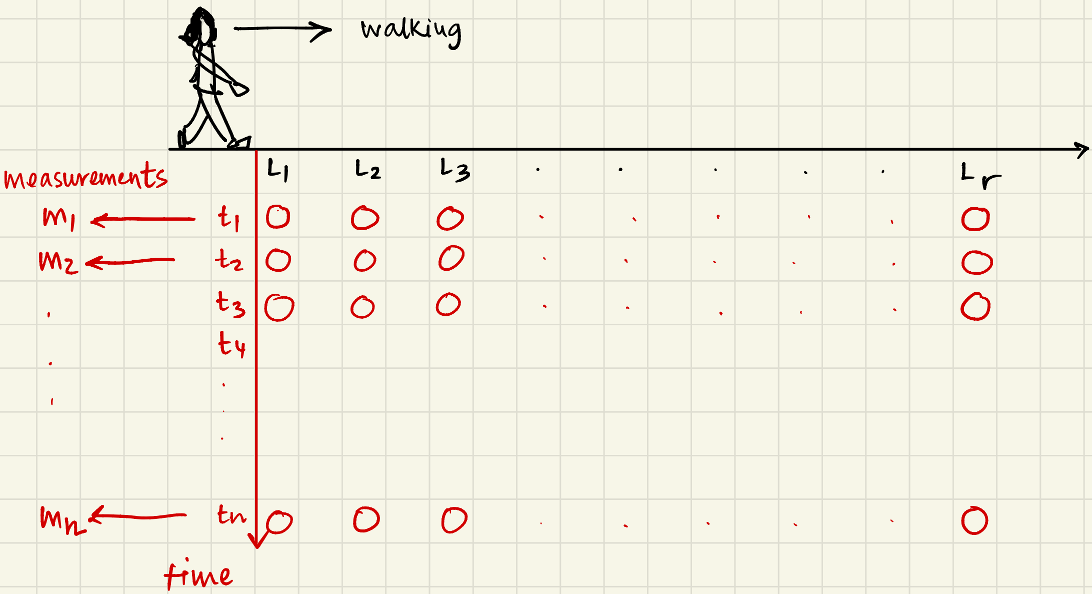
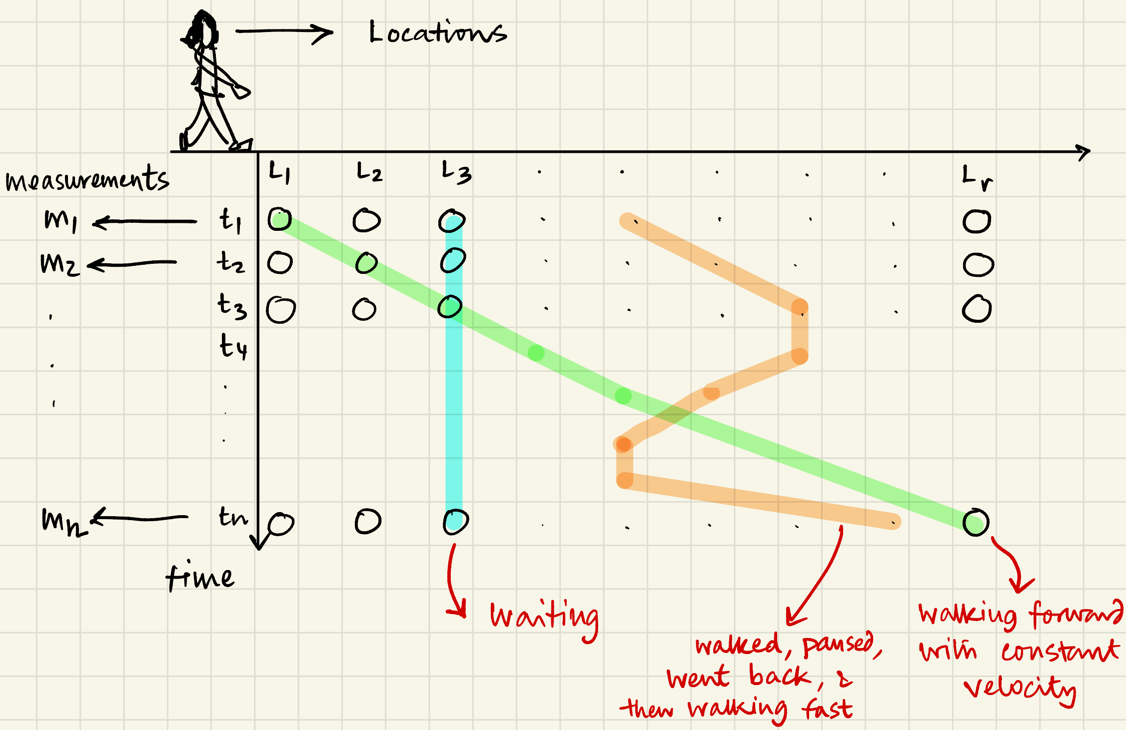
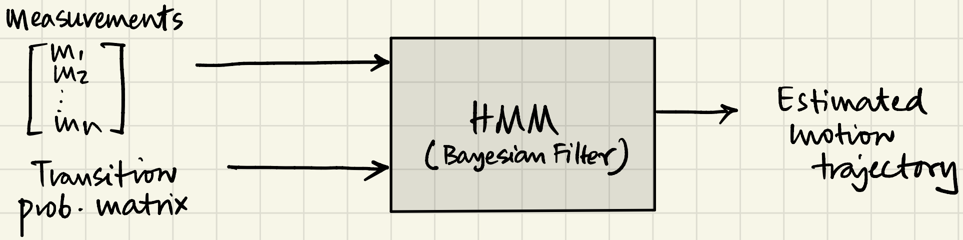
# HIDDEN MARKOV MODELS (HMM)





