CS6375: Machine Learning

Gautam Kunapuli

The Wide World of Machine Learning

Slides by various authors; acknowledged in respective sections

THE UNIVERSITY OF TEXAS AT DALLAS
Erik Jonsson School of Engineering and Computer Science

10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Matrix Factorization and Collaborative Filtering

MF Readings:
(Koren et al., 2009)

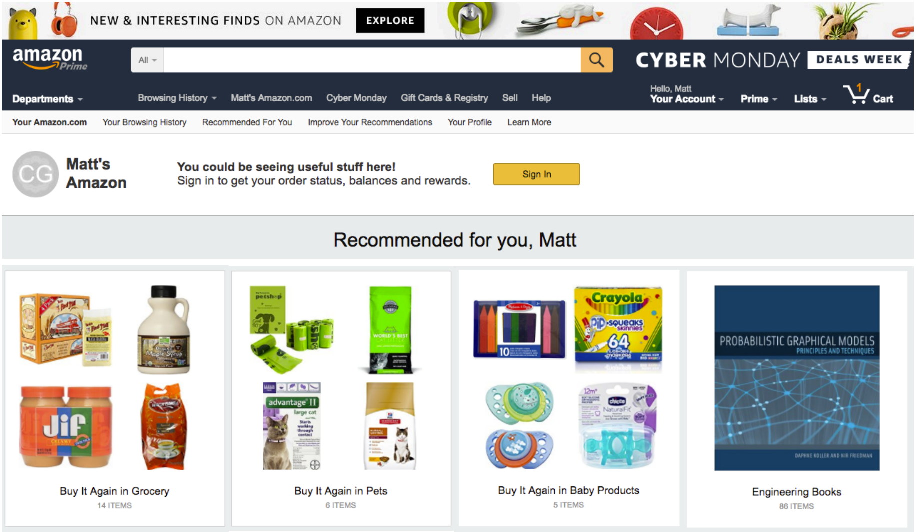
Matt Gormley
Lecture 25
April 19, 2017

Recommender Systems

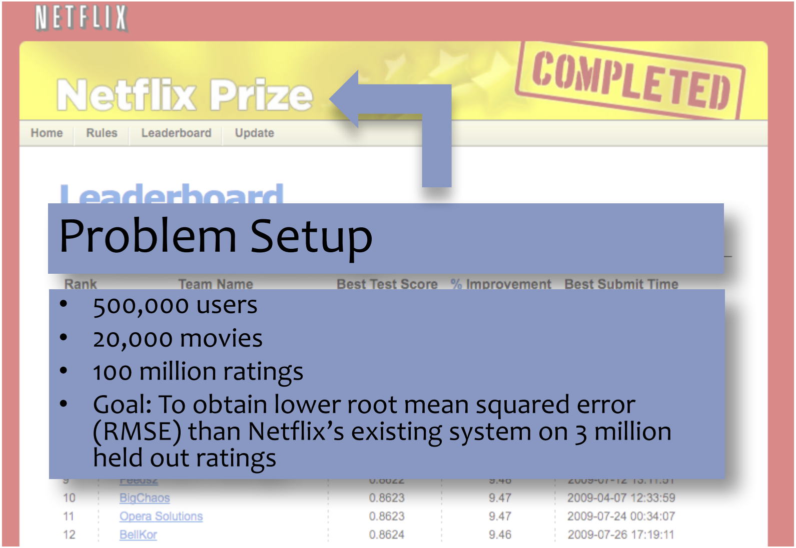
A Common Challenge:

- Assume you're a company selling **items** of some sort: movies, songs, products, etc.
- Company collects millions of **ratings** from **users** of their **items**
- To maximize profit / user happiness, you want to **recommend** items that users are likely to want

Recommender Systems



Recommender Systems



Problem Setup

- 500,000 users
- 20,000 movies
- 100 million ratings
- Goal: To obtain lower root mean squared error (RMSE) than Netflix's existing system on 3 million held out ratings

Recommender Systems

- **Setup:**
  - Items:
    movies, songs, products, etc.
    (often many thousands)
  - Users:
    watchers, listeners, purchasers, etc.
    (often many millions)
  - Feedback:
    5-star ratings, not-clicking 'next',
    purchases, etc.
- **Key Assumptions:**
  - Can represent ratings numerically as a user/item matrix
  - Users only rate a small number of items (the matrix is sparse)

	Doctor Strange	Star Trek: Beyond	Zootopia
Alice	1		5
Bob	3	4	
Charlie	3	5	2

Two Types of Recommender Systems

Content Filtering

- *Example*: Pandora.com music recommendations (Music Genome Project)
- **Con:** Assumes access to side information about items (e.g. properties of a song)
- **Pro:** Got a new item to add? No problem, just be sure to include the side information

Collaborative Filtering

- *Example*: Netflix movie recommendations
- **Pro:** Does not assume access to side information about items (e.g. does not need to know about movie genres)
- **Con:** Does not work on new items that have no ratings

