Bayesian Notebook

```
{r setup, include=FALSE} library(reticulate) use_python("C:/Users/dream/Anaconda3/python.exe")
```

Outline

Part 1

1. Probability notation: Marginal probability, joint probability, conditional probability, Bayes rule
2. Maximum likelihood, Bayesian inference
3. Normalization trick
4. Quiz and programming break(Naive bayes)

Part 2

5. Graphical model and bayesian network
6. Basic structure of graphical model
7. Programming break (Linear regression)

Part 3

8. Bayesian inference problem
9. Variable elimination
10. Latent variable
11. EM algorithm
12. Variational inference
13. Sampling method

List and function in python

List is the basic data structure in python

```
list1 = ['physics', 'chemistry', 1997, 2000]
list2 = [1, 2, 3, 4, 5 ]
list3 = ["a", "b", "c", "d"]

print("list3 =",list3)

print("list3 slice =",list3[0:3])
```

Updating lists

```
list3 = ["a", "b", "c", "d"]

list3[0] = 'new'
print(list3)
```

```
list3 = ["a", "b", "c", "d"]

list3.append('e')
print(list3)
```

List operation

```
for i in range(5):
  print(i)
```

Iterate over a list in Python

```
list1 = ['physics', 'chemistry', 1997, 2000]

for i in list1:
  print(i);
```

```
age_list = [25,32,22,35 ]

for index, age in enumerate(age_list):
  print("index", index, end = " ")
  print("age", age)
```

```
age_list = [25,32,22,35]

for index, age in enumerate(age_list):
  age_list[index] = age+20
print(age_list)
```

Question

For given input list1, write the program that find index of 2000 in list (Rule : Don't use list1.index(2000))

```
list1 = ['physics', 'chemistry', 1997, 2000]
```

```
list1 = ['physics', 'chemistry', 1997, 2000]
record = -1
for index, e_list in enumerate(list1):
  if list1[index] == 2000:
    record = index
print(record)
```

Function

Python function are defined using def keyword.

```
def function1():
  print("Hello")
  print("HEllo2")



function1()


def print_name(name):
  print("Hello",name)

print_name("Dream")


def plus(a,b):
  print("a+b =", a+b)

plus(2,6)


def plus(a,b):
  return a+b

result = plus(2,6)
print("result =",result)
```

Numpy

Basic

You can read from the doc https://docs.scipy.org/doc/numpy/user/quickstart.html .

```
import time
import numpy as np

age_list = [25,32,22,35 ]

age_numpy_array = np.array(age_list)
print("numpy array : " ,age_numpy_array)

age_numpy_array  += 20
print("numpy array +20 : " ,age_numpy_array)


print("type =", type(age_numpy_array))

print("shape =", age_numpy_array.shape)

numpy_array = np.array([[2,3],[4,5]])
```

```
print(numpy_array)

print("shape =", numpy_array.shape )
print("size =", numpy_array.size )
print("dimension =", numpy_array.ndim)
```

Basic Operations

```
arr = np.array([20,30,40,50])
arr2 = np.array([10,30,20,55])

print("arr+ arr2 =",arr+arr2)
print("arr**2 =",arr**2)
print("arr < 35 =",arr<35)
print("arr.sum() =",arr.sum())


from math import pi

rad = np.array([pi*0,pi/2,pi,pi*3/2])

result =np.cos(rad)
print(result)


#np.exp()
#np.sqrt()



A = np.array([[1,1],
             [0,1]])
B = np.array([[2,0],
              [3,4]])


print("A*B =\n",A*B)
print("A.dot(B) =\n",A.dot(B))
print("A.sum() =",A.sum())
print("A.sum(axis = 1)", A.sum(axis = 1))

print("B[0,1] =", B[0,1])
print("B[:,1] =", B[:,1])



age_list = [25,32,22,35 ]*200000

start_time = time.time()
for index,age in enumerate(age_list):
  age_list[index] += 20

print("list time =", time.time()- start_time)
```

```
age_numpy_array = np.array(age_list)
start_time = time.time()
age_numpy_array  += 20

print("numpy array time = ", time.time()-start_time)
```

Probability notation

Random variable(RV)

Random variable is a variable that can take difference values in random space. I may denoted RV in a capital letter, for example, X is random variable and lower capital x is the constant values. Note that I may use X and x interchangeably but you can understand it in the context, anyway I try to write as correct as possible.