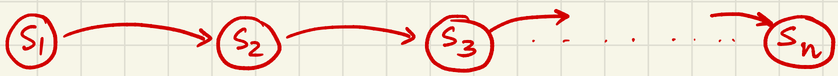

⑦ Formulating the state transition diagram :

Let $S_k$ denote the state of the subject at time $k$

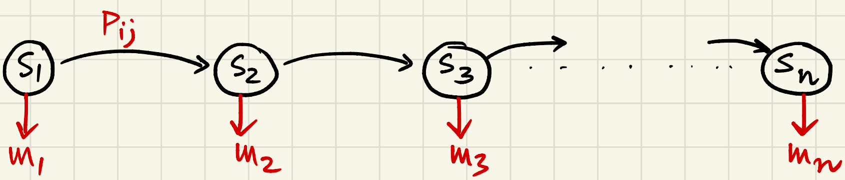
→ $S_k$ is a random variable, i.e., $S_k = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix}$ $\forall k \in [1, n]$

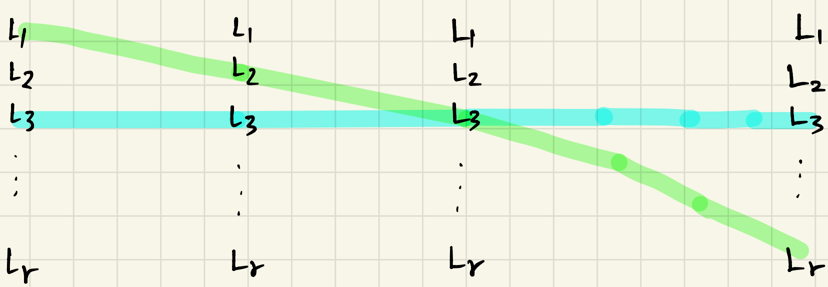
→ $S_k$ called the "state variable"

The human's walking motion is captured in




And the measurement and motion model is available for each state




⑧ Key Question: Where is/was the human at time $t_k$ ?