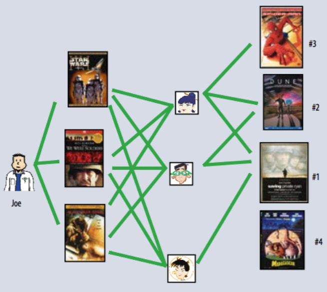
Collaborative Filtering

- **Everyday Examples of Collaborative Filtering...**
  - Bestseller lists
  - Top 40 music lists
  - The "recent returns" shelf at the library
  - Unmarked but well-used paths thru the woods
  - The printer room at work
  - "Read any good books lately?"
  - ...
- **Common insight:** personal tastes are correlated
  - If Alice and Bob both like X and Alice likes Y then Bob is more likely to like Y
  - especially (perhaps) if Bob knows Alice

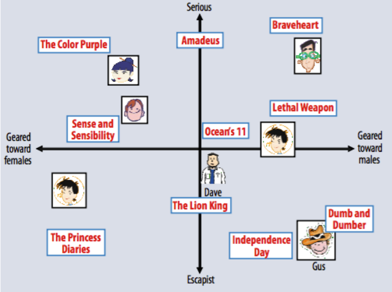
Slide from William Cohen

Two Types of Collaborative Filtering

1. Neighborhood Methods



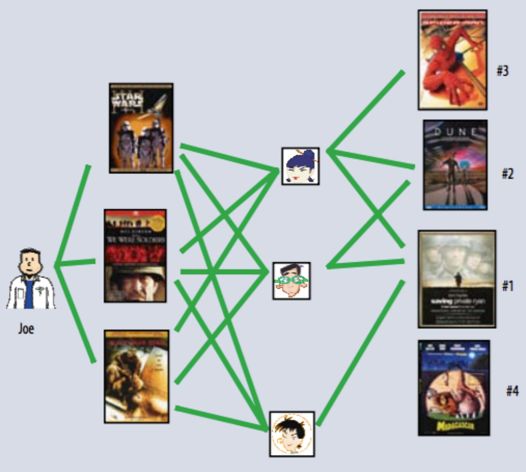
2. Latent Factor Methods



Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

1. Neighborhood Methods



In the figure, assume that a green line indicates the movie was **watched**

Algorithm:

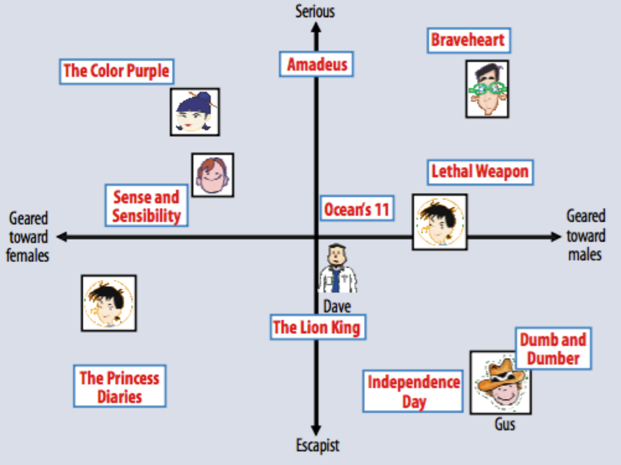
1. **Find neighbors** based on similarity of movie preferences
2. **Recommend** movies that those neighbors watched

Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

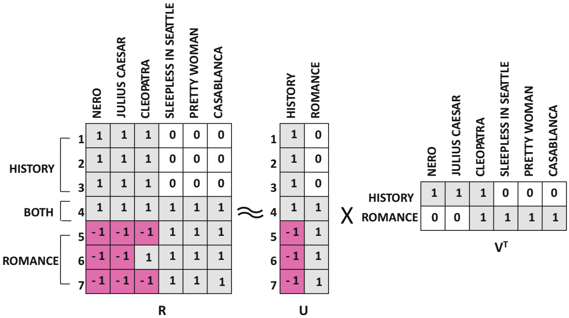
2. Latent Factor Methods

- Assume that both movies and users live in some **low-dimensional space** describing their properties
- **Recommend** a movie based on its **proximity** to the user in the latent space

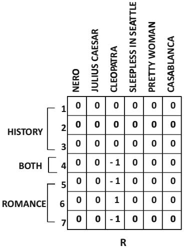


Figures from Koren et al. (2009)

Example: MF for Netflix Problem



(a) Example of rank-2 matrix factorization

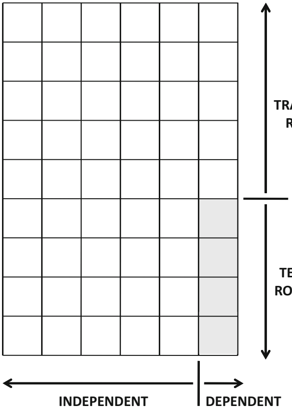


(b) Residual matrix

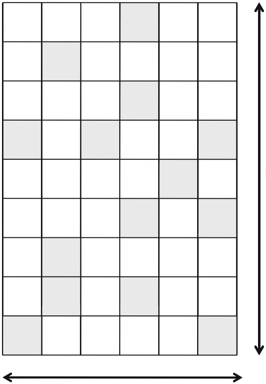
Figures from Aggarwal (2016)

Regression vs. Collaborative Filtering

Regression



Collaborative Filtering



Figures from Aggarwal (2016)

Matrix Factorization
(with matrices)

- User vectors:

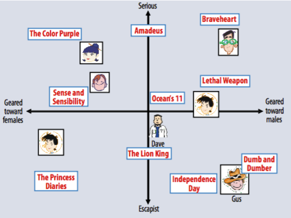
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

