Naïve Bayes Classifier

Akito Sakurai

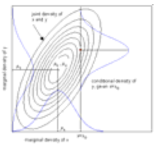
Today’s topic

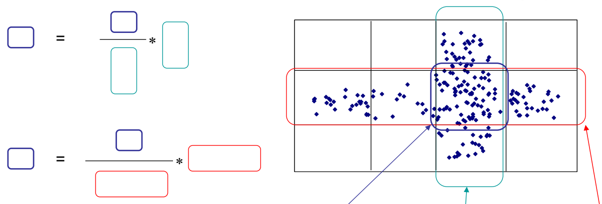
- Basics of Bayesian inference
- Principle and implementation of naïve Bayes method

Contents

- Basics of probability
  - Conditional probability and Bayes theorem
  - Bayesian inference
- Naïve Bayes
  - What is “naïve”
  - The number of features
  - Classifiers
  - A simple example
  - In R
  - Training errors

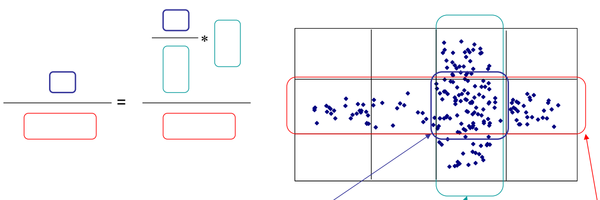
Conditional Probability





$$p(m\,|\,x)\,p(x) = p(x,m) = p(x\,|\,m)\,p(m)$$

Bayes Theorem



$$p(m\,|\,x) = \frac{p(x,m)}{p(x)} = \frac{p(x\,|\,m)\,p(m)}{p(x)}$$

Contents

- Basics of probability
  - Conditional probability and Bayes theorem
  - Bayesian inference
- Naïve Bayes
  - What is “naïve”
  - The number of features
  - Classifiers
  - A simple example
  - In R
  - Training errors

Bayesian Inference

- Bayesian inference is a method of statistical inference in which some kind of evidence or observations are used to calculate the probability that a hypothesis may be true, or else to update its previously-calculated probability.

$$p(m \mid x) = \frac{p(x \mid m)\, p(m)}{p(x)}$$

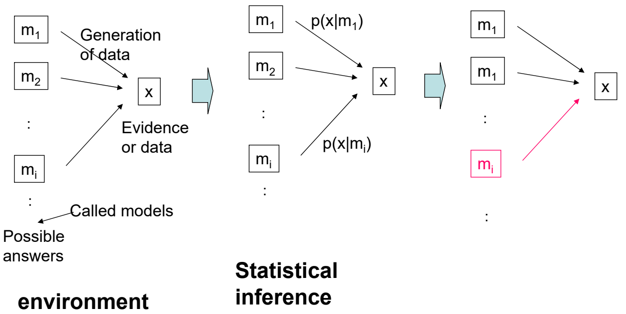
From Wikipedia

Addendum

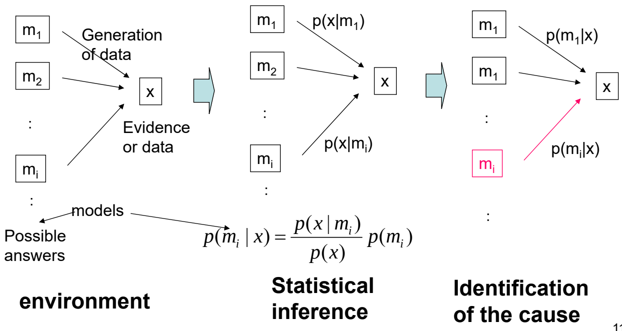
- Suppose evidence is x, and cause is m
  - Candidates of causes are: m_i
- Bayesian inference is a method to infer m from m_i by calculating $p(m_i|x)$ with a method

$$p(m \mid x) = \frac{p(x \mid m)\, p(m)}{p(x)}$$

A framework of Bayesian inference



A framework of Bayesian inference



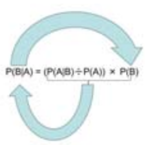
Estimation of $p(m)$ and $p(x|m)$

$$\underbrace{p(m \mid x)}_{\text{Posterior prob.}} = \frac{p(x, m)}{p(x)} = \frac{\overbrace{p(x \mid m)}^{\text{Conditional}}}{p(x)} \overbrace{p(m)}^{\text{Prior prob.}}$$

- p(m) is estimated from occurrence frequencies of the class event m
- How about p(x|m) ?
  - p(x|m) is the probability of sample x generated from model m. This is the description of the model m.
  - Maybe normal, maybe multinomial,…

Bayesian inference and naïve Bayes

- Bayesian inference



- Naïve Bayes
  - A simplified method of Bayesian inferene

Contents

- Basics of probability
  - Conditional probability and Bayes theorem
  - Bayesian inference
- Naïve Bayes
  - What is “naïve”
  - The number of features
  - Classifiers
  - A simple example
  - In R
  - Training errors

Model description by naïve Bayes

- Suppose that an evidence x is described with features
  - Very common
  - Features may be gender, age, location, weight, height, interests,…, product names, unit price, date of sales, features of customers,…
- Suppose that features are independent…
  - “No way” should be words of descent people. Therefore the assumption is called “naïve.”

