

Location Prediction



Consiglio Nazionale
delle Ricerche

Content of this lesson

- Mobility prediction: a taxonomy
- Next location prediction
 - (Hidden) Markov Model-based
 - (Frequent) Pattern-based
 - Deep Learning-based

Mobility Prediction

Target of prediction

- Individual targets
 - Trajectory
 - The next location
 - All the trip
 - Destination of trip
 - Events
 - E.g. Crashes
 - All above + time of movement
- Collective targets
 - Aggregate Flows
 - OD matrix
 - Crowd density
 - Events
 - E.g. Crashes

Perspective

- Continuous movement
 - Points expressed as latitude & longitude
 - Prediction means reconstructing the precise points & movement
- Discrete space
 - Points become “areas”
 - Mobile phone cells
 - POIs
 - Voronoi cells
 - Prediction means predicting a cell ID or a sequence of IDs

Prediction Tools

Prediction approaches

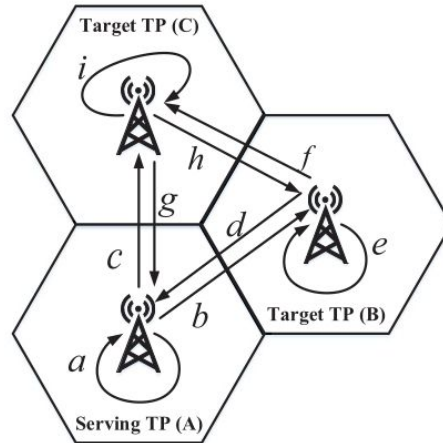
- Most typical tools adopted in mobility prediction
 - (Hidden) Markov Models
 - Pattern-based
 - Neural Networks

Markov Models

- Standard means to model sequential stochastic processes
- Assumption
 - The probability of actual event depends only on the previous event

$$P(s_i | \langle s_1, \dots, s_{i-1} \rangle) = P(s_i | s_{i-1}) \quad \rightarrow \text{all the previous states of the system are irrelevant}$$

- Consequence
 - The model can be represented as a simple “transition matrix” $P_{i,j}$
 - Example:

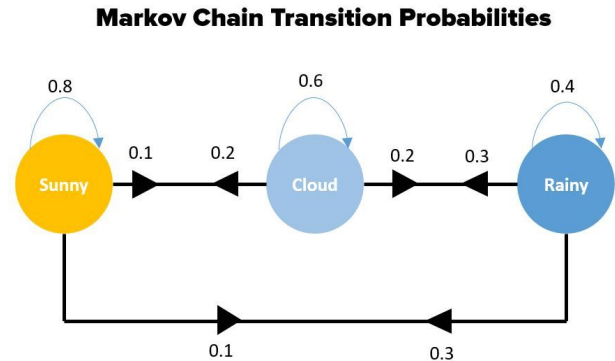


$$P = \begin{matrix} & \begin{matrix} A & B & C \end{matrix} \\ \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} & \begin{matrix} A \\ B \\ C \end{matrix} \end{matrix}$$

Markov Models

- More formally, we have
 - Set of states:
 $Q = \{q_1, q_2, \dots, q_n\}$
 - Vector of prior probabilities (probabilities of occurrence of each state):
 $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, where $\pi_i = P(S_0 = q_i)$
 - Matrix of transition probabilities
 $P = \{a_{i,j}\}$, i,j in $[1..n]$, with $a_{i,j} = P(q_j | q_i)$

- **Markov Chains (MC)**
 - The states are directly the observed values
 - Learning Π and P is straightforward: just count!



Markov Models

- More formally, we have
 - Set of states:

$$Q = \{q_1, q_2, \dots, q_n\}$$
 - Vector of prior probabilities (probabilities of occurrence of each state):

$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}, \text{ where } \pi_i = P(S_0 = q_i)$$
 - Matrix of transition probabilities

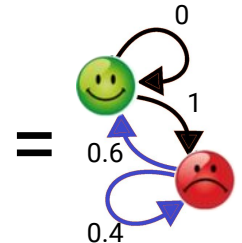
$$P = \{a_{ij}\}, i, j \text{ in } [1..n], \text{ with } a_{ij} = P(q_j | q_i)$$
- Example



What's next?

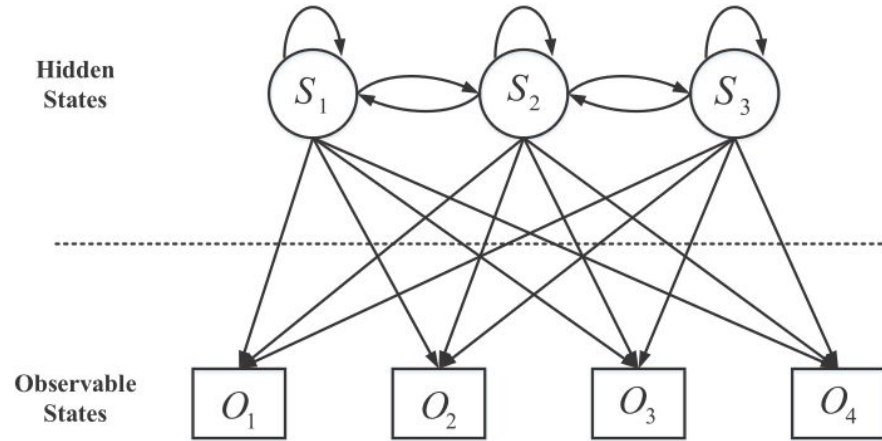
$$Q = \{ \text{😊}, \text{😞} \}$$

$$\Pi: \begin{aligned} \pi_{\text{😊}} &= 4/9 = 0.44 \\ \pi_{\text{😞}} &= 5/9 = 0.56 \end{aligned}$$

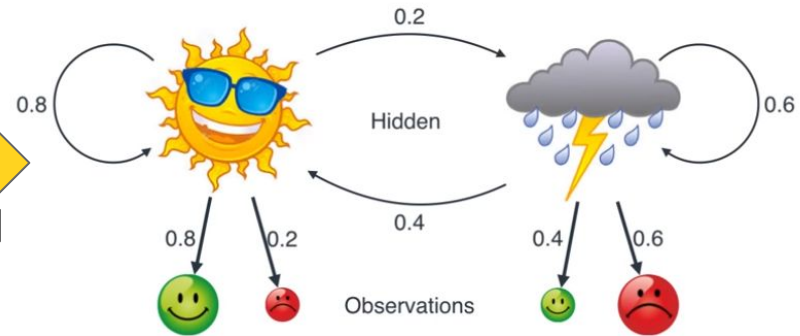
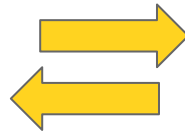
$$P: \begin{array}{c} \text{😊} \\ \text{😞} \end{array} \begin{array}{|c|c|} \hline \text{😊} & \text{😞} \\ \hline \hline \hline \end{array} = \begin{array}{|c|c|} \hline 0/3 & 3/3 \\ \hline 3/5 & 2/5 \\ \hline \end{array}$$


Markov Models

- **Hidden Markov Models (HMM)**
 - The states are not visible!
 - We can only see some “observables”
 - E.g. states = activity
= { working, leisure }
observables = places visited
= { home, cafeteria, gym, ... }
- Observables depend only on current state
 - Emission probabilities
 - $P(O_j | S_i)$