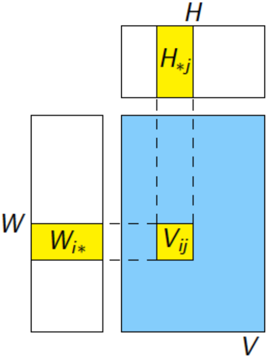
$$H_{*i} \in \mathbb{R}^r$$

- Rating prediction:

$$V_{ui} = W_{u*} H_{*i}$$
$$= [WH]_{ui}$$



Figures from Koren et al. (2009)

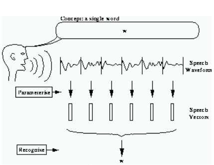


Figures from Gemulla et al. (2011)

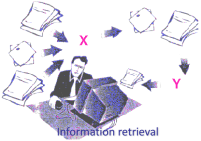
Probabilistic Graphical Model: A view from moon

Kayhan Batmanghelich

Reasoning under uncertainty!

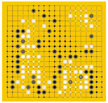


Speech recognition

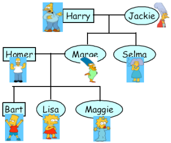


Information retrieval

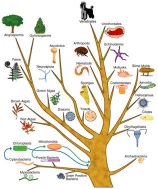

Computer vision

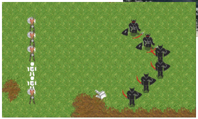

Games


Robotic control



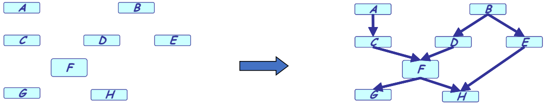
Pedigree


Evolution


Planning

So What Is a PGM After All?

- The informal blurb:
  - It is a smart way to write/specify/compose/design exponentially-large probability distributions without paying an exponential cost, and at the same time endow the distributions with ***structured semantics***



$$P(X_1,X_2,X_3,X_4,X_5,X_6,X_7,X_8)$$

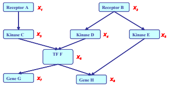
$$P(X_{18}) = P(X_1)P(X_2)P(X_3 \mid X_1X_2)P(X_4 \mid X_2)P(X_5 \mid X_2)$$
$$P(X_6|X_3,X_4)P(X_7|X_6)P(X_8|X_5,X_6)$$

- A more formal description:
  - It refers to a family of distributions on a set of random variables that are compatible with all the probabilistic independence propositions encoded by a graph that connects these variables

Two types of GMs

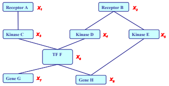
- Directed edges give causality relationships (Bayesian Network or Directed Graphical Model):

$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= P(X_1)\, P(X_2)\, P(X_3|\, X_1)\, P(X_4|\, X_2)\, P(X_5|\, X_2)$$
$$P(X_6|\, X_3, X_4)\, P(X_7|\, X_6)\, P(X_8|\, X_5, X_6)$$



- Undirected edges simply give correlations between variables (Markov Random Field or Undirected Graphical model):

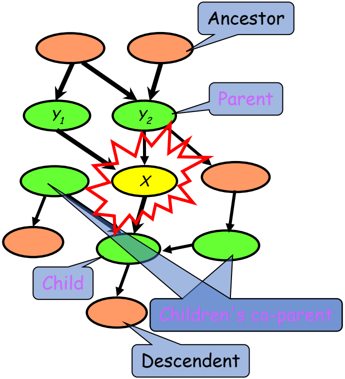
$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= 1/Z \exp\{E(X_1)+E(X_2)+E(X_3, X_1)+E(X_4, X_2)+E(X_5, X_2)$$
$$+ E(X_6, X_3, X_4)+E(X_7, X_6)+E(X_8, X_5, X_6)\}$$



Bayesian Networks

Structure: *DAG*

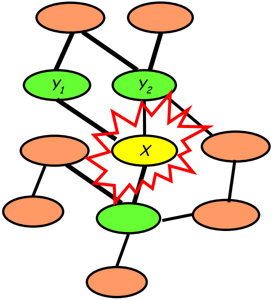
- Meaning: a node is conditionally independent of every other node in the network outside its Markov blanket
- Local conditional distributions (CPD) and the DAG completely determine the joint dist.
- Give causality relationships, and facilitate a generative process



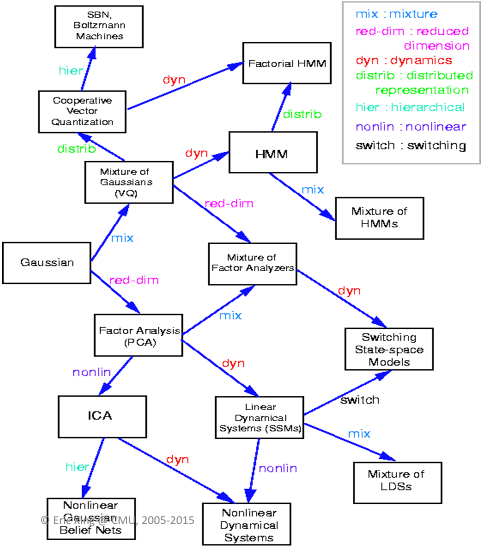
Markov Random Fields

Structure: *undirected graph*

- Meaning: a node is conditionally independent of every other node in the network given its Directed neighbors
- Local contingency functions (potentials) and the cliques in the graph completely determine the joint dist.
- Give correlations between variables, but no explicit way to generate samples



