Randomized Algorithms

Essence

- Probabilistic algorithms or sampling
- A degree of randomness is part of the logic (using a random number generator)
- Results can be different for different runs
- Algorithms may
  - Produce incorrect results (Las Vegas)
  - Fail to produce a result (Monte Carlo)

Example

- An array of n elements, half are ‘a’ and half are ‘b’
- Find an ‘a’ in the array

- findA_LV
  - Repeat
    - Randomly select one out of n elements
  - Until ‘a’ is found
- End

- findA_MC
  - count = 0
  - Repeat
    - Randomly select one out of n elements
    - count ++
  - Until ‘a’ is found or count > k
- End

Example (cont)

- An array of n elements, half are ‘a’ and half are ‘b’
- Find an ‘a’ in the array

- findA_LV
  - Always succeeds
  - Random run time (average is O(1))

- findA_MC
  - Succeed with P=1-(1/2)^k
  - Max run time is O(k)

More Example

- Quick sort
  - The selection of pivot is random
  - Always produce the correct results, but runtime is random
  - Average runtime is O(nlogn)

General Curve Fitting

$y = f(x, a_1, a_2, \cdots, a_n)$	$y = ax^2 + bx + c$
n input points $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$	3 input points $(1,1), (2,2), (3,1)$
n equations	3 equations
$y_1 = f(x_1, a_1, a_2, \cdots, a_n)$ $y_2 = f(x_2, a_1, a_2, \cdots, a_n)$ $\cdots$ $y_n = f(x_n, a_1, a_2, \cdots, a_n)$	$a + b + c = 1$ $4a + 2b + c = 2$ $9a + 3b + c = 1$
$\begin{bmatrix} f(x_1) \\ \cdots \\ f(x_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \cdots \\ y_n \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$
solve for $a_1, \cdots, a_n$	$a = -1, b = 4, c = -2$

General Least Square Regression

$$\min_{\theta=(a_0,a_1,\ldots,a_{n-1})} E$$

$$\text{where } E = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 =$$

$$\min_{\theta=(a_0,a_1,\ldots,a_{n-1})} \sum_{i=1}^{m} (y_i - (a_{n-1}x_i^{\,n-1} + a_{n-2}x_i^{\,n-2} + \ldots + a_1 x_i^{\,1} + a_0))^2$$

$$\frac{\partial E}{\partial a_j} = 0,\ j = 1,\ldots,n$$

$$\sum_{i=1}^{m} x_i^{\,j} (y_i - (a_{n-1}x_i^{\,n-1} + a_{n-2}x_i^{\,n-2} + \ldots + a_1 x_i^{\,1} + a_0)) = 0$$

General Least Square Regression

$$\sum_{i=1}^{m} x_i^{\,j}\left(y_i - \left(a_{n-1}x_i^{\,n-1} + a_{n-2}x_i^{\,n-2} + \ldots + a_1 x_i^{\,1} + a_0\right)\right) = 0$$

$$\left(\sum_{i=1}^{m} x_i^{\,j} x_i^{\,n-1}\right)a_{n-1} + \left(\sum_{i=1}^{m} x_i^{\,j} x_i^{\,n-2}\right)a_{n-2} + \ldots + \left(\sum_{i=1}^{m} x_i^{\,j} x_i^{\,1}\right)a_1 + \left(\sum_{i=1}^{m} x_i^{\,j}\right)a_0 = \sum_{i=1}^{m} x_i^{\,j} y_i$$

$$\begin{bmatrix} \sum_{i=1}^{m} x_i^{\,n-1} x_i^{\,n-1} & \sum_{i=1}^{m} x_i^{\,n-1} x_i^{\,n-2} & \cdots & \sum_{i=1}^{m} x_i^{\,n-1} \\ \sum_{i=1}^{m} x_i^{\,n-2} x_i^{\,n-1} & \sum_{i=1}^{m} x_i^{\,n-2} x_i^{\,n-2} & \cdots & \sum_{i=1}^{m} x_i^{\,n-2} \\ \cdots & \cdots & \cdots & \vdots \\ \sum_{i=1}^{m} x_i^{\,n-1} & \sum_{i=1}^{m} x_i^{\,n-2} & \cdots & \sum_{i=1}^{m} 1 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_o \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} x_i^{\,n-1} y_i \\ \sum_{i=1}^{m} x_i^{\,n-2} y_i \\ \vdots \\ \sum_{i=1}^{m} y_i \end{bmatrix}$$

