Artificial Intelligence

Machine learning - Classification

Nguyễn Văn Diêu

Ho Chi Minh City University of Transport

2022

Outline I

Classification

$$\mathcal{D} = \left\{(x_i, y_i)\right\}_1^n, \ \ x_i \in \mathbb{R}^d \quad , \quad y_i : \textbf{discrete}.$$

Binary Classification

$$y_i \in \{0, 1\} \ or \ y_i \in \{-1, 1\}$$

e.g. (E01)

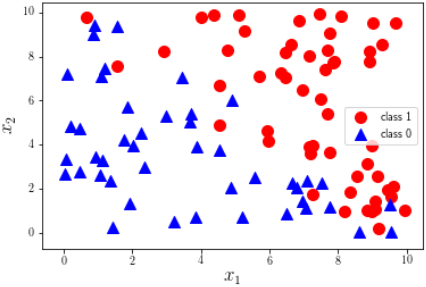
$$\left\{(x_i, y_i)\right\}_1^n, \ x_i \in \mathbb{R}^2, \ y_i \in \{0, 1\}$$

x_1	x_2	y
8.625	0.058	0
3.828	0.723	0
7.150	3.899	1
6.477	8.198	1
1.922	1.331	0

e.g. Binary Classification

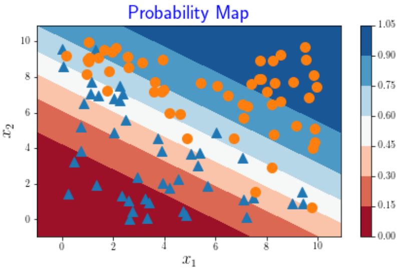
e.g. chart of dataset (E01)

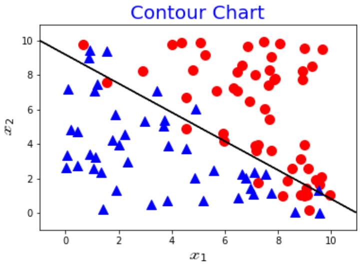
$$\{(x_i, y_i)\}_1^n,\ x_i \in \mathbb{R}^2,\ y_i \in \{0, 1\}$$



e.g. Logistic Regression

- This model predict probability of two class:
- Contour chart base on probability.



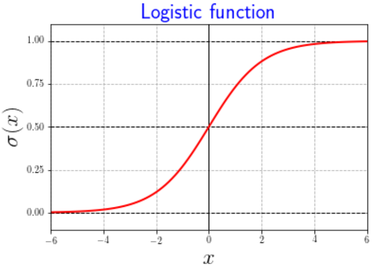


Logistic Regression for Classification

Logistic model (Logit model): Probability of a class label in dataset. **Logistic function**

$$\boxed{\sigma(x) = \frac{1}{1 + e^{-x}}}$$

$x \to +\infty \ , \ \sigma(x) \to 1$
$x \to -\infty \ , \ \sigma(x) \to 0$



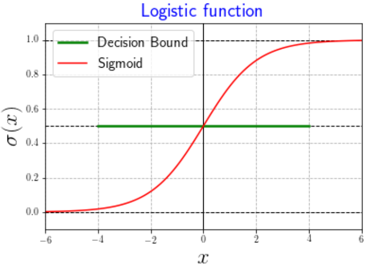
− *Continuous, has a first derivative.*

Probability return

Logistic return probability score between 0 and 1

$$\boxed{Prob = \sigma(x) = \frac{1}{1+e^{-x}}}$$

$$Prob \in (0, 1)$$
$$Prob \geq 0.5\ ,\ class = 1$$
$$Prob < 0.5\ ,\ class = 0$$



Logistic Regression Model

$$\mathcal{D} = \left\{(x_i, y_i)\right\}_1^n, \quad x_i \in \mathbb{R}^d, \quad y_i \in \{0, 1\}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix} \in \mathbb{R}^{d+1} \quad x = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \in \mathbb{R}^{d+1}$$

Model

$$\boxed{f(x) = \sigma(\beta.x) = \frac{1}{1 + e^{-\beta.x}}}$$

dot product $\quad \beta.x \equiv \sum_j \beta_j x_j$

Logistic Regression Model

e.g.

$x = (x_1, x_2)$, $y = \{0, 1\}$. imagine we know β.

$z = \beta.x = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

$$\boxed{Prob(class = 1) = \frac{1}{1 + e^{-z}}}$$

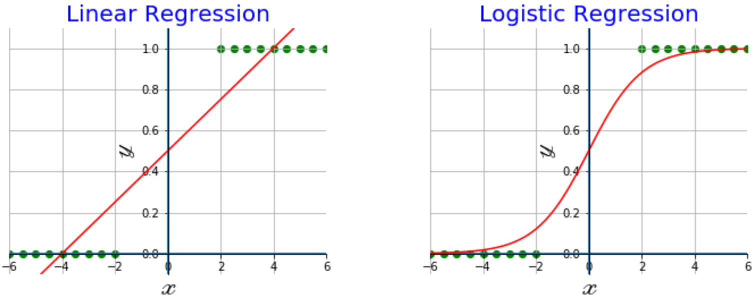
Decision boundary = .5

e.g. $Prob(class = 1)$ *return .4, only have 40% chance "class 1" or this observation as "class 0".*

Linear Regression Vs. Logistic Regression

- Linear Regression: *Continuous output.*
- Logistic Regression: *Constant output.*
- Linear Regression: *Using Ordinary Least Squares (OLS).*
- Logistic Regression: *Using Maximum Likelihood Estimation (MLE).*

Consider dataset which two class $\{0, 1\}$



Logistic Regression Model

Model

$$\boxed{f(x) = \sigma(\beta.x) = \frac{1}{1 + e^{-\beta.x}}}$$

$f(x) \in (0, 1)$
$f(x)$: Probability.