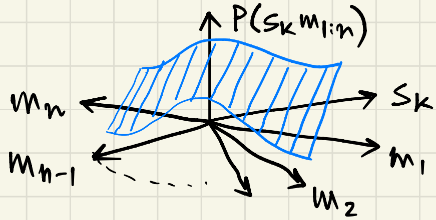
$P(S_k \mid m_{1:k})$ or $P(S_k \mid m_{1:n})$

Is this : Posterior or likelihood ?

Do you have an intuitive feel for $P(s_k \mid m_{1:n})$ ?
If not, fall back on visualizing them as vectors

$$P\left( s_k = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \middle| \; m_1 = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \; m_2 = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \cdots \; m_n = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \right)$$

$$= \frac{P(s_k, m_{1:n})}{P(m_{1:n})} =$$




$$\overline{P\left( m_1 = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \; m_2 = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \cdots \; m_n = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \right)}$$

- From this joint distribution (in numerator) you want to know which value of $s_k = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix}$ has the max probability given the $m_{1:n}$ measurements you already have.

- The denominator is same for all $s_k = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix}$, so only numerator matters. But we don't know that joint distribution.

- Turn this posterior to likelihoods : $\dfrac{P(m_{1:n} \mid s_k)\, P(s_k)}{P(m_{1:n})}$

Likelihood ... and that is not hard because it's the sensor's measurement quality

Who cares! We only want to compare the numerator, so ignore denominator.

Hmmm! This depends on where I was last. So fn of $P(s_{k-1})$

⊝ Let's do this mathematically now.

⊝ Let's start with a basic result for $P(\underbrace{s_{1:n} \mid m_{1:n}}_{\text{trajectory given measurements}})$

$$P(s_{1:n} \mid m_{1:n}) = \frac{P(s_{1:n}, m_{1:n})}{P(m_{1:n})} \propto P(s_{1:n}, m_{1:n})$$

$$P(s_{1:n}, m_{1:n}) \overset{\text{chain rule}}{=} P(m_n \mid m_{1:n-1}, s_{1:n})\, P(m_{n-1} \mid m_{1:n-2}, s_{1:n}) \cdots$$

$$\cdots P(m_1 \mid s_{1:n})\, P(s_n \mid s_{1:n-1}) \cdots P(s_2 \mid s_1) P(s_1)$$

Markov

$$= P(m_n \mid s_n)\, P(m_{n-1} \mid s_{n-1}) \cdots P(m_1 \mid s_1)\, P(s_n \mid s_{1:n-1}) \cdots P(s_2 \mid s_1) P(s_1)$$

$$\boxed{P(s_{1:n}, m_{1:n}) = P(m_1 \mid s_1) P(s_1) \prod_{i=2}^{n} P(m_i \mid s_i)\, P(s_i \mid s_{i-1})}$$

⊝ Now we want $P(s_k \mid m_{1:n})$

$$P(s_k \mid m_{1:n}) \propto P(s_k, m_{1:n}) = P(s_k, m_{1:k}, m_{k+1:n})$$

$$= P(m_{k+1:n} \mid s_k, m_{1:k})\, P(s_k, m_{1:k})$$

$$= P(m_{k+1:n} \mid s_k)\, P(s_k \mid m_{1:k})\, P(m_{1:k})$$

$$= P(s_k \mid m_{1:k}) \cdot P(m_{k+1:n} \mid s_k)$$

Forward (online)

Probability that murder suspect is at $S_k$ = green st. given $k=4$ recent surveillance camera measurements of main street → wright street → 6th street

Backward (offline)

Probability that 4th to 8th measurement are Neil St. → Kirby road → Lincoln drive → university avenue, given suspect's $k^{th}$ time location $S_k$ = green street.