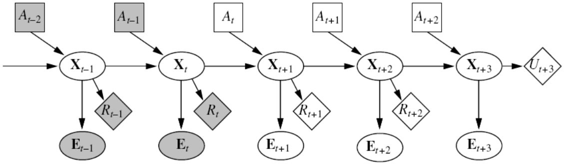
An (incomplete) genealogy of graphical models



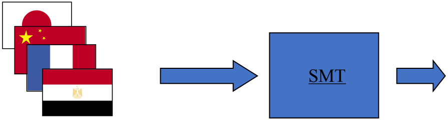
(Picture by Zoubin Ghahramani and Sam Roweis)

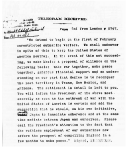
Fancier GMs: reinforcement learning

- Partially observed Markov decision processes (POMDP)



Fancier GMs:
machine translation

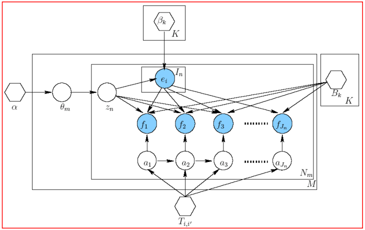




TELEGRAM RECEIVED.

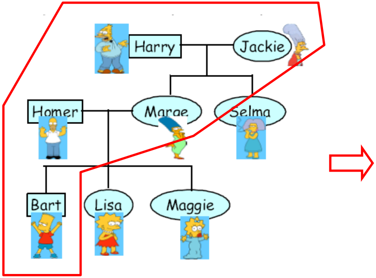
FROM 2nd from London # 5747.

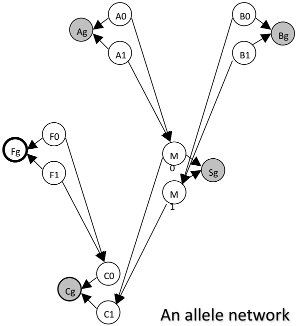
"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, [illegible]



The HM-BiTAM model (B. Zhao and E.P Xing, ACL 2006)

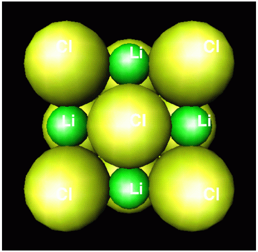
Fancier GMs: genetic pedigree

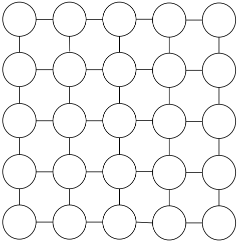




An allele network

Fancier GMs:
solid state physics





Ising/Potts model

Application of GMs

- Machine Learning
- Computational statistics

- Computer vision and graphics
- Natural language processing
- Informational retrieval
- Robotic control
- Decision making under uncertainty
- Error-control codes
- Computational biology
- Genetics and medical diagnosis/prognosis
- Finance and economics
- Etc.

Why graphical models

- A language for communication
- A language for computation
- A language for development

- Origins:
  - Wright 1920's
  - Independently developed by Spiegelhalter and Lauritzen in statistics and Pearl in computer science in the late 1980's

Why graphical models

- Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data.
- The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.
- Many of the classical multivariate probabilistic systems studied in fields such as statistics, systems engineering, information theory, pattern recognition and statistical mechanics are special cases of the general graphical model formalism
- The graphical model framework provides a way to view all of these systems as instances of a common underlying formalism.

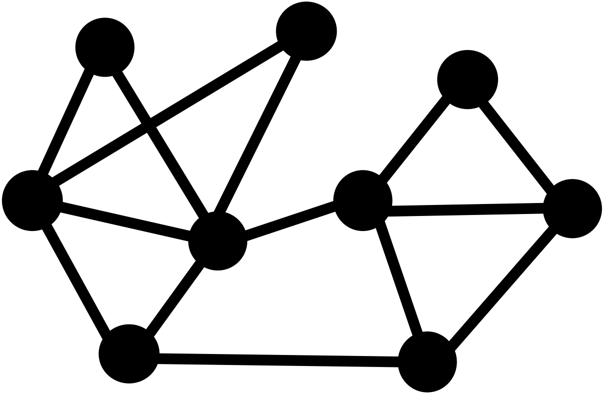
--- M. Jordan

Representation Learning on Networks

Jure Leskovec, William L. Hamilton, Rex Ying, Rok Sosic
Stanford University

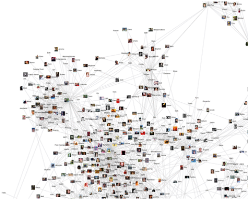
Why networks?

Networks are a general language for describing and modeling complex systems

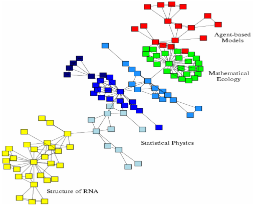


Network!