Probability return by Model

$$Prob(y = 1 \mid x; \beta) = f(x)$$

$$Prob(y = 0 \mid x; \beta) = 1 - f(x)$$

Model base on probability

$$\boxed{Prob(y \mid x; \beta) = f(x)^y \; (1 - f(x))^{1-y}}$$

The Likelihood

One sample i

$$Prob(y_i \mid x_i; \beta) = f(x_i)^{y_i} \, (1 - f(x_i))^{1-y_i}$$

Loss function $Loss(\beta)$ for all sample.

n training samples were generated independently.

Likelihood

$$Loss(\beta) = \prod_{i=1}^{n} Prob(y_i \mid x_i; \beta)$$

$$\boxed{Loss(\beta) = \prod_{i=1}^{n} f(x_i)^{y_i} \, (1 - f(x_i))^{1-y_i}}$$

Logarithm of Likelihood

- Logarithm turns a product into a sum.
- It avoid the issue of small number(typically for probability).

$$
\begin{aligned}
\mathcal{L}(\beta) &= log\ Loss(\beta) \\
&= log \prod_{i=1}^{n} f(x_i)^{y_i} \left(1 - f(x_i)\right)^{1-y_i} \\
&= \sum_{i=1}^{n} log \left\{ f(x_i)^{y_i} \left(1 - f(x_i)\right)^{1-y_i} \right\} \\
&= \sum_{i=1}^{n} \left\{ log f(x_i)^{y_i} + log(1 - f(x_i))^{1-y_i} \right\} \\
&= \sum_{i=1}^{n} \left\{ y_i\ log f(x_i) + (1 - y_i)\ log(1 - f(x_i)) \right\}
\end{aligned}
$$

Maximization into a Minimization

Maximize the Likelihood

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \left\{ y_i \, log f(x_i) \, + \, (1 - y_i) \, log(1 - f(x_i)) \right\}$$

– Negative log-likelihood (NLL).
– Use gradient descent algo.

So we minimize the negative $\mathcal{L}(\beta)$ with

$$\boxed{\mathcal{L}(\beta) = -\sum_{i=1}^{n} \left\{ y_i \, log f(x_i) \, + \, (1 - y_i) \, log(1 - f(x_i)) \right\}}$$

Minimize the negative Likelihood

$$\boxed{\mathcal{L}(\beta) = \sum_{i=1}^{n} \left\{ -y_i \; log f(x_i) \; - \; (1 - y_i) \; log(1 - f(x_i)) \right\}}$$

$$\frac{\partial}{\partial \beta_j} \mathcal{L}(\beta) = \sum_{i=1}^{n} \left\{ -y_i \frac{1}{\sigma(\beta.x_i)} + (1 - y_i) \frac{1}{1 - \sigma(\beta.x_i)} \right\} \frac{\partial}{\partial \beta_j} \sigma(\beta.x_i)$$

since $\frac{\partial}{\partial \beta_j} \sigma(\beta.x_i) = \sigma(\beta.x_i)(1 - \sigma(\beta.x_i)) \frac{\partial}{\partial \beta_j} \beta.x_i$, so

$$= \sum_{i=1}^{n} \left\{ \sigma(\beta.x_i) - y_i \right\} \frac{\partial}{\partial \beta_j} \beta.x_i$$

Since $\beta.x_i = \beta_0 + \sum_{j=1}^{d} \beta_j x_{ij} \implies \frac{\partial}{\partial \beta_j} \beta.x_i = x_{ij}$, so

$$= \sum_{i=1}^{n} (f(x_i) - y_i) x_{ij}$$

Gradient Descent

$$\boxed{\frac{\partial}{\partial \beta_j}\mathcal{L}(\beta) = \sum_{i=1}^{n}(f(x_i) - y_i)x_{ij}}$$

η : Learning rate (~ 0.1).
ϵ : Error convergence (~ 0.001).
$Epochs$: The number of times to run through the training data while updating the coefficients (~ 1000).

There are 3 loops in algorithm

1. *Loop over each epoch.*
2. *Loop over each row in the training data for an epoch.*
3. *Loop over each coefficient and update it for a row in an epoch.*

Standard Gradient Descent Algorithm

In every iteration gradients have to be computed all n training examples.

for $k = 1$ to $epocks$

for $j = 0$ to d

$$\beta_j = \beta_j - \eta \sum_{i=1}^{n} (f(x_i) - y_i) x_{ij}$$

if $\| \frac{\partial}{\partial \beta} \mathcal{L}(\beta) \|_2 < \epsilon$ then

return β

return β

$\| \frac{\partial}{\partial \beta} \mathcal{L}(\beta) \|_2$*: $l2$ norm (Euclidean norm) of* $\frac{\partial}{\partial \beta} \mathcal{L}(\beta)$

$$\| \frac{\partial}{\partial \beta} \mathcal{L}(\beta) \|_2 = \sqrt{[\frac{\partial}{\partial \beta_1} \mathcal{L}(\beta)]^2 + [\frac{\partial}{\partial \beta_2} \mathcal{L}(\beta)]^2 + ... + [\frac{\partial}{\partial \beta_d} \mathcal{L}(\beta)]^2}$$

