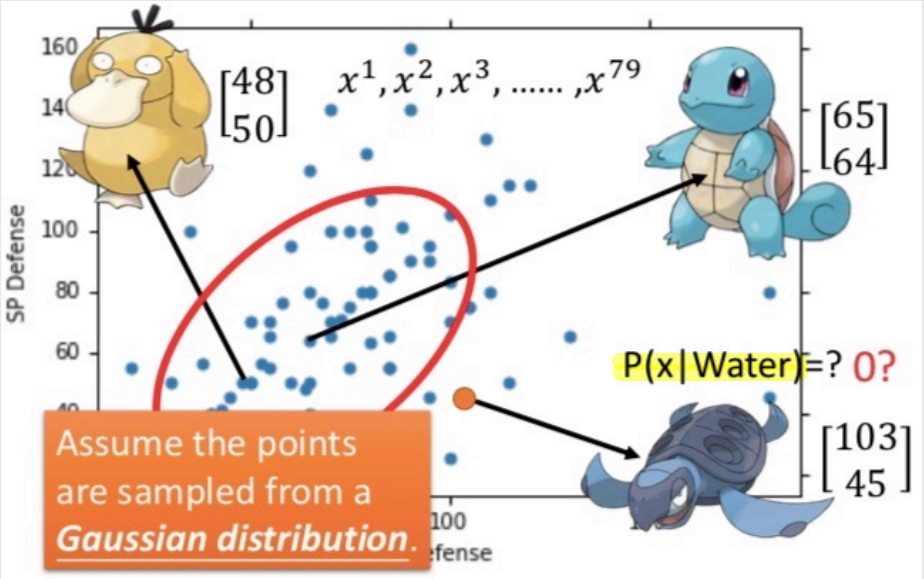
x is the feature vector of the Pokémon.



Probability from different classes

C1 water: P(x|water) → Gaussian dist $N(\mu_1, \Sigma_1)$
C2 normal: P(x|normal) → $N(\mu_2, \Sigma_2)$

Consider two features **Defense** and **SP Defense.**



Multivariate Gaussian distribution

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}_k^{-1}(x-\mu_k)}$$

The shape of the function determines by **mean μ** and **covariance matrix Σ**

For the binary classification example, k=1, 2

https://blog.slinuxer.com/tag/pca

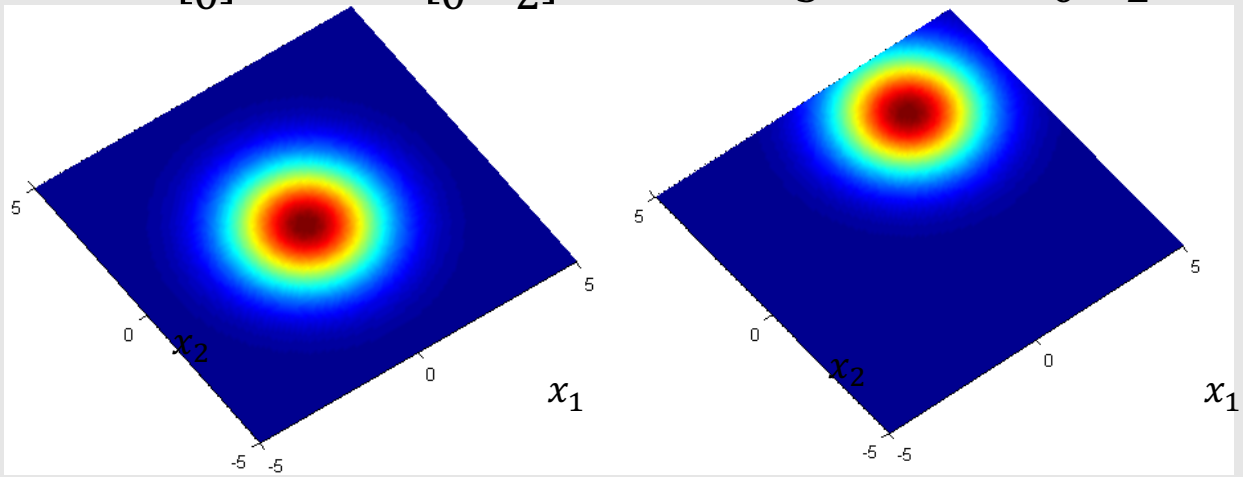
Gaussian Distribution

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

Input: vector x, output: probability of sampling x

The shape of the function determines by **mean μ** and **covariance matrix Σ**

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \qquad \mu = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$



