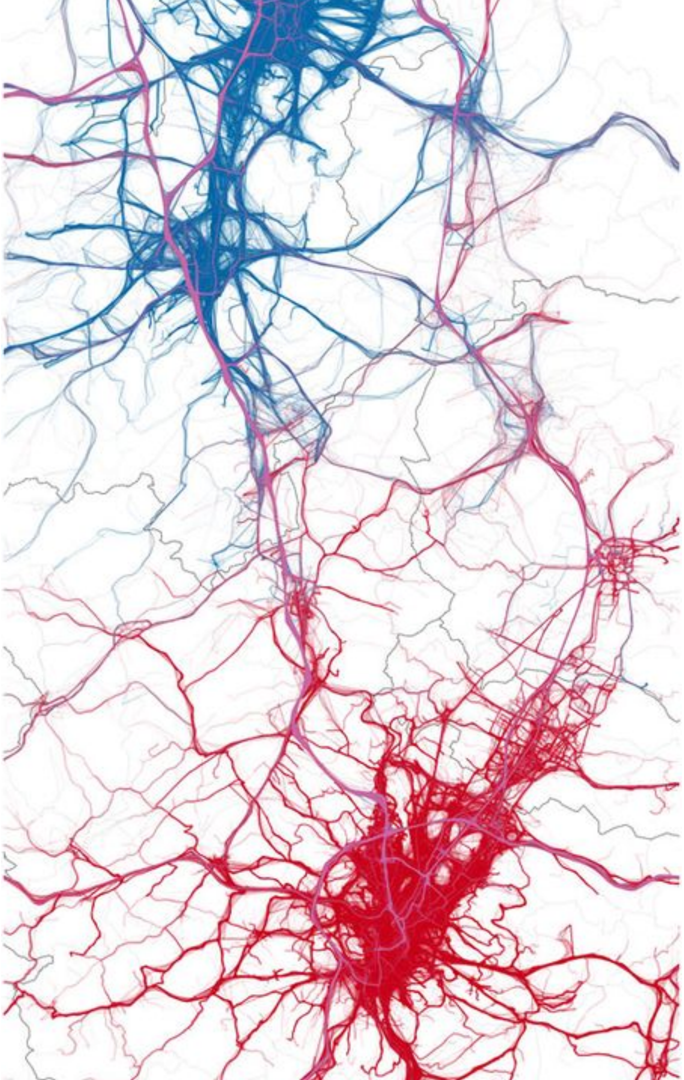


Consiglio Nazionale delle Ricerche

Location Prediction

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

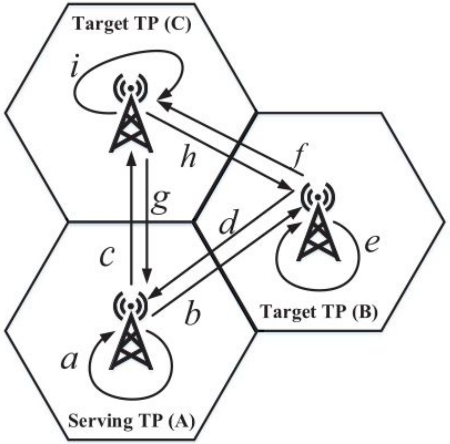
- Most typical tools adopted in mobility prediction
    - (Hidden) Markov Models
    - Pattern-based
    - Neural-Networks

Markov Models

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

    $P(s_i \mid \langle s_1, \ldots, s_{i-1} \rangle) = P(s_i \mid s_{i-1})$ → all the previous states of the system are irrelevant

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



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

Markov Models

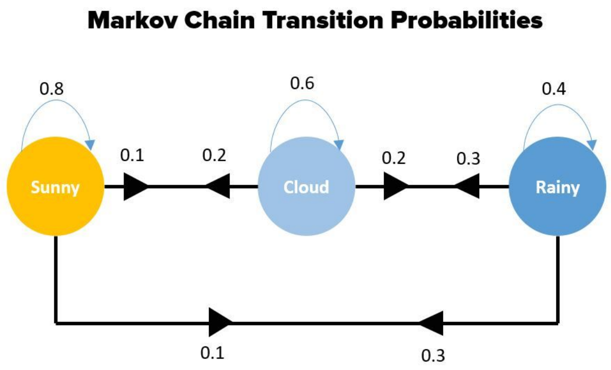
- More formally, we have
  - Set of states:

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

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

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

- **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:

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

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

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

What's next?

Q = {  ,  }

Π:

π  = 4/9 = 0.44

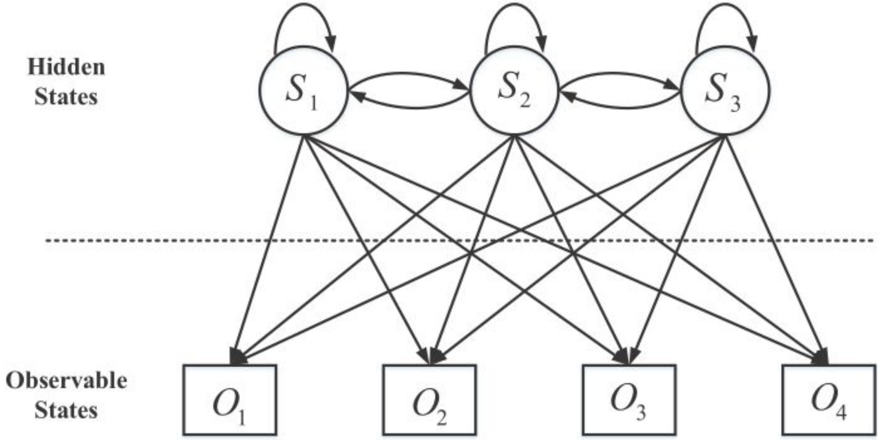
π  = 5/9 = 0.56

P:

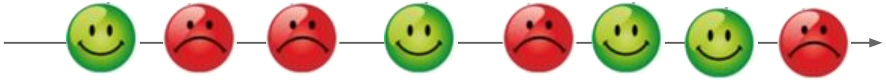
		
	0/3	3/3
	3/5	2/5

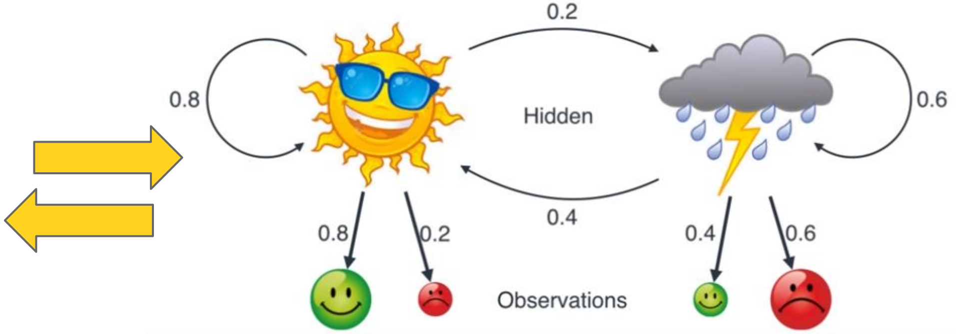
Markov Models

- **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
    - $P(O_j \mid S_i)$



Observable input sequence





Markov Models

- **Hidden Markov Models (HMM):**
  - Set of states:

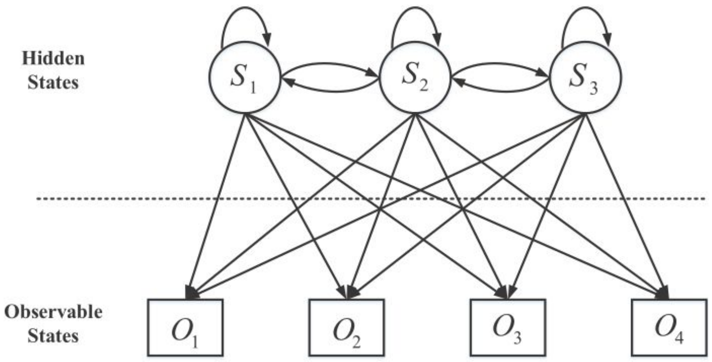
    $Q = \{q_1, q_2, \ldots, q_n\}$
  - Set of observables:

    $O = \{q_1, q_2, \ldots, q_m\}$
  - Vector of prior probabilities (probabilities of occurrence of each state):

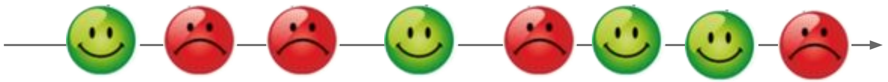
    $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$, where $\pi_i = P(S_0 = q_i)$
  - Matrix of transition probabilities

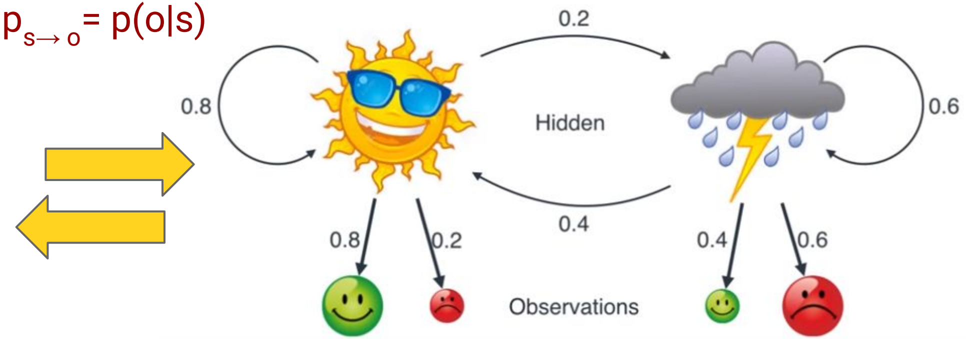
    $P = \{a_{i,j}\}$, i,j in [1..n], with $a_{i,j} = P(q_j \mid q_i)$
  - Matrix of emission probabilities

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



Observable input sequence





Markov Models

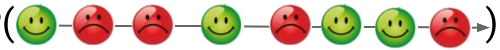
- Three basic problems with HMM
  - **Likelihood**:
    - Given an HMM λ = (A, B) and an observation sequence O, determine the likelihood P(O|λ)
  - **Decoding**:
    - Given an observation sequence O and an HMM λ = (A, B), discover the best hidden state sequence Q
  - **Learning**:
    - Given an observation 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 λ = (A, B) and an observation sequence O, determine the likelihood P(O|λ)
  - **Decoding**:
    - Given an observation sequence O and an HMM λ = (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() = ?

$$\alpha_i(t) = P(Y_1 = y_1, \ldots, Y_t = y_t, X_t = i \mid \theta)$$

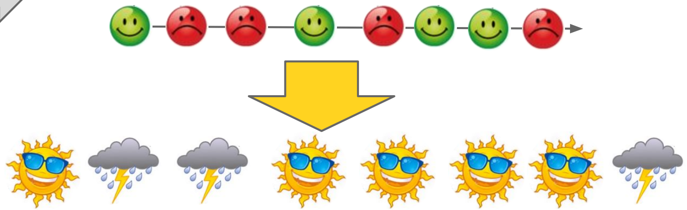
1. $\alpha_i(1) = \pi_i b_i(y_1)$,
2. $\alpha_i(t+1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) a_{ji}$.

Markov Models

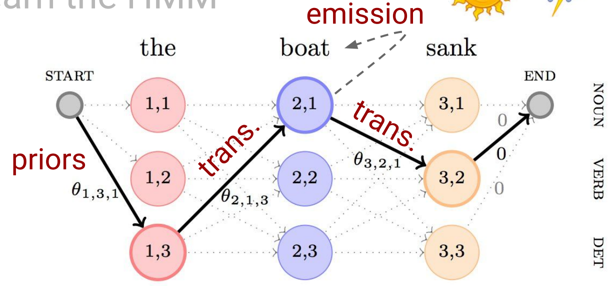
- Three basic problems with HMM
  - **Likelihood**:
    - Given an HMM λ = (A, B) and an observation sequence O, determine the likelihood P(O|λ)
  - **Decoding**:
    - Given an observation sequence O and an HMM λ = (A, B), discover the best hidden state sequence Q
  - **Learning**:
    - Given an observation 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

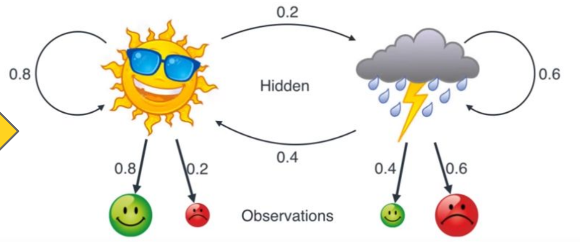


$$\langle \boldsymbol{Y}, \boldsymbol{\theta} \rangle = \theta_{1,3,1} + \theta_{2,1,3} + \theta_{3,2,1}$$