Observable input sequence



Markov Models

- **Hidden Markov Models (HMM):**

- Set of states:

$$Q = \{q_1, q_2, \dots, q_n\}$$

- Set of observables:

$$O = \{o_1, o_2, \dots, o_m\}$$

- Vector of prior probabilities (probabilities of occurrence of each state):

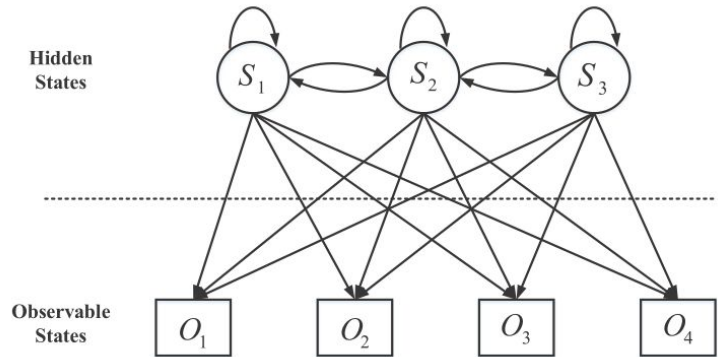
$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}, \text{ where } \pi_i = P(S_0 = q_i)$$

- Matrix of transition probabilities

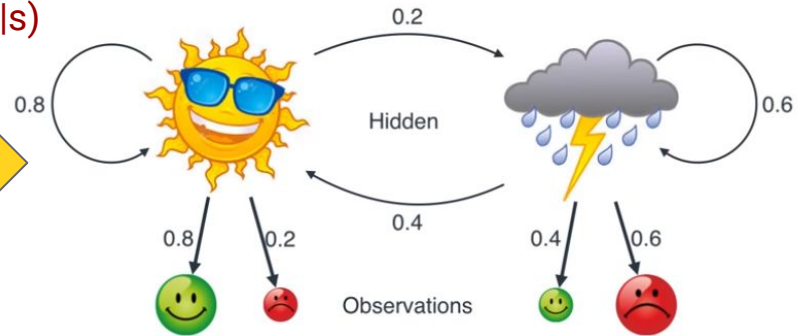
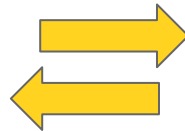
$$P = \{a_{ij}\}, i, j \text{ in } [1..n], \text{ with } a_{ij} = P(q_j | q_i)$$

- Matrix of emission probabilities

$$P_E = \{p_{s \rightarrow o}\}, s \text{ in } Q, o \text{ in } O, \text{ with } p_{s \rightarrow o} = p(o|s)$$



Observable input sequence



Markov Models

- Three basic problems with HMM
 - **Likelihood:**
 - Given an HMM $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $P(O|\lambda)$
 - **Decoding:**
 - Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q
 - **Learning:**
 - Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B

Markov Models

- Three basic problems with HMM

- **Likelihood:**

- Given an HMM $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $P(O|\lambda)$

- **Decoding:**

- Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q

- **Learning:**

- Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B



Forward Algorithm

- Dynamic programming
- Virtually tries all sequences of states & aggregate probabilities

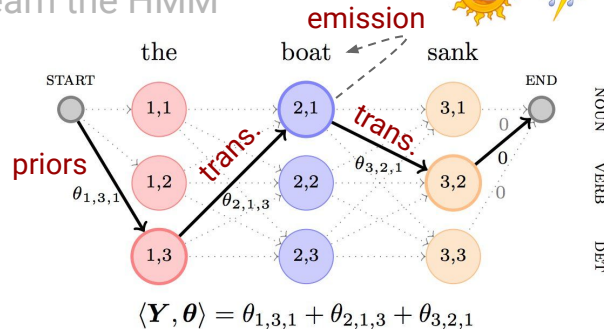
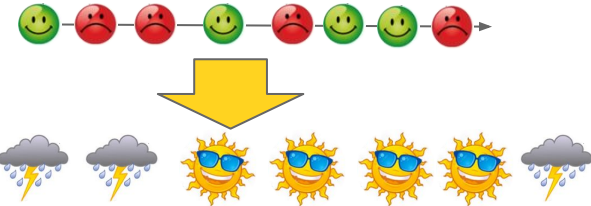
$$P(\text{😊 😞 😞 😊 😞 😊 😊 😞}) = ?$$

Markov Models

- Three basic problems with HMM
 - **Likelihood:**
 - Given an HMM $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $P(O|\lambda)$
 - **Decoding:**
 - Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q
 - **Learning:**
 - Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B

Viterbi Algorithm

- Dynamic programming
- Quite similar to the Forward Algorithm



Example on words:
 best("the boat sank") = det → noun → verb

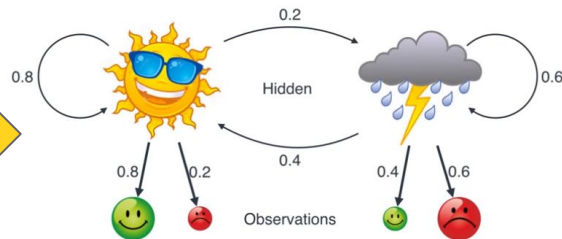
Markov Models

- Three basic problems with HMM
 - **Likelihood:**
 - Given an HMM $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $P(O|\lambda)$
 - **Decoding:**
 - Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q
 - **Learning:**
 - Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B



Forward-Backward Algorithm a.k.a. Baum-Welch Algorithm

- Similar to EM or K-means
- Iterative refinement method



Markov Models – How to make next-value predictions

- **Point estimation**

- a.k.a. Take the maximum-likely value
- For each possible value “o”:
 - Append “o” to input sequence “S”
 - Compute $L(S+o)$ = likelihood of S+o
- Take the “o” that maximizes $L(S+o)$

$$\begin{aligned}o_T &= \arg \max_o P(O_T = o \mid o_1, \dots, o_{T-1}) \\ &= \arg \max_o \frac{P(o_1, \dots, o_{T-1}, O_T = o)}{P(o_1, \dots, o_{T-1})} \\ &= \arg \max_o P(o_1, \dots, o_{T-1}, O_T = o)\end{aligned}$$

- **Conditional expectation** [for numerical data only]

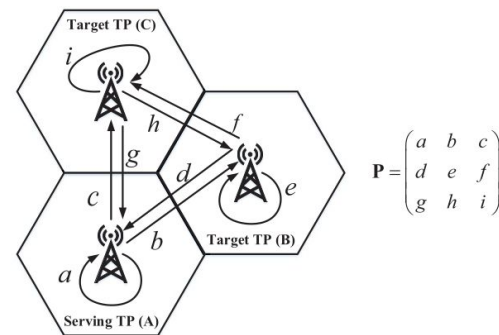
- a.k.a. Compute a mean value
- Similar computations as above
- Use likelihoods to compute a weighted average of values “o”

$$\begin{aligned}\mathbb{E}[O_T \mid o_1, \dots, o_{T-1}] &= \sum_o o P(O_T = o \mid o_1, \dots, o_{T-1}) \\ &= \frac{\sum_o o P(o_1, \dots, o_{T-1}, O_T = o)}{P(o_1, \dots, o_{T-1})}\end{aligned}$$

