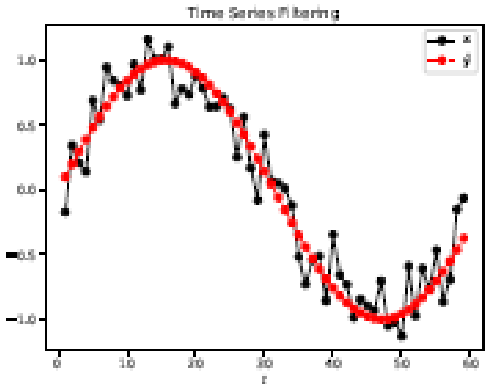
DATA ANALYTICS

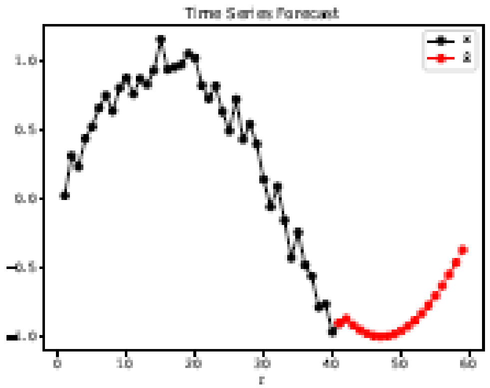
Unit 3: Time Series – Feature Extraction and Classification

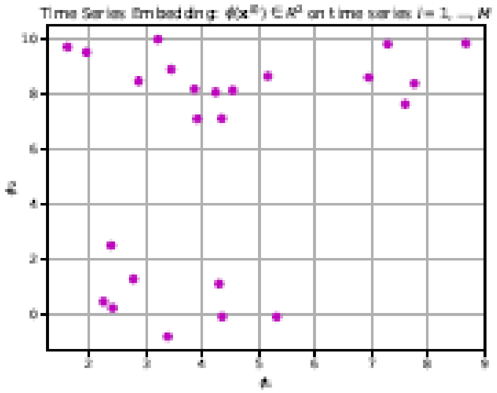
Jyothi R.
Department of Computer Science and Engineering

Time Series Tasks

- Filtering (*estimate*) $\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{y}_t$ from observations
- $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t$
- Forecasting (*predict*) $\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots$ from time t. Embedding: Describe a time series $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ as a vector $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_N]$ of fixed length N.

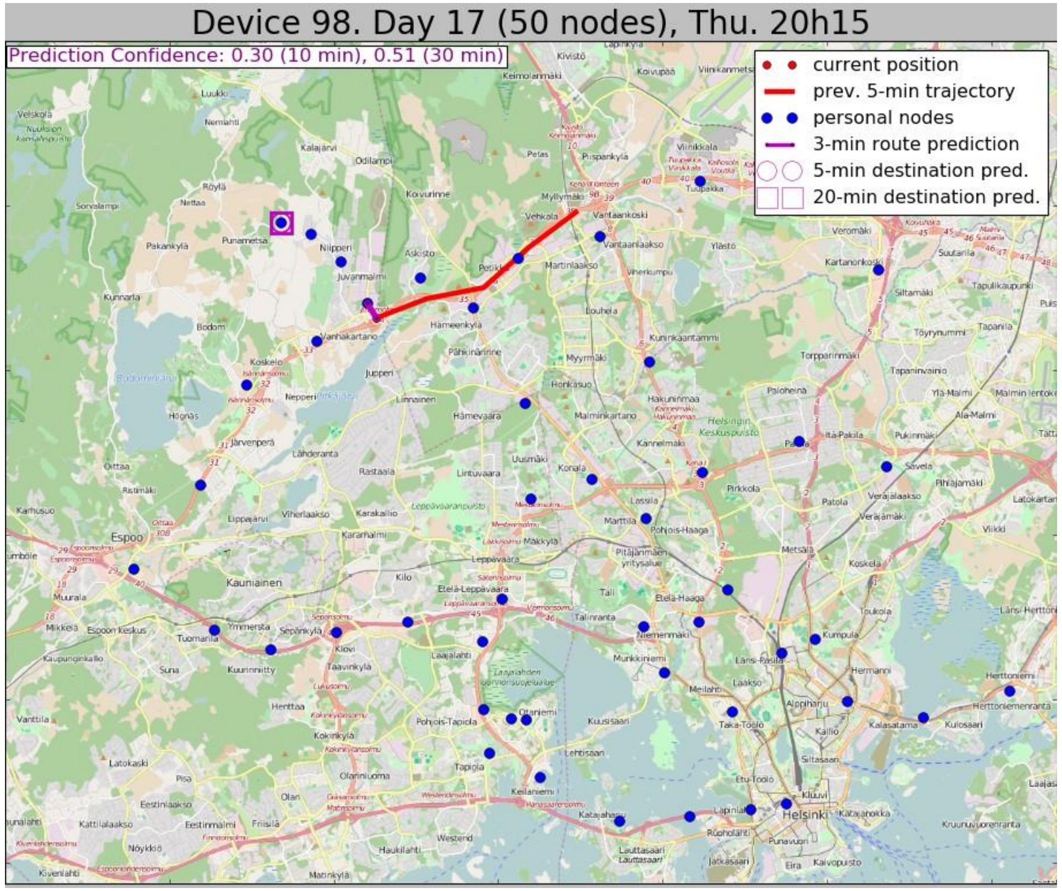






- Clustering
- Classification
- Motif extraction
- Novelty/anomaly detection
- Query by content

Machine Learning for Forecasting



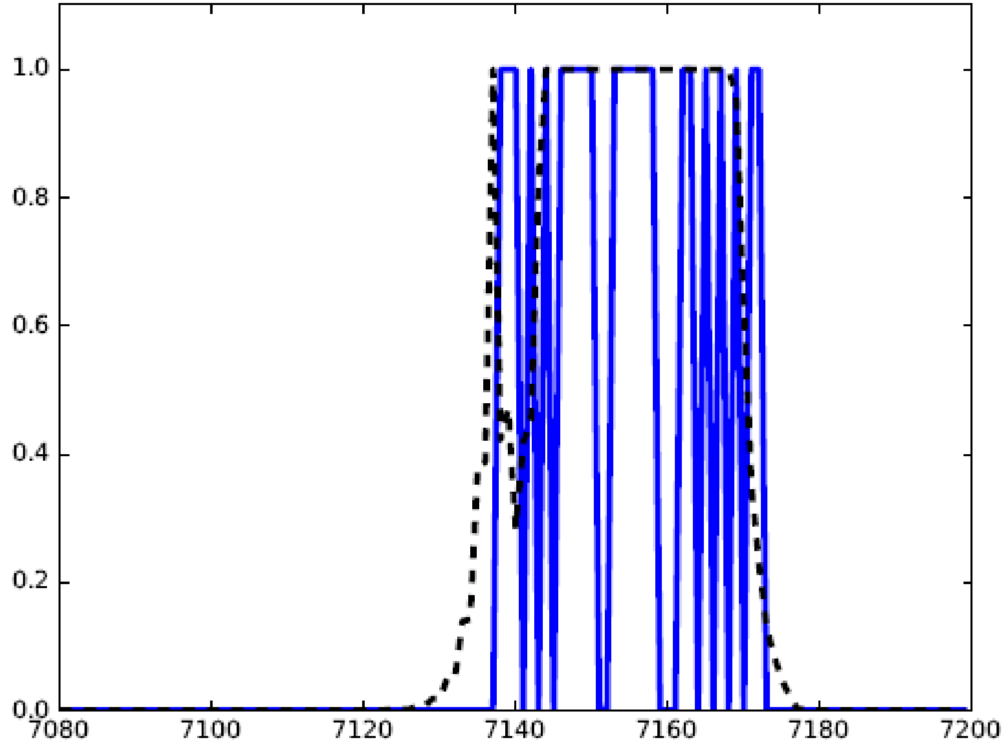
- Collected data of travellers[1]: GPS coordinates, signal strength, battery level, current time, . . .
- Predict future trajectory from current trajectory