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)

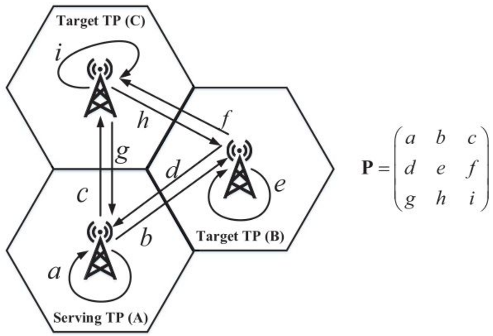
$$\begin{aligned} o_T &= \arg\max_o P(O_T = o \mid o_1, \ldots, o_{T-1}) \\ &= \arg\max_o \frac{P(o_1, \ldots, o_{T-1}, O_T = o)}{P(o_1, \ldots, o_{T-1})} \\ &= \arg\max_o P(o_1, \ldots, o_{T-1}, O_T = o) \end{aligned}$$

- **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”

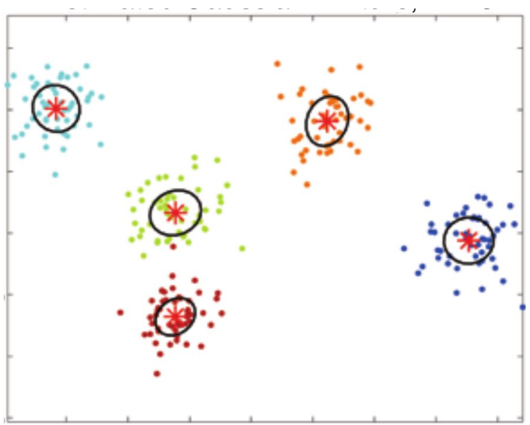
$$\begin{aligned} \mathbb{E}[O_T \mid o_1, \ldots, o_{T-1}] &= \sum_o o P(O_T = o \mid o_1, \ldots, o_{T-1}) \\ &= \frac{\sum_o o P(o_1, \ldots, o_{T-1}, O_T = o)}{P(o_1, \ldots, o_{T-1})} \end{aligned}$$

Markov Models – Applied to mobility

- **Approach 1: discretization**
  - Translate trajectories to sequences of IDs
  - E.g.:
    - Voronoi tesselation
    - Sequence of POIs visited/approached



- **Approach 2:** 2-D Gaussian distributions
  - Emissions are 2-D (lat - long)
  - Step-wise movements

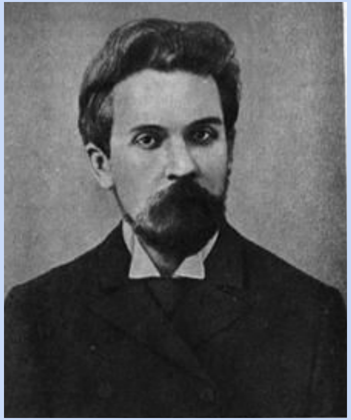


INTERVALLO

The Markovs sequence

This one!

Vladimir Markov
(1871 – 1897)



Mathematician

Andrey Markov
(???? – ????)

Father

(Not a Mathematician)

Andrey Markov
(1856 – 1922)

Mathematician

Andrey Markov
(1903–1979)

Son

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:

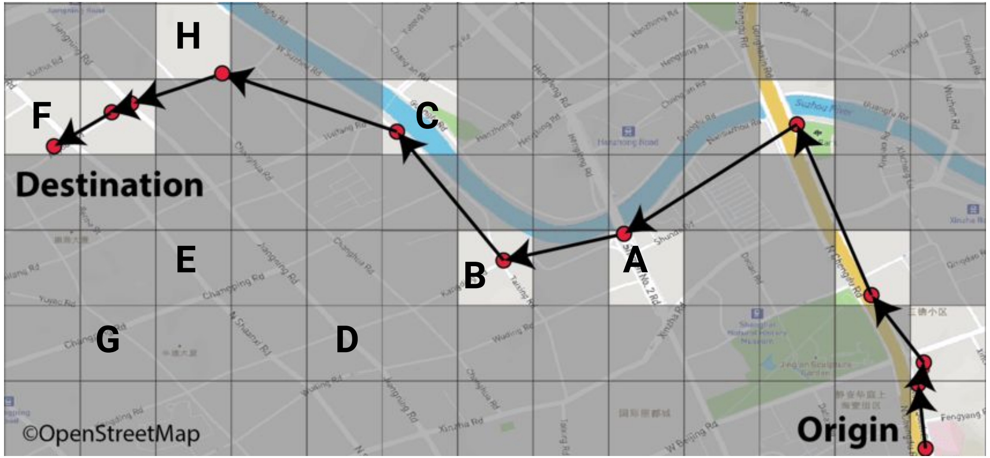
  His lectures were distinguished by an irreproachable strictness of argument, and he developed in his students that mathematical cast of mind that takes nothing for granted. He included in his courses many recent results of investigations, while often omitting traditional questions. The lectures were 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

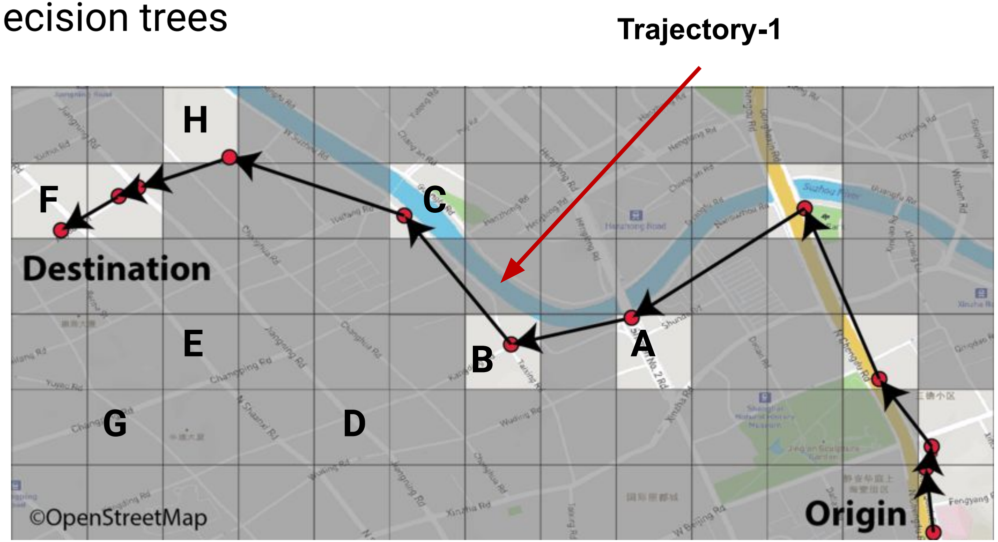
- Idea:
  - identify patterns in the trajectory
  - use them to associate prediction
- E.g.
  - Patterns: A→B→C, A→B→D, B→D→E, B→C→H
  - Predictive model:
    - If A→B→C ⇒ destination=F
    - If A→B→D ⇒ destination=E
    - If B→D→E ⇒ destination=G
    - …