Marginal probability

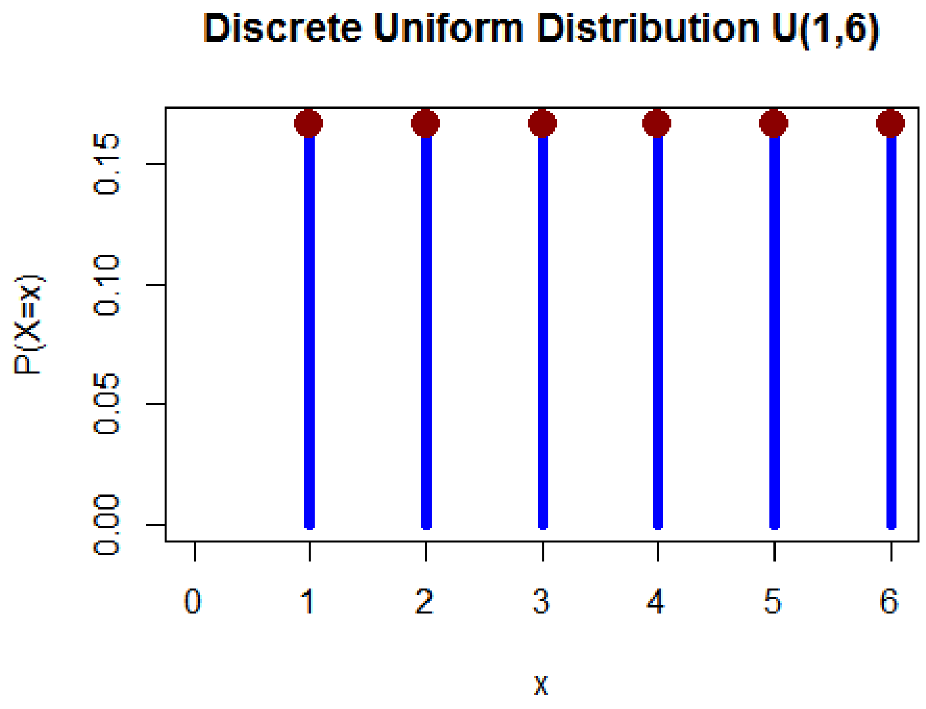
$p(X)$ denotes a marginal probability distribution of X.

Marginal probability is the probability of an event irrespective of the outcome of another variable.

Discrete case (Roll dice) $p(X) \sim \mathcal{U}(1, 6)$

You can call $p(X)$ as probability mass function (pmf).

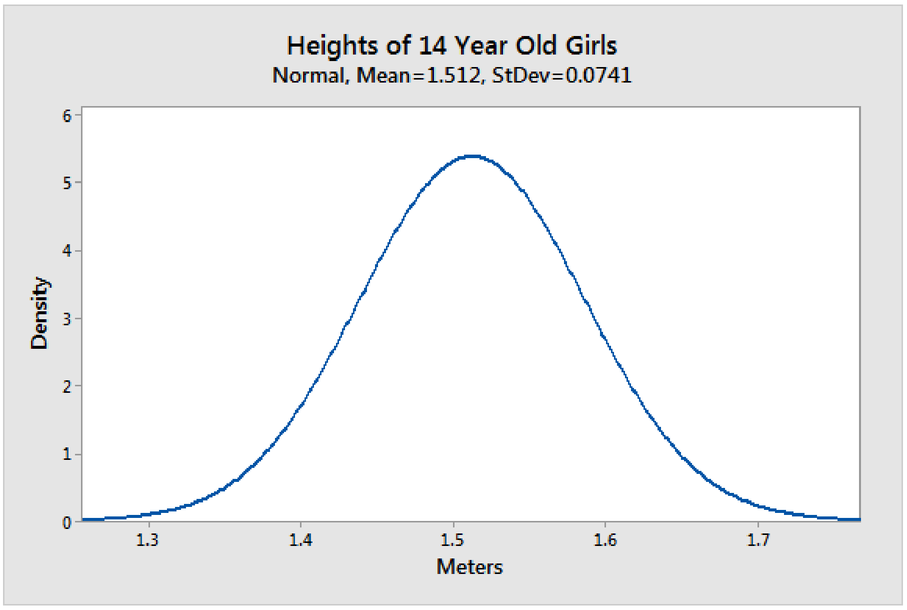
(The actual notation of pmf is $p(X = x)$)



Continuous case $p(X) \sim \mathcal{N}(\mu, \sigma)$

You can call p(X) as probability density function (pdf).

(The actual notation of pdf is $p(X = x)$)



Joint probability

$p(X, Y)$ denoted a joint probability of X and Y

Independent For example, flip the coin two times, the outcome of second coin not depend on first coin.
x denote for outcome of fliping the first coin
y denote for outcome of fliping the second coin

$$p(X, Y) = p(X)p(Y)$$

Dependent For example, this is the record of cloudy and rain in 30 day

model	Rain	No rain	Total
Cloudy	10	5	15
Not cloudy	2	13	15
Total	12	18	30

How do you calculate $p(rain)$?
In this case

$$p(rain) = p(rain, cloudy) + p(rain, notcloudy)$$

model	Rain	No rain	Total
Total	12	18	30

$$p(rain) = \sum_{all\ c} p(rain, C)$$

probability of raining $p(rain)$ is $12/30 = 0.4$
probability of cloudy $p(cloudy)$ is $15/30 = 0.5$
probability of raining and cloudy $p(rain, cloudy)$ is $10/30 = 0.333$