[1] All participants volunteered to install App; share data

Work with Jaakko Hollmèn et al. @Aalto University

Example: Predictive Maintenance of Aircraft

- Sensor readings from aircraft and textual description of observations
- Predict warnings/required replacement of components

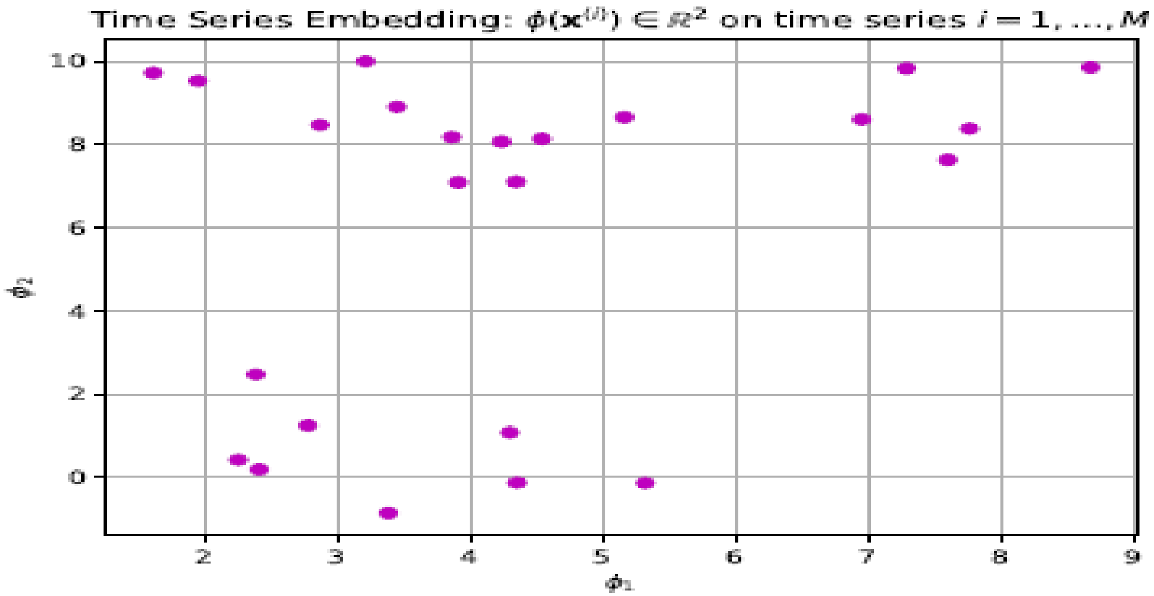


Work with DaSciM team @École Polytechnique

Embedding Time Series

We seek to turn variable-length time series $\{x_1^{(i)}, \ldots, x_{T_i}^{(i)}\}_{i=1}^{M}$

into fixed-length vectors $\boldsymbol{\varphi}^{(i)} = [\varphi_1, \ldots, \varphi_L]$.



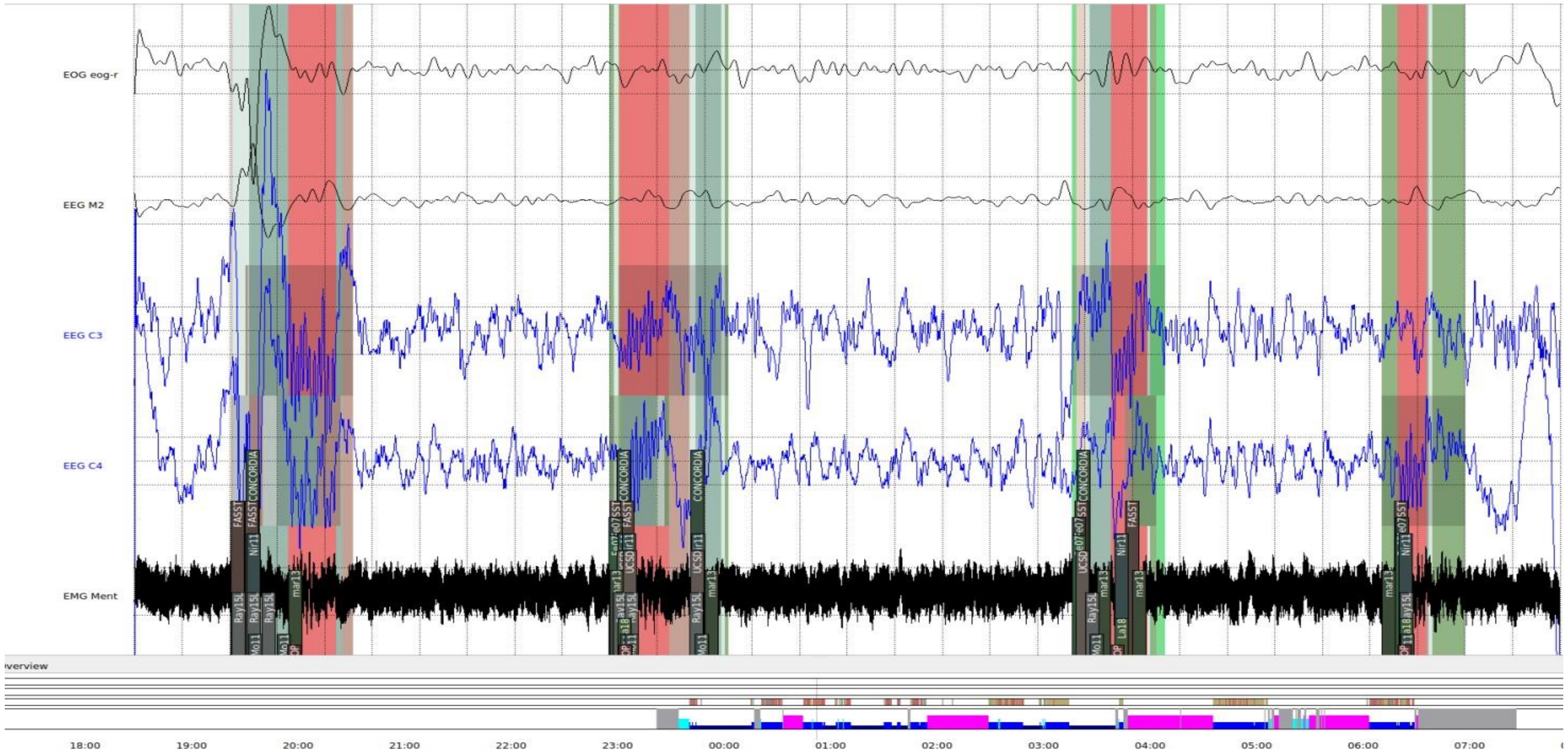
- This lets us compare and cluster time series/look for anomalies, (and classify, if we have the label): measure similarity/distance between $\varphi(\mathbf{x}^{(i)})$ and $\varphi(\mathbf{x}^{(2)})$.