https://blog.slinuxer.com/tag/pca

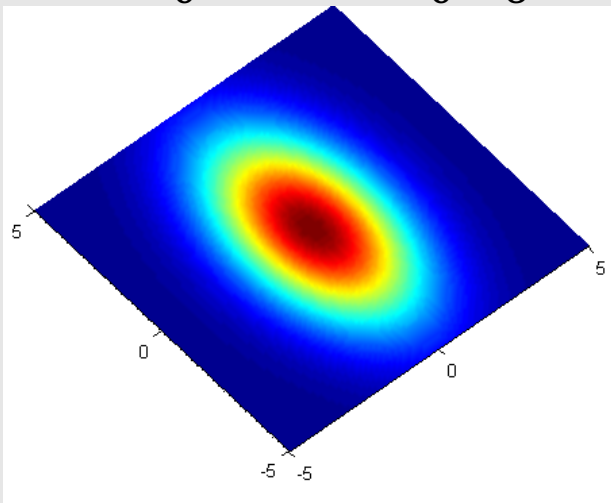
Gaussian Distribution

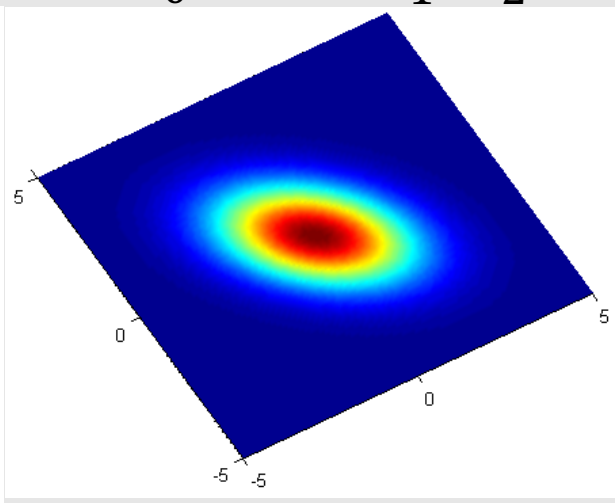
$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}$$

Input: vector x, output: probability of sampling x

The shape of the function determines by **mean μ** and **covariance matrix Σ**

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

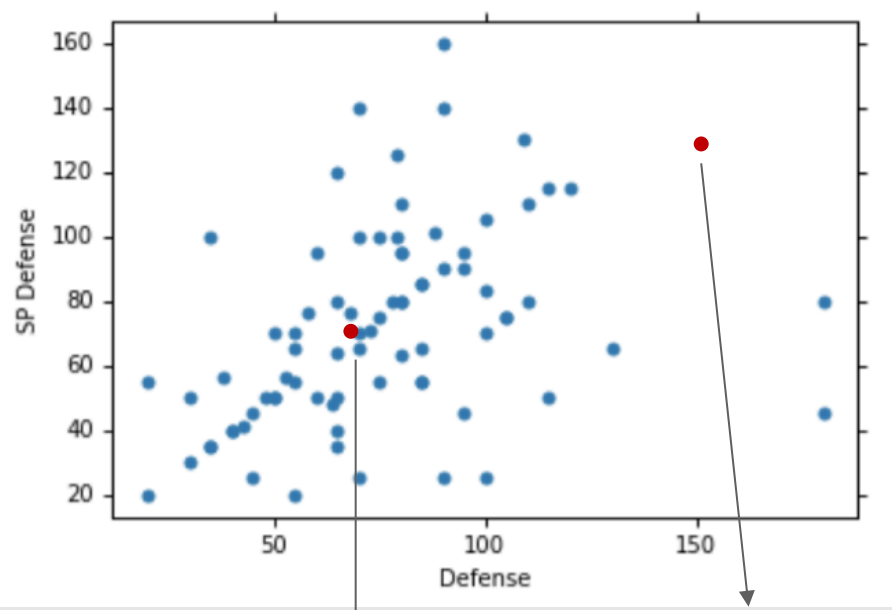




$x \mid \text{water} \sim N(\mu_1, \Sigma_1)$
$x \mid \text{Normal} \sim N(\mu_2, \Sigma_2)$
How to learn $\mu_1, \mu_2, \Sigma_1, \Sigma_2$