Homework #4

$$\min E, \text{where } E = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

$$\min E \Rightarrow \min_{\theta=(a,b)} \sum_{i=1}^{m} (y_i - (ax_i + b))^2$$

$$\frac{\partial E}{\partial a} = \frac{\partial E}{\partial b} = 0$$

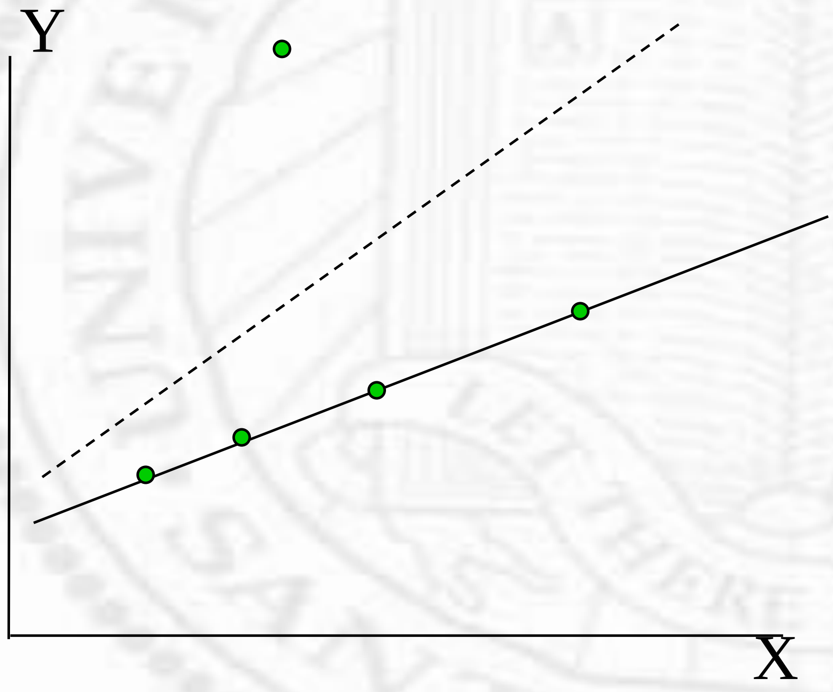
$$\frac{\partial E}{\partial a} = 2\sum_{i=1}^{m} x_i (y_i - (ax_i + b)) = 0 \Rightarrow (\sum x_i^2)a + (\sum x_i)b = \sum x_i y_i$$

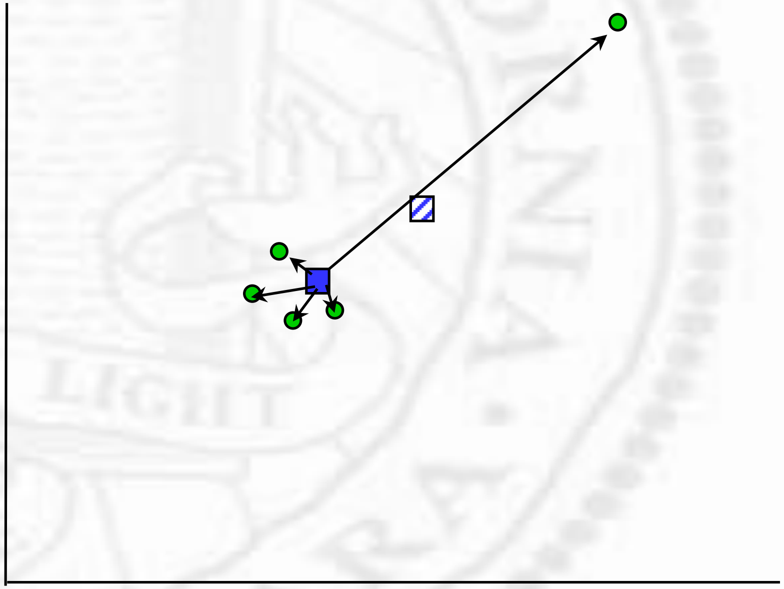
$$\frac{\partial E}{\partial b} = 2\sum_{i=1}^{m} (y_i - (ax_i + b)) = 0 \Rightarrow (\sum x_i)a + (\sum 1)b = \sum y_i$$