$$p(R, C) \neq p(R)p(C)$$

Conditional probability

$p(X \mid Y)$ denotes a conditional probability distribution of X conditioned on Y
Let look the rain & cloudy table

model	Rain	No rain	Total
Cloudy	10	5	15
Not cloudy	2	13	15
Total	12	18	30

$p(R \mid cloudy)$

model	Rain	No rain	Total
Cloudy	10	5	15
Total	10	5	15

probability of raining in cloudy day $p(rain \mid cloudy)$ is $10/15 = 0.666$

probability of cloudy $p(cloudy)$ is $15/30 = 0.5$

probability of raining and cloudy $p(rain, cloudy)$ is $10/30 = 0.333$

$$Bayes\ Theorem \qquad p(R \mid C)p(C) = p(C \mid R)p(R) = p(R, C)$$

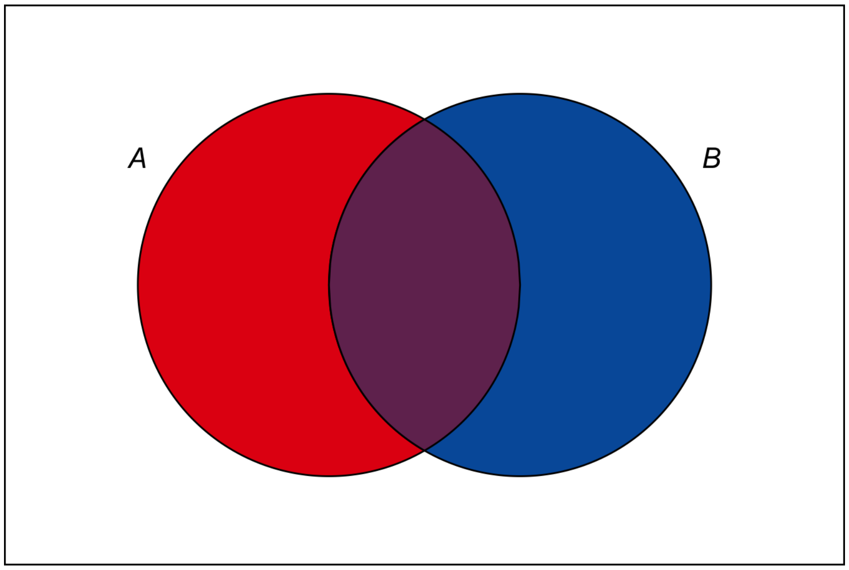
Look at the picture below

Purple color is joint probability $p(A, B)$

Red and Purple color are marginal probability $p(A)$

Blue and Purple color are marginal probability $p(B)$

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

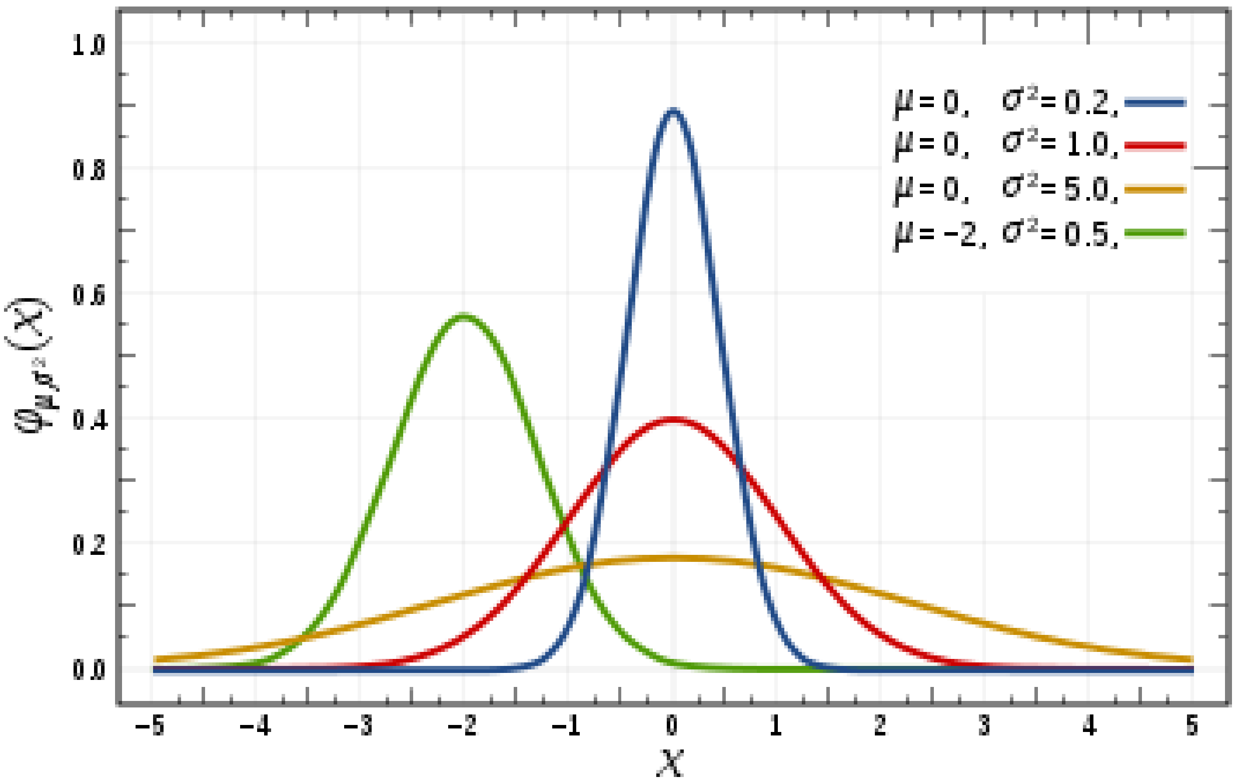


{width=500px, height=300px}

The Bayes theorem tell how often X happen given that Y happens, written as $p(R \mid C)$

Maximum likelihood estimation for parameter

For the given probabilistic model $p(X) \sim \mathcal{N}(\mu, \sigma)$ and the given data distribution as below, how do we find the best parameter μ and σ that fit the data.



Because probability of $X = x$ depend on the model parameter μ and σ we can write probability function as $p(X = x; \mu, \sigma)$.

$p(X = x; \mu, \sigma)$ mean the likelihood of observed X for given model parameter μ, σ

or we can use notation $L(\mu, \sigma; X = x)$

And we can get the model parameter μ, σ by find the μ, σ that maximize $L(\mu, \sigma; X = x)$