Probability from different classes

Water pokemons : $x^1, x^2, \ldots . x^{79}$



$\mu = (75,75)'$
$\Sigma = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$

$\mu = (150,140)'$
$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

Assume the points are sampled from a gaussian distribution

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}_k^{-1}(x-\mu_k)}$$

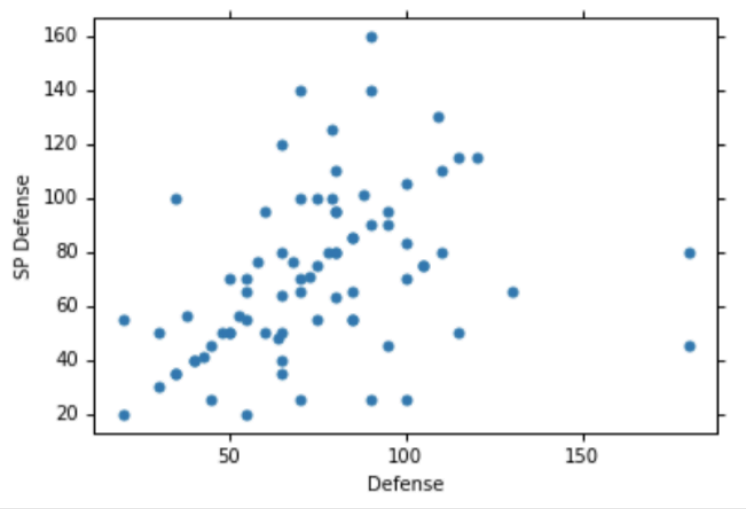
For each class we need to estimate the parameters: μ_k, Σ_k

Maximum Likelihood estimation

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

Probability from different classes

Water pokemons : $x^1, x^2, \ldots . x^{79}$



Maximum Likelihood estimation

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

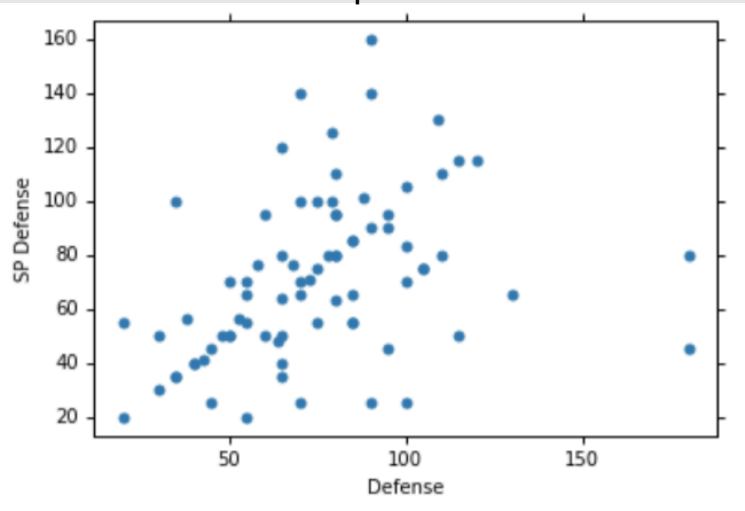
$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}$$