Stochastic Gradient Descent

The gradient is computed a single randomly chosen training example.

for $k = 1$ to $epocks$

$\quad i =$ random index between 1 and n

$\quad$ for $j = 0$ to d

$\qquad \beta_j = \beta_j - \eta\,(f(x_i) - y_i)x_{ij}$

$\quad$ if $\| \frac{\partial}{\partial\beta}\mathcal{L}(\beta) \|_2 < \epsilon$ then

$\qquad$ return β

return β

Mini-Batch Gradient Descent

In each iteration, choosing a batch of random sample from dataset.

z: batch size.

for $k = 1$ *to epocks*

$k_1, k_2, k_3, .., k_z =$ random indices between 1 and n

for $j = 0$ *to* d

$$\beta_j = \beta_j - \eta \sum_{k=1}^{z} (f(x_k) - y_k) x_{kj}$$

if $\| \frac{\partial}{\partial \beta} \mathcal{L}(\beta) \|_2 < \epsilon$ then

return β

return β

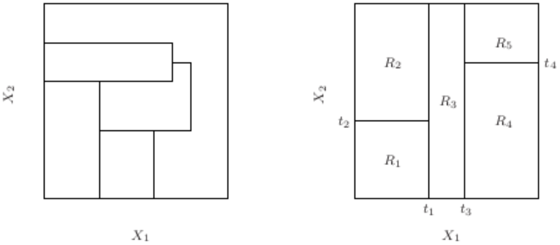
Types of Logistic Regression

- **Binary Logistic Regression** Binary class, e.g. Spam or Not Spam, Cancer or No Cancer.
- **Multinomial Logistic Regression** Many class, e.g. predicting the type of Wine.
- **Ordinal Logistic Regression** Many ordinal class, e.g. restaurant or product rating from 1 to 5.

Decision Tree

Decision Tree

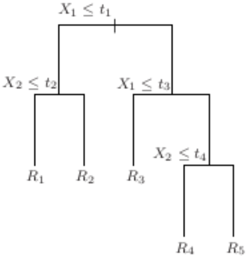
e.g. split X_1 and X_2 to 5 region

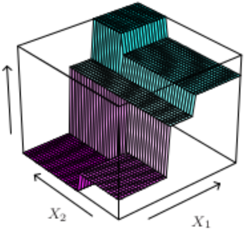


$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{\, (X_1, X_2) \in R_m \,\}$$

Decision Tree

$(X_1 X_2, Y)$





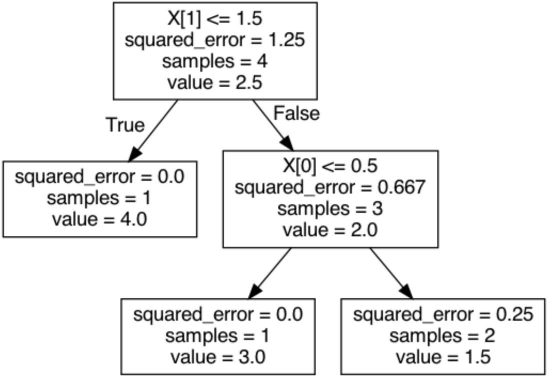
$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{\, (X_1, X_2) \in R_m \,\}$$

Decision Tree

e.g.01 Regression Decision Tree with continuous feature.

$X = \{ (3,\ 1), (1,\ 2), (0,\ 4), (4,\ 3) \}$

$y = \{ 4,\ 2,\ 3,\ 1 \}$



Predict: $(0.2,\ 3.4) \Rightarrow \hat{y} = 3.0$; $(1.3,\ 2.1) \Rightarrow \hat{y} = 1.5$

Decision Tree Model

- Dataset: $\{(x_i, y_i)\}_1^n$; $x_i \in \mathbb{R}^d$; $y_i \in \mathbb{R}$ $(or\ y_i \in\ Category)$
- Partition: M regions $R_1, R_2, ..., R_M$
- Model response as a constant c_m in each region:
- I: Indicator function

$$f(x) = \sum_{m=1}^{M} c_m I\{\ x \in R_m\ \}$$

- Data at node R with n samples
- Candidate split (j, s) : feature j , threshold s

$$R(j,s) \rightarrow R_1(j,s)\ +\ R_2(j,s)$$

Left $\quad R_1(j,s) = \{\ (x,y) \mid x_j \leq s\ \}$

Right $\quad R_2(j,s) = \{\ (x,y) \mid x_j > s\ \}$

Decision Tree algorithm

Loss Function

$$\boxed{L(j,s) = \frac{n_1}{n} L(R_1(j,s)) + \frac{n_2}{n} L(R_2(j,s))}$$

$$\boxed{(j,s)^* = \underset{(j,s)}{argmin}\ L(j,s)}$$

Recurse

$R_1(j,s)$, $R_2(j,s)$

Until maximum allowable depth is reached

Decision Tree use for both classification and regression tasks.