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & \sum 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

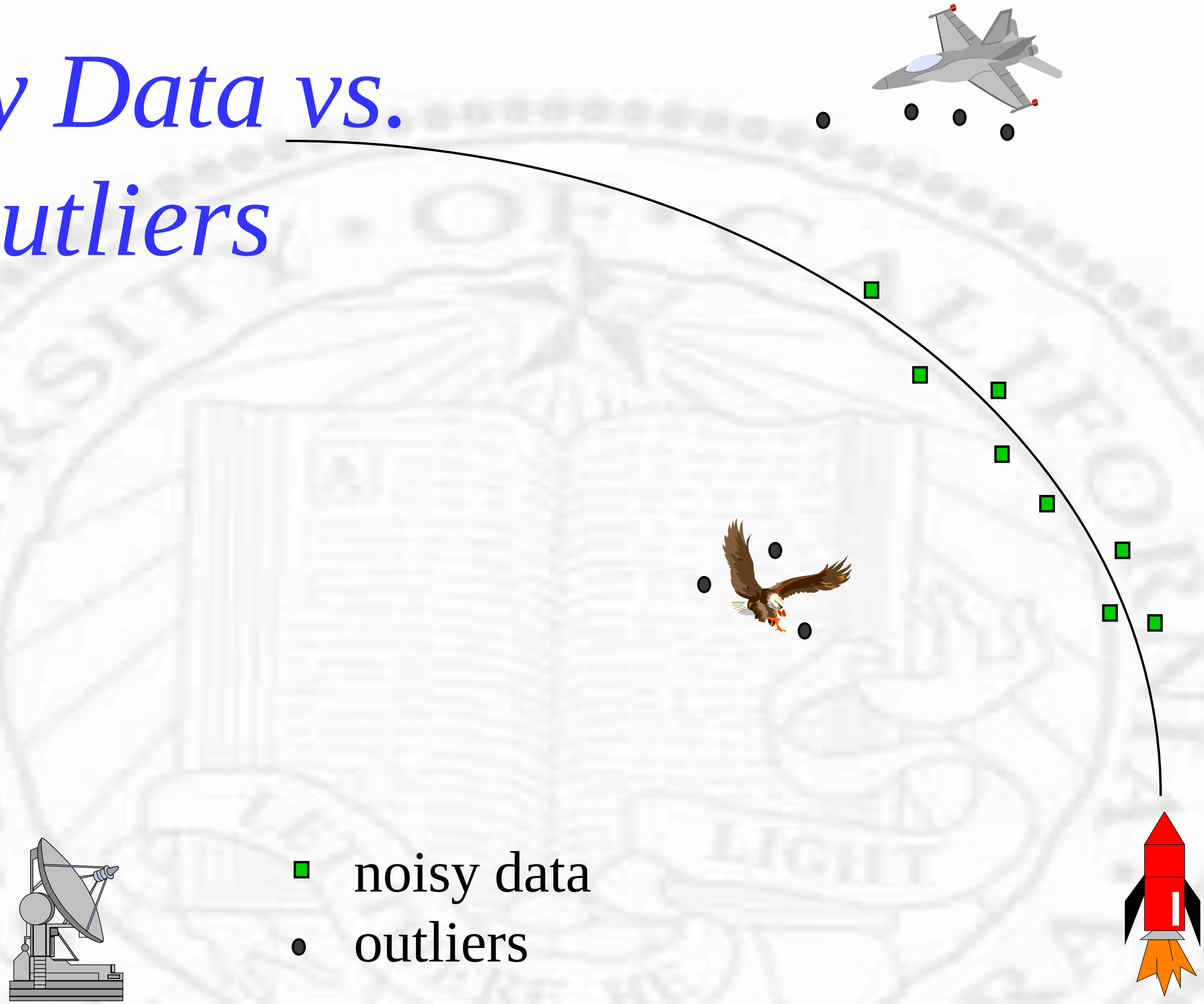
Caveats

- LS
  - Democracy, everybody gets an equal say
  - Perform badly with "outliers"