Example: Modelling and Treating Chronic Insomnia

- Goal: (semi-)automate clinical assessment; what kind of insomnia + treatment recommendation.
- Data from patients:
  - Psychological questionnaires (MMPI, CAS) EEG and ECG data overnight
  - Some labels: follow-up tests/questionnaires and *biofeedback* results (some patients found success without pharmaceutical intervention, others not)

- Questionnaire data: can take 'standard' machine learning approach, $f : X \rightarrow Y$, and inspect feature importance, statistical correlation wrt to label variable (extent of insomnia, and improvement); cluster into groups, etc.
- Time-series data: different lengths, contains artifacts, subjects fall asleep at different times, ... How to compare?

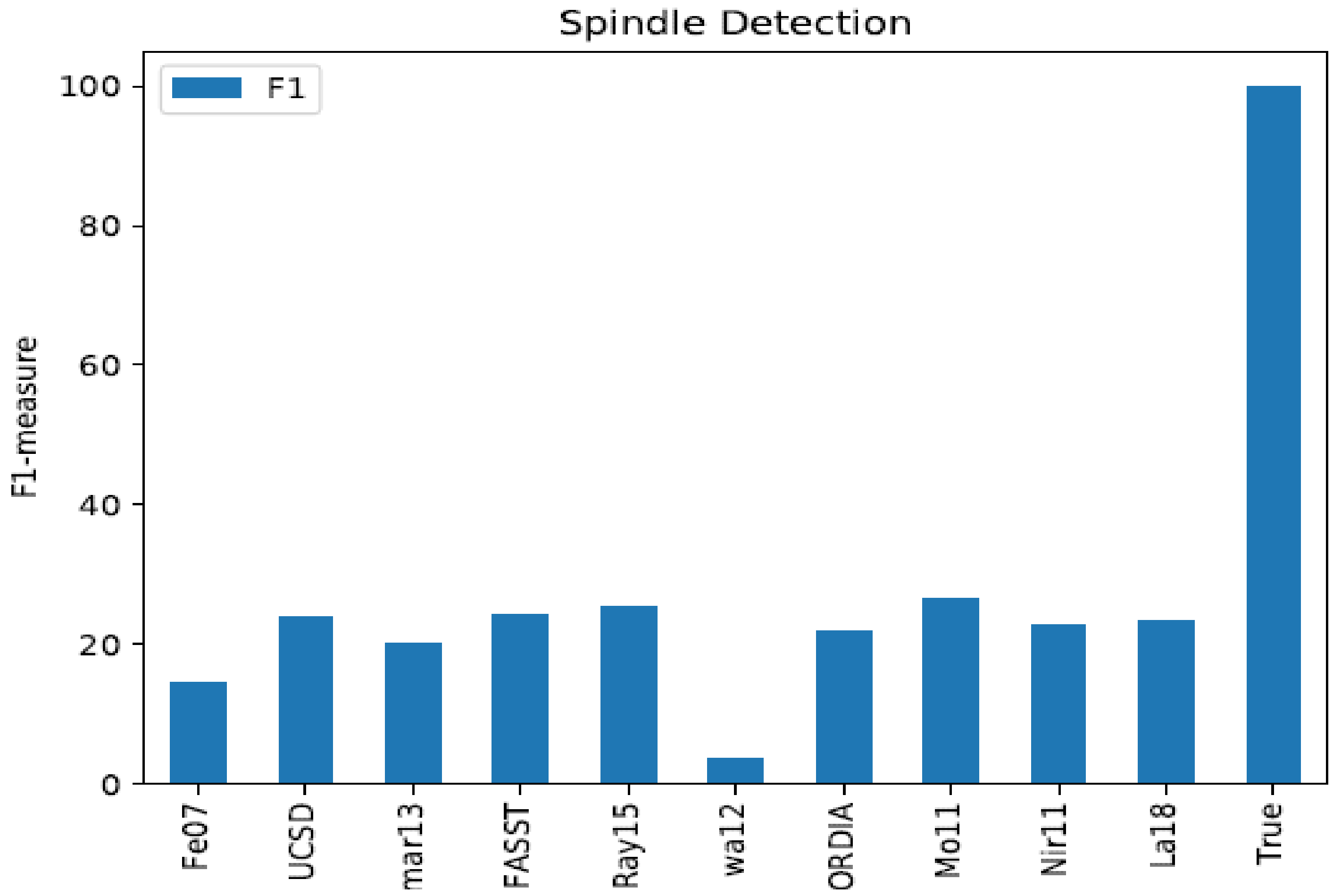
Example: Modelling and Treating Chronic Insomnia



- Certain signals are of interest: Spindles, α-waves, β-waves, . . . Simple embeddings, e.g.,
- $\varphi(\mathbf{x}^{(i)})$ = [spindles/hour, avg freq of spindle]. Detection and labelling by an expert is labour intensive.

