Recommendation Systems I

COSC 410: Applied Machine Learning
Fall 2025
Prof. Forrest Davis

September 18, 2025

Warm-up

1. Discuss with your neighbor the best bagel flavor.
2. You have a dataset with 10 samples of 2 features. You are setting up linear regression to train for 5 epochs with mini-batch gradient descent with a batch size of 2. How many times will you update your parameters?

Logistics

- Check in on Codelet 2

Learning Objectives

- Connect linear regression to probability
- Describe the basic aims of recommendation systems
- Build recommendation systems using three concrete approaches
- Reason about the scenarios different systems work well in

Summary: We wrap up linear regression with a deeper dive into its foundations. Then, we turn to recommendation systems and chart out three broad approaches and their strengths/weaknesses.

Linear Regression: The Model Generates Data

THERE IS A DEEPER CONNECTION BETWEEN minimizing MSE and modeling a linear relationship.

- Let's think about the source of our data
- In science, we propose that the universe is governed by universal laws that describe the true, unobservable nature of processes we observe

- That is, for some problems, there is a true linear relationship between some variable and some ouptut
  - A running example, Ohm's Law: The current flowing through a conductor is directly proportional to the voltage applied.
- We sample data after some manipulation
  - We want to measure the conductivity of a material, so we apply a voltage and measure the resultant current
  - Current = Conductivity of material * Voltage
- There is error in our measurement that is drawn from some Gaussian distribution
  - Our instruments are good, but they aren't perfectly able to record something (perhaps there is lack of precision in the number, e.g., a kitchen scale doesn't tell you a weight to the 10th decimal point, perhaps an issue of a non-essential interacting influence, think air resistance for gravity – in our case this could be some slight impurity in the material being evaluated)
  - Current = Conductivity of material * Voltage + error
- We assume this error in measurement is Gaussian, so the error is a normal distribution with a mean around the true measurement
  - $\mathcal{N}(\text{current}, \sigma^2)$
- We can now articulate the *likelihood* of observing labels given our model

$$P(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{Xw}, \sigma^2)$$

- Expanding this expression yields

$$P(\mathbf{y}|\mathbf{X}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{y}-\mathbf{Xw})^2}{2\sigma^2}}$$

- What makes a model a good model of reality? It assigns a high probabaility to our observations. This principle is called **maximum likelihood estimation**. In machine learning, this asserts that a good model is one whose parameters maximize the **likelihood** of the entire dataset:

$$P(\mathbf{y}|\mathbf{X}) = \prod_{i=1}^{m} P(y^{(i)}|\mathbf{x}^{(i)})$$

- There's an issue with taking the joint probability of all of our data (i.e., the product of the likelihoods). Multiplying numbers between 0 and 1 tends to smaller and smaller numbers. This can cause precision errors on our computers.
- We we can reformulate this using logs, where addition applies (e.g., log(A*B) = log(A)+log(B))
- Additionally, we frame this as a minimization problem which is the typical framing of an optimization problem

Motivation and Basic Problem Articulation

RECOMMENDATION SYSTEMS ARE A WIDELY USED machine learning algorithm. At its core the aim is to take a set of items (e.g., movies, products, classes) and a set of individuals who interact with at least some of these items and produce recommendations of items particular users have not seen. This could applications like "what to watch next" on some streaming platform, what books to read if you liked some series, or what gadget to buy next.

We will cover some basic approaches here. Table 1 provides an example, which shows how we commonly represent the problem in a matrix form with users as rows and items as columns with the cells representing ratings.

	Dune	**LOTR**	**High School Musical**	**Forgetting Sarah Marshall**
Alice	1		3	
Keisha	5	4		
Carol		3		5
Jessica			4	5

Table 1: Matrix representation of movie ratings

In real settings, to build a recommendation system we must (1) gather user ratings,[1] (2) build a model to predict unknown ratings, and (3) evaluate model predictions.[2]

Content-Based Recommendations

> **Scenario I**:
> A new movie is released on a streaming service. We would like to determine if Keisha, an active user of this service, would like this movie. We notice that the new movie is very similar to other movies Keisha has seen (it has the same director, its her favorite genre, etc.), so we predict she will probably like it too.