➔ Let's look at the forward component $P(s_k \mid m_{1:k})$

$$P(s_k \mid m_{1:k}) = \frac{P(s_k, m_{1:k})}{P(m_{1:k})} \propto P(s_k, m_{1:k})$$

By marginalizing

$$\text{RHS} = \sum_{s_{k-1}} P(s_k, s_{k-1}, m_{1:k})$$

$$= \sum_{s_{k-1}} P(m_k \mid s_k, s_{k-1}, m_{1:k-1})\, P(s_k, s_{k-1}, m_{1:k-1})$$

$$= \sum_{s_{k-1}} P(m_k \mid s_k)\, P(s_k \mid s_{k-1}, m_{1:k-1})\, P(s_{k-1}, m_{1:k-1})$$

$$P(s_k, m_{1:k}) = \underbrace{\sum_{s_{k-1}} P(m_k \mid s_k)\, P(s_k \mid s_{k-1})\, P(s_{k-1}, m_{1:k-1})}_{\text{call this } \alpha_k}$$

$$\therefore \quad \alpha_k = \sum_{s_{k-1}} P(m_k \mid s_k)\, P(s_k \mid s_{k-1})\, \alpha_{k-1}$$

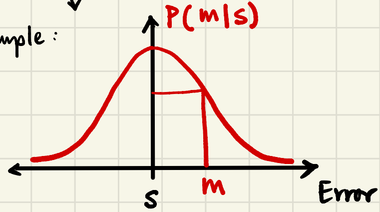
Dynamic program

➔ Initial condition $P(s_1, m_1) = P(m_1 \mid s_1) P(s_1)$ needs to be known.

Errors of sensors derived from their data sheets.

perhaps all locations are equally probable

Example:



→ Now let's look at the backward part : $P(m_{k+1:n} \mid s_k)$

$$P(m_{k+1:n} \mid s_k) = \frac{P(m_{k+1:n}, s_k)}{P(s_k)}$$

$$= \frac{1}{P(s_k)} \sum_{s_{k+1}} P(m_{k+1:n}, s_k, s_{k+1})$$

$$= \frac{1}{P(s_k)} \sum_{s_{k+1}} P(m_{k+2:n} \mid s_{k+1}, s_k, m_{k+1}) \cdot P(m_{k+1} \mid s_{k+1}, s_k) \cdot P(s_{k+1} \mid s_k) P(s_k)$$

Markov

$$= \sum_{s_{k+1}} P(m_{k+2:n} \mid s_{k+1}) P(m_{k+1} \mid s_{k+1}) P(s_{k+1} \mid s_k)$$

Say LHS $= P(m_{k+1:n} \mid s_k) = \beta_k$

$$\therefore \quad \beta_k = \sum_{s_{k+1}} \beta_{k+1} \, P(m_{k+1} \mid s_{k+1}) \, P(s_{k+1} \mid s_k)$$

Dynamic program again — sensor error distribution — Transition probability matrix

→ How should we initialize this $\beta_k$ ?

$$\beta_{n-1} = P(m_{n:n} \mid s_{n-1}) = \sum_{s_n} \frac{P(m_n, s_{n-1}, s_n)}{P(s_{n-1})}$$

$$= \frac{1}{\cancel{P(s_{n-1})}} \sum_{s_n} P(m_n \mid s_n, s_{n-1}) P(s_n \mid s_{n-1}) \cancel{P(s_{n-1})}$$

$$\beta_{n-1} = \sum_{s_n} P(m_n \mid s_n) P(s_n \mid s_{n-1})$$ ⇒ Both terms known

$$\beta_{n-2} = \sum_{s_{n-1}} \beta_{n-1} \, P(m_{n-1} \mid s_{n-1}) \, P(s_{n-1} \mid s_{n-2})$$

Recall original goal : $P(s_k \mid m_{1:n})$ $\Rightarrow$ offline version

$$P(s_k \mid m_{1:n}) = P(s_k \mid m_{1:k}) \, P(m_{k+1:n} \mid s_k)$$

$\downarrow$ Dynamic prog. $\qquad$ $\downarrow$ Dynamic prog

Hide

$$\left( \alpha_k = \sum_{s_{k-1}} P(m_k \mid s_k) \, P(s_k \mid s_{k-1}) \, \alpha_{k-1} \right)$$

$$\left( \beta_k = \sum_{s_{k+1}} \beta_{k+1} \, P(m_{k+1} \mid s_{k+1}) \, P(s_{k+1} \mid s_k) \right)$$