$$parameter_{ML} = argmax_{parameter} p(X \mid parameter)$$

Bayesian Inference

For the maximum likelihood estimation we assume the parameter is not random variable. But in bayesian we assume the model parameter as random variable. $p(X, Y) = p(Y, X)$
$p(Y \mid X)p(X) = p(X \mid Y)p(Y)$

$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}$$

$$p(parameter \mid X) = \frac{p(X \mid parameter)p(parameter)}{p(X)}$$

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

Because the evident is just constant $(p(X = x))$

$$Posterior \propto Likelihood \ \times \ Prior$$

Prior show the probability before we know the data (distribution over possible parameter values)

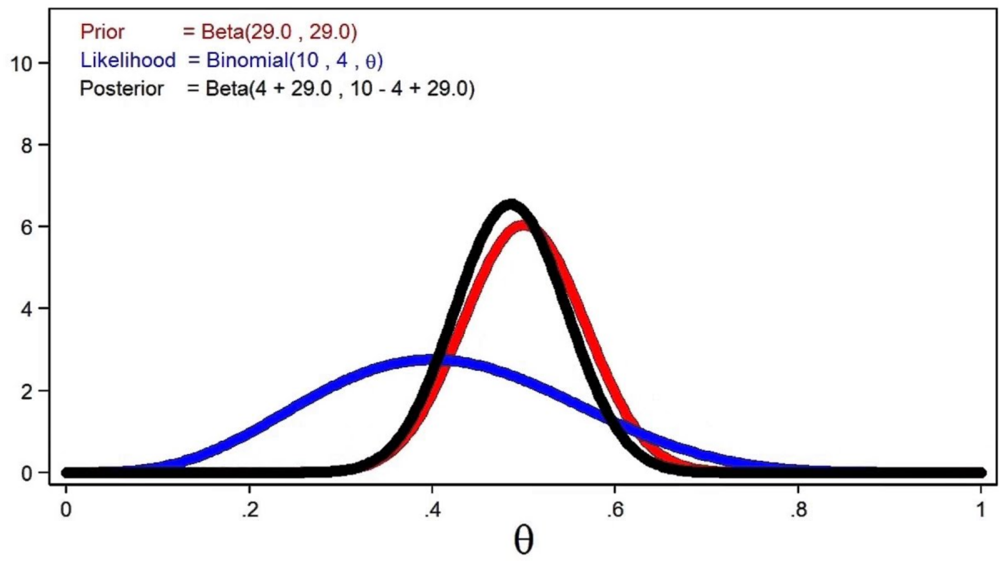
Likelihood give us a best parameter for given data.

Posterior is the distribution that conbine the information from prior knowledge and the data.

And we can find the best parameter by find the maximum posterior

$$paramater_{MAP} = argmax_{parameter} p(parameter \mid X)$$

Example : fliping coin



Normalization trick

model	Rain	No rain	Total
Cloudy	10	5	15
Not cloudy	2	13	15
Total	12	18	30

model	Rain	No rain	Total
Cloudy	0.333	0.166	0.5
Not cloudy	0.066	0.433	0.5
Total	0.4	0.6	1

We can calculate $p(R \mid cloudy)$ by calculate only $p(R, cloudy)$

model	Rain	No rain	Total
Cloudy	0.333	0.166	0.5

And normalize the table to get $p(R \mid cloudy)$

model	Rain	No rain	Total
Cloudy	0.333/0.5	0.166/0.5	1

Because

$$p(R \mid C = c) = \frac{p(R, C = c)}{p(C = c)}$$

$p(C = c)$ is just constant

$$p(R \mid C = c) \propto p(R, C = c)$$

Quiz true or false ?