Are weight and height independent?
No! i.e., features may be dependent

Features

- If $<a_1,\ldots,a_n>$ is a vector of features of “evidence” x , we may describe it by x and also by $<a_1,\ldots,a_n>$.
- Under such circumstances, a feature vector is the sample itself
  - Ex.
  - If Jim’s feature vector is <172, 63, computer science, 19>, <172, 63, computer science, 19> is Jim himself

Features are independent, if…

- Suppose $<a_1,\ldots,a_n>$ is the feature vector of evidence x. The features are independent if:

$$p(X=x)=p(A_1=a_1,\ldots,A_n=a_n)$$
$$=\prod_{i=1}^{n} p(A_i=a_i)$$

- whereas “conditional independence” is defined as

$$p(X=x\mid C=c)=p(A_1=a_1,\ldots,A_n=a_n\mid C=c)$$
$$=\prod_{i=1}^{n} p(A_i=a_i\mid C=c)$$

Model description by Naïve Bayes

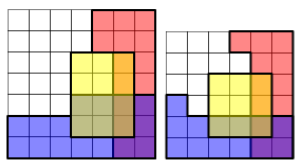
is

- Describe evidence x by its features as $<a_1,\ldots,a_n>$
- And suppose that:

$$p(X=x)=p(A_1=a_1,\ldots,A_n=a_n)$$
$$=\prod_{i=1}^{n} p(A_i=a_i)$$
$$p(X=x\mid C=c)=p(A_1=a_1,\ldots,A_n=a_n\mid C=c)$$
$$=\prod_{i=1}^{n} p(A_i=a_i\mid C=c)$$

Conditional independence

- Independence and cond. ind. are different



Illustrations. Each rectangle is an event. Each event has the same probability of occurrence. Events R, B and Y are in red, blue, yellow. Overlaps of events R and B are in purple. In both of these,
$\Pr(R\cap B\mid Y)=\Pr(R\mid Y)\Pr(B\mid Y)$ and $\Pr(R\cap B\mid\neg Y)\neq\Pr(R\mid\neg Y)\Pr(B\mid\neg Y)$
Therefore $\Pr(R\cap B)\neq\Pr(R)\Pr(B)$

https://en.wikipedia.org/wiki/Conditional_independence

Coming back

- What we want is $p(m|x)$.

$$p(m\,|\,x) = \frac{p(x,m)}{p(x)} = \frac{p(x\,|\,m)}{p(x)}\,p(m) = \frac{p(a_1,\ldots,a_n\,|\,m)}{p(x)}\,p(m)$$

Therefore

$$p(m\,|\,x) = \frac{\prod_{i=1}^{n} p(a_i\,|\,m)}{p(x)}\,p(m)$$

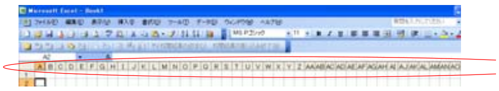
by naïve Bayes

Why is it good?

- We want to circumvent a problem caused by the number of features.

Is it a problem to have large set of features?

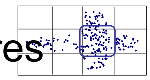
- Yes. If there are many features, large dataset is required to estimate the parameters.



Contents

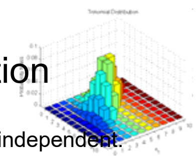
- Basics of probability
  - Conditional probability and Bayes theorem
  - Bayesian inference
- Naïve Bayes
  - What is “naïve”
  - The number of features
  - Classifiers
  - A simple example
  - In R
  - Training errors

The number of features



- Suppose that the variables take discrete values. Let us use an example (not in general formulae)
- In $<A_1,A_2,A_3,A_4>$, the four variables take values high, middle, and low (abbreviated as 0,1, and 2).
- No distribution is assumed (no a priori knowledge). In such a case, if for any of 3^4=81 $<A_1,A_2,A_3,A_4>$ combinations one probability $p_{<A_1,A_2,A_3,A_4>}$ is determined, the distribution is determined. Since the sum of them is restricted to be1, 80 values are to be determined.
- How large should be the dataset to estimate these values from data?