CART for Regression Tree

Loss function base on Mean Squared Error (MSE or L2 error)

$$MSE(.) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

The best $f(x_i)$ is average of y_i on region R

$$\boxed{c_m = ave(y_i \mid x_i \in R_m)}$$

$$L(R_m) = MSE(R_m) = \frac{1}{n}\sum_{y \in R_m}(y - c_m)^2$$

$$L(j,s) = \left\{ \frac{n_1}{n}\frac{1}{n_1}\sum_{y \in R_1}(y - c_1)^2 \ + \ \frac{n_2}{n}\frac{1}{n_2}\sum_{y \in R_2}(y - c_2)^2 \right\}$$

CART for Regression Tree

$$L(j,s) = \frac{1}{n}\left\{ \sum_{y \in R_1} (y - c_1)^2 \; + \; \sum_{y \in R_2} (y - c_2)^2 \right\}$$

$$\boxed{(j,s)^* = \underset{(j,s)}{argmin} \left\{ \sum_{y \in R_1(j,s)} (y - c_1)^2 + \sum_{y \in R_2(j,s)} (y - c_2)^2 \right\}}$$

CART for Regression Tree

CART (Classification And Regression Tree)

CART($R,\ stop!$)

1. $list\ W = \{\ \}$
2. $for\ all\ j:\ \ feature\ x_j$
   - $sort\ Domain\ \{x_j\}$
   - $for\ all\ t_k \in Domain\ \{x_j\}$
     - $choose\ s:\ \ s = \frac{(t_k\ +\ t_{k+1})}{2}$
     - $w(j,s) = \sum\limits_{y \in R_1(j,s)} (y - c_1)^2 + \sum\limits_{y \in R_2(j,s)} (y - c_2)^2$
     - $add\ w(j,s)\ to\ list\ W$
3. $w(j,s)\ =\ min\{W\}$
4. **CART**($R_1(j,s),\ stop!$) , **CART**($R_2(j,s),\ stop!$)

Regression Tree Algorithm

Problem

– How large should we grow the tree?

– Very large tree might overfit the data.

– Small tree might not show the important structure.

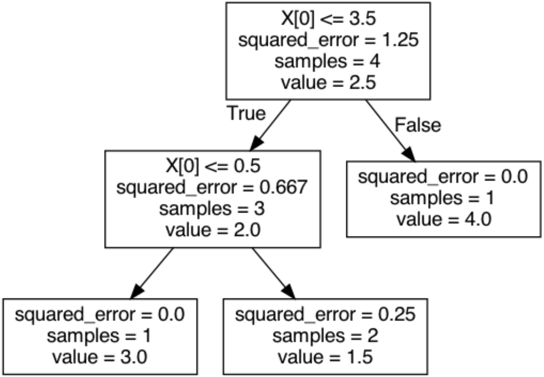
– Optimal tree size, we can choose from the dataset.

– Real domain x !. We have to partition it.

e.g. Regression Tree

e.g.02:

$X = \{\ 3,\ 1,\ 0,\ 4\ \}$
$y = \{\ 2,\ 1,\ 3,\ 4\ \}$



Predict:

$(x = 4.2) \Rightarrow \hat{y} = 4.0$
$(x = 1.3) \Rightarrow \hat{y} = 1.5$

CART Classification Tree

Loss function base on proportion p_k of class k in region R:

$$p_k = \frac{1}{n} \sum_{x_i \in R} I(y_i = k) = \frac{n_k}{n}$$

Metric:

– **Gini index** also called **Gini impurity**

$$Gini(R) = \sum_{p_k \in R} p_k(1 - p_k) = 1 - \sum_{p_k \in R} p_k^2$$

– **Entropy**

$$E(R) = -\sum_{k} p_k log(p_k)$$

CART Classification Tree

Gini index

$$Gini(R) = 1 - \sum_{p_k \in R} p_k^2 \quad ; \quad \{p_k \in R\} = \frac{n_k}{n_R}$$

$$(j,s)^* = \underset{(j,\ s)}{argmin} \left\{ \frac{n_1}{n} Gini(R_1) \ + \ \frac{n_2}{n} Gini(R_2) \right\}$$

$$(j,s)^* = \underset{(j,\ s)}{argmin} \ \frac{1}{n} \left\{ n_1 Gini(R_1) \ + \ n_2 Gini(R_2) \right\}$$

$$\boxed{(j,s)^* = \underset{(j,\ s)}{argmin} \left\{ n_1 Gini(R_1) \ + \ n_2 Gini(R_2) \right\}}$$

CART algorithm (gini)

CART (Classification And Regression Tree)

CART($R,\ stop!$**)**

1. $list\ W = \{\ \}$
2. $for\ all\ j:\ feature\ x_j$
   - $sort\ Domain\ \{x_j\}$
   - $for\ all\ t_k \in Domain\ \{x_j\}$
     - $choose\ s:\ s = \frac{(t_k\ +\ t_{k+1})}{2}$
     - $w(j, s) =\ n_1 Gini(R_1)\ +\ n_2 Gini(R_2)$
     - $add\ w(j, s)\ to\ list\ W$
3. $w(j, s)\ =\ min\{W\}$
4. **CART(**$R_1(j, s),\ stop!$**)** , **CART(**$R_2(j, s),\ stop!$**)**

e.g.03:

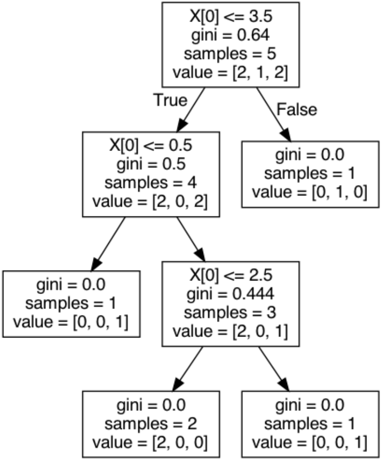
$$X = \{\ 3,\ 1,\ 0,\ 4,\ 2\ \}$$

$$y = \{\ 2,\ 0,\ 2,\ 1, 0\ \}$$

Predict:

$(x = 4.5) \Rightarrow class = 1$

$(x = 1.3) \Rightarrow class = 0$



e.g. Classification Tree

e.g.04:

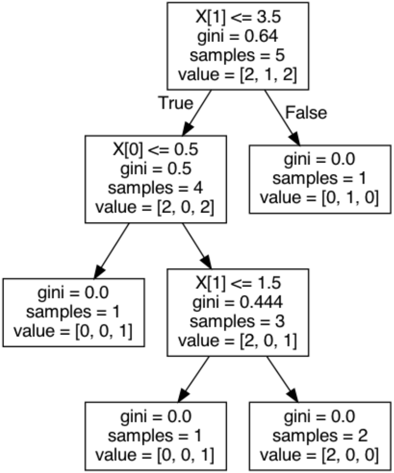
$X = \{(2,1),(1,2),(0,3),(3,4),(4,3)\}$

$y = \{2,0,2,1,0\}$

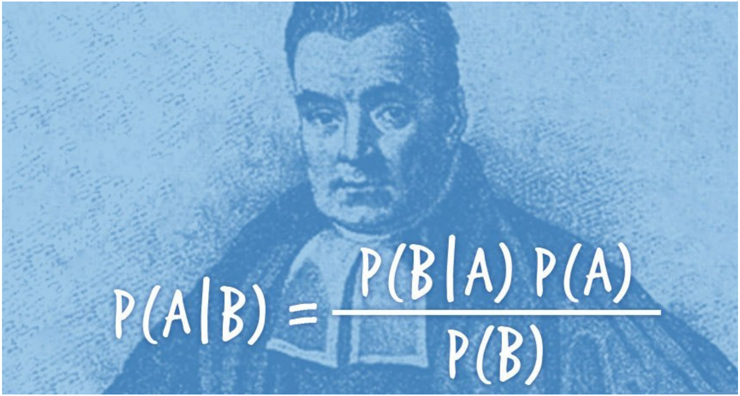
Predict:

$x = (3,4) \Rightarrow class = 1$

$x = (1.5,1.5) \Rightarrow class = 2$



Bayes's Theorem

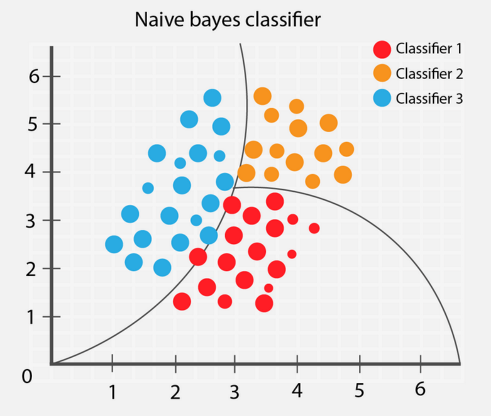


$p(A)$: *Prior*
$p(A|B)$: *Posterior*
$p(B|A)$: *Likelihood*
$p(B)$: *Evidence*

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

$$\boxed{P(class/data) = \frac{P(data/class) \times P(class)}{P(data)}}$$

Naïve Bayes Classifier



$$\boxed{P(class/data) = \frac{P(data/class) \times P(class)}{P(data)}}$$

review: Joint Probability

Tossing two coins: Independent

– A: Means the first coin lands face up
– B: Means the second coin lands face up

- $p(A) = p(B) = 0.5$
- $p(A \text{ } and \text{ } B) = p(A)\, p(B) = 0.25$
- $p(B|A) = p(B)$

Events are not independent

– A: Mean it rains today
– B: Means it rains tomorrow

- It rained today, it more likely rain tomorrow
- $p(B|A) > p(B)$
- $p(A \text{ } and \text{ } B) = p(A)\, p(B|A)$

e.g. Cookie problem

– *Suppose there are two bowls of cookies*
  + Bowl 1:
    30 vanilla
    10 chocolate
  + Bowl 2:
    20 vanilla
    20 chocolate

– *Now suppose you choose*
  + One of the bowls at random
  + Without looking, select a cookie at random

This is a conditional probability

$$p(Bowl\ 1 \mid vanilla)$$

$p(vanila \mid Bowl\ 1) = 3/4$
$\neq p(Bowl\ 1 \mid vanilla)$

Bayes's Theorem

Any events A and B

$- \; p(A \; and \; B) = p(B \; and \; A)$

$- \; p(A \; and \; B) = p(A) \; p(B|A)$

$- \; p(B \; and \; A) = p(B) \; p(A|B)$

$\implies \; p(B) \; p(A|B) = p(A) \; p(B|A)$

Bayes's Theorem

$$\boxed{p(A|B) = \frac{p(B|A) \; p(A)}{p(B)}}$$

Cookie problem

+ B_1: Hypothesis of cookie came from Bowl 1
+ V: Vanilla cookie

$$p(B_1|V) = \frac{p(V|B_1) \; p(B_1)}{p(V)}$$

e.g. Cookie problem

$$p(B1|V) = \frac{p(V|B1)\ p(B1)}{p(V)}$$

$p(B1)$: Probability chose Bowl 1

$$p(B1) = 1/2$$

$p(V|B1)$: Probability vanilla cookie from Bowl 1

$$p(V|B1) = 3/4$$

$p(V)$: Probability vanilla cookie from either bowl

$$p(V) = 5/8$$

$$p(B1|V) = \frac{(3/4)\ (1/2)}{(5/8)} = 3/5$$

e.g. Elderly Fall and Death

- Elderly person is died: 10%
- Elderly people falling: 5%
- All elderly people die, they had fall: 7%