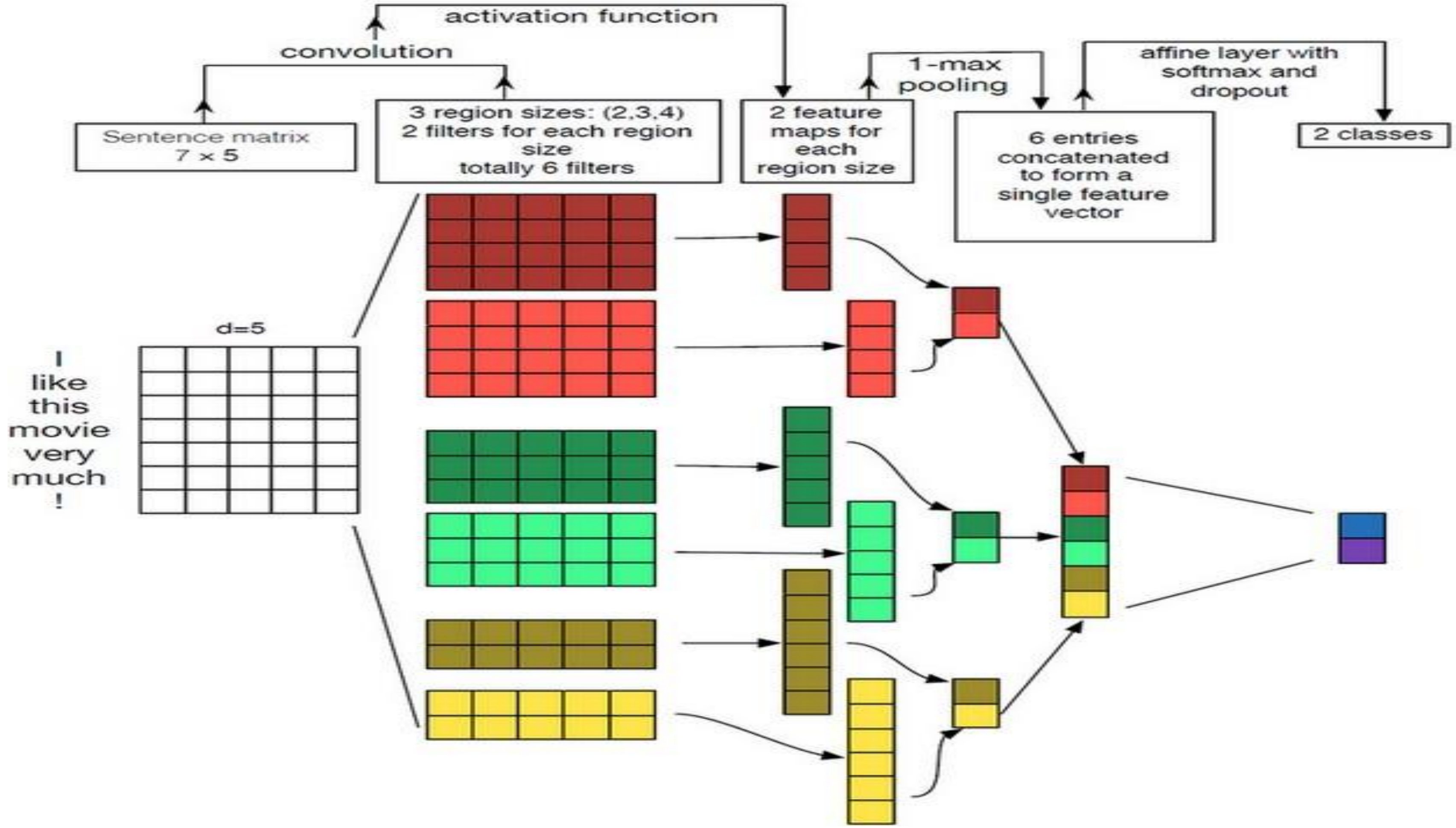
Outline

There exist many rule-based methods, e.g., wavelet analysis But predictive performance is insufficient in many practical settings

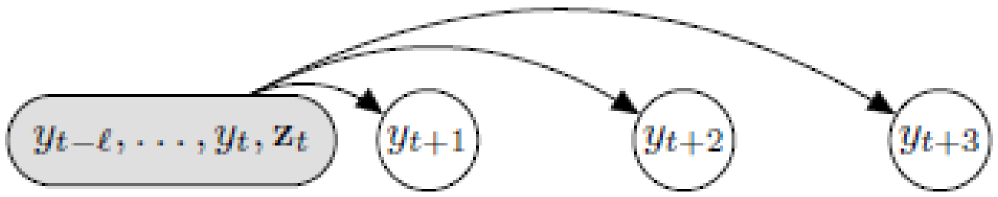


Deep learning

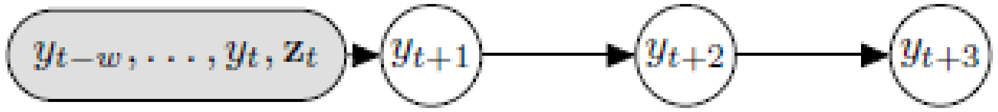
- Many current solutions are inspired by / related to NLP.
- Similar to a ‘simple’ embedding, but more data-driven.



