AI for Digital Retail

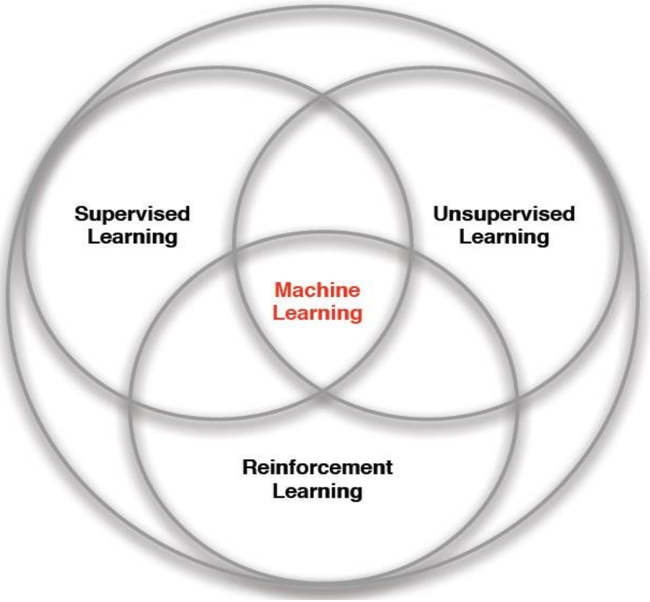
Ashwin Rao

Stanford University

A bit about me

- Co-Founder of CX Score: AI to enhance Digital Customer Experience
- Adjunct Professor, Applied Mathematics (ICME), Stanford University
- Past: VP of AI at Target Corporation (ML for Operations and Digital)
- Past: MD at Morgan Stanley, Trading Strategist at Goldman Sachs
- I direct Stanford's Mathematical & Computational Finance program
- Research & Teaching in: *RL and it's applications in Finance & Retail*
- Book: Foundations of RL with Applications in Finance
- CX Score: AI enabling retailers to deliver great Customer Experience

Machine Learning Branches



Machine Learning Overview

- ML is roughly classified into Supervised, Unsupervised and RL
- Supervised: Predicts by learning relationships between variables
- Unsupervised: Unearths patterns & structures within data
- RL: Optimal Sequential Decisioning under Uncertainty
- ML has been a breakthrough for images, text, games
- Largely due to the past decade of success of Deep Neural Networks
- We are now adapting Deep Learning techniques to other domains
- Works well when we have plenty of data and plenty of compute
- ML practice tends to be quite laborious on Big Data Engineering
- I've seen promising results in Finance and Retail applications

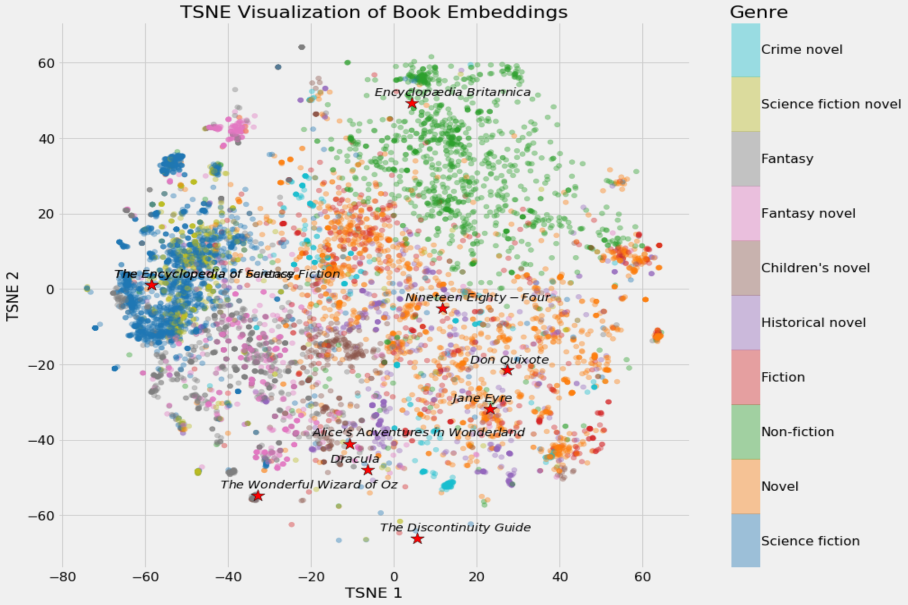
AI for Digital Customer Engagement & Profitability

- AI has been used extensively in Digital Retail
- Primarily for Search, Recommendations, Marketing/Ads
  - Search Relevance - Language Models, Product *Embeddings*
  - Personalized Recommendations/Ads - Customer *Embeddings*
  - Tactical Promotion of Products (eg: Profitability, Inventory, Clearance)
- Moving on to personalizing/optimizing entire page content/structure
- Natural progression to personalizing/optimizing web/mobile *journeys*
- Balancing multiple objectives of customer engagement & profitability
- Techniques: *Multi-Armed Bandits* and *Reinforcement Learning*

Similar/Substitutable Products identified with *Embeddings*

- *Embeddings* automate identification of similarity/substitutability
- Similarity by Visuals or by Descriptions or by Customer Interest
- Throw this heterogeneous data into deep neural network learning
- Deep inside the neural network, we find encodings of these products
- Similar/Substitutable products have similar encodings
- The technical term for these encodings is Embeddings
- Embeddings are low-dimensional numerical representations (*Vectors*)
- Capturing the most important features of products
- Captures various relationships between products, eg: complementarity
- Based on product images, descriptions, customer interest
- Embeddings can be used as ML features for transfer learning
- Embedding vectors have powerful algebraic/geometric properties