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

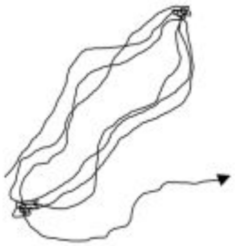


Pattern-based prediction

- 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

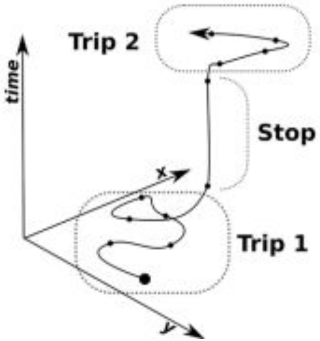
Pattern-based prediction

Mobility profiles as “frequent prediction patterns”



User history

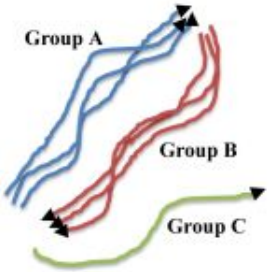
An ordered sequence of spatio-temporal points.



Trips construction

Cutting the user history when a **stop** is detected

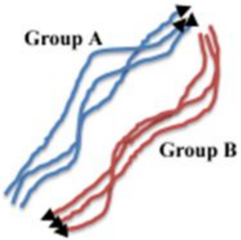
Stops Spatial Threshold
Stops Temporal Threshold



Grouping

Performing **clustering** equipped with a **spatio temporal** distance function

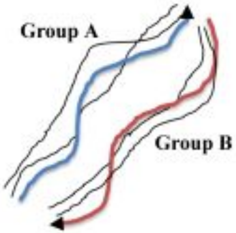
Spatial Tollerance
Temporal Tollerance
Spatio temporal distance



Pruning

Groups with few trips are pruned

Support Threshold

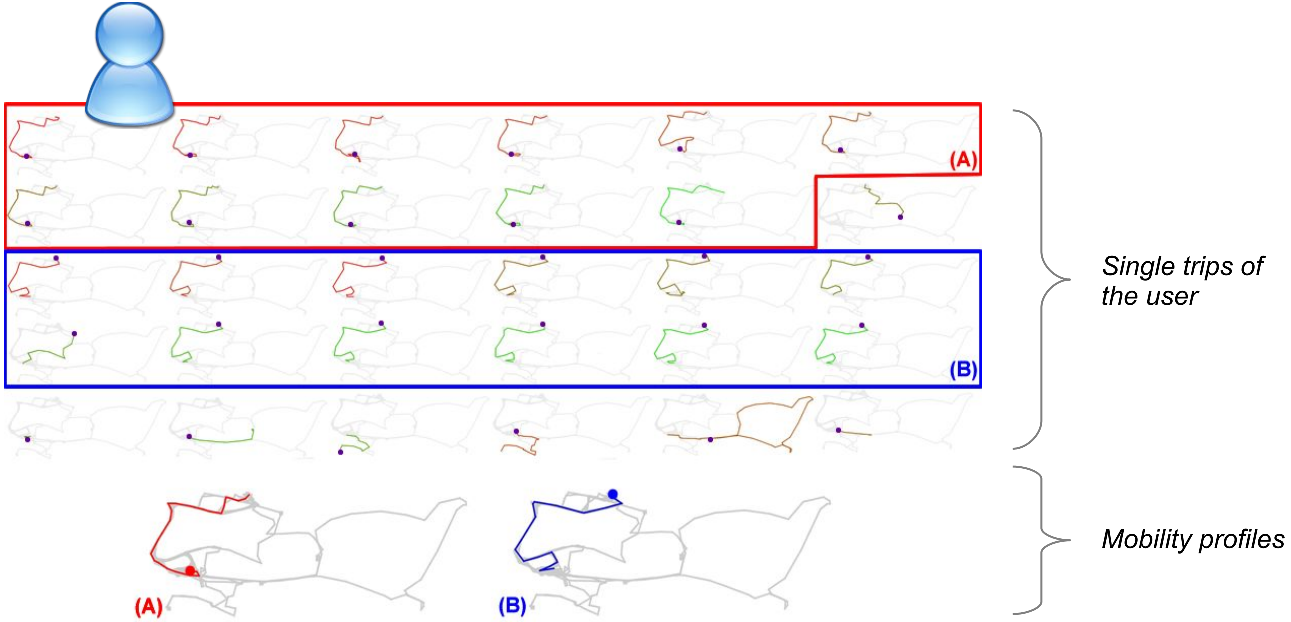


Profile extraction

The **medoid** of each group becomes user’s **routines**

Pattern-based prediction

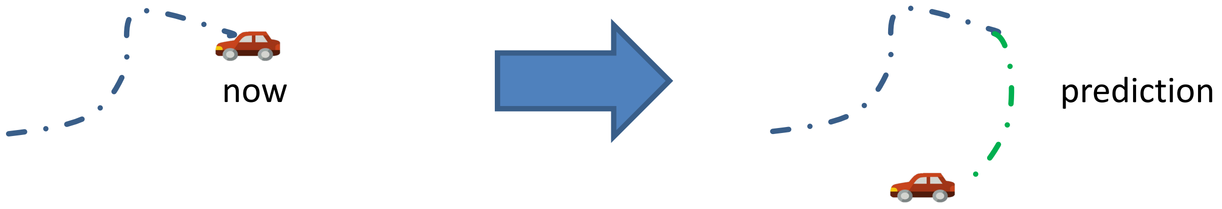
Mobility profiles as “frequent prediction patterns”