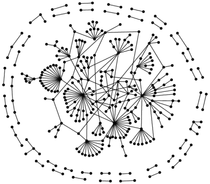
Many Data are Networks



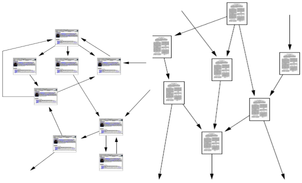
Social networks



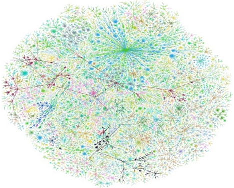
Economic networks



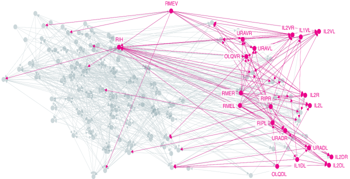
Biomedical networks



Information networks:
Web & citations



Internet



Networks of neurons

Why Networks? Why Now?

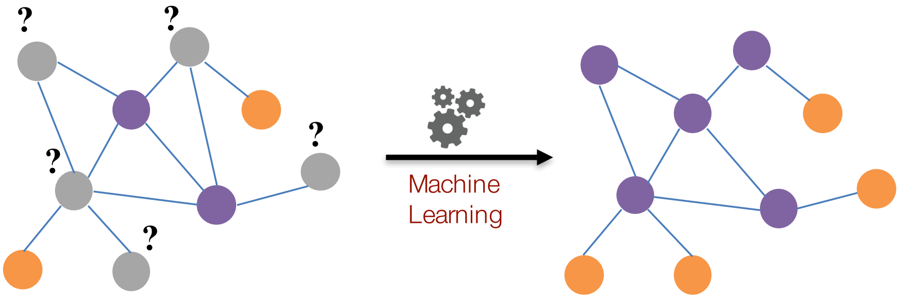
- Universal language for describing complex data
  - Networks from science, nature, and technology are more similar than one would expect
- Shared vocabulary between fields
  - Computer Science, Social science, Physics, Economics, Statistics, Biology
- Data availability (+computational challenges)
  - Web/mobile, bio, health, and medical
- Impact!
  - Social networking, Social media, Drug design

Machine Learning with Networks

Classical ML tasks in networks:

- Node classification
  - Predict a type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Network similarity
  - How similar are two (sub)networks

Example: Node Classification



Example: Node Classification

Classifying the function of proteins in the interactome!

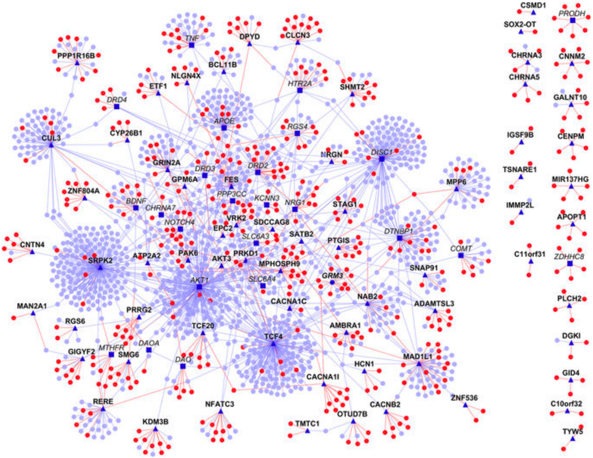
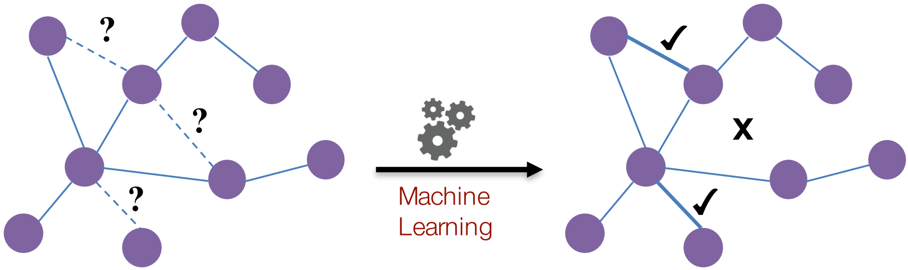


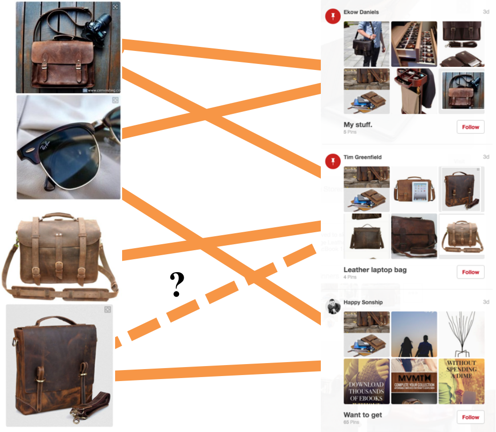
Image from: Ganapathiraju et al. 2016. [Schizophrenia interactome with 504 novel protein–protein interactions](). *Nature*.