Markov Models – Applied to mobility

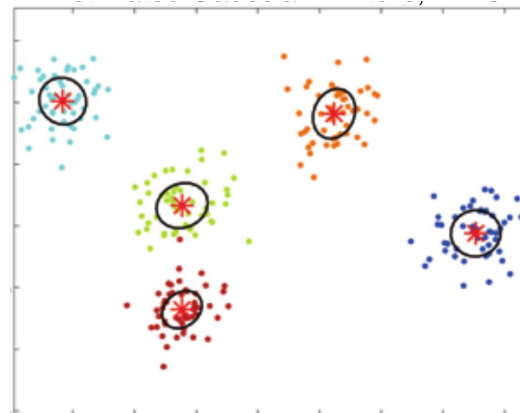
- **Approach 1: discretization**

- Translate trajectories to sequences of IDs
- E.g.:
 - Voronoi tessellation
 - Sequence of POIs visited/approached



- **Approach 2: 2-D Gaussian distributions**

- Emissions are 2-D (lat - long)
- Step-wise movements

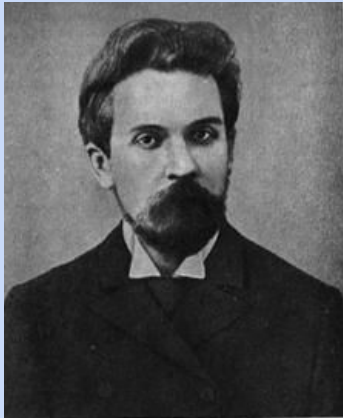


INTERVALLO

The Markovs sequence

This one!

Vladimir Markov
(1871 – 1897)



Mathematician

Andrey Markov
(???? – ????)



(Not a Mathematician)

Andrey Markov
(1856 – 1922)



Mathematician

Andrey Markov
(1903–1979)



Mathematician



INTERVALLO

The Markovs sequence

- Known in his youth as a rebellious student
- Very good in Math, very bad in anything else
- Excommunication:
 - 1912: the Russian Orthodox Church excommunicates Leo Tolstoy
 - Markov responds by requesting his own excommunication
 - The Church complied with his request
- Early retirement:
 - In 1905, he was appointed merited professor
 - That granted the right to retire, whatever the age!
 - Markok retired immediately – at 49! (though he kept teaching...)

Andrey Markov
(1856 – 1922)



INTERVALLO

The Markovs sequence

- Also interested in poetry
 - he made studies of poetic style
 - Kolmogorov (father of probability theory) had similar interests...
- He applied his theory of Markov chains to chains of two states, namely vowels and consonants, in literary texts

- As a lecturer, Markov demanded much of his students:

His lectures were distinguished by an irreproachable strictness of argument, and he developed in his students that mathematical cast of mind that takes nothing for granted. He included in his courses many recent results of investigations, while often omitting traditional questions. The lectures were difficult, and only serious students could understand them. ... During his lectures he did not bother about the order of equations on the blackboard, nor about his personal appearance.

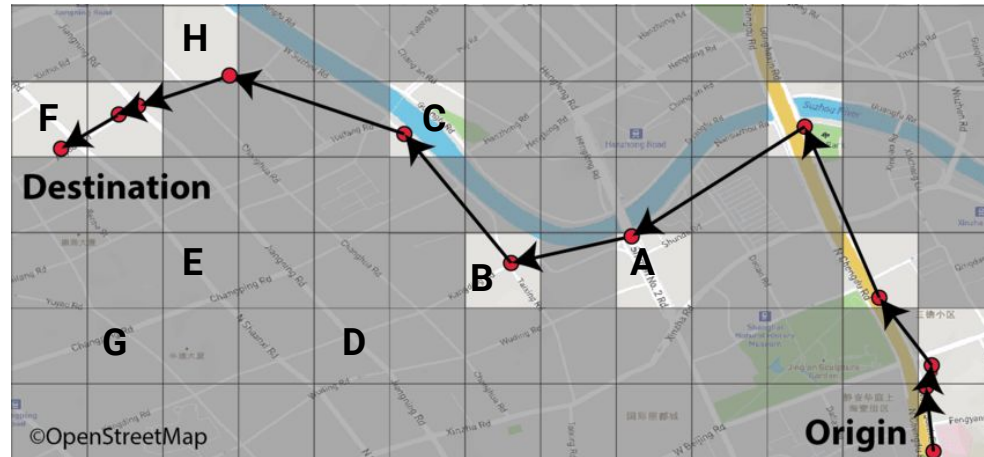
A A Youschkevitch, Biography in Dictionary of Scientific Biography (New York 1970-1990).

Andrey Markov
(1856 – 1922)



Pattern-based prediction

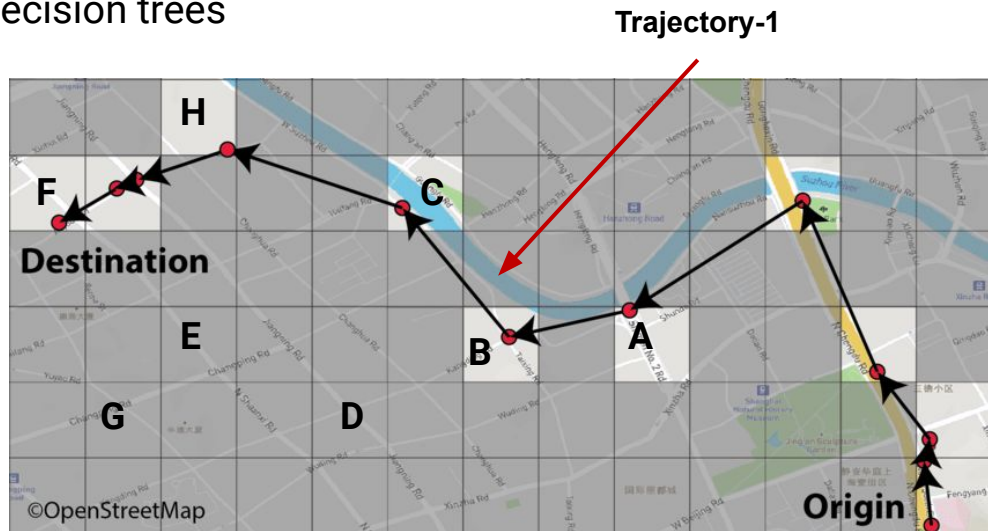
- Idea:
 - identify patterns in the trajectory
 - use them to associate prediction
- E.g.
 - Patterns: $A \rightarrow B \rightarrow C$, $A \rightarrow B \rightarrow D$, $B \rightarrow D \rightarrow E$, $B \rightarrow C \rightarrow H$
 - Predictive model:
 - If $A \rightarrow B \rightarrow C \Rightarrow$ destination=F
 - If $A \rightarrow B \rightarrow D \Rightarrow$ destination=E
 - If $B \rightarrow D \rightarrow E \Rightarrow$ destination=G
 - ...