$$\mu^* = \frac{1}{79}\sum_{i=1}^{79} x^i$$

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

Probability from different classes

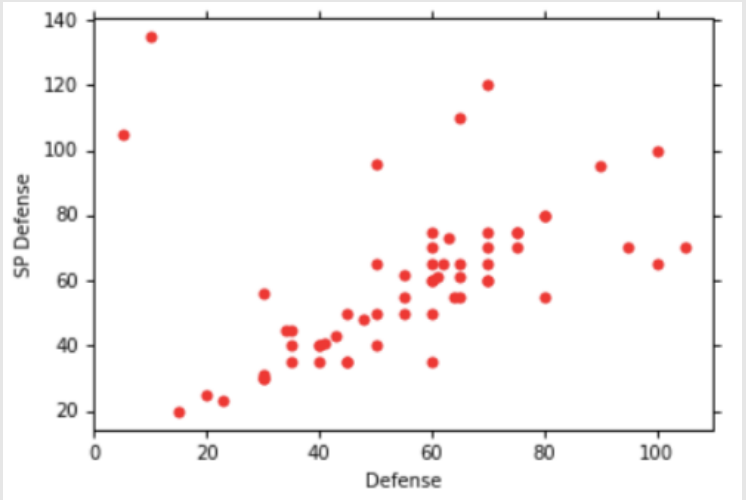
Water pokemon



$$\mu_1 = (75.0, 71.3)^T$$

$$\Sigma_1 = \begin{pmatrix} 874 & 327 \\ 327 & 929 \end{pmatrix}$$

Normal pokemon



$$\mu_2 = (55.6, 59.8)^T$$

$$\Sigma_2 = \begin{pmatrix} 847 & 422 \\ 422 & 685 \end{pmatrix}$$

Classification

$$P(x|C_1) = \frac{1}{(2\pi)^{d/2}|\Sigma_1|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right\} \quad P(C_1) = 79/(79+61) = 0.56$$

Classification based on: $P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1)+P(x|C_2)P(C_2)}$

$$P(x|C_2) = \frac{1}{(2\pi)^{d/2}|\Sigma_2|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right\} \quad P(C_2) = 61/(79+61) = 0.44$$

If $P(C_1|x) > 0.5$, then the observation belongs to class 1

Classification

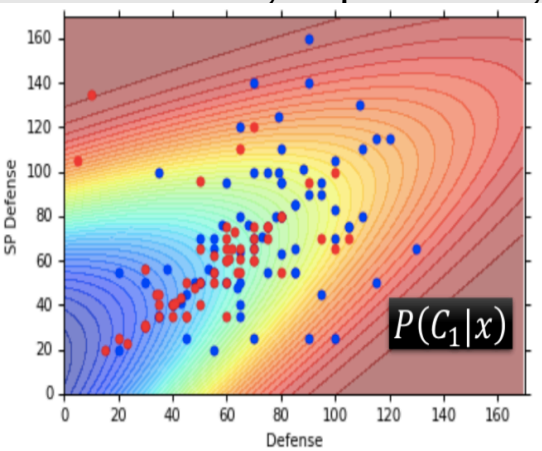
Classification based on: $P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1)+P(x|C_2)P(C_2)}$

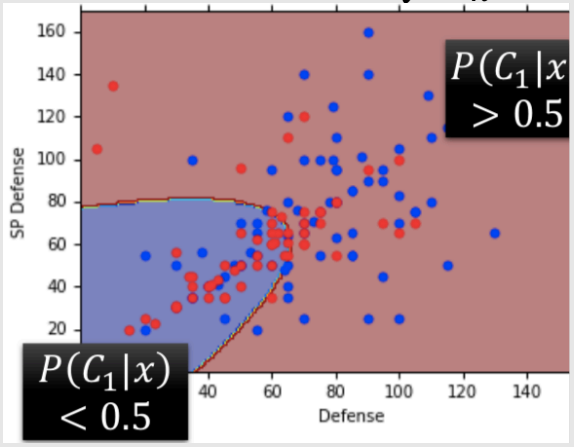
Assign the observation which maximize $P(C_1|x)$ or equivalently

$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + logp_k$ Quadratic Function

Where $p_k = P(C_k)$

Two classes by separated by the decision boundary: $\delta_k(x) = \delta_l(x)$





Quadratic Discriminant Analysis

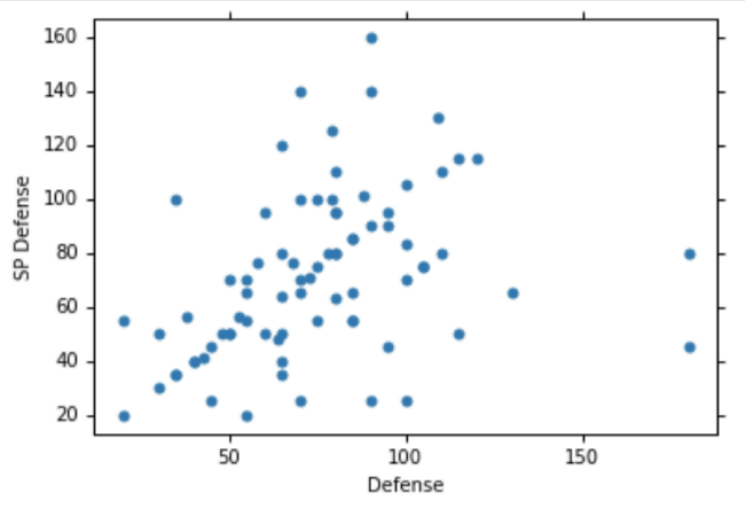
Assume $x \mid \text{Water}$, $x \mid \text{Normal}$ share same variance

