# Bayesian decision
# first encounter with Bayesian theory

YT YANG

# Axiom of probability

1. $0 \le P(E) \le 1$. If $E_1$ is an event that cannot possibly occur then $P(E_1) = 0$. If $E_2$ is sure to occur, $P(E_2) = 1$.

2. $S$ is the sample space containing all possible outcomes, $P(S) = 1$.

3. If $E_i, i = 1, \ldots, n$ are mutually exclusive (i.e., if they cannot occur at the same time, as in $E_i \cap E_j = \emptyset, j \neq i$, where $\emptyset$ is the *null event* that does not contain any possible outcomes) we have

(A.1) $$P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{i=1}^{n} P(E_i)$$

For example, letting $E^c$ denote the *complement* of $E$, consisting of all possible outcomes in $S$ that are not in $E$, we have $E \cap E^C = \emptyset$ and

$$P(E \cup E^c) = P(E) + P(E^c) = 1$$
$$P(E^c) = 1 - P(E)$$

If the intersection of $E$ and $F$ is not empty, we have

(A.2) $$P(E \cup F) = P(E) + P(F) - P(E \cap F)$$

### A.1.2 Conditional Probability

$P(E|F)$ is the probability of the occurrence of event $E$ given that $F$ occurred and is given as

$$P(E|F) = \frac{P(E \cap F)}{P(F)} \tag{A.3}$$

Knowing that $F$ occurred reduces the sample space to $F$, and the part of it where $E$ also occurred is $E \cap F$. Note that equation A.3 is well-defined only if $P(F) > 0$. Because $\cap$ is commutative, we have

$$P(E \cap F) = P(E|F)P(F) = P(F|E)P(E)$$

which gives us *Bayes' formula:*

$$P(F|E) = \frac{P(E|F)P(F)}{P(E)} \tag{A.4}$$

**Bayes' Rule**

When two random variables are jointly distributed with the value of one known, the probability that the other takes a given value can be computed using *Bayes' rule:*

$$P(y|x) = \frac{P(x|y)P_Y(y)}{P_X(x)} = \frac{P(x|y)P_Y(y)}{\sum_y P(x|y)P_Y(y)}$$

Or, in words

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

Note that the denominator is obtained by summing (or integrating if $y$ is continuous) the numerator over all possible $y$ values. The "shape" of $p(y|x)$ depends on the numerator with denominator as a normalizing factor to guarantee that $p(y|x)$ sum to 1. Bayes' rule allows us to modify a prior probability into a posterior probability by taking information provided by $x$ into account.

Bayes' rule inverts dependencies, allowing us to compute $p(y|x)$ if $p(x|y)$ is known. Suppose that $y$ is the "cause" of $x$, like $y$ going on summer vacation and $x$ having a suntan. Then $p(x|y)$ is the probability that someone who is known to have gone on summer vacation has a suntan. This is the *causal* (or predictive) way. Bayes' rule allows us a *diagnostic* approach by allowing us to compute $p(y|x)$: namely, the probability that someone who is known to have a suntan, has gone on summer vacation. Then $p(y)$ is the general probability of anyone's going on summer vacation and $p(x)$ is the probability that anyone has a suntan, including both those who have gone on summer vacation and those who have not.

# Bayesian is “elephant in the room”

# Bayes' Rule

$$\underbrace{P(C \mid \mathbf{x})}_{\text{posterior}} = \frac{\overbrace{P(C)}^{\text{prior}}\,\overbrace{p(\mathbf{x} \mid C)}^{\text{likelihood}}}{\underbrace{p(\mathbf{x})}_{\text{evidence}}}$$

$$P(C=0) + P(C=1) = 1$$

$$p(\mathbf{x}) = p(\mathbf{x} \mid C=1)P(C=1) + p(\mathbf{x} \mid C=0)P(C=0)$$

$$p(C=0 \mid \mathbf{x}) + P(C=1 \mid \mathbf{x}) = 1$$

# The prior

$P(h = f \mid \mathcal{D})$ requires an additional probability distribution:

$$P(h = f \mid \mathcal{D}) \;=\; \frac{P(\mathcal{D} \mid h = f)\, P(h = f)}{P(\mathcal{D})} \;\propto\; P(\mathcal{D} \mid h = f)\, P(h = f)$$