1. $\sum_{all\ x} p(X) = 1$
2. $\sum_{all\ x, all\ y} p(X, Y) \neq 1$
3. $\sum_{all\ x, all\ y} p(X \mid Y) = 1$
4. $argmax_{parameter} p(parameter \mid X)$ is only depend on $argmax_{parameter} p(X; parameter)$ sometimes.

Programming break

In this example we have the data of New China Virus and the symptom of patients. The given data will be look like table below.

Patients number	New China Virus	Fever	Vomitting
1	1	1	1
2	0	1	0
3	1	1	1
4	1	1	1
5	0	1	1
6	0	1	0
7	1	0	1
8	0	1	0
9	1	0	1
10	0	1	0

Naive bayes

```
import numpy as np

def generate():

  dataset = []
  for i in range(10):
    patient = np.random.randint(0,2, 3)
    dataset.append(patient)
  dataset = np.array(dataset)
  return dataset

dataset = generate()
```

$$\textit{Bayes Theorem} \qquad p(X \mid Y)p(Y) = p(X,Y)$$

$$p(virus \mid fever, vomit) = \frac{p(virus, fever, vomit)}{p(fever, vomit)}$$

Because naive bayes assume independent variable.

$$p(virus \mid fever, vomit) = \frac{p(fever, vomit \mid virus)p(virus)}{p(fever)p(vomit)}$$

$$p(virus \mid fever, vomit) = \frac{p(fever \mid virus)p(vomit \mid virus)p(virus)}{p(fever)p(vomit)}$$

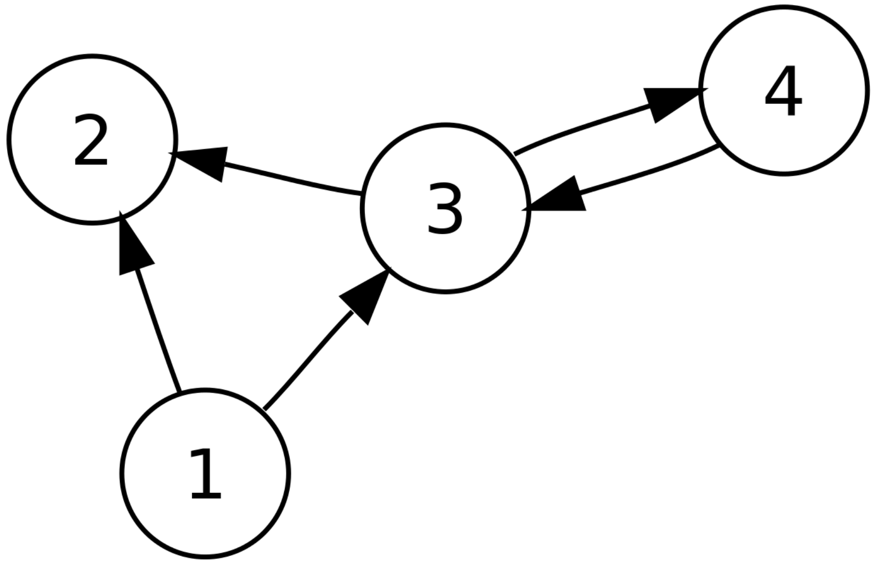
```
print("dataset = \n",dataset)

fever = dataset[:, 1]
virus = dataset[:, 0]
vomit = dataset[:, 2]


p_vomit = vomit.sum()/len(vomit)
p_virus = virus.sum()/len(virus)
p_fever = fever.sum()/len(fever)

dataset_virus_positive = dataset[virus == 1]

print("dataset virus positive = \n" ,dataset_virus_positive)

vomit_virus_pos = dataset_virus_positive[:,2]
fever_virus_pos = dataset_virus_positive[:,1]

p_vomit_condition_virus = vomit_virus_pos.sum()/len(vomit_virus_pos)
p_fever_condition_virus = fever_virus_pos.sum()/len(fever_virus_pos)

posterior = p_vomit_condition_virus*p_vomit_condition_virus*p_virus/p_vomit/p_fever

print( "posterior = ",posterior)
```

Example above you will know that if the patient have fever and come from china the probability that they have virus is the “posterior”