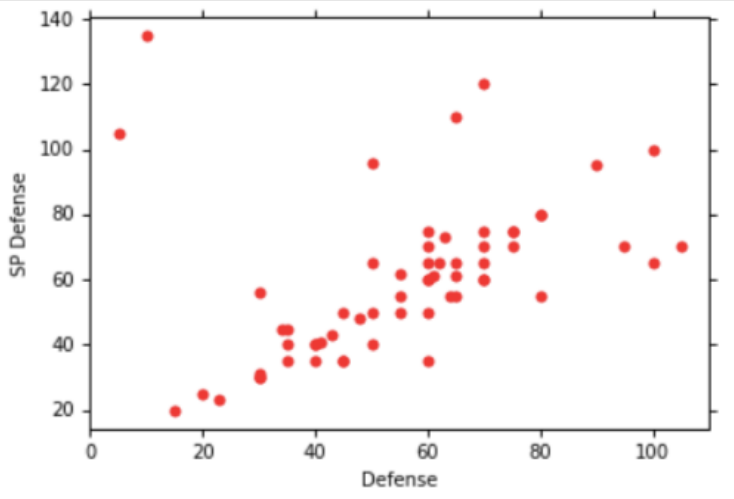
$x \mid \text{Water} \sim N(\mu_1, \Sigma)$

$x \mid \text{Normal} \sim N(\mu_2, \Sigma)$

parameters: μ_1, μ_2, Σ

Modify the model





$$\mu_1 = (75.0, 71.3)^T$$

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

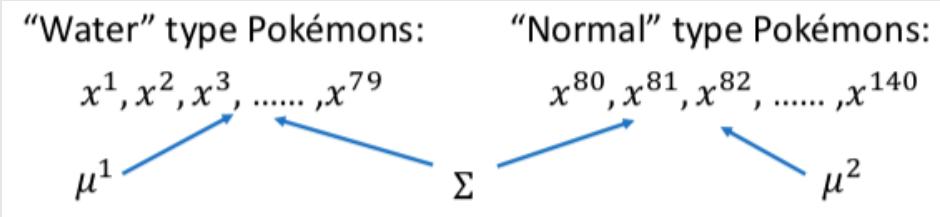
The Same Σ

$$\mu_2 = (55.6, 59.8)^T$$

$$\Sigma = \begin{pmatrix} 847 & 422 \\ 422 & 685 \end{pmatrix}$$

Modify the model

Maximum likelihood to estimate μ^1, μ^2 and Σ



Classification based on: $P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1)+P(x|C_2)P(C_2)}$

