Data Schools

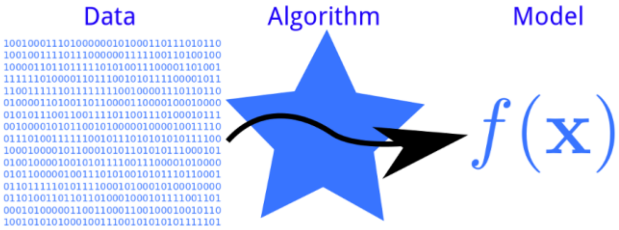
Machine Learning
1. Summary of the course

Ekpe Okorafor ekpe@okorafor.com

Topics & What We Learned

1. What Machine Learning is
2. When to Leverage Machine learning
3. Machine Learning algorithms & methodology
4. What a recommender system is
5. Types of recommender systems – collaborative filtering
6. Limitations of recommender systems
7. Fundamental concepts
8. Hands-On Exercise: Implementing a Basic Recommender

What Is Machine Learning

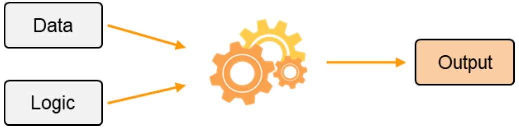


“Machine Learning is the science of getting computers to act without being explicitly programmed.” – Andrew Ng (Coursera)

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at task in T, as measured by P, improves with experience E.” – Tom M. Mitchell (1997)

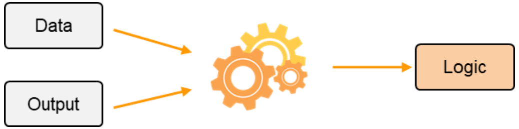
- Machine Learning is a particularly interesting technique because it represents a paradigm shift within AI

Traditional AI techniques



- **Static** – hard-coded set of steps and scenarios
- **Rule Based** – expert knowledge
- **No generalization** – handling special cases is difficult

Machine Learning



- **Dynamic** – evolves with data, finds new patterns
- **Data driven** – discovers knowledge
- **Generalization** – adapts to new situations and special cases

When Do We Leverage ML

- Machine learning is particularly good at solving **2 types of problems** where other AI techniques fail

Tasks programmers can’t describe

Complex multidimensional problems that can’t be solved by numerical reasoning

Handwriting

Cognitive Reasoning

Weather Forecasting

Health Care Outcomes

Network Intrusion

Movie Recommendation

Machine Learning Algorithms / Methods

Supervised Learning: Predicting values. **Known** targets.
User inputs correct answers to learn from. Machine uses the information to guess new answers.

Regression

- Linear Regression
- Ordinary Least Squares Regression
- LOESS (Local Regression)
- Neural Networks

Classification

- Decision Trees
- K-Nearest Neighbors
- Support Vector Machine
- Logistic Regression
- Naïve Bayes
- Random Forests

Unsupervised Learning: Search for structure in data. Unknown targets.
User inputs data with undefined answers. Machine finds useful information hidden in data

Cluster Analysis

- K-Means Clustering
- Hierarchical Clustering

Dimension Reduction

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

Machine Learning – Recommender Sys

Recommender systems deal with making recommendations based upon previously collected data and leveraging ML techniques.

Formal Model

X = set of **Customers**

S = set of **Items**

Utility function u: $X \times S \rightarrow R$

R = set of ratings

R is a totally ordered set

e.g., **0-5** stars, real number in **[0,1]**

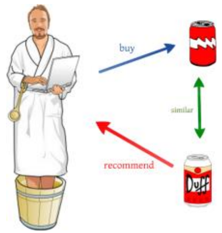
Three approaches to recommender systems:

1) Content-based

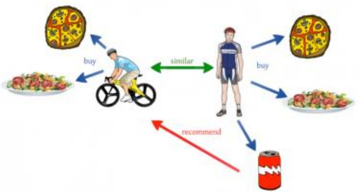
2) Collaborative

3) Latent factor based

Content Based (Features)
Modified Linear Regression



Non-content Based (No Features)
Collaborative Filtering
Matrix Factorization



Collaborative Filtering

Collaborative filtering can be subdivided into two main types

- **User-based: "What do users similar to you like?"**
  - For a given user, find other people who have similar tastes
  - Then, recommend items based on past behavior of those users

- **Item-based: "What is similar to other items you like?"**
  - Given items that a user likes, determine which items are similar
  - Make recommendations to the user based on those items

User-based

- **Consider user x**
- Find set **N** of other users whose ratings are ***"similar"*** to ***x***'s ratings
- Estimate ***x***'s ratings based on ratings of users in **N**

Item-based

- For item ***i***, find other similar items
- Estimate rating for item ***i*** based on ratings for similar items
- Can use same similarity metrics and prediction functions as in user-user model

We dug a little deeper – pros & cons

+ Works for any kind of item. No feature selection needed

- Cold Start: Need enough users in the system to find a match

- Sparsity: The user/ratings matrix is sparse. Hard to find users that have rated the same items

- First rater: Cannot recommend an item that has not been previously rated. New items, Esoteric items

- Popularity bias: Cannot recommend items to someone with unique taste. Tends to recommend popular items

Ok, some practical implementations

You built a movie recommendation engine. You used both the Collaborative Filtering approach and Content-based Filtering approach and to build a basic movie recommendation engine.

1. We used the ***MovieLens*** dataset, explored movie / ratings files and examined the data structures
2. I know you remember this - Now, wow us! – Do your thing, create an amazing visualization plots (2-D, 3-D, etc)
3. We used User-Based Collaborative Filtering to generate a top-10 recommendation list for users using the ***recommenderlab*** package available in R.
4. Next, we built a basic content-based recommender engine based on movie genres only.
5. Finally, we created the recommendation by forming groups and presenting your results
   - We chose the content-based approach
   - You used Jaccard Distance to measure the similarity between user profiles, and the movie genre matrix. Consider only the FIRST user in the dataset.
   - Finally, you stated your recommendation

Summary and next topics

1. You were introduced to the basic concept of Machine Learning
2. You also implemented a recommender engine using collaborative filtering approach in R
3. What’s next? What’s New? What has changed?
4. Tips on how you can teach this course

Data Schools