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

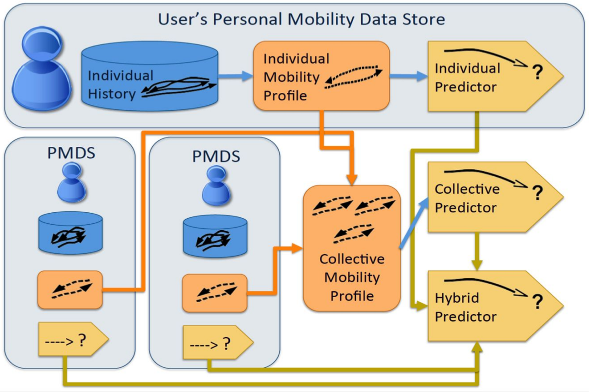
Mobility profiles as “frequent prediction patterns”

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

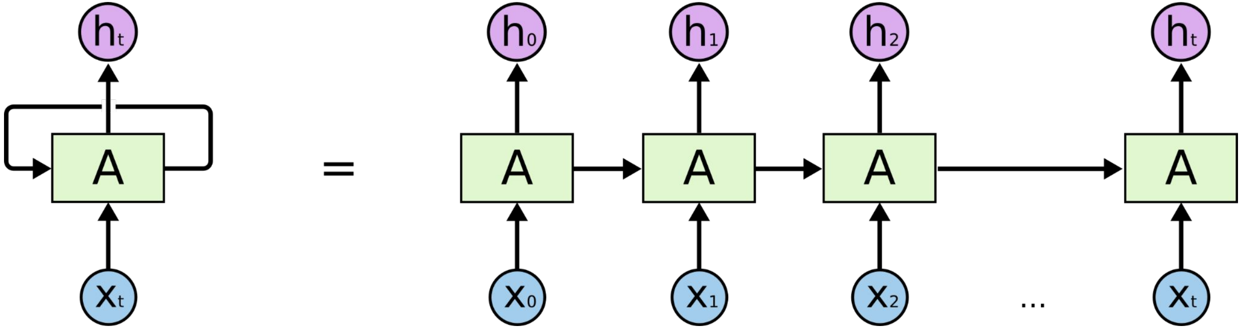


Deep Learning for trajectory prediction

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

RNN

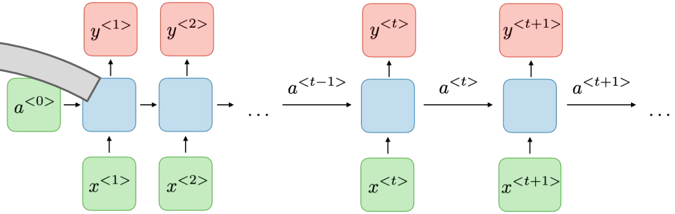
- NN where the hidden layer has a “loop” connection
- The link works as extra input for the next training data instance
    - Makes the model apt for sequential data
- Notice: “A” is always the same

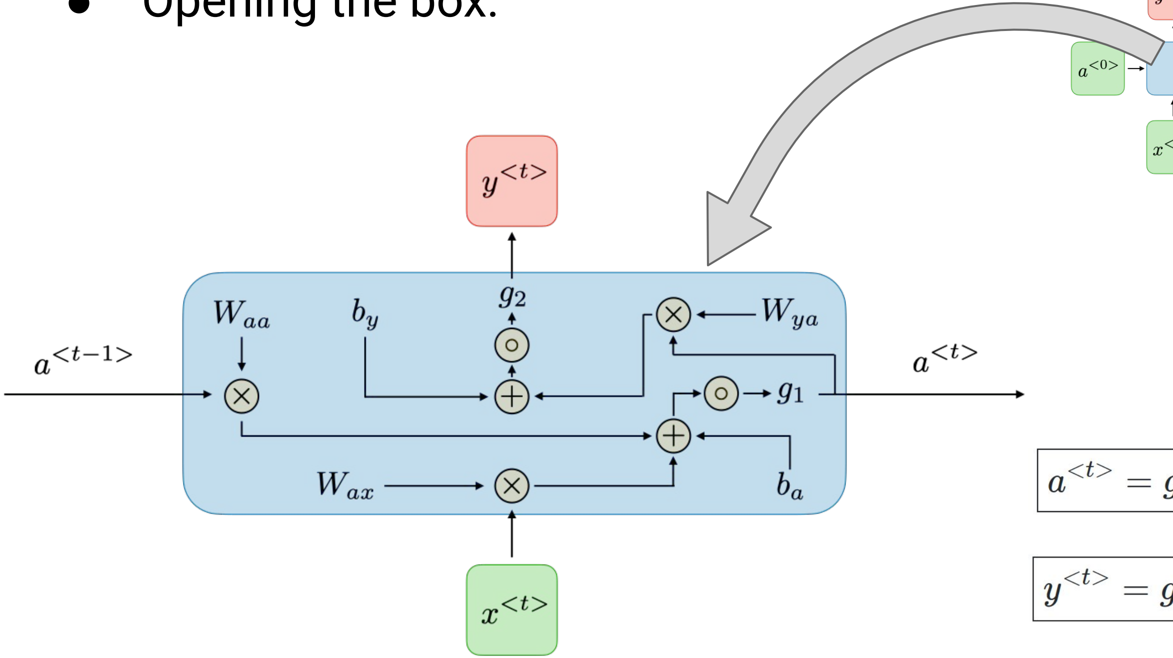


- E.g. $<x_0, \ldots, x_t>$ = input trajectory, $<h_0, \ldots, h_t>$ = output trajectory

RNN

- Opening the box:



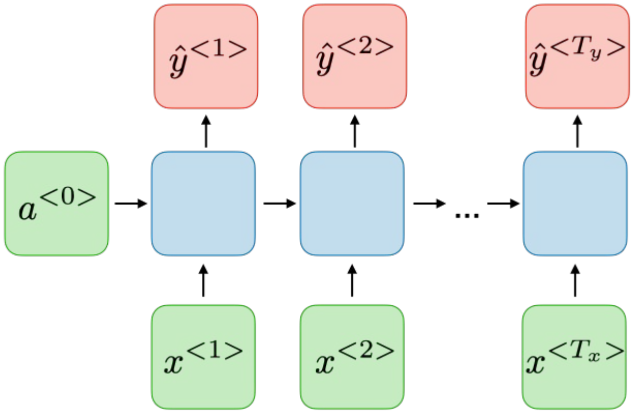


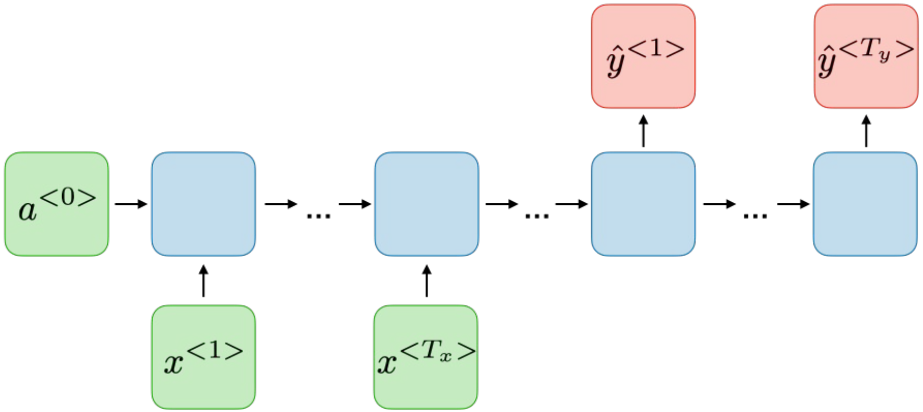
$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

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

RNN

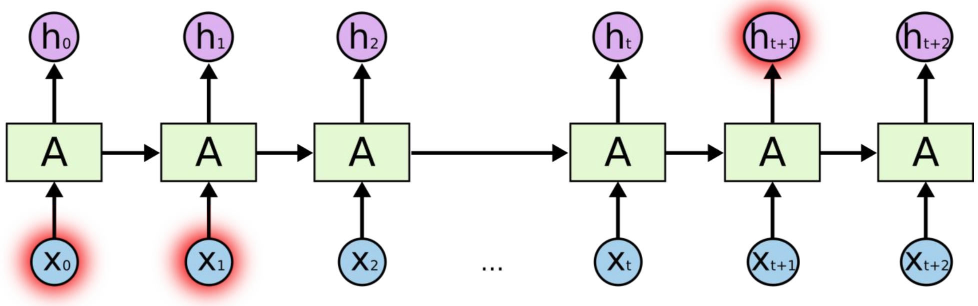
- Two main “many-to-many” architectures:



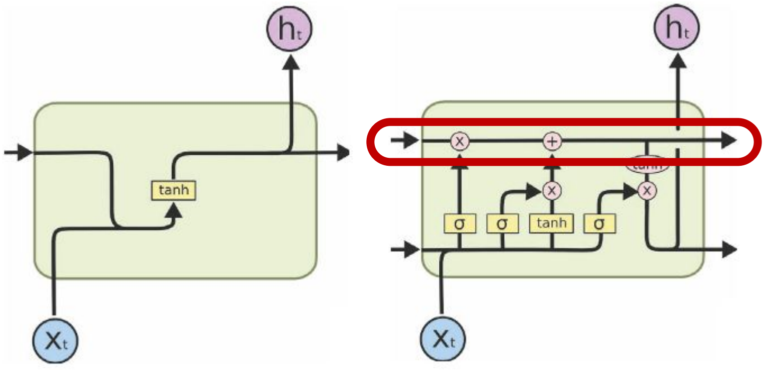


RNN - LSTM

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



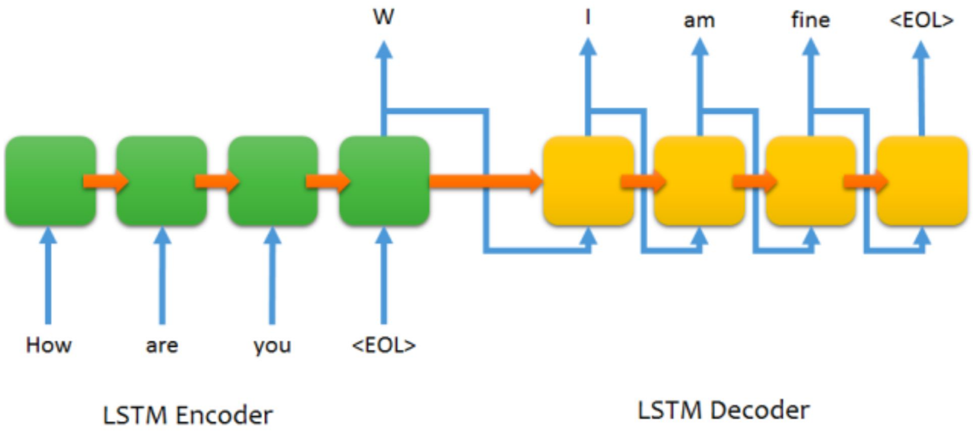
- Solution: pass a state information that is incrementally updated



(a) RNN (b) LSTM

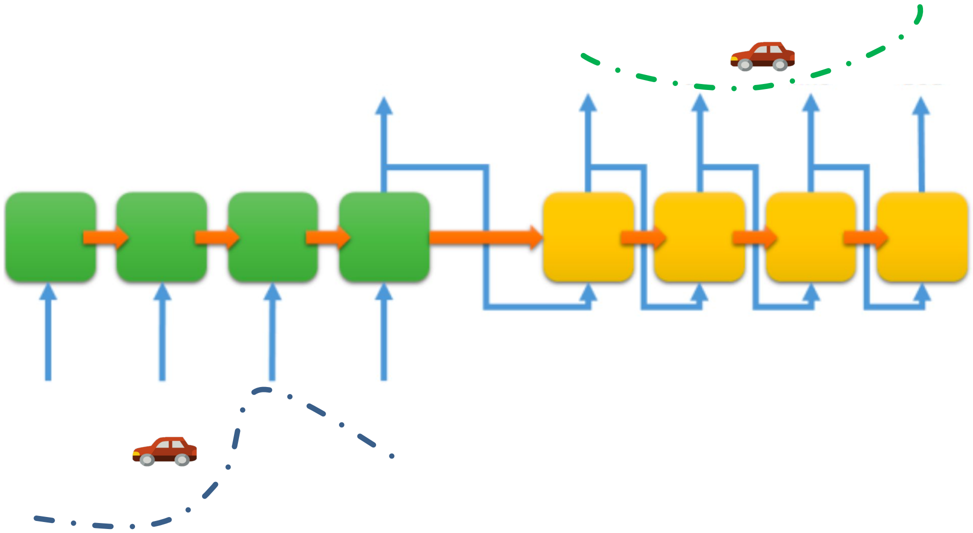
RNN - LSTM - Encoder-Decoder schema

- **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)