Example: Link Prediction



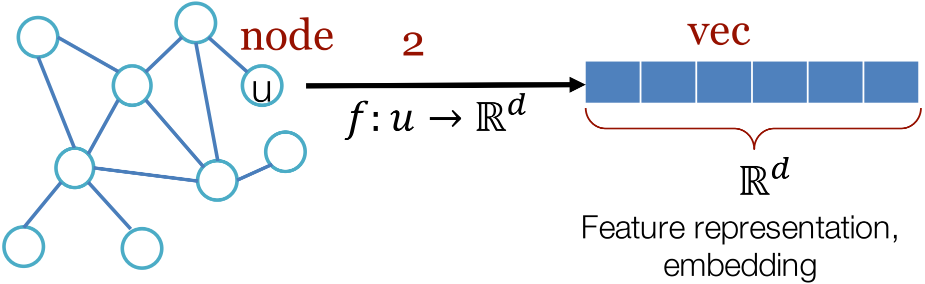
Example: Link Prediction

Content recommendation is link prediction!



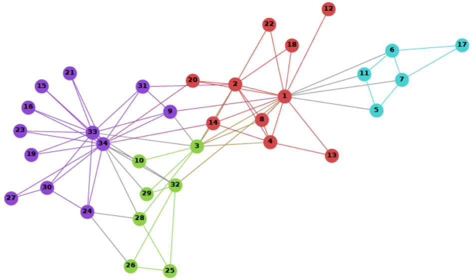
Feature Learning in Graphs

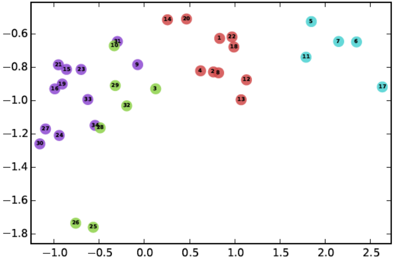
Goal: Efficient task-independent feature learning for machine learning in networks!



Example

- Zachary's Karate Club Network:



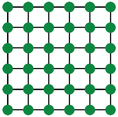


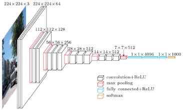
Input **Output**

Image from: [Perozzi et al. 2014](). DeepWalk: Online Learning of Social Representations. *KDD.*

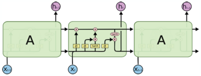
Why Is It Hard?

- Modern deep learning toolbox is designed for simple sequences or grids.
  - CNNs for fixed-size images/grids….



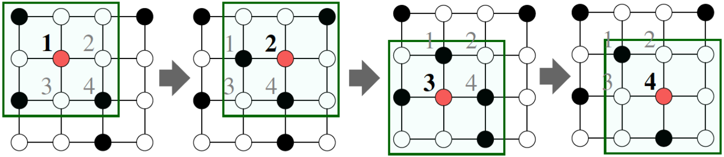


  - RNNs or word2vec for text/sequences…



Why Is It Hard?

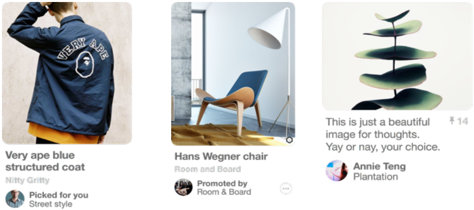
- But networks are far more complex!
  - Complex topographical structure (i.e., no spatial locality like grids)



  - No fixed node ordering or reference point (i.e., the isomorphism problem)
  - Often dynamic and have multimodal features.

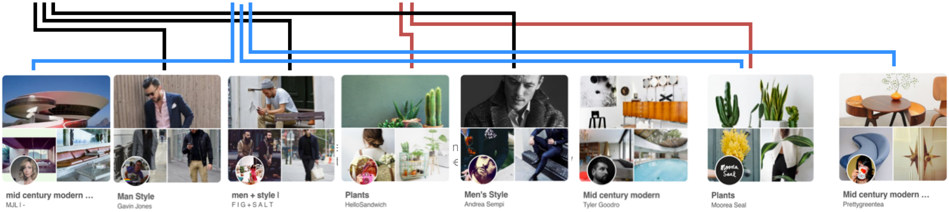
Application: Pinterest

Human curated collection of pins



Pins: Visual bookmarks someone has saved from the internet to a board they've created.

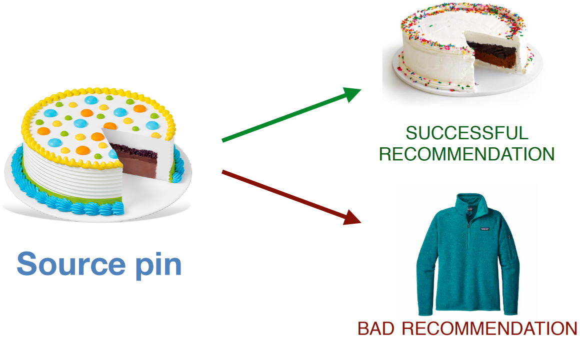
Pin features: Image, text, link



Boards

Application: Pinterest

Task: Recommend related pins to users.