Pattern-based prediction

- Tabular representation of trajectories
- Each pattern can become
 - a Boolean (the pattern occurs/does not occur)
 - a numerical value (how strongly the pattern occurs)
- Standard ML tools can be applied, e.g. decision trees

	A→B→C	A→B→D	B→D→E	B→C→H
Trajectory-1	1	0	0	1
Trajectory-2	0	0	1	0

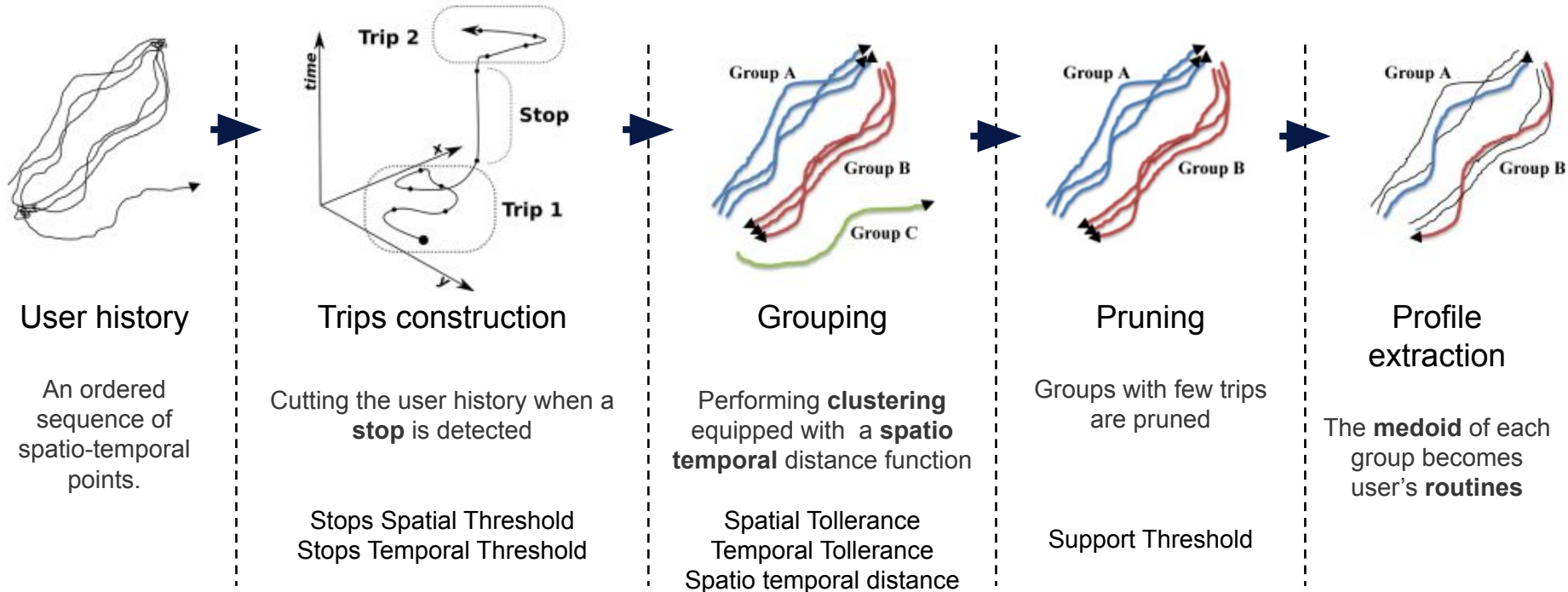


Pattern-based prediction

- Patterns can be extracted through various criteria
 - Based on frequency
 - General frequent pattern / clustering methods can be applied
 - Frequent patterns expected to express significant features
 - Based on discrimination power
 - Requires ad hoc solutions
 - Might find infrequent yet useful patterns

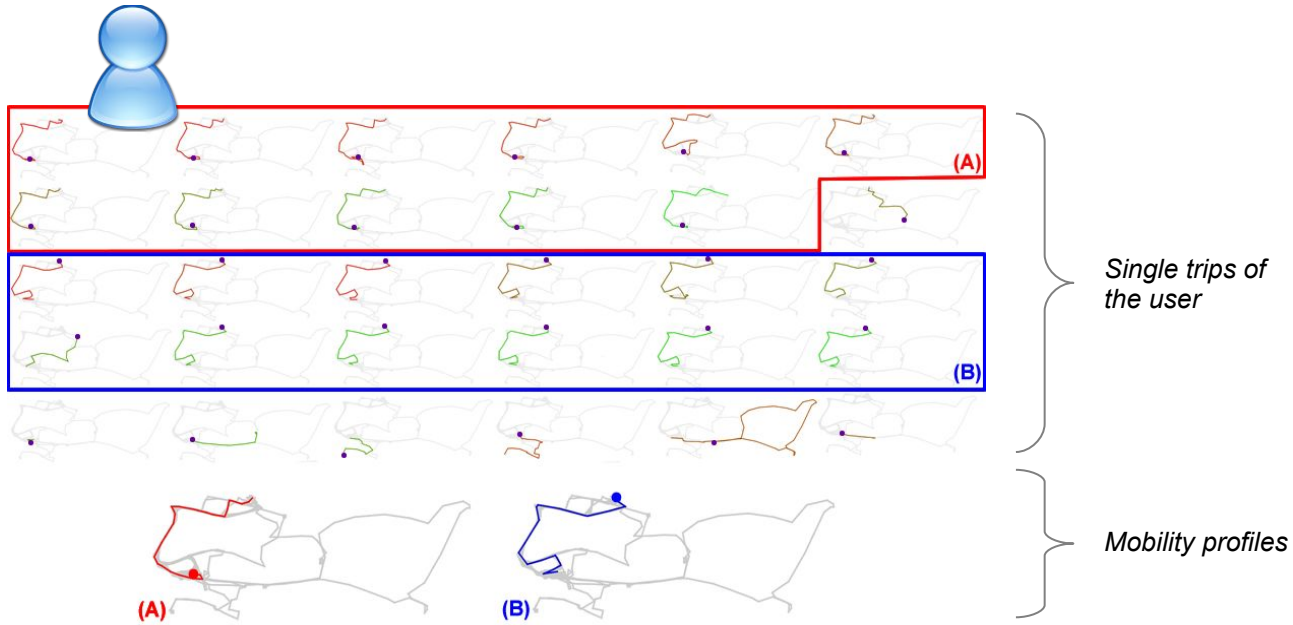
Pattern-based prediction

Mobility profiles as “frequent prediction patterns”



Pattern-based prediction

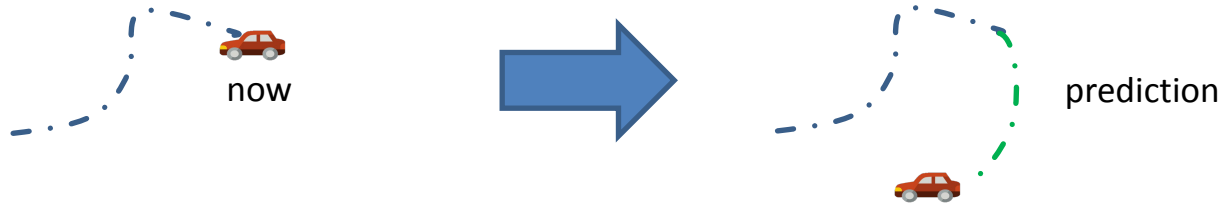
Mobility profiles as “frequent prediction patterns”