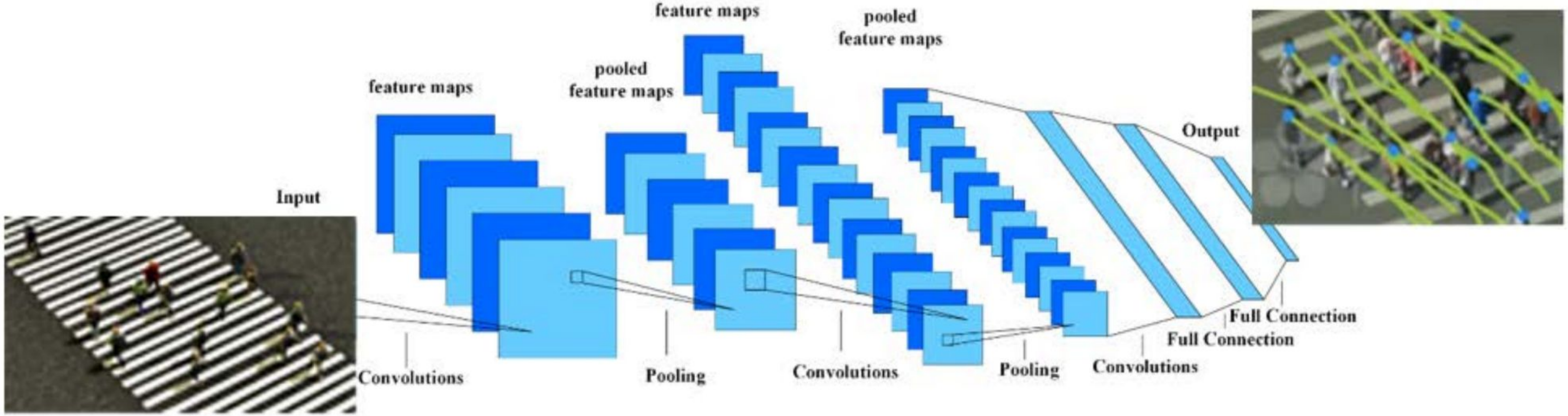
RNN - LSTM - Encoder-Decoder schema

- **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)



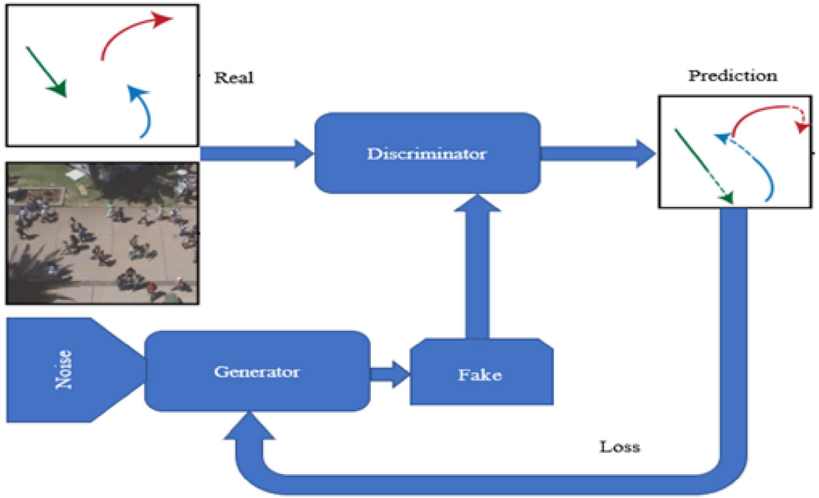
CNN for movement prediction

- 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 Lots 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

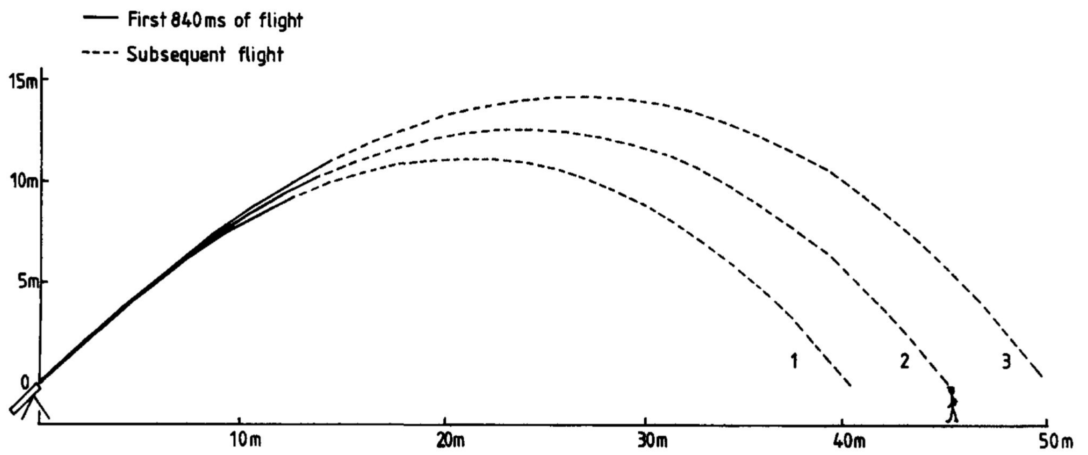
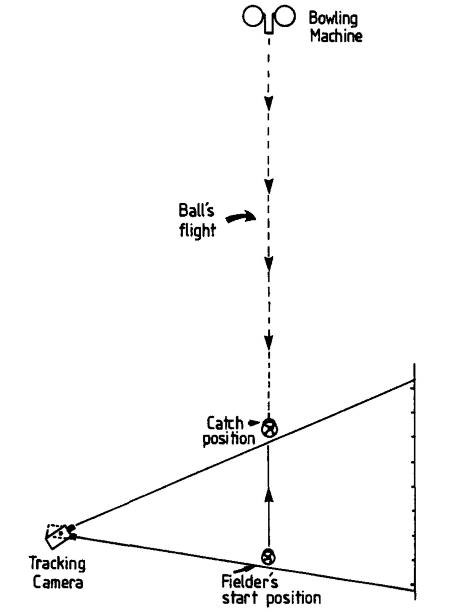