Assign the observation which maximize $P(C_1|x)$ or equivalently

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log p_k$$

A Linear Function of x

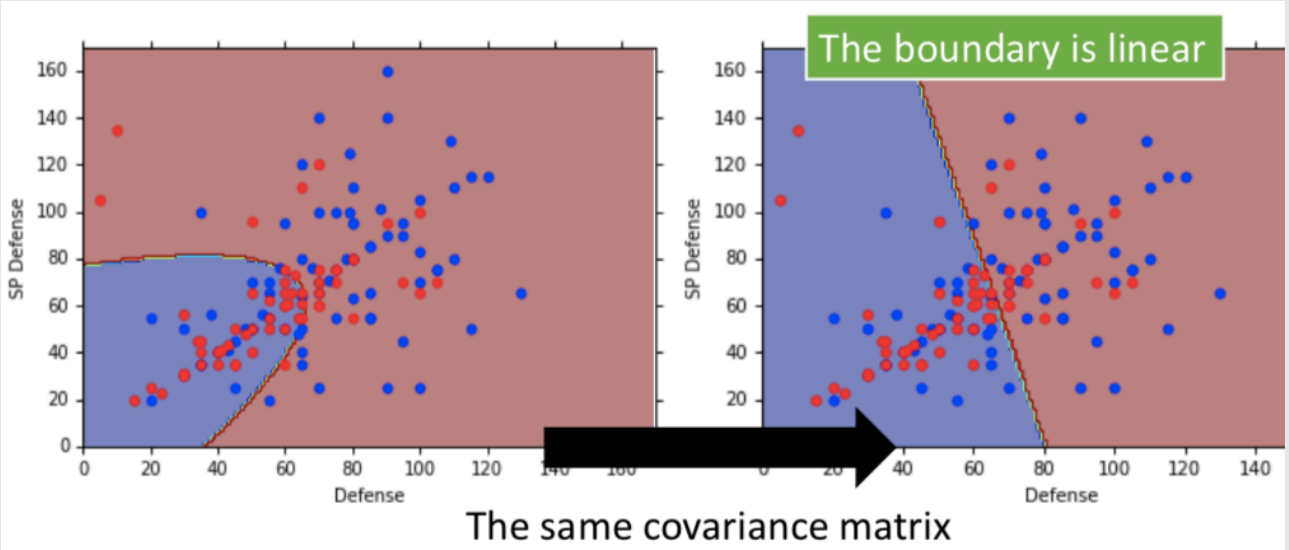
Where $p_k = P(C_k)$

Classification

Two classes by separated by the decision boundary: $\delta_k(x) = \delta_l(x)$

Linear boundary

Linear Discriminant Analysis



Three Steps

- Function Set (Model):

x ⇨

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

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

Otherwise, output: class 2

- Goodness of a function:

  The mean μ and covariance Σ that maximizing the likelihood (the probability of generating data)

- Find the best function: easy

Probability Distribution

You can use the distribution that you like

$$P(x|C_1) = P(x_1|C_1)P(x_2|C_1, x_1) \dots P(x_p|C_1, x_1 \dots x_{p-1})$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}$$

1-D Gaussian

For binary features, you may assume they are from Bernoulli distributions.
If you assumes that all dimensions are independent, then you are using Naïve Bayes Classifier.

Naive Bayes

In addition we assume different predictors are independent ~ Naive Bayes Assumption

$$P(x|C_1) = P(x_1|C_1)P(x_2|C_1\ ) \dots P(x_p|C_1)$$

then Σ is a diagonal matrix, this is equivalent to Naïve Bayes Classifier.

With normal distribution over the features, Naïve Bayes Classifier is a special case of QDA with diagonal Σ.

- Useful when p is large.
- Can be used for mixed feature vectors (qualitative and quantitative). If X is qualitative, replace f(x) with probability mass function over discrete categories.

Naive Bayes

Classify the email as spam or non-spam.

Only consider the words from a predefined dictionary/vocabulary set.
We can present each email in terms of a feature vector whose length is equal to size of the dictionary

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

Naive Bayes

Naïve Bayes classifier assumes that words are independent from each other given y.

$$p(x_1, \dots x_{5000}|y) = p(x_1|y)p(x_2|y) \dots p(x_{5000}|y)$$

Make classification based on

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

The model parameters that we need to learn:

$$p_1 = P(y = 1) \quad \text{Prior Prob}$$

$$\phi_j = P(x_j = 1|y = 1) \quad \text{Conditional Prob} \quad x|y=1$$

$$\theta_j = P(x_j = 1|y = 0) \quad \text{Conditional Prob} \quad x|y=0$$

Naive Bayes

train data $\{(\vec{x}_i, y_i)\}_{i=1}^{N}$

$\vec{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix}$ D → size of your vocab

Use MLE to estimate the parameters

$$L(p_1, \phi_j, \theta_j) = \prod_{i=1}^{n} p(x_i, y_i)$$

$$L(p_1,\phi,\theta) = \prod_{i=1} P(x_i, y_i) = \prod_{i=1}^{n} P(y_i)\cdot P(x_i|y_i)$$

$$= \prod_{i=1} \left[p_1 \prod_{j=1}^{D} \phi_j^{x_{ij}} (1-\phi_j)^{1-x_{ij}} \right]^{y_i} \cdot \left[(1-p_1) \prod_{j=1}^{D} \theta_j^{x_{ij}} (1-\theta_j)^{1-x_{ij}} \right]^{1-y_i}$$

$$P(x_i, y_i) = P(y_i)\cdot P(x_i|y_i)$$

$$= \begin{cases} p_1 \prod_{j=1}^{D} \phi_j^{x_{ij}} (1-\phi_j)^{1-x_{ij}} & \text{if } y_i = 1 \\ p_2 \prod_{j=1}^{D} \theta_j^{x_{ij}} (1-\theta_j)^{1-x_{ij}} & \text{if } y_i = 0 \end{cases}$$