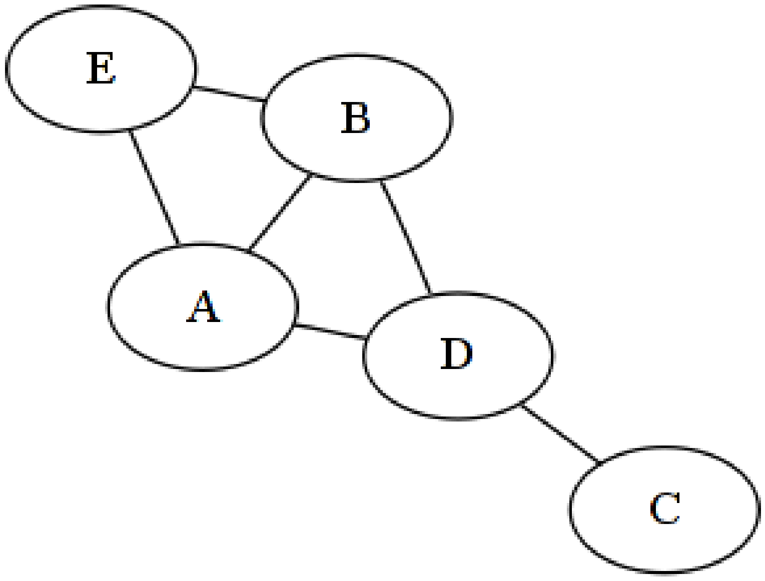
Graphical models

1. Directed graph

- Bayesian network (today)
- Naive Bayes classifier (today)



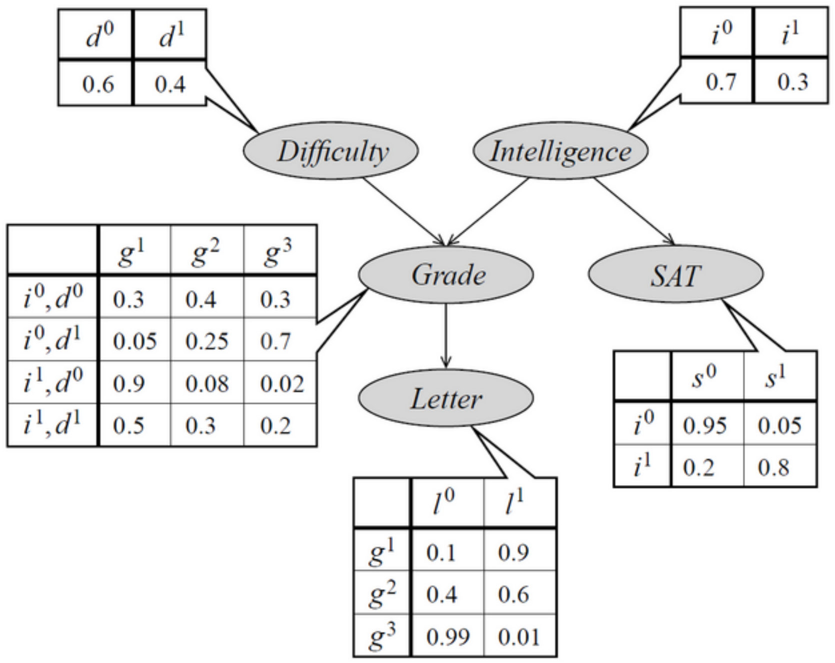
2. Undirected graph



Bayesian network (BN)

BN model the random variable(RV) using directed acyclic graph. Let start by modeling the graph of student network.

We have RV : G(grade), D(school difficalty), I(intelligence), S(SAT score), L(recommend Letter)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Bayesian network encode the joint distribution as conditional distribution.

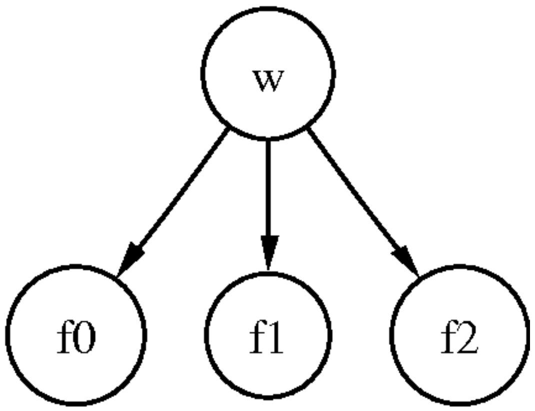
$$p(X_1, X_2, X_3, ...X_n) = \prod_{i=1}^{n} p(X_i \mid parent(X_i))$$

We can factor the joint probability of student network by looking at the graph.

$$p(G, D, I, S, L) = p(G \mid D, I)p(D)p(I)p(S \mid I)p(L \mid G)$$

Naive bayes (revisit)

```
| Patients number | New China Virus | Fever | Come from china |
|-----------------|-----------------|-------|-----------------|
| 1               | 1               | 1     | 1               |
| 2               | 0               | 1     | 0               |
| 3               | 1               | 1     | 1               |
| 4               | 1               | 1     | 1               |
| 5               | 0               | 1     | 1               |
| 6               | 0               | 1     | 0               |
| 7               | 1               | 0     | 1               |
| 8               | 0               | 1     | 0               |
| 9               | 1               | 0     | 1               |
| 10              | 0               | 1     | 0               |
```