Multinomial distribution



- Each sample (evidence) supposed to be independent.
- Frequency of occurrences of $<A_1,A_2,A_3,A_4>$ distributes according to multinomial distribution.
- Multinomial Dist.: Suppose that event e_i occurs with probability p_i (sum of p_i is 1). In n repetitions, the probability that event e_i occurs n_i times is

$$p(n_1,\ldots,n_k;n,p_1,\ldots,p_k) = \frac{n!}{n_1!\cdots n_k!}\,p_1^{n_1}\cdots p_k^{n_k}$$

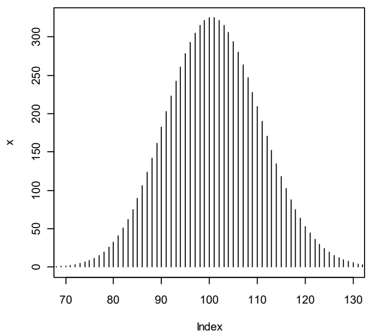
- Note that its expectation, variance, and covariance are

$$E(N_i) = np_i,\ \mathrm{var}(N_i) = np_i(1-p_i),\ \mathrm{cov}(N_i,N_j) = -np_ip_j$$

The number of features

- Because $p_{<A_1,A_2,A_3,A_4>}$ occurs 81 times, suppose true value $p_{<0,0,0,0>}$ =1/81 and let us estimate it.
- $<0,0,0,0>$ follows binomial distribution. Then for n=8100, mean $np_{<0,0,0,0>}$ =100, variance $np_{<0,0,0,0>}(1-p_{<0,0,0,0>})\approx 98.8$, SD ≈ 9.9
- Therefore to estimate $p_{<0,0,0,0>}$, if n=8100 ,the probability that the occurrences of $<0,0,0,0>$ is in 100±10 (error rate is lower than 10%) about 68% (approx. 1σ)
  - bad : - (
- But if we suppose the features are independent, since $p_{<0,0,0,0>} = \prod p_{A_i=0}$ $p_{Ai=0}$ are only to be estimated, we can use all the data (i.e., n=8100)
- Then: if $p_{Ai=0}$ =1/3, for n=8100, mean 2700, variance 1800, SD ≈42.4. thTe probability that it is in 2700±270 (error rate less than 10%) is greater than about 1 – 2/one billion (6σ)
  - For n=300, mean 100, variance≈66.7, SD≈8.16, therefore the probability that is in 100±10 (error rate less than 10%) is greater than 68%, but approximately the same (greater than 1σ)

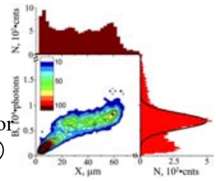
```
> x<-dbinom(0:200,8100,1/81)*8100
> plot(x,type="h",xlim=c(70,130))
>
```



The number of features

In summary

- To estimate $p_{<A_1,A_2,A_3,A_4>}$, for n=8100, the probability that the error rate is less than 10% is about 68% (approx. 1σ)
- On the other hand, if we suppose independence of features as naïve Bayes for n=300, the probability that the error rate is less than 10% is greater than 68% but approx. the same
  - For n=8100, the probability that the error rate is less than 10% is >1 – 2/10^9 (6σ)



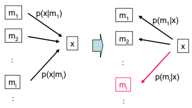
Is everything OK?

- If the independence is really true, everything is OK
- But it never is.
  - Suppose that you have to diagnose influenza or not.
  - Clearly three features <cough, soar throat, fever> are not independent
- If we suppose independence, although they are not, what will happen?
- We cannot know what happens
  - In fact, the probability estimated by naïve Bayes is completely garbage
- But in reality, naïve Bayes works well quite often, because
  - Increase of errors caused by erroneous assumption of independence is canceled out by the increase of accuracy of parameter estimation based on the erroneous assumption
    - Distribution is not estimated. We estimate class probability.