Probability that elderly people die when they fall?

$$P(Die|Fall) = \frac{P(Fall|Die) \times P(Die)}{P(Fall)}$$

$P(Die) = 0.10$
$P(Fall) = 0.05$
$P(Fall|Die) = 0.07$

$$P(Die|Fall) = \frac{0.07 \times 0.10}{0.05}$$

$$P(Die|Fall) \mathbf{= 0.14}$$

- If an elderly person falls
- There is a 14% probability that they will die from the fall

e.g. Email and Spam Detection

- Email receive is spam: 2%
- Spam detector accuracy: 99%
- When an email is not spam, it will mark it as spam: 0.1%

Probability that fact spam email in spam folder?

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

$$P(B) = P(B|A) \times P(A) \ + \ P(B|not\ A) \times P(not\ A)$$

e.g. Email and Spam Detection

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

$$P(B) = P(B|A) \times P(A) \; + \; P(B|not\ A) \times P(not\ A)$$

$$\begin{aligned}
P(A|B) &= P(Spam|Detected) = ? \\
P(B|A) &= P(Detected|Spam) = 0.99 \\
P(A) &= P(Spam) = 0.02 \\
P(not\ A) &= 1 \text{ - } P(Spam) = 0.98 \\
P(B|not\ A) &= P(Detected|not\ Spam) = 0.001
\end{aligned}$$

$$P(Spam|Detected) = \frac{0.99 \times 0.02}{0.99 \times 0.02 + 0.001 \times 0.98} = 0.952$$

– *Probability fact spam email in spam folder, is* **95.2%**.

e.g. Liars and Lie Detectors

- Lie Detector test persons: *if positive result $\Longrightarrow$ they are lying*.
- People are tested:
  - Telling the truth: 98%
  - Liars: 2%
- Liar people is tested: positive result 72%
- When the machine says they are *not lying*: this is true 97%

Probability that they are indeed lying?

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

$$P(B) = P(B|A) \times P(A) \ + \ P(B|not\ A) \times P(not\ A)$$

e.g. Liars and Lie Detectors

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

$$P(B) = P(B|A) \times P(A) \ + \ P(B|not\ A) \times P(not\ A)$$

$$\begin{aligned}
P(A|B) &= P(Lying|Positive) = ? \\
P(B|A) &= P(Positive|Lying) = 0.72 \\
P(A) &= P(Lying) = 0.02 \\
P(not\ A) &= 1 \text{ - } P(Lying) = 0.98 \\
P(not\ B|not\ A) &= P(not\ Positive|not\ Lying) = 0.97 \\
P(B|not\ A) &= 1 \text{ - } P(not\ B|not\ A) = 1 \text{ - } 0.97 = 0.03
\end{aligned}$$

$$P(Lying|Positive) = \frac{0.72 \times 0.02}{0.72 \times 0.02 + 0.03 \times 0.98} = 0.328$$

– *Probability fact lying when positive test result, is* **32.8%**.

e.g. Medical test

- People have a certain genetic defect: 1%
- Testing to genetic defect (true positives): 90%
- Testing have false positives: 9.6%

Probability genetic defect when get a positive test result?

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B|A) \times P(A) + P(B|not\ A) \times P(not\ A)}$$

$$\begin{aligned}
P(A|B) &= P(GeneticDefect|Positive) = ? \\
P(B|A) &= P(Positive|GeneticDefect) = 0.9 \\
P(A) &= P(GeneticDefect) = 0.01 \\
P(not\ A) &= 1 \text{ - } P(GeneticDefect) = 0.99 \\
P(B|not\ A) &= P(Positive|not\ GeneticDefect) = 0.096
\end{aligned}$$

$$P(GeneticDefect|Positive) = \frac{0.9 \times 0.01}{0.9 \times 0.01 + 0.096 \times 0.99} = 0.0865$$

- *Probability faulty gene on positive result, is* **8.65%**.

e.g. Breast Cancer test

– Women over 50 have breast cancer: 1%
– Women who have breast cancer, had positive result test: 90%
– Women will have false positives: 8%

Probability woman has cancer if she has a positive result?

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B|A) \times P(A) + P(B|not\ A) \times P(not\ A)}$$

$$\begin{aligned}
P(A|B) &= P(Cancer|Positive) = ?\\
P(B|A) &= P(Positive|Cancer) = 0.9\\
P(A) &= P(Cancer) = 0.01\\
P(not\ A) &= 1 \text{ - } P(Cancer) = 0.99\\
P(B|not\ A) &= P(Positive|not\ Cancer) = 0.08
\end{aligned}$$

$$P(Cancer|Positive) = \frac{0.9 \times 0.01}{0.9 \times 0.01 + 0.08 \times 0.99} = 0.10$$

– *Probability cancer, given a positive test result, is* **10%**.