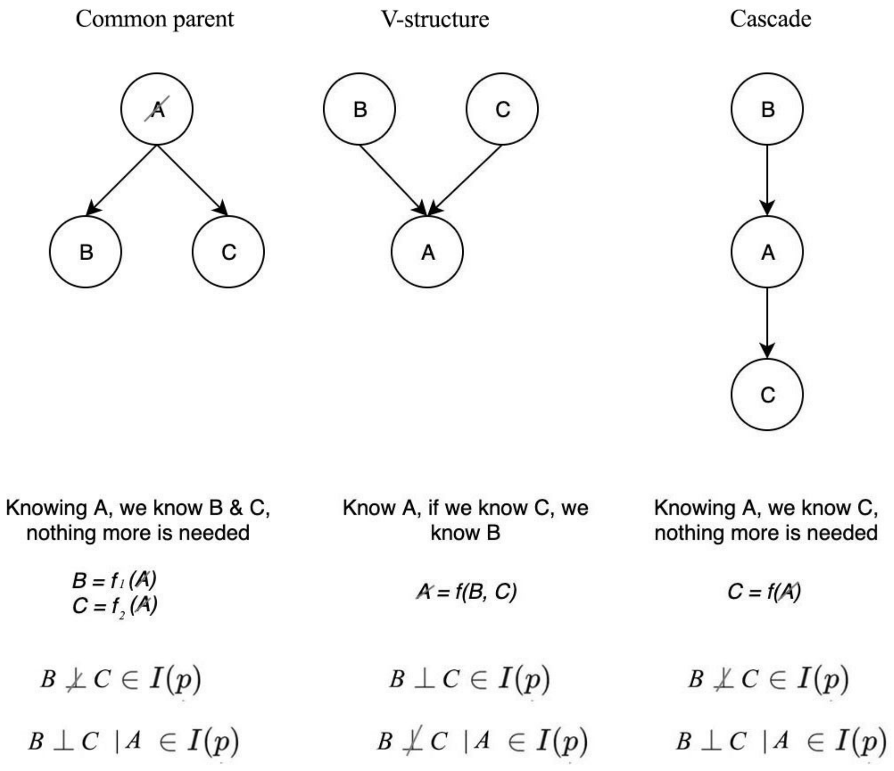


Naive Bayes model:

$$p(W \mid f_1, f_2, f_3) \propto p(W) \prod_i p(f_i|W)$$

Structure in graphical model

In this picture we want to know if B can influence C or C can influence B by given the basic structure of graphical model


Common parent
V-structure
Cascade
$\cancel{A}$
B
C
Knowing A, we know B & C, nothing more is needed
$B = f_1(\cancel{A})$
$C = f_2(\cancel{A})$
$B \not\perp C \in I(p)$
$B \perp C \mid A \in I(p)$
Know A, if we know C, we know B
$\cancel{A} = f(B, C)$
$B \perp C \in I(p)$
$B \not\perp C \mid A \in I(p)$
Knowing A, we know C, nothing more is needed
$C = f(\cancel{A})$
$B \not\perp C \in I(p)$
$B \perp C \mid A \in I(p)$

model	B can give information about C not given A	B can give information about C given A
B->A->C	yes	no
C->A->B	yes	no

model	B can give information about C not given A	B can give information about C given A
C->A<-B	no	yes
B<-A->C	yes	no

Programming break (linear regression)

I will show you bayesian approach for linear regression.

$$y = mx + c$$

$$y = \beta_1 x + \beta_2 x^2 + \beta_0$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}$$