Therefore naïve Bayes

- Shall we use it? (old people thought so)
- In fact it works often.
  - Do not use it for probability estimation
  - Works only for classification
- Let us use it for classification

Naïve Bayes classifier



- In a previous slide, we have

$$p(m \mid x) = \frac{p(x \mid m)}{p(x)} p(m)$$

suppose m_1 is class1, m_2 is class2

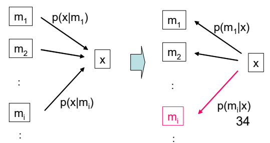
- Evidence x is a set of observations (only 1 sample) , each sample is described as $<A_1,\ldots,A_n>$.
  - Each attribute values are discrete
- Each class is statistically independent
- Class is characterized by the distribution of attributes
  - For each class, A_i 's value $a_{i1},\ldots,a_{ik}$ distributes according to the probability $p_{i1},\ldots,p_{ik}$ (to estimate them is to learn)

Naïve Bayes classifier

- Under these assumptions

$$\begin{aligned} p(m_j \mid x) &= \frac{p(x \mid m_j)}{p(x)} p(m_j) \\ &\approx p(x \mid m_j) p(m_j) \\ &= p(a_1,\ldots,a_n \mid m_j) p(m_j) \\ &= p(m_j) \prod_{i=1}^{n} p(a_i \mid m_j) \end{aligned}$$

$$m_{\text{MAP}} = \arg\max_j \; p(m_j \mid x)$$



Naïve Bayes classifier

- The parameters (probabilities $p_{i1},\ldots,p_{ik}$) to describe a model m are estimated as follows.
- Suppose the model m generated n-dimensional samples $<y_{j1},\ldots,y_{jn}>$ $(j=1,\ldots,N)$
- Build a histogram of $<y_{1i},\ldots,y_{Ni}>$ for the attributes A_i $(i=1,\ldots,n)$, i.e., if an A_i takes three values $1,2$, and 3, count occurrences of 1, 2, and 3.
- Based on this, estimate p_{i1}, p_{i2}, p_{i3} , i.e., p_{i1}=counts of 1/N, p_{i2}=counts of 2/N, p_{i3}=counts of 3/N.

Contents

Play tennis

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	No	No
Sunny	Hot	High	Yes	No
Overcast	Hot	High	No	Yes
Rainy	Mild	High	No	Yes
Rainy	Cool	Normal	No	Yes
Rainy	Cool	Normal	Yes	No
Overcast	Cool	Normal	Yes	Yes
Sunny	Mild	High	No	No
Sunny	Cool	Normal	No	Yes
Rainy	Mild	Normal	No	Yes
Sunny	Mild	Normal	Yes	Yes
Overcast	Mild	High	Yes	Yes
Overcast	Hot	Normal	No	Yes
Rainy	Mild	High	Yes	No

Two classes: Play=Yes to play tennis and Play=No for not to play tennis

Predict whether Play=Yes or Play=No for the following unseen sample i.e., a sample not in the training dataset.

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

From **Tom Mitchell**'s book **Machine Learning**. Often used to help students to estimate by hand.

First, divide samples into classes

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
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

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

Count and estimate

	A1=Outlook		.2=Temperatur		A3=Humidity		A4=Windy	
frequency	Sunny	2	Hot	2	High	3	False	6
	Overcast	4	Mild	4	Normal	6	True	3
	Rainy	3	Cool	3				
	Sum	9	Sum	9	Sum	9	Sum	9
estimation	Sunny	2/9	Hot	2/9	High	3/9	False	6/9
	Overcast	4/9	Mild	4/9	Normal	6/9	True	3/9
	Rainy	3/9	Cool	3/9				

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
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

	A1=Outlook		.2=Temperatur		A3=Humidity		A4=Windy	
frequency	Sunny	3	Hot	2	High	4	False	2
	Overcast	0	Mild	2	Normal	1	True	3
	Rainy	2	Cool	1				
	Sum	5	Sum	5	Sum	5	Sum	5
estimation	Sunny	3/5	Hot	2/5	High	4/5	False	2/5
	Overcast	0/5	Mild	2/5	Normal	1/5	True	3/5
	Rainy	2/5	Cool	1/5				

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

Put them into one table

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
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

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

p(m) is this

A1=Outlook			A2=Temperature			A3=Humidity			A4=Windy			m=Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Inference

$$p(m_j \mid x) = p(x \mid m_j)\,p(m_j) \big/ p(x)$$
$$= p(a_1, \ldots, a_n \mid m_j)\,p(m_j) \big/ p(x)$$
$$= \left(\prod_{i=1}^{n} p(a_i \mid m_j)\right) p(m_j) \Big/ p(x)$$

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Unseen *x*

p(Play=yes | *x*)
= p(Outlook=Sunny | Play=yes)
  * p(Temp=Cool | Play=yes)
  * p(Humidity=High | Play=yes)
  * p(Windy=True | Play=yes)
  * p(Play=yes) / p(x)
= (2/9) * (3/9) * (3/9) * (3/9)
  *(9/14) / p(x)
= 0.0053 / p(x)

p(Play=no | *x*)
= p(Outlook=Sunny | Play=no)
  * p(Temp=Cool | Play=no)
  * p(Humidity=High | Play=no)
  * p(Windy=True | Play=no)
  * p(Play=no) / p(x)
= (3/5) * (1/5) * (4/5) * (3/5)
  *(5/14) / p(x)
= 0.0206 / p(x)