- **Challenges:**
  - **Massive size:** 3 billion pins and boards, 16 billion interactions
  - **Heterogeneous data:** Rich image and text features

AAAI-19

Deep Reinforcement Learning with Applications in Transportation

Zhiwei (Tony) QIN

DiDi AI Labs

DiDi Labs

Jian TANG

DiDi AI Labs

Syracuse University

Jieping YE

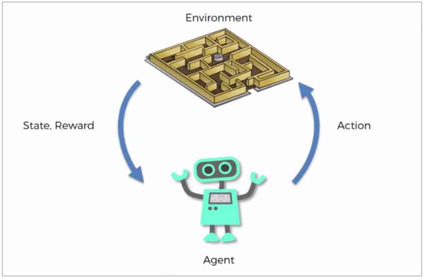
DiDi AI Labs

Univ. of Michigan, Ann Arbor

Reinforcement Learning

Problem

- Agent interacts with environment
  - Executes an action based on its state at each step
  - Receives a reward from environment
- Want to find an optimal policy π^* to achieve maximum cumulative rewards in the long run.



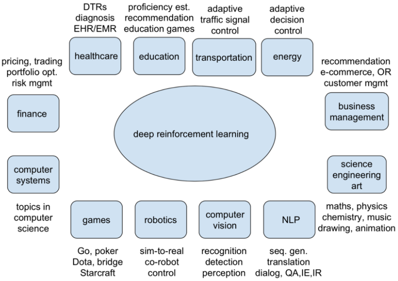
Different from the other paradigms

- No supervision on long-term reward, only immediate feedback
- Feedback is often delayed
- Sequential decisions
- Agent's action affects subsequent data received

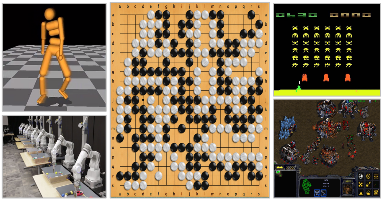
The Rise of RL

Success stories

- Chess, Board game: AlphaGo, AlphaZero
- Atari games: DQN
- Robotics



Yuxi Li, Deep Reinforcement Learning, arXiv, 2018



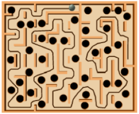
More applications

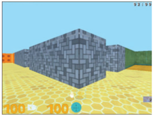
- Transportation
- Recommendation system
- Industrial control
- Education
- ...

Route Planning

- Planning a route for a trip on map
  - Distance, traffic
  - Road network known
  - Shortest travel time, avoid congestion

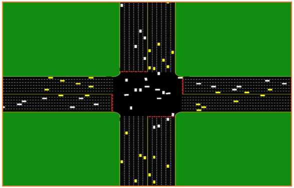
- Planning a route for robot navigation
  - With or without map
  - Perception as input

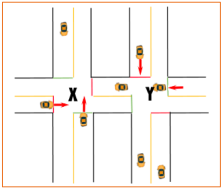




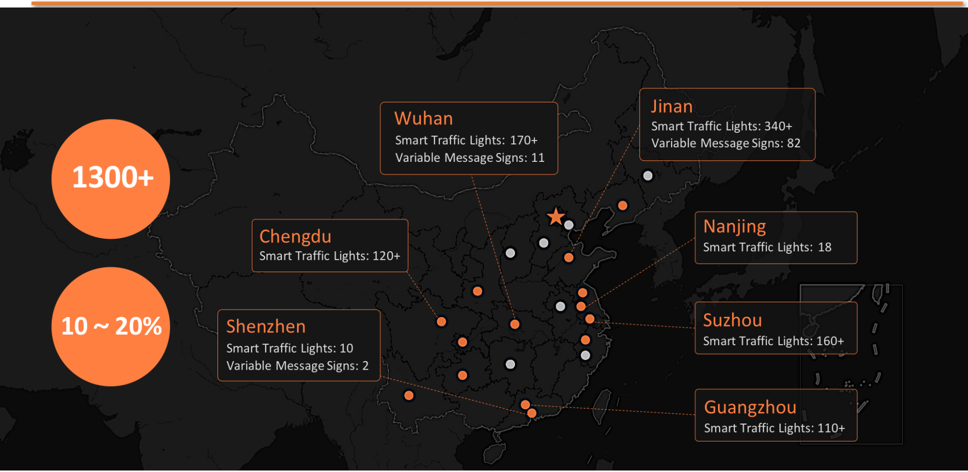
Traffic Signals Control

- Background
  - Traffic lights control traffic flow at intersections.
  - Affects throughput, delay, waiting time, etc
- Traditional methods
  - Fixed-time intervals for red-yellow-green
  - Traffic model-based methods
- Road network
  - Multiple intersections: control at one intersection has impact on neighboring intersections.



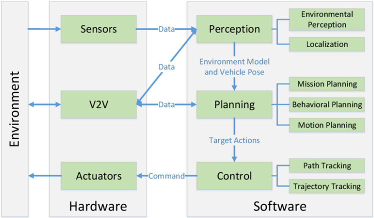


Improving Traffic Conditions in over 20 cities

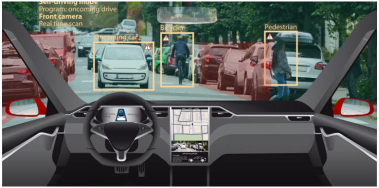


Autonomous Vehicle Control

- Framework
  - Perception: visual & sensory signals
  - Planning: behavior planning, motion planning
  - Control: path tracking
- Challenges
  - Complexity of the environment: color, shape of objects, type of objects, background, viewpoint, …
  - Smooth control is hard, e.g. smooth turning
  - Control has to adapt to fast changes in environment
  - Strict safety requirement