Pattern-based prediction

Mobility profiles as “frequent prediction patterns”

- Step 1: perform “partial match” of a trajectory with a set of systematic trips
- Step 2: check that the best match is similar enough
- Step 3: use the “rest” of the systematic trip as continuation of the trip



Pattern-based prediction

Mobility profiles as “frequent prediction patterns”

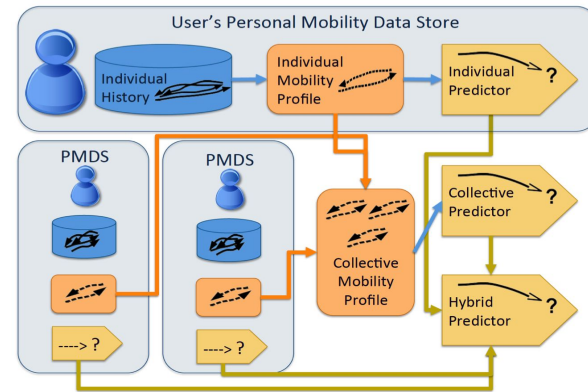
Which systematic trips to use?

- Priority: those of the same user/device/vehicle
- If fails: those of the whole population analyzed

1. Check my personal systematic mobility



2. Check others' systematic mobility

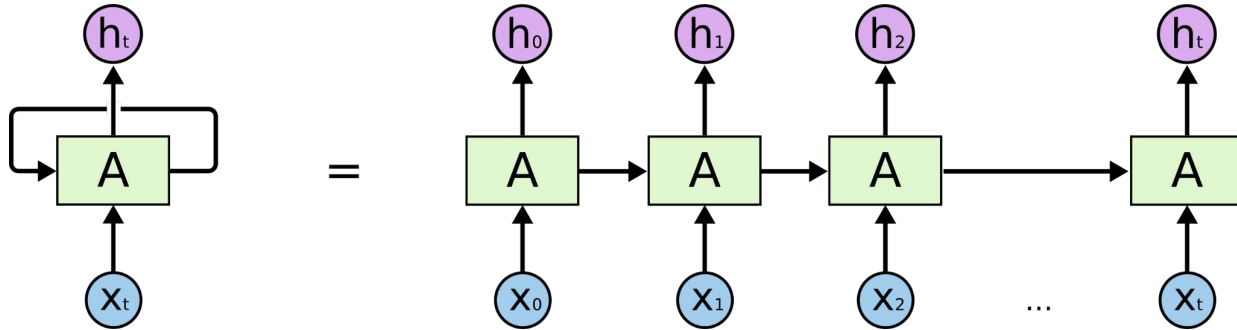


Deep Learning for trajectory prediction

- Three main architectures (+ hundreds of variants)
 - RNN: Recurrent Neural Network / LSTM: Long-Short Term Memory
 - CNN: Convolutional Neural Network
 - GAN: Generative Adversarial Network

RNN

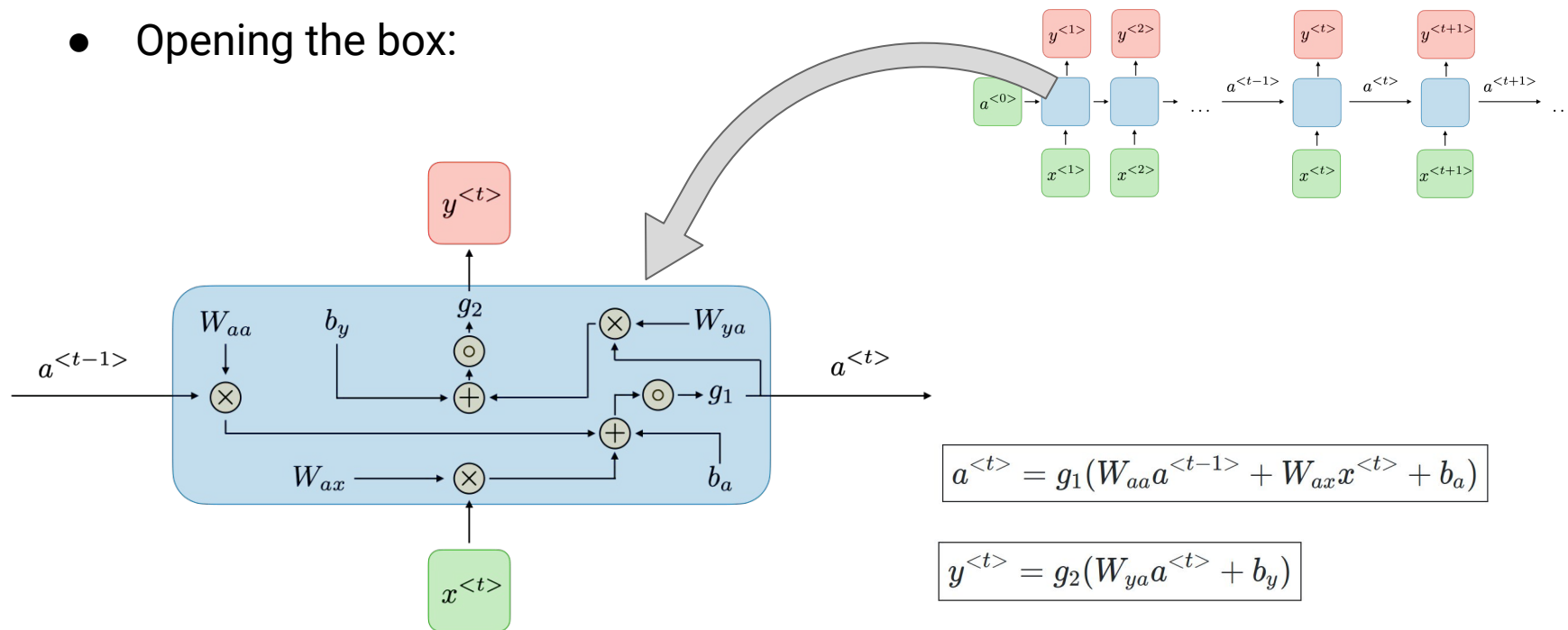
- NN where the hidden layer has a “loop” connection
- The link works as extra input for the next training data instance
 - Makes the model apt for sequential data
- Notice: “A” is always the same



- E.g. $\langle x_0, \dots, x_t \rangle =$ input trajectory, $\langle h_0, \dots, h_t \rangle =$ output trajectory

RNN

- Opening the box:

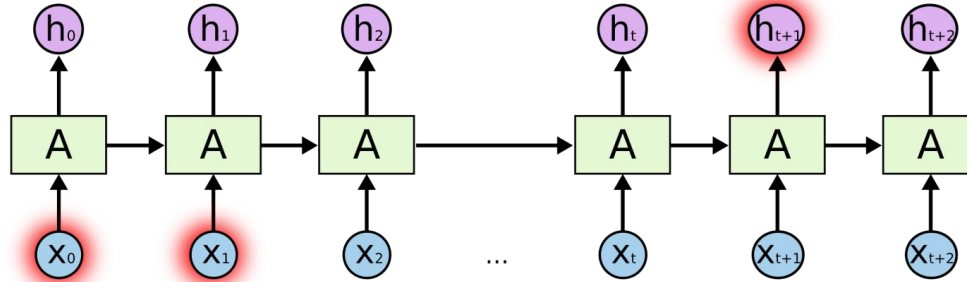


$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

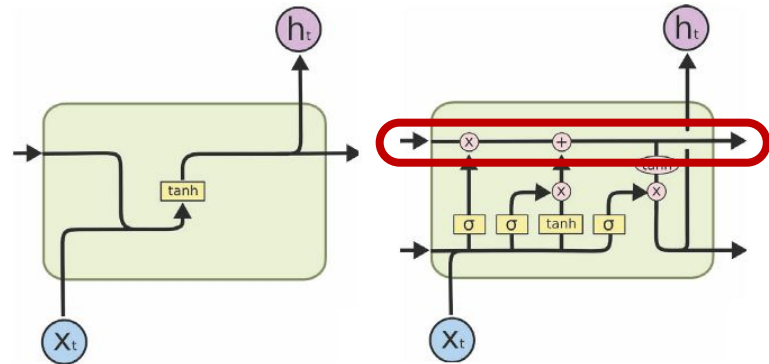
$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

RNN - LSTM

- RNN has issues with long-term relations between input and output



- Solution: pass a state information that is incrementally updated

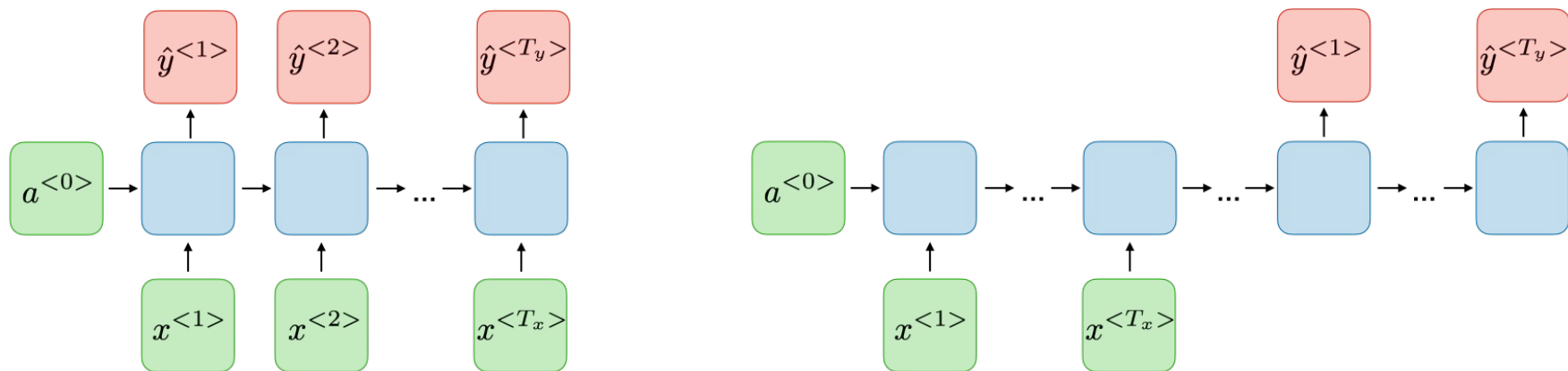


(a) RNN

(b) LSTM

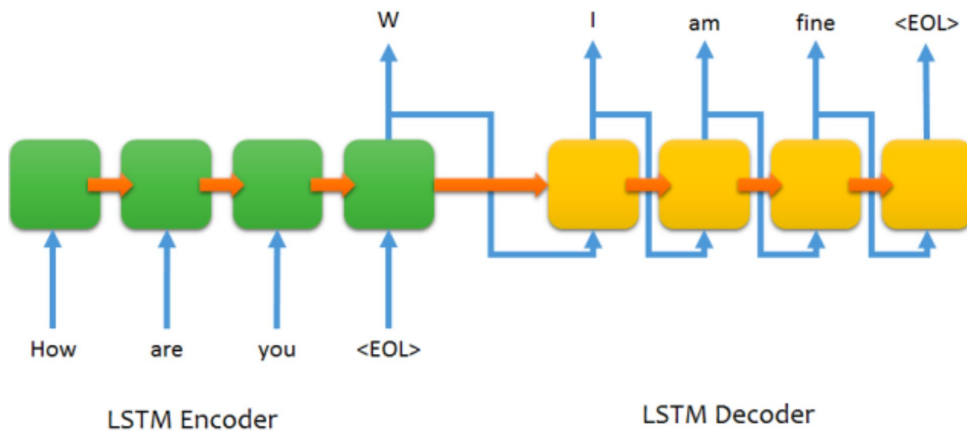
RNN

- Two main “many-to-many” architectures:



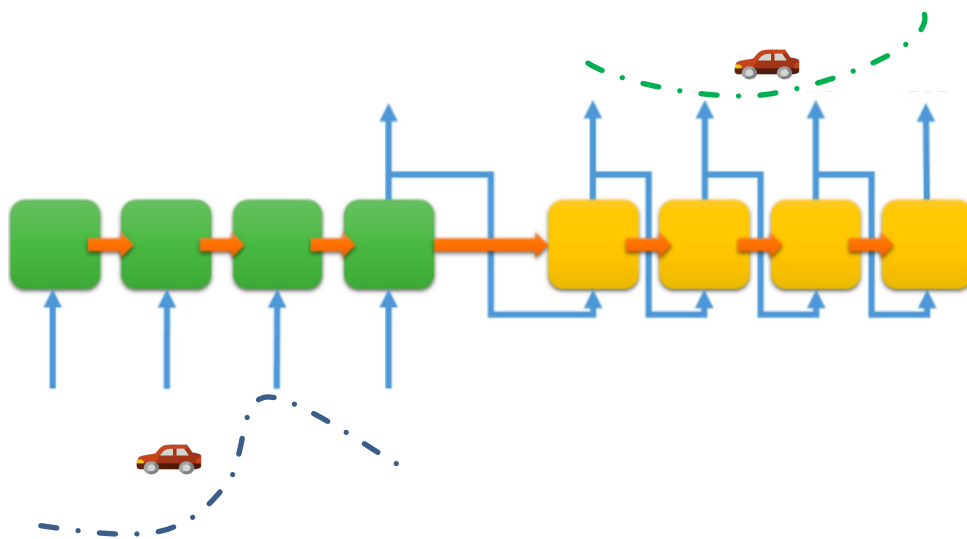
RNN - LSTM - Encoder-Decoder schema

- **Encoder step:** during input elaboration the output (W) is a latent representation of the whole sequence
- **Decoder step:** W becomes the input of LSTM, which is fired till generating a special output (stop)