Figure 1. The trajectories of three balls projected at 45° and a velocity (v) of 22.3, 24.0, and 25.7 m/s, respectively, toward a fielder 45 m away. They experienced a deceleration due to aerodynamic drag proportional to v^2. The constant of proportionality was 0.007 m^{-1}, a value typical of objects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

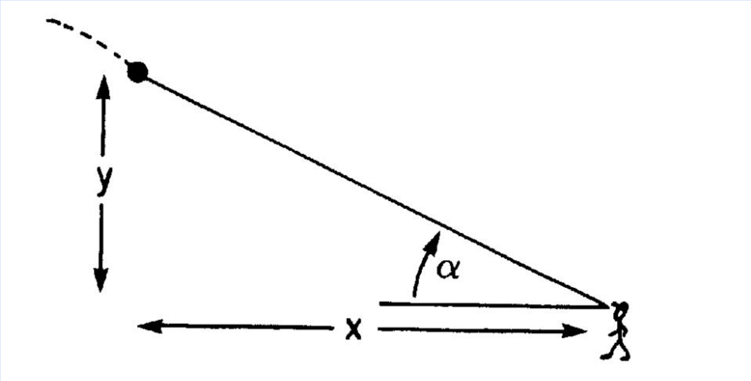


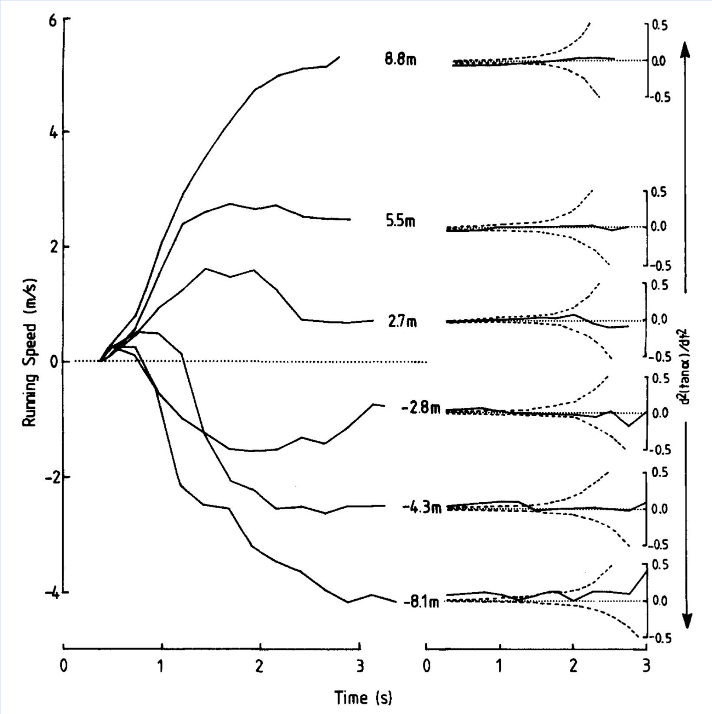
INTERVALLO

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



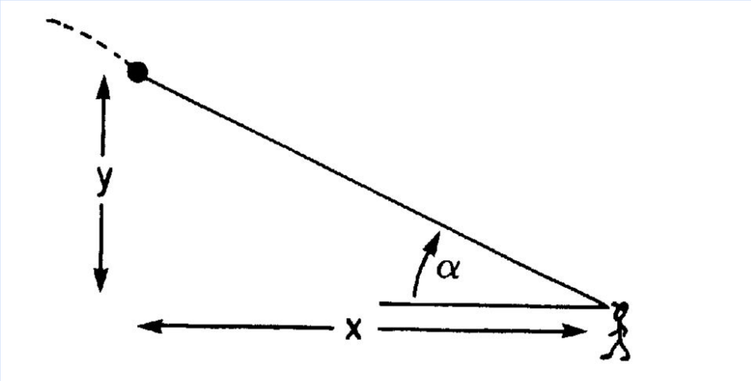


INTERVALLO

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

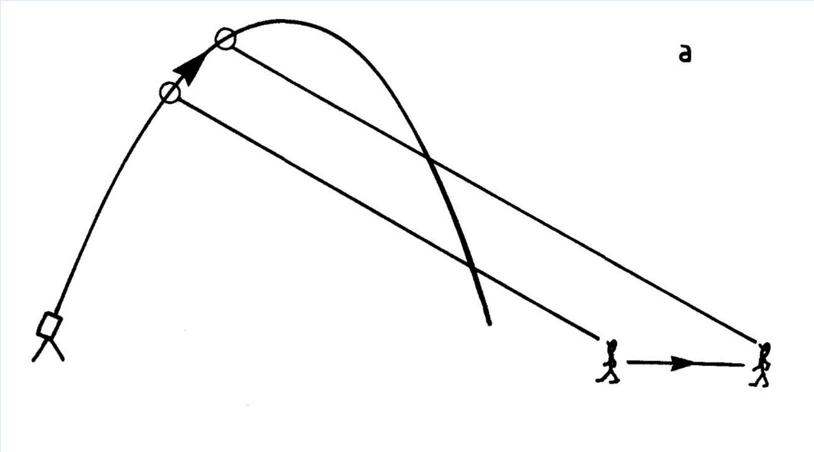
Discovery

- Fielders adjust movement in order to keep $\mathbf{d^2(\tan\alpha)/dt^2 = 0}$
- 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

Homework 10.1

HMM-based trajectory generation

Select a sample of taxi data in SF and train a HMM (for instance discretize trips to sequences of cells and then use CategoricalHMM). Take 5 random trips T (not used in the training), cut them in two equal length parts T1 and T2, use the HMM to understand what is the most likely final state of T1 and then randomly generate a possible continuation T3 of the trip. Finally, compare T2 and T3.

- Write a (well commented) python notebook

Homework 10.2

Pattern-based prediction

Randomly select a set MP of 100 trips in SF taxi data (and pretend they are our representative mobility profiles), and another set TS of 5 trips (our test set). (1) Cut all trips T in MP U TS in two equal length parts T_1 and T_2 ; (2) for each T in TS find the 3 trips T in MP that minimize the dist(T_1,T_1*), where dist() is a trajectory distance of your choice (e.g. Hausdorff); (3) compare each T_2 with the three corresponding T_2* "predicted".*

- Write a (well commented) python notebook