The results say p(Play=yes | *x*) < p(Play=no | *x*)
i.e., didn't (or won't) "play tennis"

Note: 1/p(x) turns out to be no head ache; any counterpart have it

Contents

In R

```
# after installing package e1071
> library(e1071)
> setwd("D:/R/Sample")
> xy<-read.csv("04PlayTennis.csv", header=TRUE)
> xyt<-read.csv("04PlayTennisTest01.csv",header=TRUE,as.is=TRUE)
> tt<-data.frame(factor(xyt[,1],levels=levels(xy[,1])))
> for (i in 2:5) {
>   tt<-data.frame(tt,factor(xyt[,i],levels=levels(xy[,i])))
> }
> names(tt)<-names(xy)
> tt
  Outlook Temp. Humidity Windy Play
1   Sunny  Cool     High  True <NA>
> m <- naiveBayes(xy[,-5], xy[,5])
> predict(m, tt)
[1] No
Levels: No Yes
>
```

We cannot use xyt as xy（as as.is=FALSEで）. The reason is that unseen sample, here, is just one sample and all the values for each categorical attribute are not included and those levels in xy could not be referred.

Note

Apply cannot be used for a For loop because levels are combined when apply is used.

```
# package e107 をインストールした後、
> library(e1071)
> setwd("D:/R/Sample")
> xy<-read.csv("04PlayTennis.csv", header=TRUE)
> xyt<-read.csv("04PlayTennisTest01.csv", header=TRUE, as.is=TRUE)
> tt<-apply(as.data.frame(1:5),1,
        function(i) factor(xyt[,i],levels=levels(xy[,i])))
> tt
[1] Sunny Cool  High  True  <NA>
Levels: Overcast Rainy Sunny Cool Hot Mild High Normal False True No Yes
>
```


Note 2:

We can get prediction probability by just adding type="raw" as an argument to the function "predict"

```
> predict(m, tt, type="raw")
            No       Yes
[1,] 0.7954173 0.2045827
>
```


Contents

Parameters and training error

```
> m

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = xy[, -5], y = xy[, 5])

A-priori probabilities:
xy[, 5]
       No       Yes
0.3571429 0.6428571

Conditional probabilities:
       Outlook
xy[, 5]  Overcast     Rainy     Sunny
    No  0.0000000 0.4000000 0.6000000
    Yes 0.4444444 0.3333333 0.2222222

       Temp.
xy[, 5]      Cool       Hot      Mild
    No  0.2000000 0.4000000 0.4000000
    Yes 0.3333333 0.2222222 0.4444444

       Humidity
xy[, 5]      High    Normal
    No  0.8000000 0.2000000
    Yes 0.3333333 0.6666667

       Windy
xy[, 5]     False      True
    No  0.4000000 0.6000000
    Yes 0.6666667 0.3333333
```

confusion matrix:

```
> table(predict(m, xy[,-5]), xy[,5])

      No Yes
  No   4   0
  Yes  1   9
>
```

Yes is predicted

No is the truth

A1=Outlook			A2=Temperature			A3=Humidity			A4=Windy			m=Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

An exercise

- Suppose you are given a training dataset at the left and as an unseen sample a sample at the right is given. Use Naïve Bayes and bet “go skiing” value
- The dataset is in: http://www.sakurai.comp.ae.keio.ac.jp/classes/IntInfProc-class/2017/04PlaySkii.zip

snow	weather	season	physical condition	go skiing
sticky	foggy	low	rested	no
fresh	sunny	low	rested	yes
fresh	foggy	low	rested	yes
frosted	foggy	low	injured	no
fresh	sunny	low	injured	no
sticky	sunny	low	rested	yes
fresh	foggy	low	rested	yes
sticky	sunny	mid	rested	yes
fresh	sunny	high	rested	yes
fresh	windy	low	rested	yes
frosted	foggy	mid	rested	no
fresh	windy	low	rested	yes
fresh	sunny	mid	rested	yes
frosted	windy	high	tired	no

snow	wether	season	sical condi	go skiing
sticky	windy	mid	tired	?

Summary

- Bayesian inference
  - Get the posterior probability of causes (models) based on gathered evidence, and infer the cause
- Difficulty
  - (if complex models are to be used) the number of data to be used to determine the parameters is large
- Naïve Bayes
  - A good solution to address it
  - Assumes attributes (to describe samples) are conditionally independent
    - May not be true but works.
  - Is not old fashoned