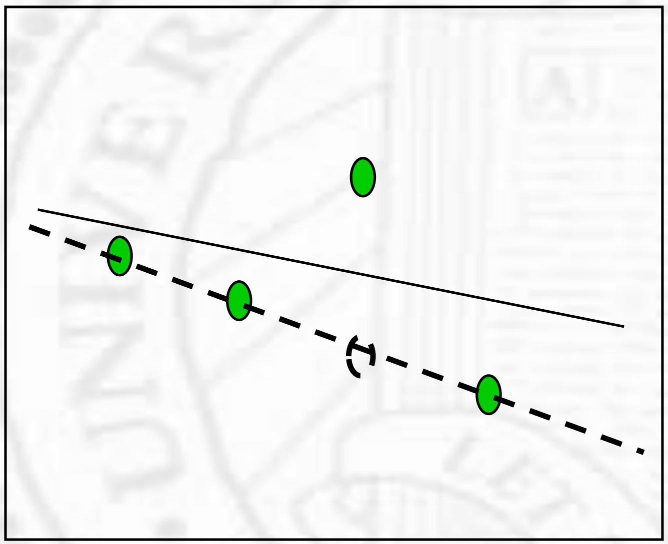


Noisy Data vs. Outliers

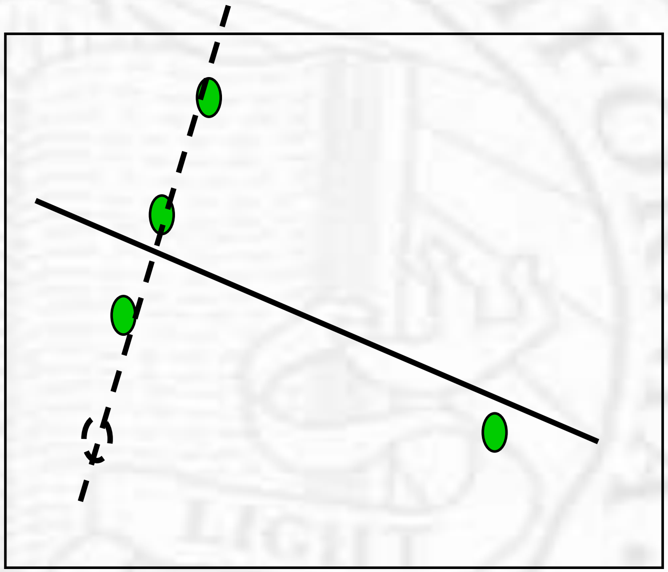


Outliers

- Outliers (y)



- Outliers (x, leverage points)



Randomized Algorithm

- Choose p points at random from the set of n data points
- Compute the fit of model to the p points
- Compute the median of the fitting error for the remaining n-p points
- The fitting procedure is repeated until a fit is found with sufficiently small median of squared residuals or up to some predetermined number of fitting steps (Monte Carlo Sampling)

How Many Trials?

- Well, theoretically it is $C(n,p)$ to find all possible p-tuples
- Very expensive

$1-(1-(1-\varepsilon)^p)^m$

ε : fraction of bad data

$(1-\varepsilon)$: fraction of good data

$(1-\varepsilon)^p$: all p samples are good

$1-(1-\varepsilon)^p$: at least one sample is bad

$(1-(1-\varepsilon)^p)^m$: got bad data in all m tries

$1-(1-(1-\varepsilon)^p)^m$: got at least one good p set in m tries

How Many Trials (cont.)