Naïve Bayes Classifier

Given dataset $D = \{\ (x_i, y_i)\ \}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$, $y_i \in C$

given (x, y) , we find $y \in C$, with maximum $p(y|x)$

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

- $p(y)$: Prior probability of class y in dataset D (we have C class)
- $p(y|x)$: Posterior probability of class y given **one** evidence $x = (x_1, x_2, ..., x_d)$
- $p(x|y)$: Likelihood which is the probability of evidence given class $y \in C$
- $p(x)$: Prior probability of **one** evidence in D $x = (x_1, x_2, ..., x_d)$

Naïve Bayes Model

$$p(y|x_1, x_2, ..., x_d) = \frac{p(y)\ p(x_1, x_2, ..., x_d|y)}{p(x_1, x_2, ..., x_d)}$$

$(x_1, x_2, ..., x_d)$ are stochastically independent, given y:

$$p(x_1, x_2, ..., x_d|y) = p(x_1|y)\ p(x_2|y)\ ...\ p(x_d|y)$$

$$p(y|x_1, x_2, ..., x_d) = \frac{p(y) \prod_{i=1}^{d} p(x_i|y)}{p(x_1, x_2, ..., x_d)}$$

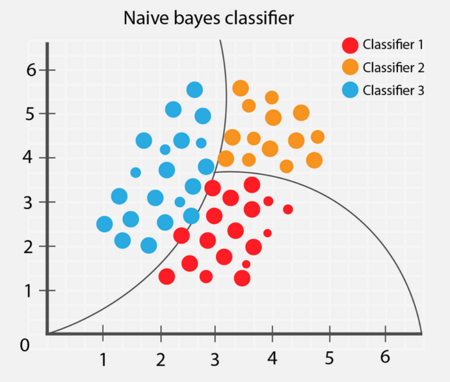
$p(x_1, x_2, ..., x_d)$ is constant given the Data set,

$$p(y|x_1, x_2, ..., x_d)\ \propto\ p(y) \prod_{i=1}^{d} p(x_i|y)$$

Algorithm:

$$\boxed{\hat{y} = \underset{y\ \in\ C}{argmax}\ p(y) \prod_{i=1}^{d} p(x_i|y)}$$

Naïve Bayes Classifier algorithm



$$\hat{y} = \underset{y \in C}{argmax}\ p(y) \prod_{i=1}^{d} p(x_i|y)$$

Very Easy!

Example

Outlook	Temperature	Huminity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Give a new instance x :

Outlook	Temperature	Huminity	Windy	Play
sunny	*cool*	*high*	*true*	**?**

y = **yes**

$p(yes) = \frac{9}{14}$
$p(sunny|yes) = \frac{2}{9}$
$p(cool|yes) = \frac{3}{9}$
$p(high|yes) = \frac{3}{9}$
$p(true|yes) = \frac{3}{9}$
$p(y = yes) = \frac{9}{14} \times \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9}$
$= 0.00529$

y = **no**

$p(no) = \frac{5}{14}$
$p(sunny|no) = \frac{3}{5}$
$p(cool|no) = \frac{1}{5}$
$p(high|no) = \frac{4}{5}$
$p(true|no) = \frac{3}{5}$
$p(y = no) = \frac{5}{14} \times \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5}$
$= 0.02057$

$p(y = no) > p(y = yes)$. Predict result:

Outlook	Temperature	Huminity	Windy	Play
sunny	*cool*	*high*	*true*	**no**

review: Mean, Variance, Standard Deviation

$$x = \{x_1, x_2, ..., x_n\} \ , \ x_i \in \mathbb{R}$$

Population Mean

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Population Variance

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

Population Standard Deviation Measure of how spread out numbers are.

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

review: Mean, Variance, Standard Deviation

$$x = \{x_1, x_2, ..., x_n\} \ , \ x_i \in \mathbb{R}$$

– Samuel Johnson: *"You don't have to eat the whole animal to know that the meat is tough."*

– Bessel's correction: Using $n - 1$ instead n sample

Sample Mean

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Sample Variance

$$v^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)^2$$

Sample Standard Deviation Measure of how spread out numbers are.

$$s = \sqrt{v^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$

review: Median, Mode

Median M Midpoint of a distribution, the number such that half the observations are smaller and the other half are larger.
To find the median of a distribution:

1. Arrange all observations is increase.

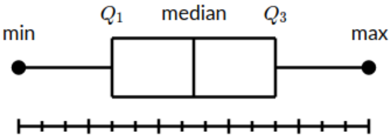
2. – If observations n is odd, the median M is the center observation in the ordered list.

   – If observations n is even, the median M is midway between the two center observations in the ordered list.

3. Always locate the median: $\frac{n+1}{2}$ is location of the median in the ordered list.

Mode The value that appears most often in a set of data.

review: The Quartiles, Box plot (Whisker plot)



First quartile Q_1 Median in the left of the overall median

Third quartile Q_3 Median in the right of the overall median

Example: Finding the five-number summary

25, 28, 29, 29, 30, 34, 35, 35, 37, 38

Step 1 Order the data from smallest to largest

25, 28, 29, 29, 30, 34, 35, 35, 37, 38

Credit: Khan Academy

review: The Quartiles, Box plot (Whisker plot)

Step 2 Find the median.