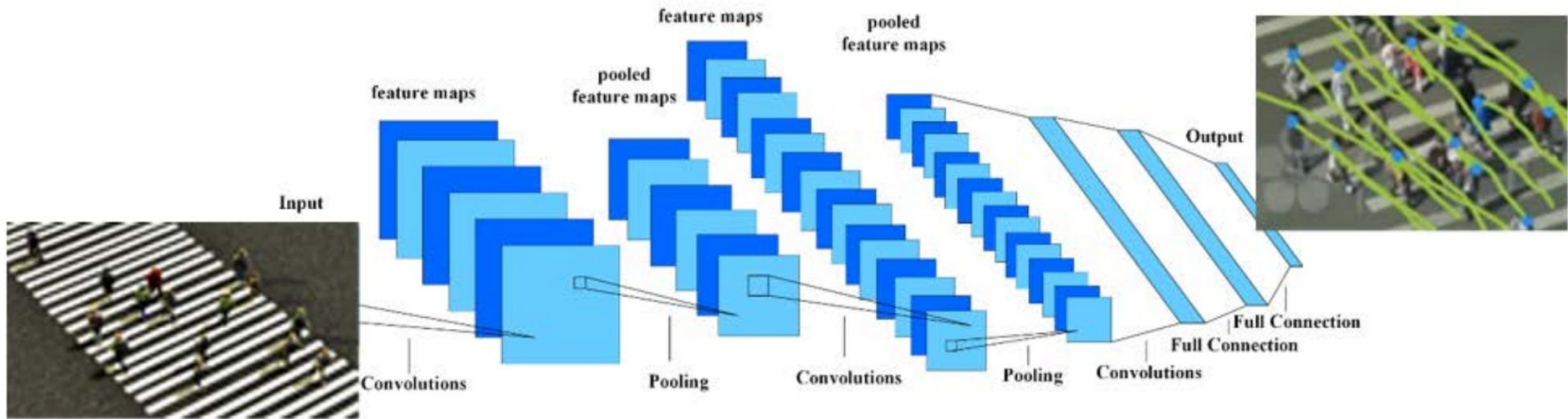
RNN - LSTM - Encoder-Decoder schema

- **Encoder step:** during input elaboration the output (W) is a latent representation of the whole sequence
- **Decoder step:** W becomes the input of LSTM, which is fired till generating a special output (stop)



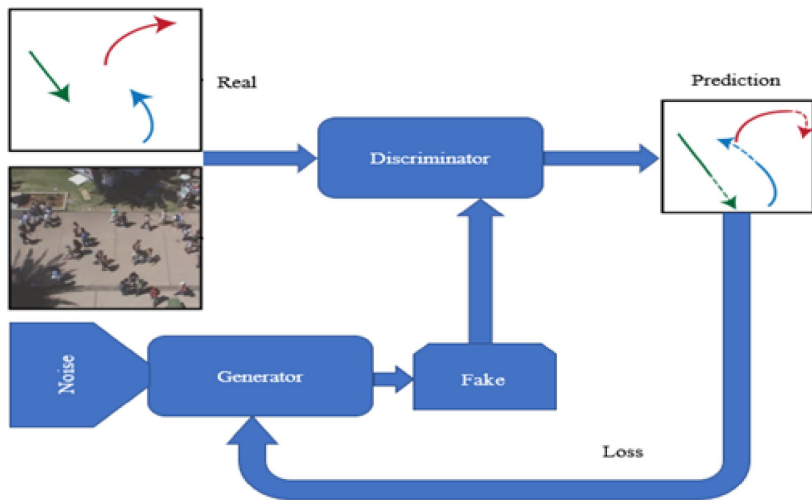
CNN for movement prediction

- Typically used when images are involved, e.g. camera data
- Usually in conjunction with LSTM or other sequence-based models
 - CNN captures spatial relations & identifies objects / features
 - LSTM captures temporal relation & movement



GAN

- Trains 2 models at the same time:
 - Generator: one that generates fake objects
 - Discriminator: one that can distinguish real vs fake objects
- The generator can be seeded with a (representation of a) partial trajectory
- Both G and D can be any suitable model, e.g. LSTM



INTERVALLO

**Baseball: Do Fielders Know Where to Go to Catch
the Ball or Only How to Get There?**

or

**When Psychologists (With Lot of Spare Time)
Meet Sport Analytics**



Peter McLeod & Zoltan Dienes (1996).
J. of Experimental Psychology: Human Perception and Performance.
Vol. 22, No. 3, 531-543.

INTERVALLO

Baseball: Do Fielders Know Where to Go to Catch the Ball...?

Experimental setup

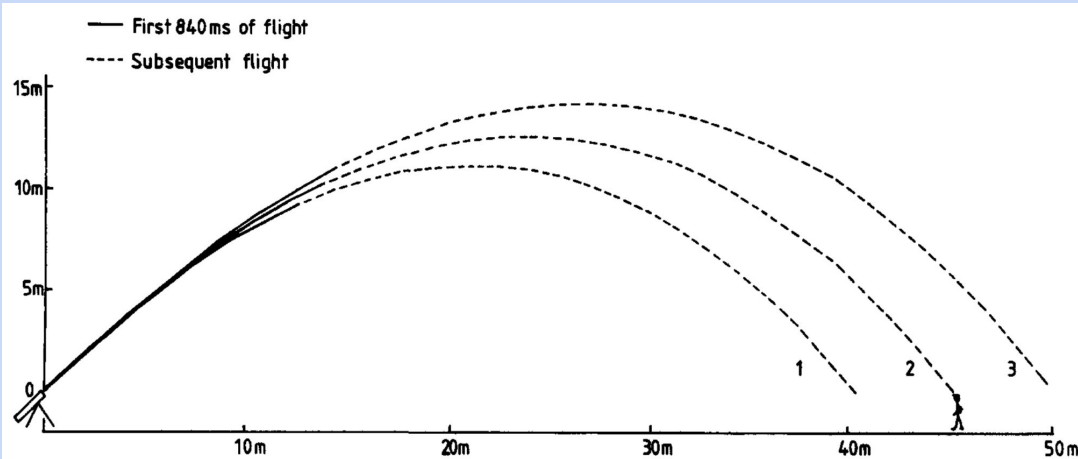
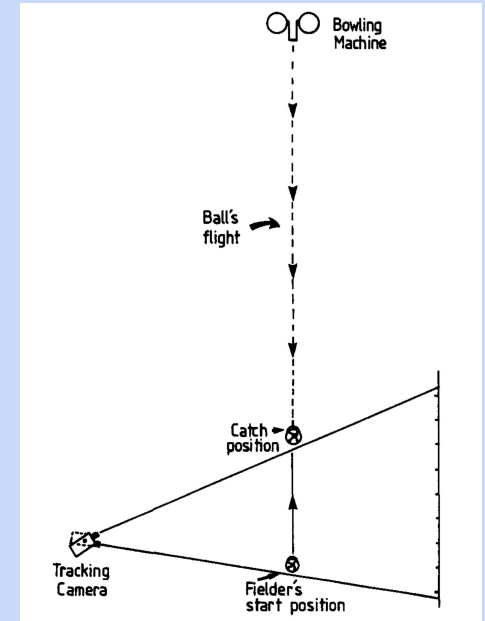


Figure 1. The trajectories of three balls projected at 45° and a velocity (v) of 22.3, 24.0, and 25.7 m/s, respectively, toward a fielder 45 m away. They experienced a deceleration due to aerodynamic drag proportional to v^2 . The constant of proportionality was 0.007 m^{-1} , a value typical of objects such as cricket balls (Daish, 1972).