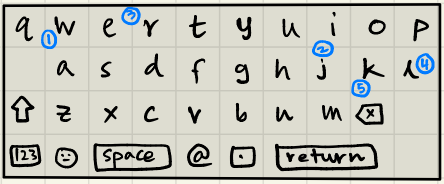
HMM's $\Rightarrow$ Efficiently identifying the most likely value of a state variable from a large space of computation and possibilities

$\Rightarrow$ Possible to also compute the full trajectory

↳ called Viterbi Decoding

## Some other applications of HMM (informal discussion)

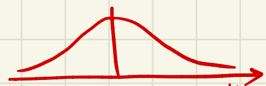
### (I) Auto-correction in smartphone keyboard.



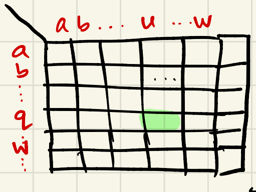

$\left.\begin{array}{l}\text{wield}\\ \text{quell}\\ \text{qualm}\end{array}\right\}$ hide

$$P(m_1|s_1) = P\left(m_1 = \text{location of } ① \;\middle|\; s_1 = \begin{bmatrix} a\\ b\\ c\\ d\\ \vdots\\ z\end{bmatrix}\right) =$$

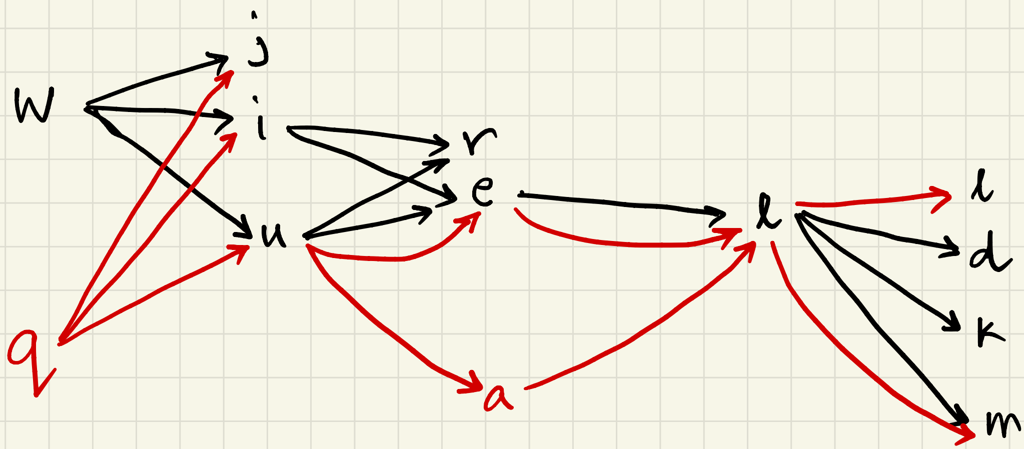
2D Gaussian




$$P(s_2|s_1) = P\left(s_2 = \begin{bmatrix} j\\ i\\ u\\ h\\ o\\ k\\ \vdots\end{bmatrix} \;\middle|\; s_1 = \begin{bmatrix} w\\ q\\ a\\ e\\ s\\ \vdots\end{bmatrix}\right) =$$




} from English dictionary

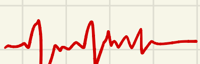


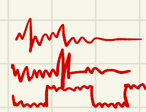

$\Downarrow$

Decodes to "quell" or "qualm"

### Similar application in speech recognition

$$P(m_K|s_K) = P\left(m_K = \text{[waveform]} \;\middle|\; s_K = \begin{bmatrix} a' =\\ b' =\\ K' =\\ \vdots\end{bmatrix}\right)$$





# Questions