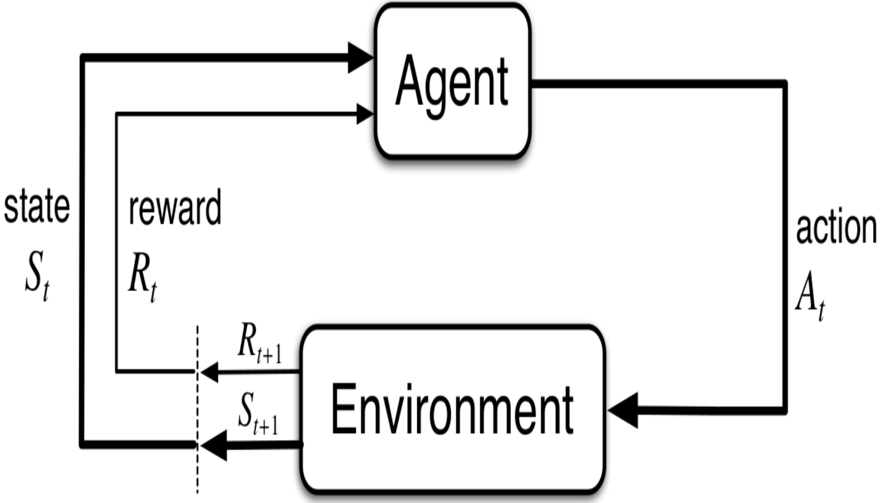
Book Embeddings flattened to 2-D Vectors



RL for Optimized Web/Mobile Journeys

- Today almost all retail companies use ML for Personalization
- Product Recommendations/Marketing based on customer interest
- So product displays are customized to generate clicks/engagement
- Uses Customer Embeddings and Contextual Bandits techniques
- Now this is being extended to personalizing the entire page
- The natural progression is to personalize web/mobile journeys
- Objective is to blend customer engagement & profitability
- Core Problem: Optimal Sequential Decisioning under Uncertainty
- This problem cries out for Reinforcement Learning (RL)
- RL based on the powerful framework of Markov Decision Processes

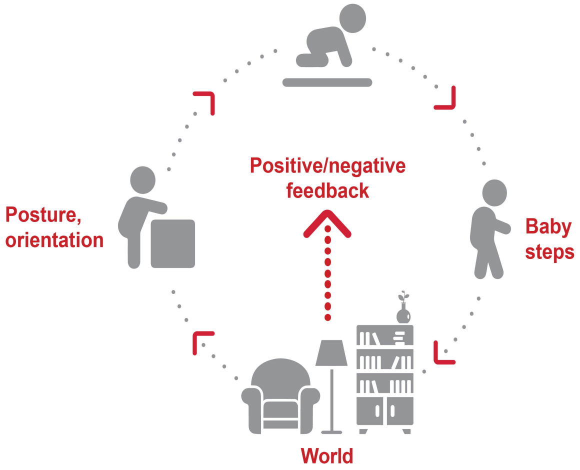
The Markov Decision Process Framework



Components of the MDP Framework

- The *Agent* and the *Environment* interact in a time-sequenced loop
- *Agent* responds to [*State*, *Reward*] by taking an *Action*
- *Environment* responds by producing next step's (random) *State*
- *Environment* also produces a (random) number we call *Reward*
- Goal of *Agent* is to maximize *Expected Sum* of all future *Reward*s
- By controlling the ($Policy : State \rightarrow Action$) function
- This is a dynamic (time-sequenced control) system under uncertainty
- MDP framework enables modeling problem of optimized app journeys

How a baby learns to walk



Many real-world problems fit this MDP framework

- Self-driving vehicle (speed/steering to optimize safety/time)
- Game of Chess (Boolean *Reward* at end of game)
- Inventory Replenishment to ensure high availability at low cost
- Make a humanoid robot walk/run on difficult terrains
- Manage an investment portfolio (covered in depth in my book)
- Control a power station
- Hyper-Personalized Apps for optimizing user engagement
- Optimal decisions during a football game
- Strategy to win an election (high-complexity MDP)

Self-Driving Vehicle

Web/Mobile Journey Optimization as an MDP

- MDP *State* is past and current data from the customer
- *State* also includes location, trends, pricing, inventory etc.
- *Action* is the content and structure of next page to display
- *Reward* function blends customer engagement and profitability
- State transitions governed by uncertainty in customer "clicks"
- Solve: Dynamic Programming or Reinforcement Learning
- Curse of Dimensionality and Curse of Modeling $\Rightarrow$ RL

How RL Works: Learning from Samples of Data

- RL incrementally learns from state/reward transitions data
- Typically served by a simulator acting as a *Simulated Environment*
- RL is a "trial-and-error" approach linking *Actions* to *Rewards*
- Try different actions & learn what works, what doesn't
- Deep Neural Networks are typically used for function approximation
- Big Picture: Sampling and Function Approximation come together
- RL algorithms are clever about balancing "explore" versus "exploit"
- Promise of modern A.I. is based on success of RL algorithms
- Potential for automated decision-making in many industries
- In 10-20 years: Bots that act or behave more optimal than humans
- RL already solves various low-complexity real-world problems
- RL has many applications in Retail, in Operations and Digital
- RL covered in detail in my book, with applications in Finance/Retail