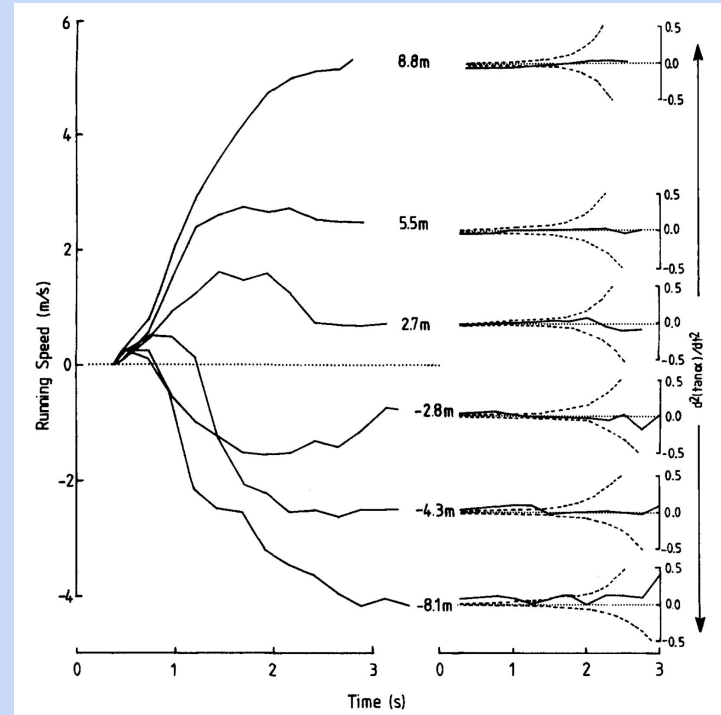
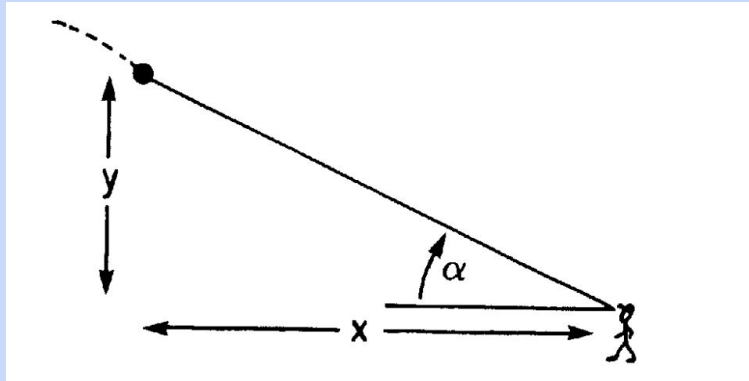


INTERVALLO

Baseball: Do Fielders Know Where to Go to Catch the Ball...?

Discovery

- Fielders adjust movement in order to keep $d^2(\tan \alpha)/dt^2 = 0$
- α = angle of gaze

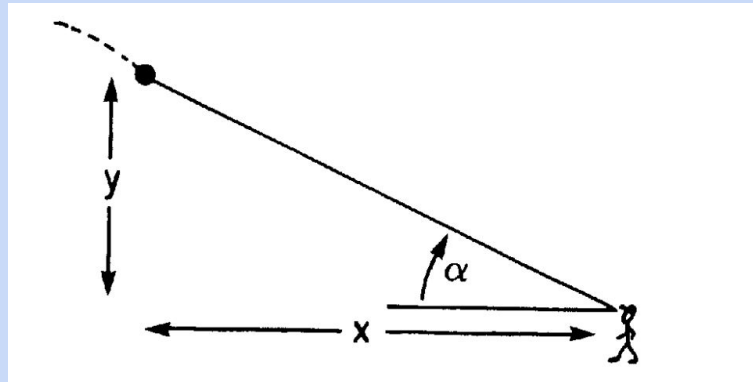


INTERVALLO

Baseball: Do Fielders Know Where to Go to Catch the Ball...?

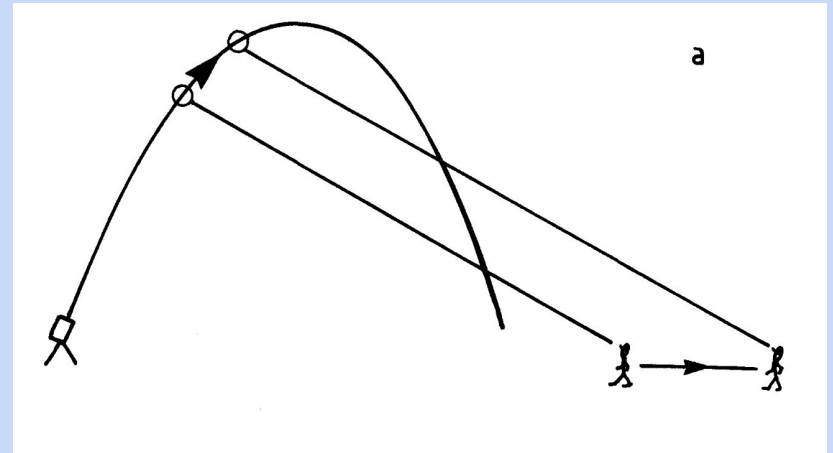
Discovery

- Fielders adjust movement in order to keep $d^2(\tan \alpha)/dt^2 = 0$
- α = angle of gaze



Remark

- Simple solutions like keeping α constant would not work



INTERVALLO

Baseball: Do Fielders Know Where to Go to Catch the Ball...?

Question 1: do fielders predict where the ball will fall?

- TL;DR: most likely no
 - They never reach the spot before the ball
 - They appear to dynamically adjust the movement to ball position

Question 2: do people really evaluate/compute $d^2(\tan \alpha)/dt^2$?

- Answer: most likely no

Homeworks

Food for thought

- **MC or HMM?** We can represent a HMM as a (larger) Markov Chain. Will the transition matrix have anything special?
- **HMMs and next location:** by modeling locations as states and specific positions as observables (maybe obtained through some Gaussian distribution) for the next location prediction, we need a lot of states. Is there a better way to model it? Maybe states might be something different.

Hint: traditional time series prediction does that a lot.

- **Next Location as collective phenomenon:** my driving behaviour greatly depends on how the people around me drive. Think about how we could take into account in next location prediction.

Material

- [paper] [Mobility Prediction: A Survey on State-of-the-Art Schemes and Future Applications](https://ieeexplore.ieee.org/document/8570749), Zhang and Dai (2019), <https://ieeexplore.ieee.org/document/8570749>
 - Sections I-IV
- [paper] [Hidden Markov Models](https://web.stanford.edu/~jurafsky/slp3/A.pdf), Daniel Jurafsky & James H. Martin (2024), draft book chapter, <https://web.stanford.edu/~jurafsky/slp3/A.pdf>
- [library] [HMMlearn library](https://hmmlearn.readthedocs.io/en/latest/), <https://hmmlearn.readthedocs.io/en/latest/>
 - See main functionalities

Material

- [paper] [A Survey on Trajectory-Prediction Methods for Autonomous Driving](#), Huang et al. (2022), <https://ieeexplore.ieee.org/document/9756903>
 - Sections IV and V
- [paper] [Do Fielders Know Where to Go to Catch the Ball...?](#), Mcleod & Dienes (1996), J. Exp. Psychology. [no easy download link; local copy on DidaWiki]