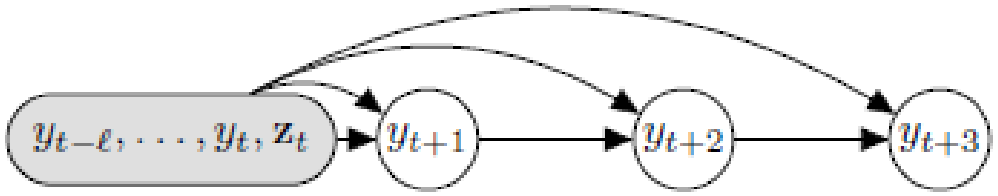
Multi-Step-Ahead Forecasting



Direct

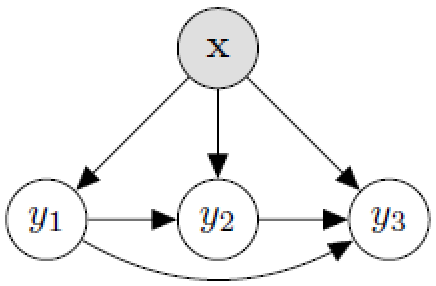


Iterated

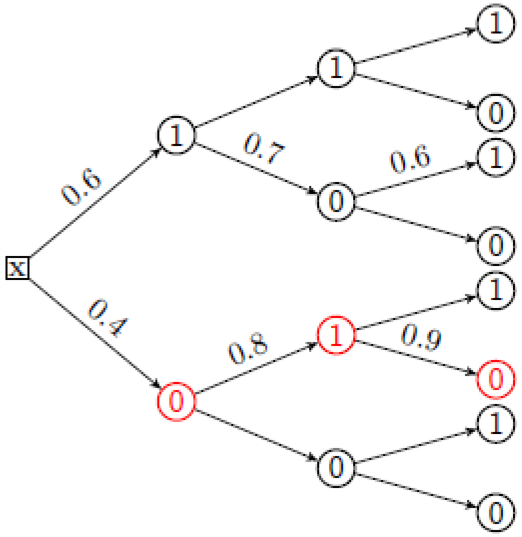


Classifier/Regressor Chain cascade

Classifier Chains



For example, where each $y_t \in \{0, 1\}$



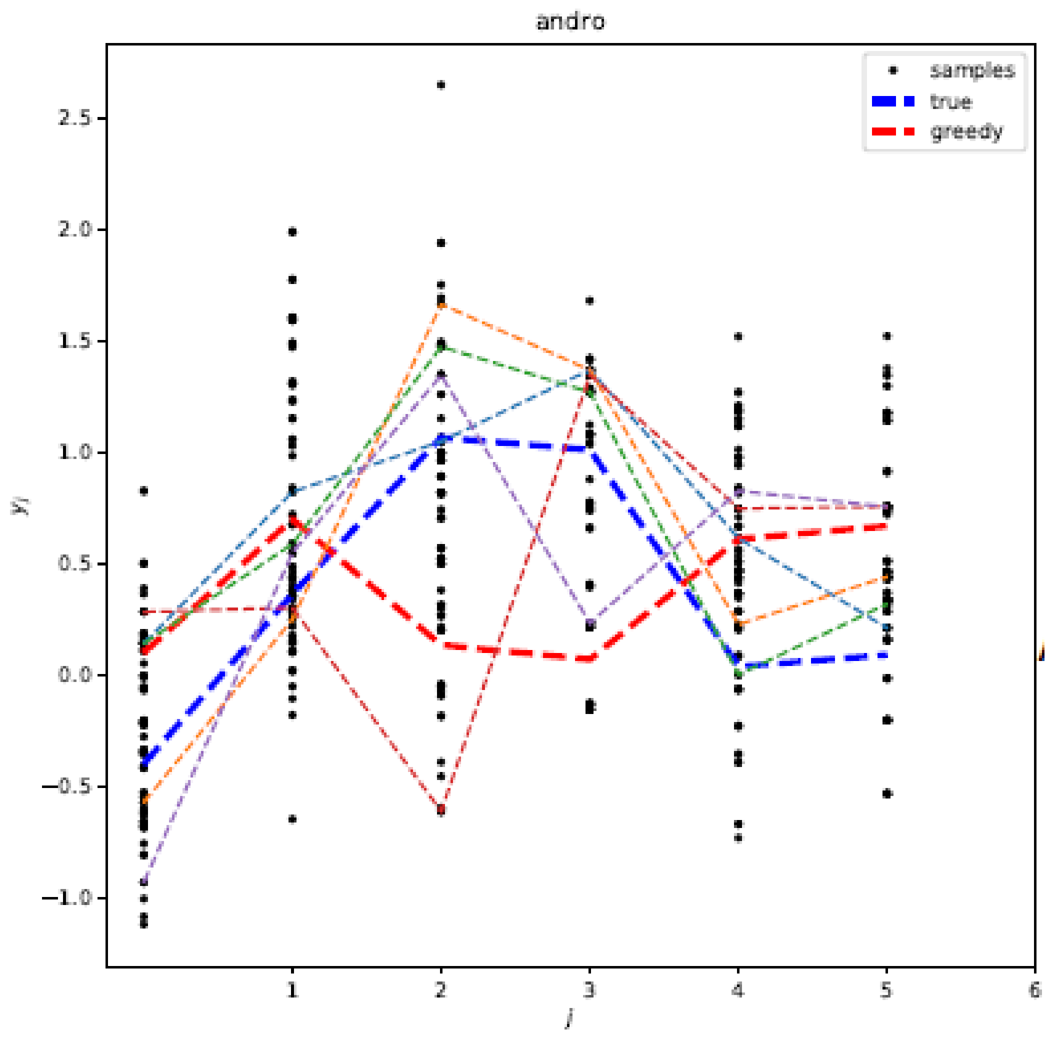
- Predictions become input, across a cascade/chain
- Efficient
- Probabilistic interpretation:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P(y_t|\mathbf{x}, y_1, \ldots, y_{t-1})$$

$$\hat{\mathbf{y}} = f(\mathbf{x}) = \underset{\mathbf{y}\in\{0,1\}^3}{\operatorname{argmax}} P(\mathbf{y}|\mathbf{x})$$

- Search probability tree (for best prediction) with AI-search techniques (Monte-Carlo search, beam search, A* search, . . .)
- Explore structure

Regressor Chains



- e.g., where $\mathbf{y} \in R^6$,
  - Sample down the chain
  - $y_{t+1} \sim p(y_{t+1}|y_1, \ldots, y_t, \mathbf{x})$
  - More samples = more hypotheses
  - Consider different loss functions
- Applications:
  - Multi-output regression
    Tracking
  - Forecasting

One-Step Decision Theory

Under uncertainty, we wish to assign $y^* = f^*(\mathbf{x})$, the best label/hypothesis, $y^* \in Y$, given $\mathbf{x} \in R^D$

.Minimizing conditional expected loss

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \underbrace{\sum_{y \in \mathcal{Y}} \ell(f(\mathbf{x}), y) P(y|\mathbf{x})}_{\mathbb{E}_{Y \sim P(Y|\mathbf{x})}[\ell(\hat{y}, Y)|\mathbf{x}]}$$

under loss function A, which describes our preferences. In the case of 0/1 loss (1 if $y \neq \hat{y}$, else 0),

Maximum a Posteriori

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(\mathbf{x}|y) P(y) = \underset{y \in \{0,1\}}{\operatorname{argmax}}\, P(y|\mathbf{x})$$

We can estimate P from the training data.

Expected Utility

- An intelligent agent wishes to make a decision to achieve a goal.

  The decision which involves the least risk. Another way of looking at the problem: utility.

  A rational agent maximizes their expected utility, not necessarily a simple *payoff* (e.g., amount of money): **Expected Utility** $U(y) = \sum_{y \in \mathcal{Y}} u(y)p(y)$

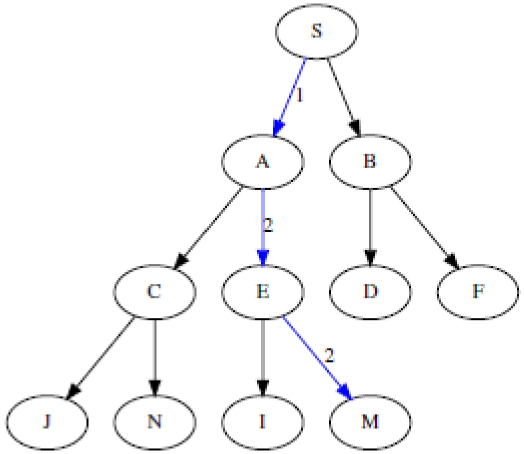
- with satisfaction/utility *u*(*y*) for outcome *y* . Different agents may have different utility functions, even when ‘payoff’ is the same item. Instead of labels given input, we can deal with actions given evidence and belief.

  - A risk-prone agent will tend to gamble higher stakes A conservative (risk-adverse) agent will not
  - A risk-neutral agent only cares about payoff *y* directly

What about sequential decisions?

In a Deterministic Environment

(e.g., board games – chess, etc.)