- Make sure the probability is high (e.g. >95%)
- given p and epsilon, calculate m

p	5%	10 %	20 %	25 %	30 %	40 %	50 %
1	1	2	2	3	3	4	5
2	2	2	3	4	5	7	11
3	2	3	5	6	8	13	23
4	2	3	6	8	11	22	47
5	3	4	8	12	17	38	95

Best Practice

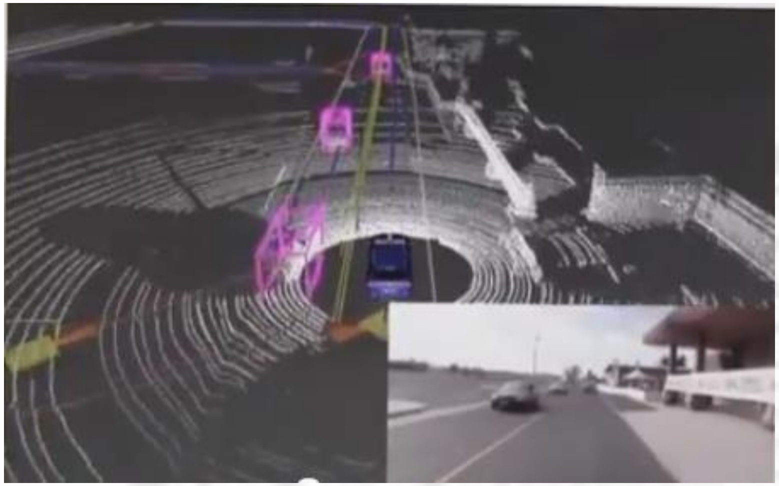
- Randomized selection can completely remove outliers
- "plutocratic"
- Results are based on a small set of features

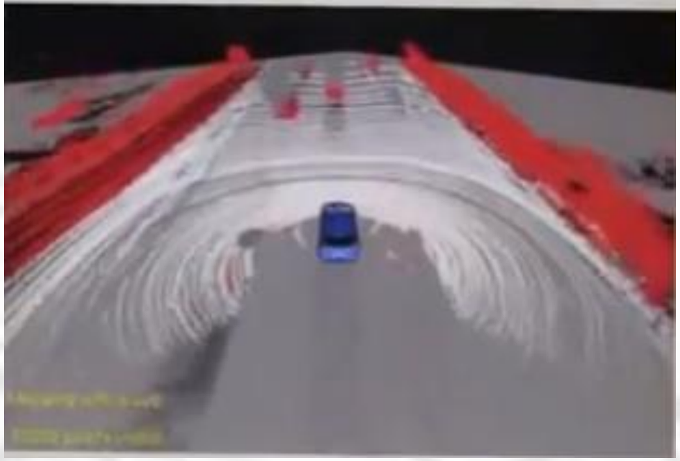
- LS is most fair, everyone get an equal say
- "democratic"
- But can be seriously influenced by bad data

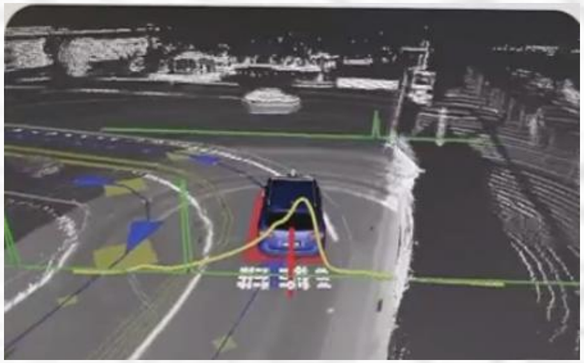
- Use randomized algorithm to remove outliers
- Use LS for final "polishing" of results (using all "good" data)
- Allow up to 50% outliers theoretically