$P(h = f)$ is the **prior**

$P(h = f \mid \mathcal{D})$ is the **posterior**

Given the prior, we have the full distribution


A randomized experiment
Can be deterministic
If we know more information
Of the process

Suppose z is unobservable,
X is the outcome (head, tail)

X= f(z)

Z: initial condition of flipping a coin

# Probability and Inference

- Result of tossing a coin is $\in$ {Heads,Tails}
- Random var $X \in \{1,0\}$

  Bernoulli: $P\{X=1\} = p_o$ $P\{X=0\} = 1 - p_o$

  $P\{X\} = p_o^X (1 - p_o)^{(1-X)}$
- Sample: $\mathbf{X} = \{x^t\}_{t=1}^N$

  Estimation: $p_o = \#\{\text{Heads}\}/\#\{\text{Tosses}\} = \sum_t x^t / N$
- Prediction of next toss:

  Heads if $p_o > ½$, Tails otherwise

# Classification

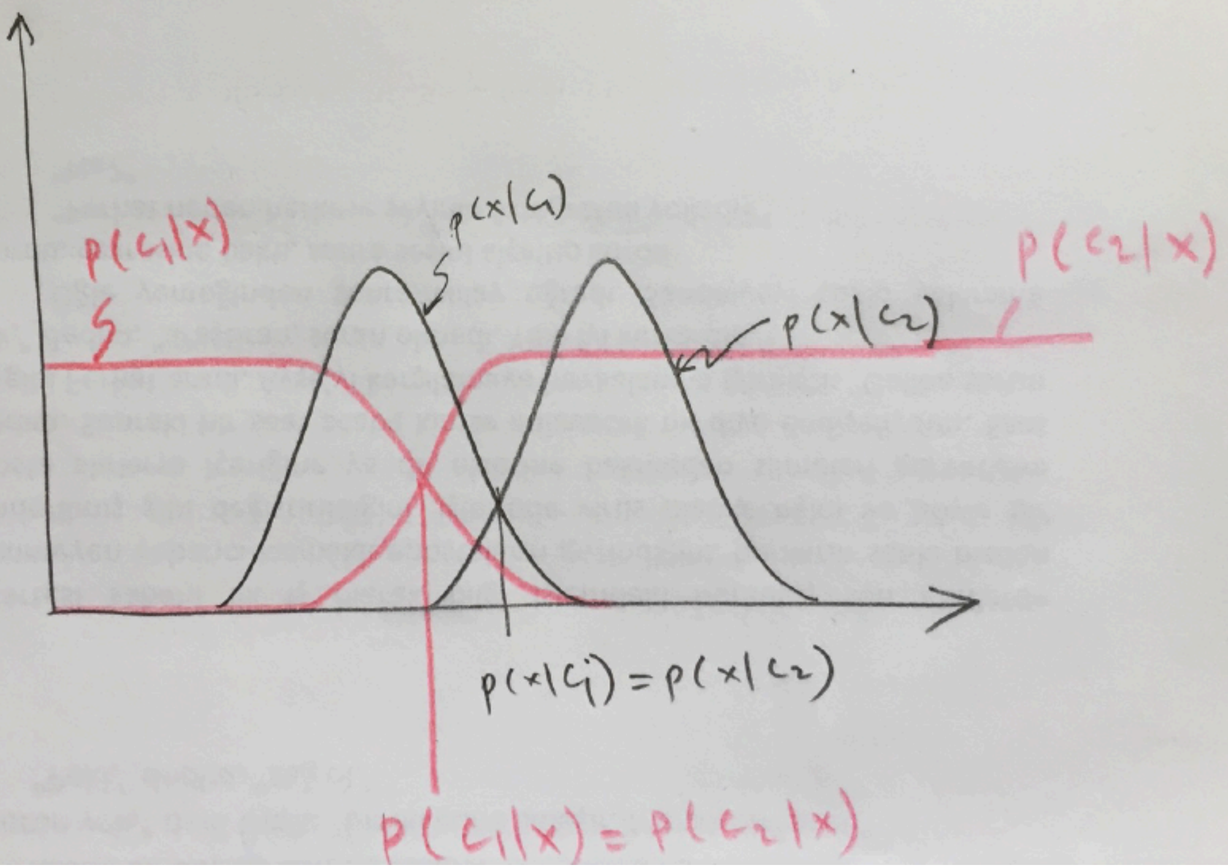
- Credit scoring: Inputs are income and savings.
  Output is low-risk vs high-risk
- Input: $\mathbf{x} = [x_1, x_2]^T$ ,Output: $C \in \{0,1\}$
- Prediction:

$$\text{choose } \begin{cases} C = 1 \text{ if } P(C=1 \mid x_1, x_2) > 0.5 \\ C = 0 \text{ otherwise} \end{cases}$$

or

$$\text{choose } \begin{cases} C = 1 \text{ if } P(C=1 \mid x_1, x_2) > P(C=0 \mid x_1, x_2) \\ C = 0 \text{ otherwise} \end{cases}$$

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



# Bayes' Rule: $K>2$ Classes

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x} \mid C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x} \mid C_k)P(C_k)}$$

$P(C_i) \geq 0$ and $\sum_{i=1}^{K} P(C_i) = 1$

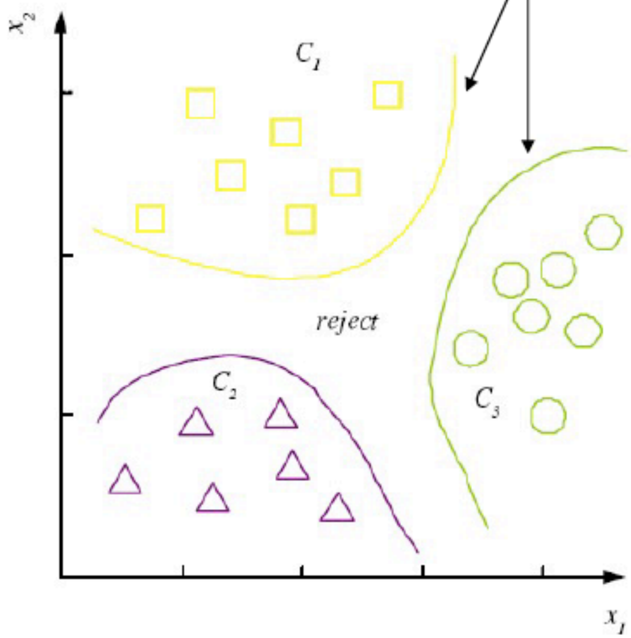
choose $C_i$ if $P(C_i \mid \mathbf{x}) = \max_k P(C_k \mid \mathbf{x})$

# Discriminant Functions

choose $C_i$ if $g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})$

$$g_i(\mathbf{x}) = \begin{cases} -R(\alpha_i \mid \mathbf{x}) \\ P(C_i \mid \mathbf{x}) \\ p(\mathbf{x} \mid C_i) P(C_i) \end{cases}$$

$g_i(\mathbf{x}), i = 1, \dots, K$




$K$ decision regions $\mathcal{R}_1, \dots, \mathcal{R}_K$

$$\mathcal{R}_i = \{\mathbf{x} \mid g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})\}$$

# $K$=2 Classes

- Dichotomizer ($K$=2) vs Polychotomizer ($K$>2)
- $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$

$$\text{choose} \begin{cases} C_1 \text{ if } g(\mathbf{x}) > 0 \\ C_2 \text{ otherwise} \end{cases}$$

- *Log odds:* $\log \dfrac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})}$

# Losses and Risks

- Actions: $\alpha_i$
- Loss of $\alpha_i$ when the state is $C_k$ : $\lambda_{ik}$
- Expected risk (Duda and Hart, 1973)

$$R(\alpha_i \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k \mid \mathbf{x})$$

$$\text{choose } \alpha_i \text{ if } R(\alpha_i \mid \mathbf{x}) = \min_k R(\alpha_k \mid \mathbf{x})$$

# Risk CIA case: non-symmetric situation

Action 1 you

Action 2 intruder

Risk is defined as

$\lambda$

| | k=1 | k=2 |
|---|---|---|
| $\alpha 1$ | 0 | 1000 |
| $\alpha 2$ | 1 | 0 |

# Association Rules

- Association rule: $X \rightarrow Y$
- *People who buy/click/visit/enjoy X are also likely to buy/click/visit/enjoy Y.*
- A rule implies association, not necessarily causation.