The state space, e.g., $s_t \in \{A, B, \ldots, M\}$

An initial state, e.g., $s_0 = S$

A goal state, e.g., $s_t = M$

A set of actions, e.g., $a_t \in \{1, 2\}$

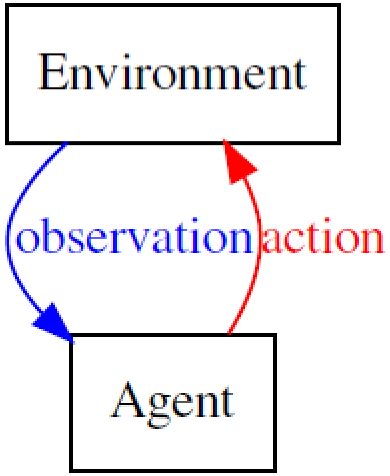
A cost for each branch, e.g., $\text{Cost}(S, A) = 1$

It's just a search! AI-search techniques applicable (DFS, A^*, . . .).

Markov Decision Processes (MDP)

MDPs are models that seek to provide optimal solutions for stocastic sequential decision problems.

$$\text{MDP} = \text{Markov Chain} + \text{One-step Decision Theory}$$



Outline

Now we have a model with

$P(s^j|s, a)$ transition function

$R(s^j, a, s)$ reward function

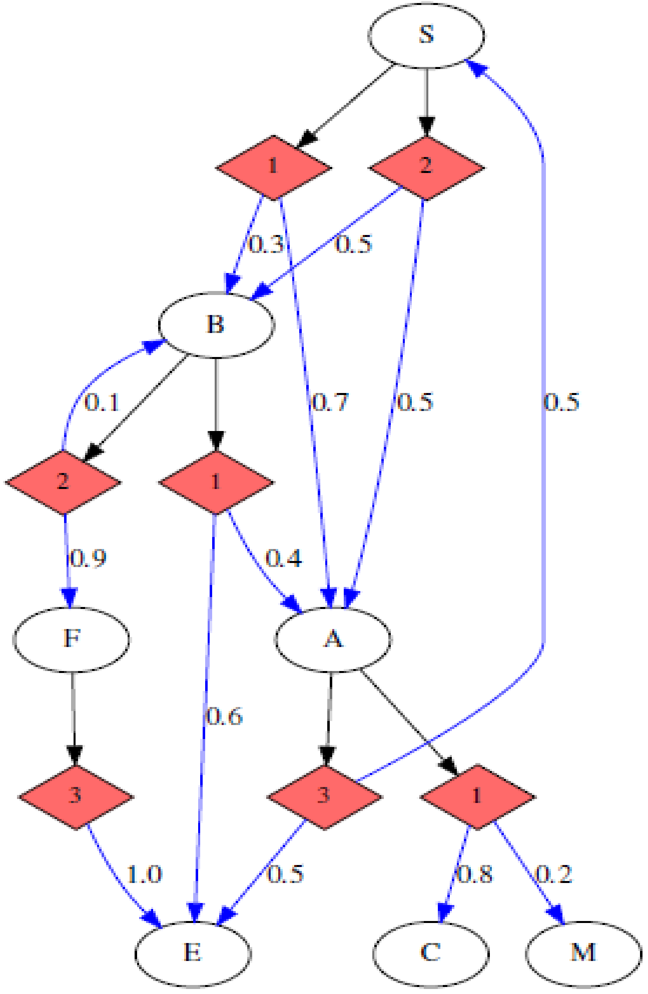
Objective: obtain a policy

$$\pi : \mathcal{S} \mapsto \mathcal{A}$$

which maximizes expected reward:

$$\mathbb{E}[R_0|s_0 = s] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)\right]$$

solution can be found via dynamic programming!
Just need the model . . .



Reinforcement Learning

- We don’t have the model!
- Don’t have transition/reward functions.
- No input-output training pairs, just reward signal.
- The agent needs to experiment! Exploration vs exploitation. Deep neural net can learn a model
- . . .over millions of iterations. Emerging applications:
  - Gameplay
  - Robotics (usually trained in simulation) Parameter-tuning, etc. (as a tool)
- Transfer learning is promising

Image Courtesy

https://www.abs.gov.au/websitedbs/D3310114.nsf/home/Time+Series+Analysis:+The+Basics

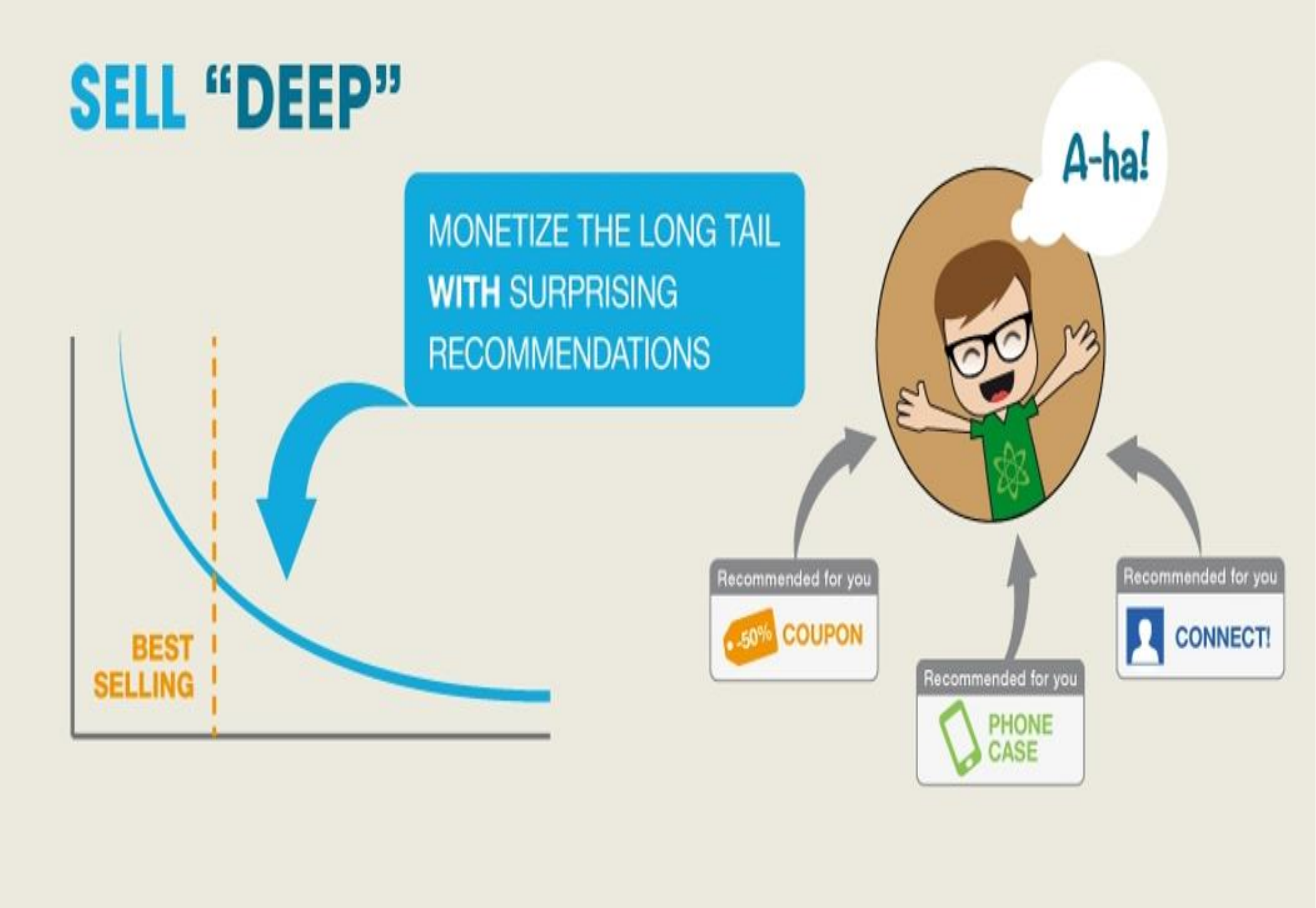
https://otexts.com/fpp2/stationarity.html
https://www.abs.gov.au/websitedbs/D3310114.nsf/home/Time+Series+Analysis:+The+Basics
https://bookdown.org/rdpeng/timeseriesbook/spectral-analysis.html
https://www.stat.berkeley.edu/~bartlett/courses/153-fall2010/lectures/15.pdf
https://astrostatistics.psu.edu/su07/fricks_2timeseries07.pdf)
https://blog.octo.com/en/time-series-features-extraction-using-fourier-and-wavelet-transforms-on-ecg-data/
https://jmread.github.io/talks/Time_Series_AI.pdf