Navigation

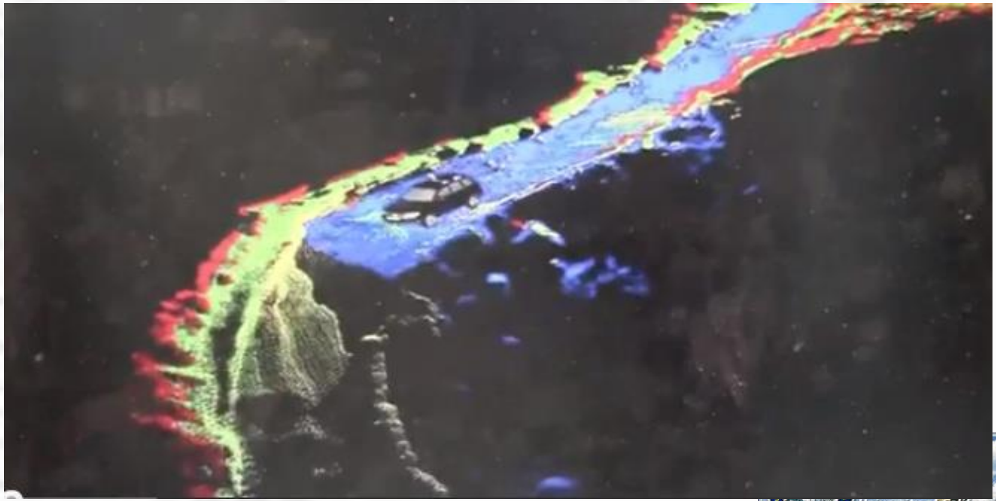
- Autonomous Land Vehicle (ALV), Google's Self-Driving Car, etc.
- One important requirement: track the position of the vehicle
- Kalman Filter, loop of
  - (Re)initialization
  - Prediction
  - Observation
  - Correction








intel
MDV
NXP

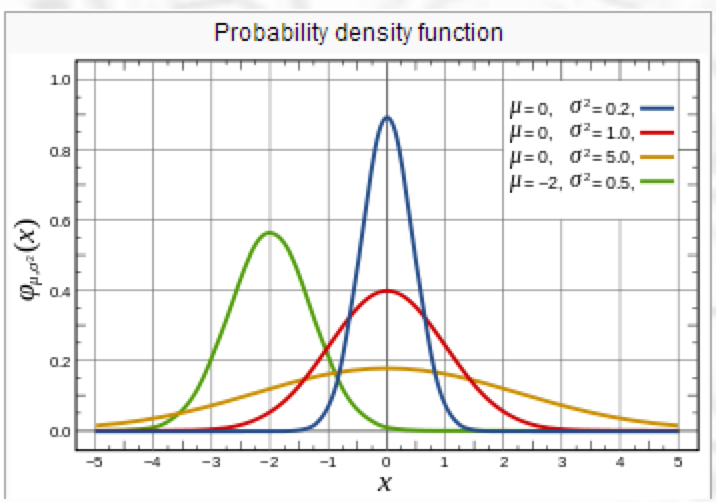


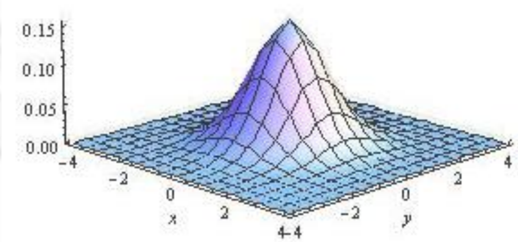
Navigation

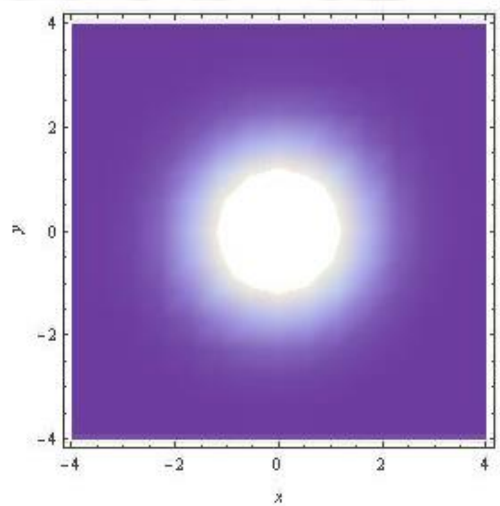
- Hypothesis and verification
  - Classic Approach like Kalman Filter maintains a single hypothesis
  - Newer approach like particle filter maintains multiple hypotheses (Monte Carlo sampling of the state space)

Single Hypothesis

- If the “distraction” – noise – is white and Gaussian
- State-space probability profile remains Gaussian (a single dominant mode)
- Evolving and tracking the mean, not a whole distribution







Multi-Hypotheses

- The distribution can have multiple modes
- Sample the probability distribution with “importance” rating
- Evolve the whole distribution, instead of just the mean