25, 28, 29, 29, **30**, **34**, 35, 35, 37, 38

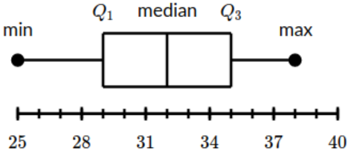
$\frac{30+34}{2} = 32$ The median is 32.

Step 3: Find the quartiles.

25, 28, **29**, 29, 30 $\quad Q_1 = 29$

34, 35, **35**, 37, 38 $\quad Q_3 = 35$

Step 4: Complete the five-number summary by finding the min and the max: **25, 29, 32, 35, 38**



Credit: Khan Academy

review: Probability Distribution

– A probability distribution is a statistical function that describes all the possible values that a random variable can take within a given range.

– This range will be bounded between the minimum and maximum possible values.

– Probability distribution depends on factors:

+ Mean (average)
+ Standard deviation
+ Skewness
+ Kurtosis

review: Probability Distribution

- Many different classifications of probability distributions.
- It serve different purposes and data processes.

e.g.

Normal distribution.

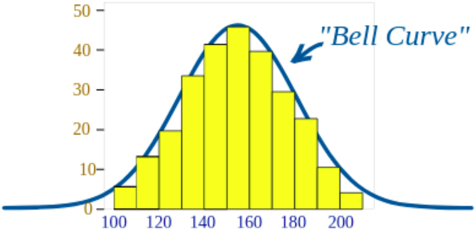
Chi square distribution.

Binomial distribution.

Poisson distribution.

review: Normal Distribution

Many cases, data tends to be around a central value:



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

σ : Standard deviation of x

μ : Mean of x

$\pi \approx 3.14159...$

$e \approx 2.71828...$

review: Normal Distribution

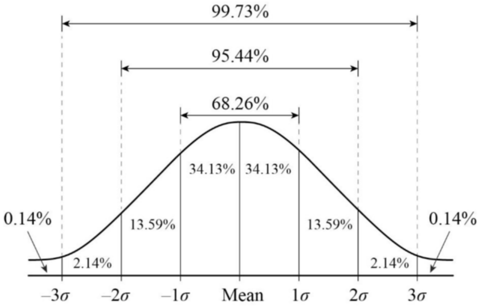
Many things closely follow a Normal Distribution:

- Heights of people
- Size of things produced by machines
- Errors in measurements
- Blood pressure
- Marks on a test

The 68 - 95 - 99.7 Rule

- Mean μ
- Standard deviation σ

- Approximately 68% observations fall within σ of the mean μ.
- Approximately 95% observations fall within 2σ of σ.
- Approximately 99.7% observations fall within 3σ of σ.



Naïve Bayes Classifier, Continuous variables

Gaussian Naive Bayes algorithm for classification

If features are continuous values, the likelihood of the features is assumed to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_{iy}^2}}\; e^{-\frac{1}{2}\frac{(x_i-\mu_{iy})^2}{\sigma_{iy}^2}}$$

x_i: feature $i\ of\ x$
μ_{iy} mean of feature $i\ of\ x$ with label $= y$
σ_{iy}^2 variance of feature $i\ of\ x$ with label $= y$

Algorithm

$$\boxed{\hat{y} = \underset{y\ \in\ C}{argmax}\ p(y)\prod_{i=1}^{d} p(x_i|y)}$$

Very easy!

Naïve Bayes Classifier, Continuous variables

With:

Sample Mean of feature i

$$\mu_i = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Sample Variance of feature i

$$\sigma_i^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_i)^2$$

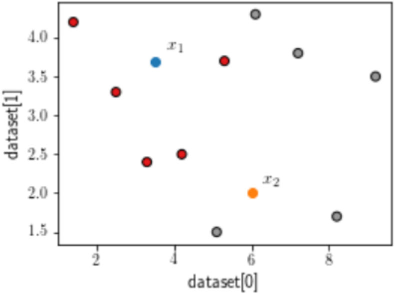
Sample Standard Deviation of feature i

$$\sigma_i = \sqrt{\sigma_i^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_i)^2}$$

e.g. Gaussian Naive Bayes Classifier

dataset:

(3.3 , 2.4 , 0),
(2.5 , 3.3 , 0),
(4.2 , 2.5 , 0),
(1.4 , 4.2 , 0),
(5.3 , 3.7 , 0),
(6.1 , 4.3 , 1),
(9.2 , 3.5 , 1),
(7.2 , 3.8 , 1),
(5.1 , 1.5 , 1),
(8.2 , 1.7 , 1)



$x_1 = (3.5\ ,\ 3.7)$
$x_2 = (6\ ,\ 2)$

e.g. Gaussian Naive Bayes Classifier

	$y = 0$	$y = 1$
feature 1		
μ	3.34	7.16
σ	1.50	1.63
feature 2		
μ	3.22	2.96
σ	0.77	1.28

$x_1 = (3.5\ ,\ 3.7)$

$y = 0$:

$$p(x_1|y) = p(y) \times p(x_{10} = 3.5|y) \times p(x_{11} = 3.7|y)$$
$$= 0.5 \times 0.26 \times 0.43 = 0.056$$

$y = 1$:

$$p(x_1|y) = p(y) \times p(x_{10} = 3.5|y) \times p(x_{11} = 3.7|y)$$
$$= 0.5 \times 0.02 \times 0.26 = 0.026$$

So, predict Class of x_1 is $y = 0$