DATA ANALYTICS

Unit 4: Introduction to Recommendation System

Jyothi R.
Department of Computer Science and Engineering

DATA ANALYTICS
Why Recommendation Systems?



"We are leaving the age of information and entering the age of recommendation"

-Chris Anderson in "The Long Tail"

DATA ANALYTICS
Age of Recommendation

Search:

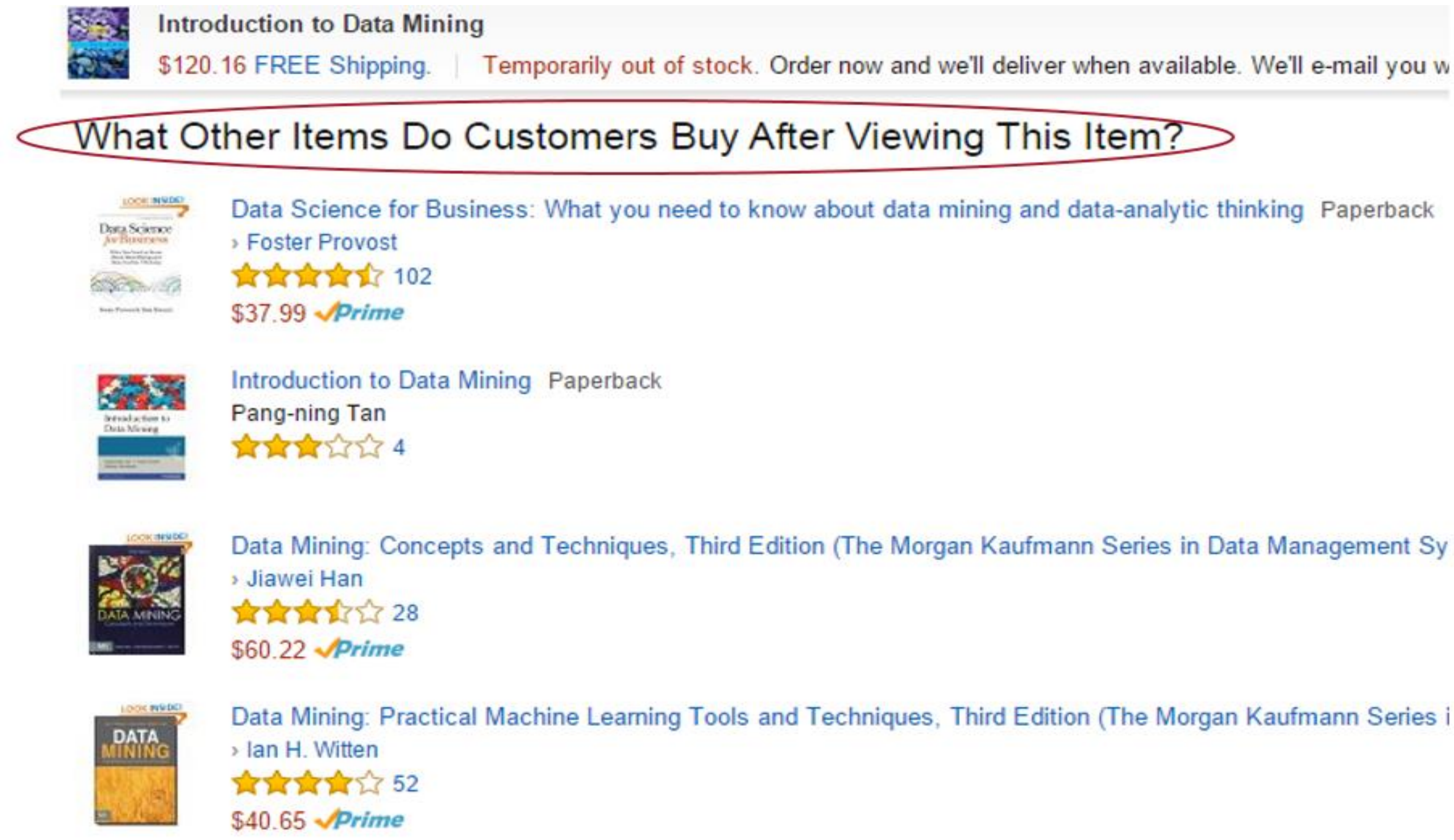
User → Items

Recommend:

Items → User

DATA ANALYTICS
Amazon A personalized online store

DATA ANALYTICS
Amazon A personalized online store



Recommender Problem

A Good recommender

- Show programming titles to a software engineer and baby toys to a new mother
- Don’t recommend items, which user already knows or would find anyway.
- Expand User’s taste without offending or annoying him/her..