Key – Baeys Rule

$$P(s_i \mid o) = \frac{p(o, s_i)}{p(o)} = \frac{p(o \mid s_i) P(s_i)}{p(o)} \approx p(o \mid s_i) P(s_i)$$

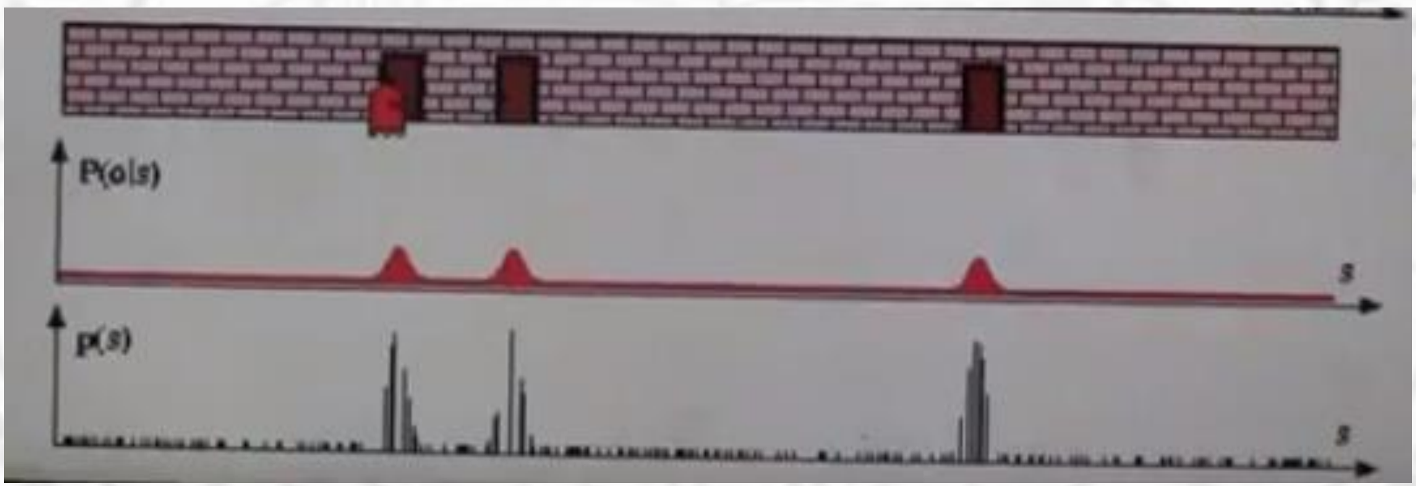
$s : state$

$o : observation$

- In the day time, some animal runs in front of you on the bike path, you know exactly what it is (p(o|si) is sufficient)
- In the night time, some animal runs in front of you on the bike path, you can hardly distinguish the shape (p(o|si) is low for all cases, but you know it is probably a squirrel, not a lion because of p(si))

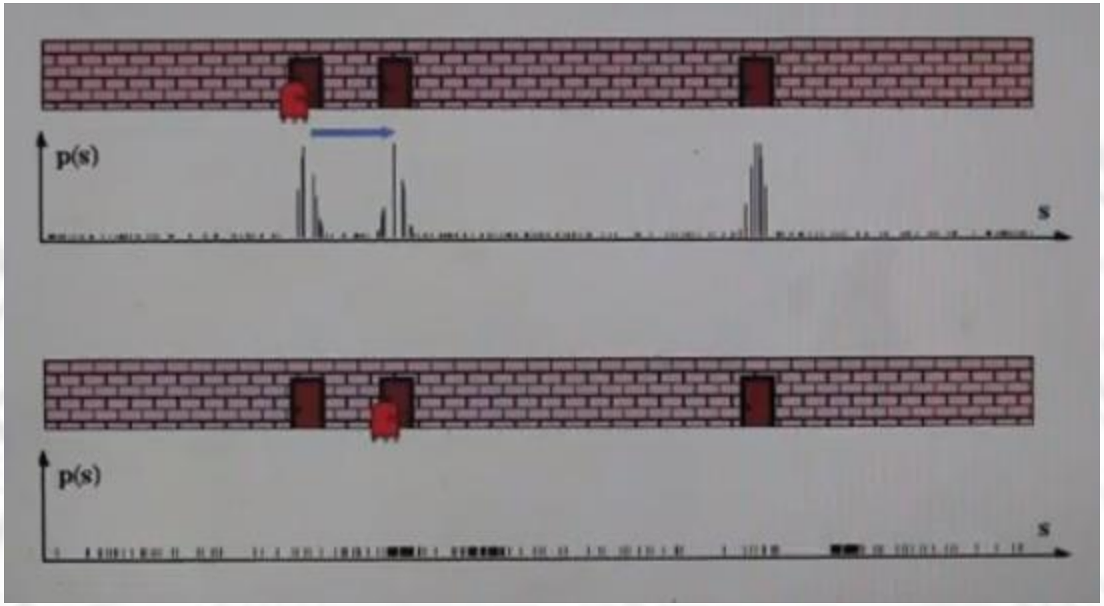
Initialization: before observation and measurement