Log-likelihood

$$\ell(p_1,\phi,\theta) = \sum_i y_i \log p_1 + \sum_i y_i \left[\sum_{j=1}^{D} (x_{ij} \log \phi_j + (1-x_{ij}) \log(1-\phi_j)) \right]$$

$$+ \sum_i (1-y_i) \log(1-p_1) + \sum_i (1-y_i) \left[\sum_j (x_{ij} \log \theta_j + (1-x_{ij}) \log(1-\theta_j)) \right]$$

Naïve Bayes

MLE for the parameters

$$p_1 = \frac{\sum_{i=1}^{n} I(y_i=1)}{n}$$

$$\phi_j = \frac{\sum_{i=1}^{n} I(y_i=1 \; and \; x_i=1)}{\sum_{i=1}^{n} I(y_i=1)}$$

$$\theta_j = \frac{\sum_{i=1}^{n} I(y_i=0 \; and \; x_i=1)}{\sum_{i=1}^{n} I(y_i=0)}$$

Then we based on the following posterior probability to make prediction

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

Discriminative model vs. generative model

- Discriminative model: Map directly from feature vector X to the labels {0,1}. Try to predict $p(y|x)$ directly
- Generative model: learns joint $p(x|y)$ or the joint probability $p(x, y)$
    - Make decision based on Bayes rules:

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

Gaussian Discriminant Analysis

Model assumption:

$$Y \sim Bernoulli(p)$$
$$X|Y = 0 \sim N(\mu_0, \Sigma_0)$$
$$X|Y = 1 \sim N(\mu_1, \Sigma_1)$$

Under the assumption,

$$P(y) = p^y (1-p)^{1-y}$$

$$P(x|Y=0) = \frac{1}{(2\pi)^{d/2}|\Sigma_0|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_0)^T \Sigma_0^{-1}(x-\mu_0)\right\}$$

$$P(x|Y=1) = \frac{1}{(2\pi)^{d/2}|\Sigma_1|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1)\right\}$$

Gaussian Discriminant Analysis

Parameters $(p, \mu_0, \Sigma_0, \mu_1, \Sigma_1)$

Log-likelihood

$$L(p, \mu_0, \Sigma_0, \mu_1, \Sigma_1) = log \prod_{i=1}^{N} P\left(x^{(i)}, y^{(i)}; p, \mu_0, \Sigma_0, \mu_1, \Sigma_1\right)$$
$$= log \prod_{i=1}^{N} P\left(x^{(i)} \middle| y^{(i)}; p, \mu_0, \Sigma_0, \mu_1, \Sigma_1\right) P\left(y^{(i)}; p\right)$$

MLE of the parameters

$$p = \frac{1}{N} \sum_{i=1}^{N} I\left(y^{(i)} = 1\right)$$

$$\mu_0 = \frac{\sum_{i \in C_0} x^{(i)}}{|C_0|}, \Sigma_0 = \frac{1}{|C_0|} \sum_{i \in C_0} \left(x^{(i)} - \mu_0\right) \left(x^{(i)} - \mu_0\right)^T$$

$$\mu_1 = \frac{\sum_{i \in C_1} x^{(i)}}{|C_1|}, \Sigma_1 = \frac{1}{|C_1|} \sum_{i \in C_1} \left(x^{(i)} - \mu_1\right) \left(x^{(i)} - \mu_1\right)^T$$

Gaussian Discriminant Analysis

Quadratic discriminative Analysis:

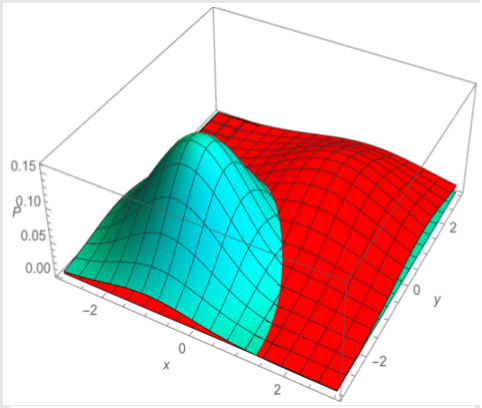
Different Covariance Matrix

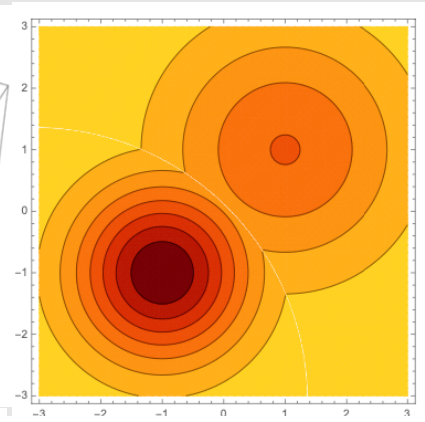
$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log p_k$$

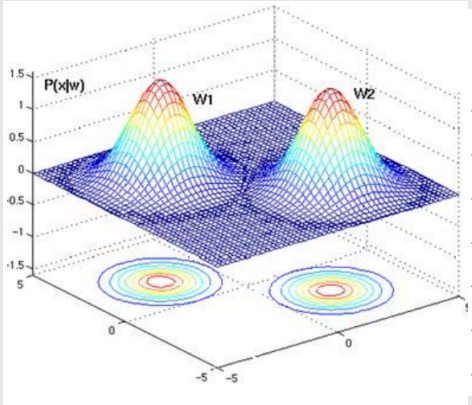
Linear discriminative Analysis

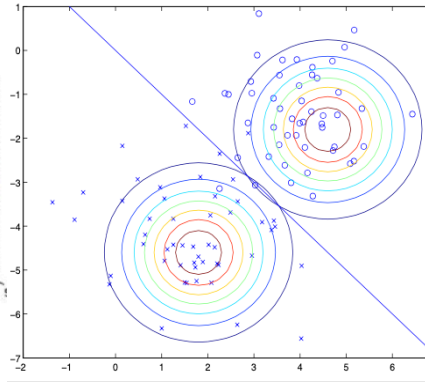
Same Covariance matrix

$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma| + \log p_k$$









Linear discriminant analysis and Logistic regression

The posterior probability:

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

$$= \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}} = \frac{1}{1 + \exp(-z)}.$$

$P(y=a|x) = \frac{1}{1+e^{-z}}$

$z = w^T x$

where $z = ln \frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$ is it a linear function of x

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

Posterior Probability

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

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

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

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

Posterior Probability

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

When we assume equal covariance matrix

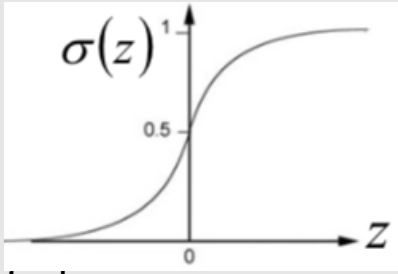
$$\Sigma_1 = \Sigma_2 = \Sigma$$

$$z = (\mu^1 - \mu^2)^T\Sigma^{-1}x - \frac{1}{2}(\mu^1)^T\Sigma^{-1}\mu^1 + \frac{1}{2}(\mu^2)^T\Sigma^{-1}\mu^2 + ln\frac{N_1}{N_2}$$

LDA's format is similar to a logistic regression classifier

Linear discriminant analysis and Logistic regression

We have $P(C_1|x) = \sigma(w^T x + b)$



- LDA has the same form as logistic regression.
- The difference is in how the parameters are estimated.

  Logistic regression uses the conditional likelihood based on P(Y |X) (known as discriminative learning).

  LDA uses the full likelihood based on P(X, Y) (known as generative learning).

- Despite these differences, in practice the results are often very similar.

Footnote: logistic regression can also fit quadratic boundaries like QDA, by explicitly including quadratic terms in the model.

Generative vs. Discriminative

Benefit of generative model

With the assumption of probability distribution,

- Less training data is needed
- With the assumption of probability distribution, more robust to the noise
- Priors can be estimated from different sources.

Reference

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