# Association measures

- Support ($X \to Y$):

$$P(X,Y) = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}}$$

- Confidence ($X \to Y$):

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)} = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought } X\}}$$

- Lift ($X \to Y$):

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

# Example

| Transaction | Items in basket |
|---|---|
| 1 | milk, bananas, chocolate |
| 2 | milk, chocolate |
| 3 | milk, bananas |
| 4 | chocolate |
| 5 | chocolate |
| 6 | milk, chocolate |

milk $\to$ bananas: Support: $\dfrac{\#\{\text{milk, bananas}\}}{N}$ $\dfrac{2}{6}$ $P(\text{milk} \wedge \text{bananas})$

Confidence: $\dfrac{\#\{\text{milk, bananas}\}}{\#\{\text{milk}\}}$ $\dfrac{2}{4}$ $P(\text{bananas} \mid \text{milk}\}$

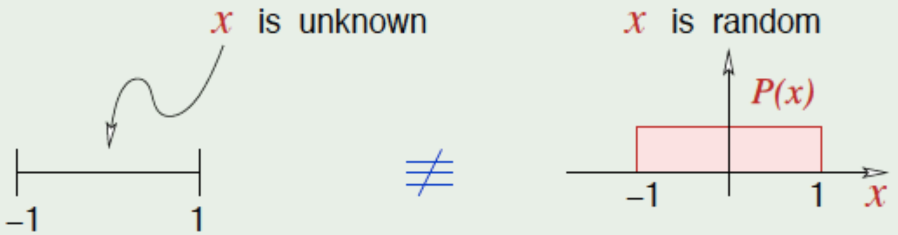
bananas $\to$ milk Confidence $= \dfrac{\#\{\text{milk, bananas}\}}{\#\{\text{bananas}\}} = \dfrac{2}{2} = 1.$

# Bayesian update as an algorithm

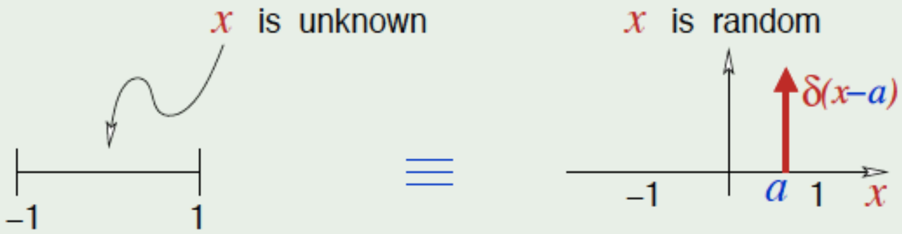
# A simple case to make a point

## A prior is an assumption

Even the most "neutral" prior:




The true equivalent would be:



## When is Bayesian learning justified?

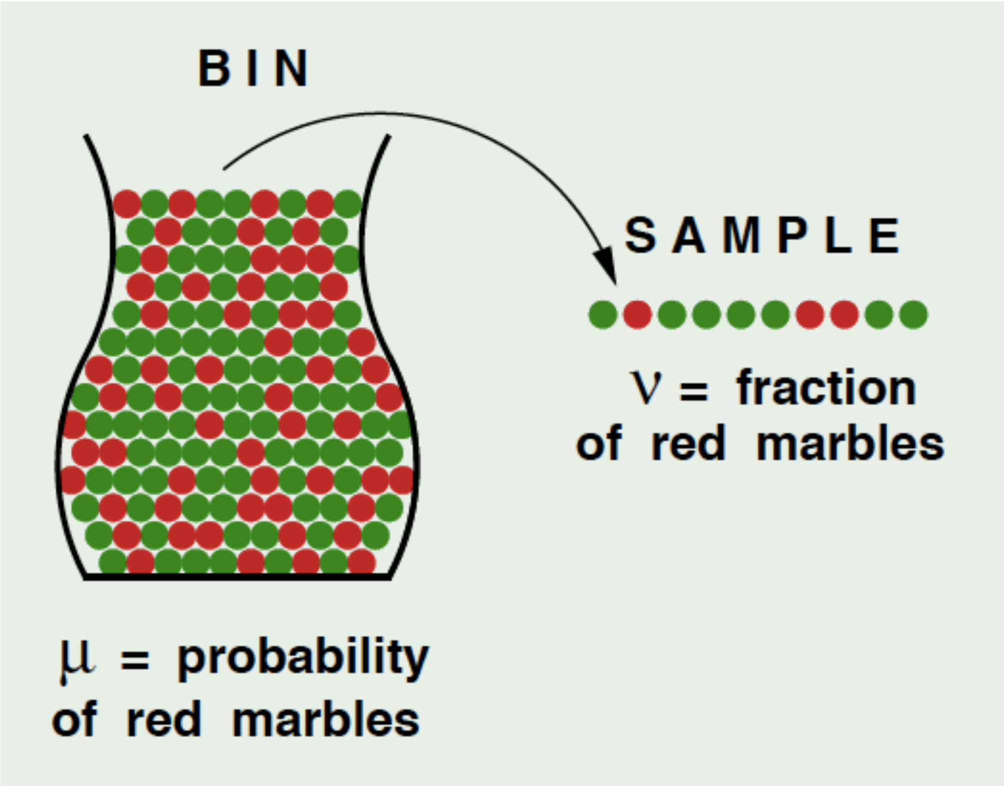
1. The prior is **valid**

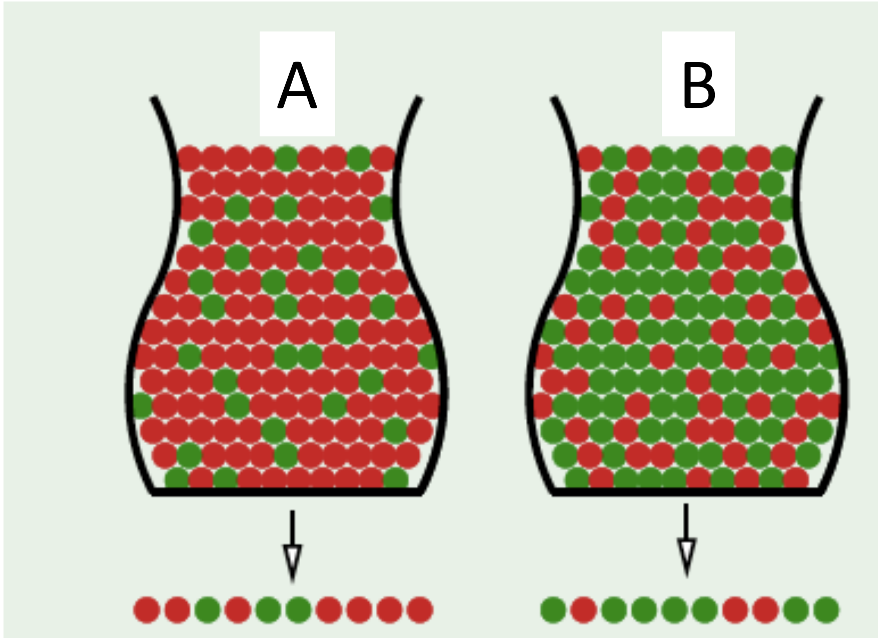
   trumps all other methods

2. The prior is **irrelevant**

   just a computational catalyst

# Bin example




まなびのずかん統計学の図鑑, [涌井良幸](), [涌井貞美]()

Suppose we have two possible bins from A and B. If the bin is A, the probability is given

$P(\text{ red } | H=A) = 0.6$

$P(\text{green } | H=A) = 0.4$

If the bin is B, the probability is given by

$P(\text{ red } | H=B) = 0.3$

$P(\text{green } | H=B) = 0.7$

How can we distinguish the bin is A or B?

# Bayesian update approach

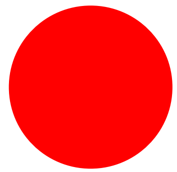
- First, we model with prior probability and assume some initial prior, e.g.

$$P(H=A) = 1/2$$
$$P(H=B) = 1/2$$

- Second we draw the ball from the bin
- Third step is calculation of the posterior probability
- Four step, set the new prior and go to the first step with this new prior

If the ball is red

$$P(\,H = A|red) = \frac{P(red|H{=}A)P(H = A)}{P(red)}$$



$$P(\,H = B|red) = \frac{P(red|H{=}B)P(H = B)}{P(red)}$$

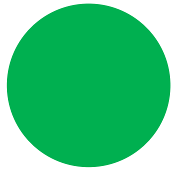
$$P(red) = P(red \mid H{=}A)P(H{=}A) + P(red \mid H{=}B)P(H{=}B)$$