Pendleton et al., 2017



CS6375: Machine Learning

Gautam Kunapuli

Reinforcement Learning

THE UNIVERSITY OF TEXAS AT DALLAS
Erik Jonsson School of Engineering and Computer Science

Reinforcement Learning

Supervised learning: Given **labeled** data $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$, learn a function $f : \boldsymbol{x} \rightarrow y$

- Categorical y : **classification**
- Continuous y : **regression**

Rich feedback from the environment: the learner is told exactly what it should have done

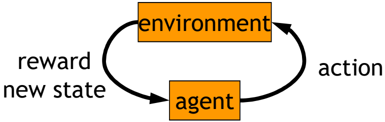
Unsupervised learning: Given **unlabeled** data $\boldsymbol{x}_i$, $i = 1, \ldots, n$, can we infer the underlying structure?

- Clustering
- dimensionality reduction,
- density estimation

No feedback from the environment: the learner receives no labels or any other information

Reinforcement Learning is learning from Interaction: learner (agent) receives feedback about the appropriateness of its actions while interacting with an environment, which provides numeric reward signals

Goal: Learn how to take actions in order to maximize reward



Example: Grid World

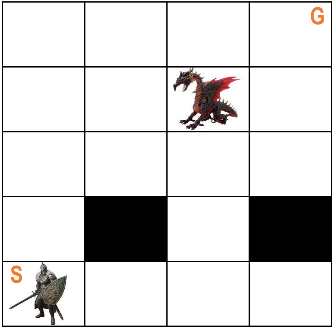
Example: Learn to navigate from beginning/start state (S) to goal state (G), while avoiding obstacles

Autonomous "**agent**" interacts with an **environment** through a series of **actions**

- trying to find the way through a maze
- actions include turning and moving through maze
- agent earns rewards from the environment under certain (perhaps unknown) conditions

The agent's goal is to maximize the reward

- we say that the agent learns if, over time, it improves its performance



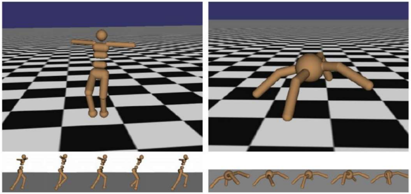
actions are what the agent actually wants to do

effects are what actually happens after the agent executes the chosen action

Actions	**Effects**
→ (right)	→ (60%), ↓ (40%)
↑ (up)	↑ (100%)
← (left)	← (100%)
↓ (down)	↓ (70%), ← (30%)

Image from Fang et al., https://arxiv.org/abs/1802.03248

Applications of Reinforcement Learning


Schulman et al (2016)

Robot Locomotion (and other control problems)

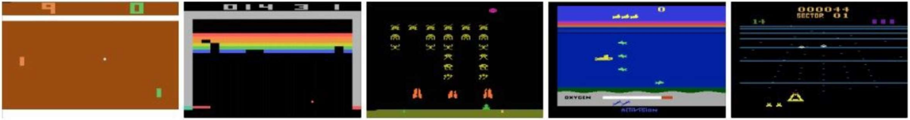
Objective: Make the robot move forward

State: Angle and position of the joints
Action: Torques applied on joints
Reward: 1 at each time step upright + forward movement

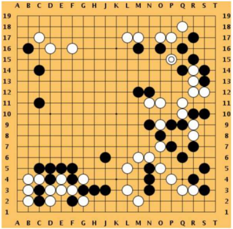
Atari Games

Objective: Complete the game with the highest score

State: Raw pixel inputs of the game state
Action: Game controls e.g. Left, Right, Up, Down
Reward: Score increase/decrease at each time step



Applications of Reinforcement Learning



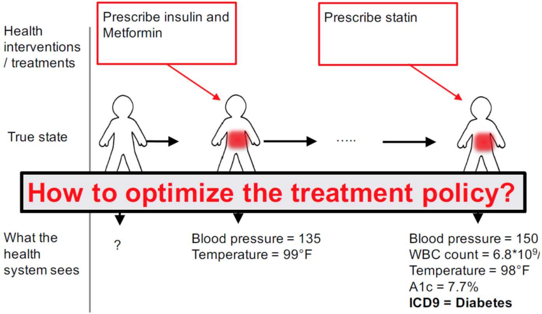
Go!

Objective: Win the game!

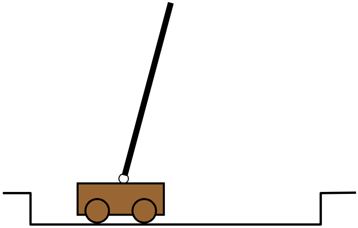
State: Position of all pieces
Action: Where to put the next piece down
Reward: 1 if win at the end of the game, 0 otherwise

Treatment Planning
Objective: Find the best treatment policy

State: Patient health data every 6 months
Action: Clinical interventions and treatment
Reward: negative rewards for deterioration positive rewards for improvement



Reinforcement Learning



Other examples
- pole-balancing
- TD-Gammon [Gerry Tesauro]
- helicopter [Andrew Ng]

General challenge: no teacher who would say "good" or "bad"
- is reward "10" good or bad?
- rewards could be delayed
- similar to control theory
  - more general, fewer constraints
- ***explore the environment and learn from experience***
  - not just blind search, try to be smart about it