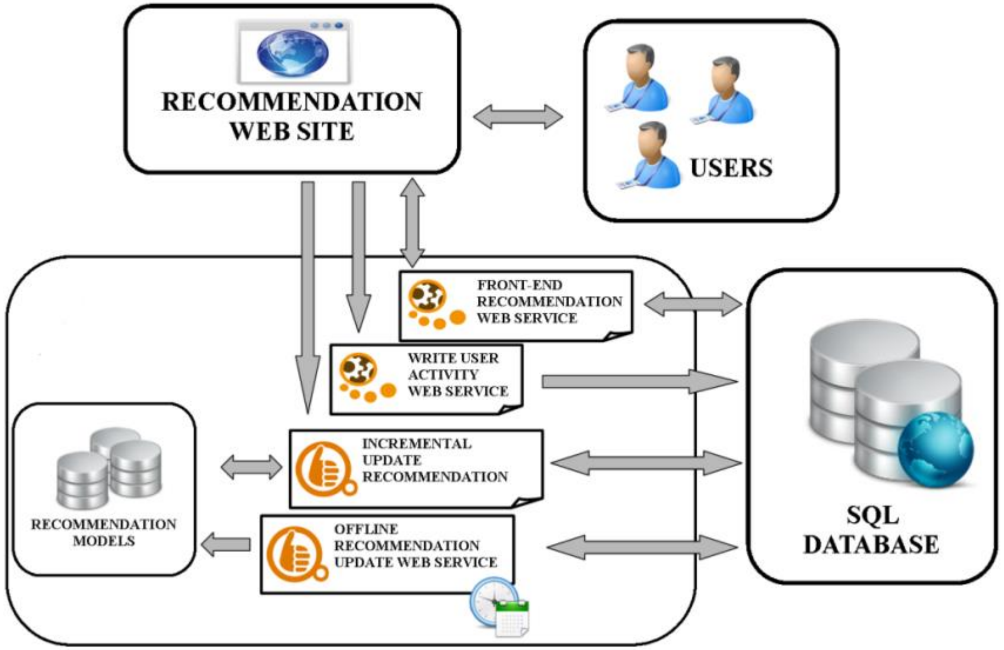
Challenges

- Huge amounts of data, tens of millions of customers and millions of distinct catalog items.
- Results are required to be returned in real time.
- New customers have limited information.
- Customer data is volatile.

DATA ANALYTICS
Amazon's solution

1. Amazon Recommendation Engine: Amazon's model that implements recommendation algorithm. Recommendation algorithm is designed to personalize the online store for each customer.

2. Recommendation Engine Workflow:



Recommendations:

Introduction to Recommendation system

- Recommendation system - problem of information filtering

- Enhance user experience
  - Assist users in finding information
  - Reduce search and navigation time

- Recommender systems are the most popular applications of data science today, to Increase productivity, Increase credibility.

- They are used to predict the “rating” or “preference” that a user would give to an item.

- Amazon uses it to suggest products to customers.

- YouTube uses it to decide which video to play next on auto play, and,

- Facebook uses it to recommend pages to like and people to follow.

- Most of the companies business model and its success revolves around the potency of their recommendations.

Goals of Recommender Systems

1. **Prediction version of problem:** the first approach is to predict the rating value for a user-item combination. It is assumed that training data is available, indicating user preferences for items. For m users and n items, this corresponds to an incomplete mxn matrix, where the specified values are used for training.

2. **Ranking version of problem:** In practice, it is not necessary to predict the ratings of users for specific items in order to make recommendations to users. The determination of the top-k items is more common than the determination of top-k users.

Goals of Recommender Systems Contd.

In order to achieve broader business-centric goal of increasing revenue, the operational and technical goals of recommender systems are as follows

1. **Relevance**: Users are more likely to consume items they find interesting, rating value for a user-item combination.

2. **Novelty:** Recommender systems are truly helpful when the recommended item is something that the user has not seen in the past. For example, Popular movies of a preferred genre would rarely be novel to the user.

3. **Serendipity:** The items recommended are somewhat unexpected, and therefore there is a modest element of lucky discovery. Recommendations are truly surprising to the user. It leads to sales diversity or beginning a new trend of interest in the user.

4. **Increasing Recommendation Diversity:** It has the benefit of ensuring that the user does not get bored by repeated recommendation of similar items.

Goals of Recommender Systems Contd.

- Aside from these concrete goals, a number of soft goals are also met by the recommendation process both from the perspective of the user and merchant.
- The broad diversity of recommender systems that were built either as research prototype, or are available today as commercial systems in various problem settings

1. GroupLens Recommender System
2. Amazon.com Recommender System
3. Netflix Movie Recommender System
4. Google News Personalization System
5. Facebook Friend Recommendations

The Spectrum of Recommendation Applications

1. Collaborative Filtering Models
   i) Memory - based collaborative filtering
   a) User-Based collaborative filtering
   b) Item-based collaborative filtering
   ii) Model-Based Methods
   a) Types of Ratings: Implicit and Explicit Ratings
   b) Relationship with missing values.

2. Content-Based Recommender systems

3. Knowledge-Based Recommender Systems
   i) Constraint-based recommender systems
   ii) Case-based recommender systems

4. Demographic Recommender systems

5. Hybrid and Ensemble-Based Recommender Systems

Gathering Ratings

Types of Ratings

- Explicit
- Ask people to rate items
- Doesn’t scale: only a small fraction of users leave ratings and reviews

- Implicit
- Learn ratings from user actions
- E.g., purchase implies high rating
- What about low ratings?

Domain-specific Challenges in Recommender Systems

1. Context-Based Recommender Systems: It could include time, location, or social data. For example, the types of clothes recommended by a retailer might depend both on the season and location of the customer. Even particular type of festival or holiday affects the underlying customer activity.

2. Time-Sensitive Recommender Systems:

i. The rating of an item might evolve with time, as community attitudes evolve and the interests of users change over time. User interests, likes, dislikes, and fashions inevitably evolve with time.

ii. The rating of an item might be dependent on the specific time of day, day of week, month, or season.

iii. For example, it makes little sense to recommend winter clothing during the summer, or Raincoats during the dry season.

Domain-specific Challenges in Recommender Systems

3. Location-Based Recommender Systems
    - i) User-Specific Locality
    - ii) Item-specific Locality

4. Social Recommender Systems
    - i) Structural Recommendation of Nodes and Links
    - ii) Product and Content Recommendations with social influence.
    - iii)Trustworthy Recommender Systems
    - iv) Leveraging Social Tagging Feedback for Recommendations

References

Text Book:

“Business Analytics, The Science of Data-Driven Decision Making”, U. Dinesh Kumar, Wiley 2017

“Recommender Systems, The text book, Charu C. Aggarwal, Springer 2016

Section 1.

Image Courtesy

https://www.scribd.com/presentation/414445910/CS548S15-Showcase-Web-Mining

https://towardsdatascience.com/image-recommendation-engine-leverage-transfert-learning-ec9af32f5239

https://www.youtube.com/watch?v=1JRrCEgiyHM

www.amazon.com

http://elico.rapid-i.com/recommender-extension.html

THANK YOU

Jyothi R.
Assistant Professor,
Department of Computer Science
jyothir@pes.edu