Observation: after seeing a door

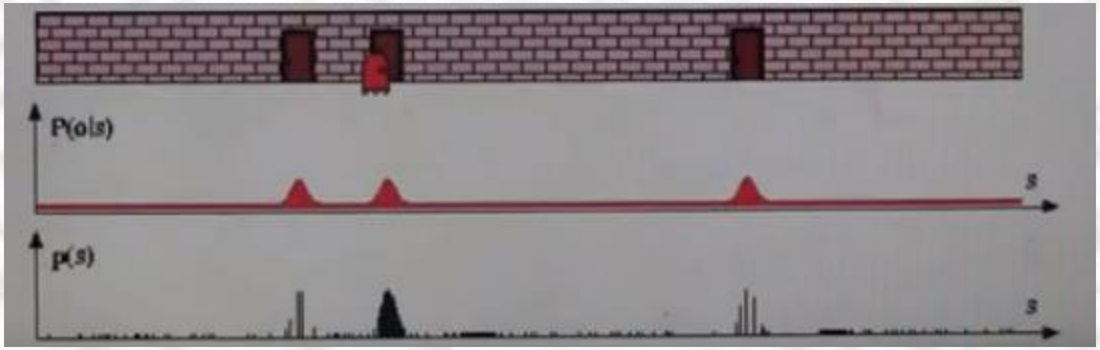


P(s): probability of state
P(o|s): probably of observation given current state

Prediction : internal mechanism saying that robot moves right



Correction : prediction is weighed by confirmation with observation



PARTICLE FILTERS FOR LOCALIZATION

MONTE CARLO LOCALIZATION

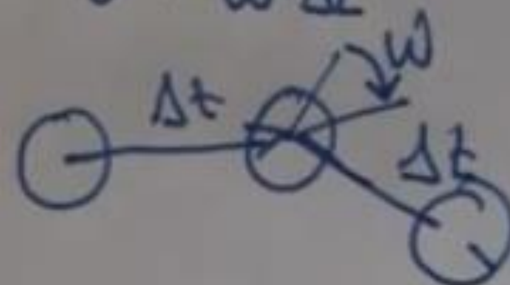
$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$

Δt

velocity v

turning velocity ω

$$\left.\begin{aligned} x' &= x + v \cdot \Delta t \cdot \cos\theta \\ y' &= y + v \cdot \Delta t \cdot \sin\theta \\ \theta' &= \theta + \omega \cdot \Delta t \end{aligned}\right]$$



Particles + weights

controls

measurements

ALGORITHM PARTICLE FILTER(S, U, Z)

$S' = \emptyset$ $\quad \eta = 0$ Total weights

new particles + weights

For $i = 1 \ldots n$

Sample $j \sim \{w\}$ with replacement

$x' \sim p(x' \mid u, s_j)$

$w' = p(z \mid x')$ $\qquad \eta = \eta + w'$

$S' = S' \cup \{\langle x', w' \rangle\}$

end

For $i = 1 \ldots n$

$w_i = \frac{1}{\eta} w_i$

Why Be Stochastic?

- More Choices – remove bad data
- More Alternatives – sample the problem states based on likelihood