$$Y = X \times W$$

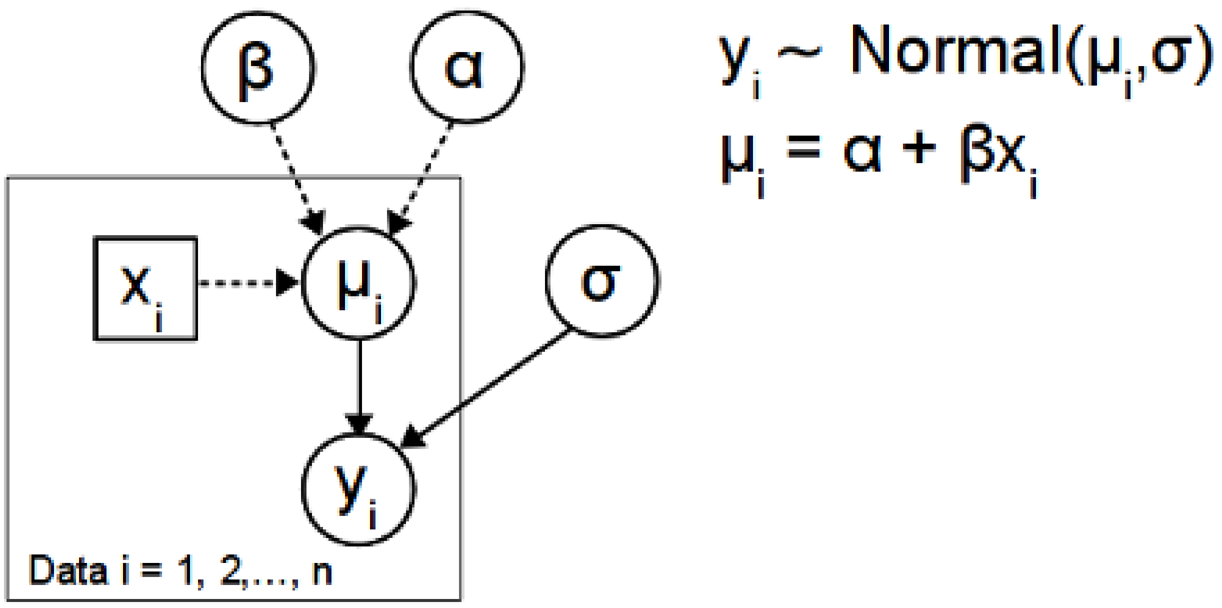
$$X -> \ Y \ <- W$$

$$p(W \mid X, Y) = \frac{p(W, X, Y)}{p(X, Y)}$$

$$p(W \mid X, Y) \propto p(W, X, Y)$$

$$p(W \mid X, Y) \propto p(W) p(Y \mid W, X)$$

We don't know anything about the actual model of $p(W)$ and $p(Y \mid X, W)$, but we can assume it.


β
α
x_i
μ_i
σ
y_i
Data i = 1, 2,..., n
$y_i \sim \text{Normal}(\mu_i, \sigma)$
$\mu_i = \alpha + \beta x_i$

$$p(Y \mid W, X) = \mathcal{N}(Y \mid \mu, \sigma)$$

$$p(Y \mid W, X) = \mathcal{N}(Y \mid XW, \sigma)$$

and

$$p(W) = \mathcal{N}(W \mid 0, \gamma)$$

you get

$$p(W \mid X, Y) \propto p(W)p(Y \mid W, X)$$

$$p(W \mid X, Y) \propto \mathcal{N}(W \mid 0, \gamma)\mathcal{N}(Y \mid \mu, \sigma)$$

The multivariate guassian function is

$$\mathcal{N}(Y \mid \mu, \sigma I) = \frac{1}{(2\pi)^{d/2}}|\Sigma|^{-1/2} exp[-\frac{1}{2}(y-\mu)\Sigma^{-1}(y-\mu)^T]$$

Problem about bayesian inference

Let define the random varibles first

- Evidence variable: $E_1 = e_1 ... E_k = e_k$
- Query variable: Q
- Hidden variable: $H_1 ... H_r$

We want $p(Q \mid e_1 ... e_k)$

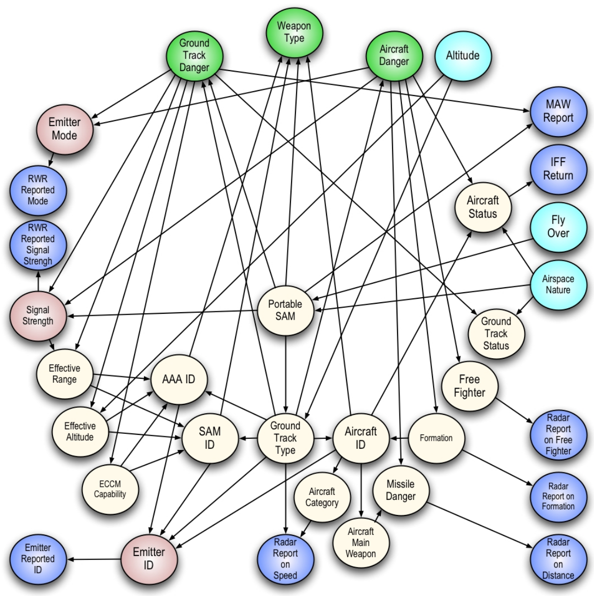
$$p(Q, e_1...e_k) = \sum_{h_1...h_r} p(Q, h_1...h_r e_1...e_k)$$

And

$$p(Q \mid e_1...e_k) = \frac{p(Q, e_1...e_k)}{p(e_1...e_k)}$$

$$p(Q \mid e_1...e_k) = \frac{\sum_{h_1...h_r} p(Q, h_1...h_r e_1...e_k)}{p(e_1...e_k)}$$

If you have a very large network, it very slow and hard to compute.



Solution Variable elimination

Latent variable

EM algorithm

Variational inference

Sampling method

Python Library

1. PyMC3, PyMC4
2. pyStan
3. pyro
4. Tensorflow probability
5. Edward 2

Reference

1. https://medium.com/@jonathan_hui
2. https://www.coursera.org/specializations/probabilistic-graphical-models
3. https://www.youtube.com/playlist?list=PLe5rNUydzV9QHe8VDStpU0o8Yp63OecdW
4. http://ai.berkeley.edu/lecture_slides.html