We then set new prior to be

$$P(H{=}A) = P(H{=}A \mid red)$$
$$P(H{=}B) = P(H{=}B \mid red)$$

If the ball is green

$$P(\,H = A|green) = \frac{P(green|H{=}A)P(H = A)}{P(green)}$$



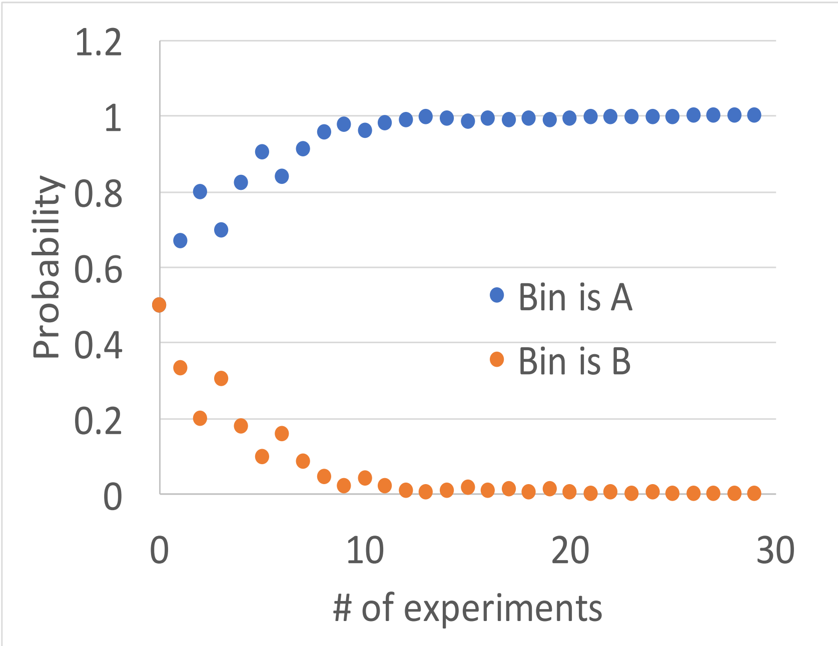
$$P(\,H = B|green) = \frac{P(green|H{=}B)P(H = B)}{P(green)}$$

P( green ) = P(green | H=A)P(H=A)+ P(green | H=B)P(H=B)

We then set new prior to be

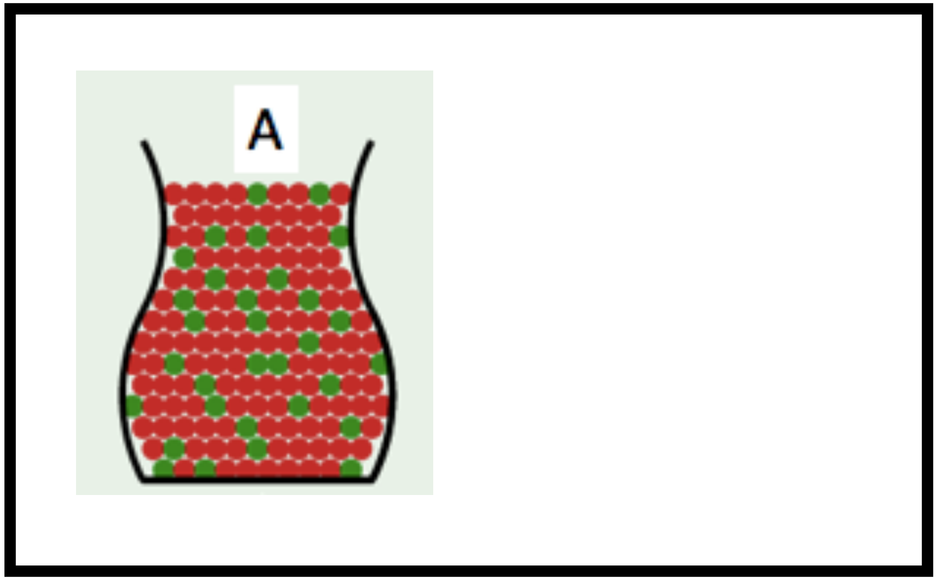
P(H=A)= P(H=A | green)
P(H=B)= P(H=B | green)

# Simulation example




Posterior probablity calculated from the simulation.

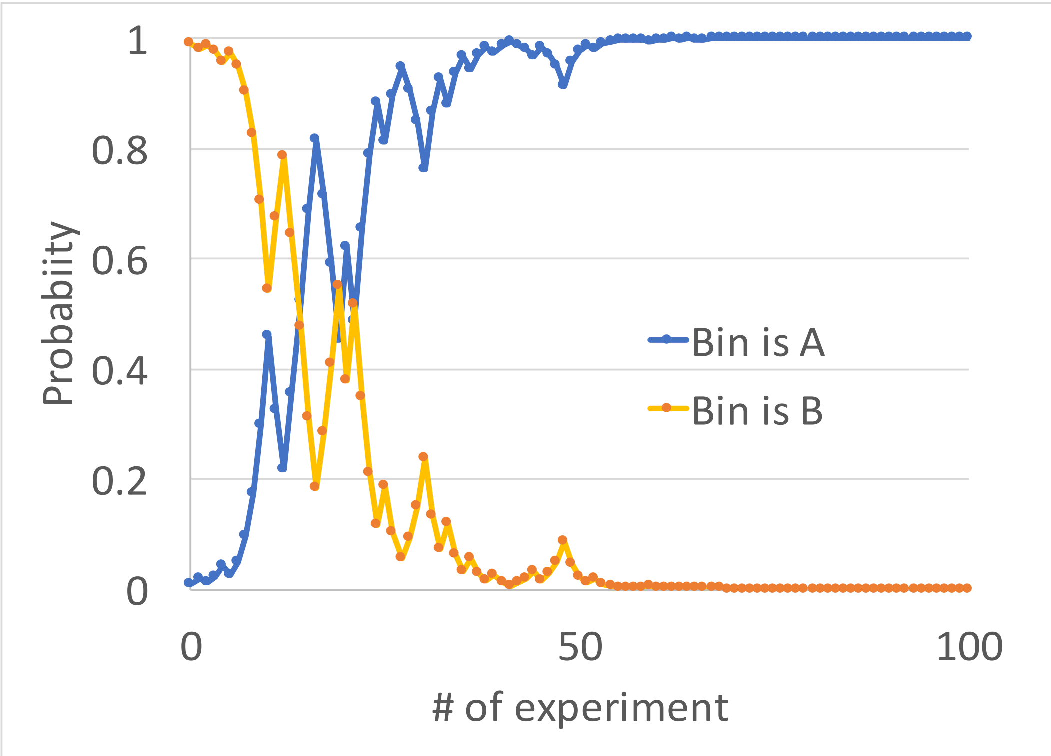
## Black box




Initial prior

P(H=A) =1/2
P(H=B)= 1/2

# A more radical case




WHAT IF THE PRIOR IS FAR FROM TRUE?

$$P(H=A) = 0.01$$
$$P(H=B) = 0.99$$

# TO SUMMERIZE

- Bayesian probability formalism automatically make a learning algorithm.
- We first model the system by making assumption on the prior and use Bayesian formula to update our current belief.
- The process is iterated as new evidence is generated from our experiment.
- We establish this algorithm indeed works regardless which initial prior we start with.
- Therefore we can consider prior is “computational catalyst” to get the computation going.

```python
import random

#initial prior probabity
priorA = 0.01    #prior probability
priorB = 0.99    # prior proability

Pgreen_A= 4/10   # Probability of green given Bin A
Pred_A=6/10      # Probability of red given Bin A
Pgreen_B= 7/10   # Probability of green given Bin B
Pred_B= 3/10     # Probability of red given Bin B
#define these probabity and set to zero for convenience
P_A_red=0
P_A_green=0
P_B_red=0
P_B_green=0
num_seq=100

for seq in range(num_seq):
    x=random.uniform(0,1)

    if x>=0 and x<=Pgreen_A:
        P_A_green= Pgreen_A*priorA/(Pgreen_A*priorA+Pgreen_B*priorB)
        P_B_green=1-P_A_green
        priorA=P_A_green
        priorB=P_B_green
        print( 'green', seq+1, round(P_A_green,4))

    else:
        P_A_red= Pred_A*priorA/(Pred_A*priorA+Pred_B*priorB)
        P_B_red=1-P_A_red
        priorA=P_A_red
        priorB=P_B_red
        print( 'red', seq+1, round(P_A_red,4))
```