To do this, we need to:

1. Characterize items
2. Characterize the user
3. Rate new new items

There are many different ways to answer these questions. We will chart one, so you can have a concrete example.

[1] We could gather ratings explicitly by asking users to rate items they've interacted with on some scale, or implicitly via some proxy signal like how long they watched a movie for, whether they purchased the item, or whether they shared it on social media.

[2] We are usually interested surfacing predictions that would be positively rated by a user.

Characterizing Items

Using Table 1 as our running example, we need to determine a representation for movies. One approach could be facts about the movie:

- Genre
- Lead Actor
- Year
- Director

For example, Dune would be ['Sci-Fi', 'Timothy Chalamet', 2021, 'Denis Villeneuve'].[3] Each item gets a vector with consistent features.

Characterizing Users

Using Table 1 as our running example, we need to determine a representation for users (to capture their preferences). One approach is representing the user as a item vector (in this case movie vector) that is composed of the weighted average of the items (in this case movies) they have watched.

Concretely for Keisha, we would take the vector for Dune and the vector for Lord of the Rings (LOTR) because these were watched by her. We would then combine them, weighted by her rating of them:

$$\frac{5 * \vec{\text{Dune}} + 4 * \vec{\text{LOTR}}}{2}$$

Predicting Ratings of New Items

Given our representations of items and our representation of users, we can plot the vectors in space, as in Figure 1.

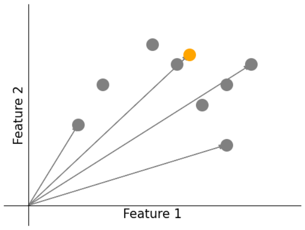

Figure 1: Plotting new items (in grey) and user (in orange) vectors where the vectors have two features.

We can sketch out how we would predict a rating for a new item. If the new item vector is very similar to the user vector, then we expect the user would like it more.[4].

---

[3] In fact, we would have to **factorize** some of these features so that we can do math. In particular, the strings for Genre, Lead Actor, and Director, would have to be mapped to numbers (e.g., maybe 'Rom-Com' maps to 1 and 'Sci-Fi' to 2).

[4] There are quite a few ways to quantify similarity, which we will turn to shortly. An immediately applicable example could be the dot product

Question

1. What do we predict about the ratings for Keisha for the movies High School Musical and Forgetting Sarah Marshall?

2. Consider the strengths and limitations of content-based approaches to recommender systems. Concretely, address the following question:
   - (a) Do we need information from other users?
   - (b) Are broadly unpopular items as likely to be recommend as popular items?
   - (c) How does the user's behavior contribute to the recommend items?
   - (d) In what sample situations would this approach be useful? How about not useful?

User-based Collaborative Filtering

> **Scenario II**:
> We would like to determine what movies to recommend to Keisha. We notice that she watches very similar movies to Amy, though there are some movies Amy has seen that Keisha hasn't. We predict that Keisha will feel similarly to Amy about movies that she hasn't seen but Amy has.

We need to establish at least three things to operationalize this scenario:

- Characterize a user
- Define the similarity between two users
- Predict ratings

We will use Table 2 as our running example.

	Fast and Furious	**Faster and Furiouser**	**Fastest and Furiousest**	**Twilight**	**Saw I**	**Saw II**	**Saw III**
Alice	4			5	1		
Keisha	5	5	4				
Carol				2	4	5	
Jessica	3						3

Table 2: Movie ratings by individuals

Characterizing a User

We will represent a user $\mathbf{x}$ as a vector $\mathbf{r}^{(x)}$ of their ratings. For example, Alice is [4, 0, 0, 5, 1, 0, 0] and Keisha is [5, 5, 4, 0, 0, 0, 0].[5]

Question

What is the vector for Jessica based on Table 2?

Defining Similarity

Let $\mathbf{r}^{(x)}$ be the vector of user $\mathbf{x}$'s ratings, $\mathbf{r}^{(y)}$ be the vector of user $\mathbf{y}$'s ratings, and $\mathbf{r}^{(x)} = [1,0,0,1,3]$ and $\mathbf{r}^{(y)} = [1,0,2,2,0]$. We will consider two ways to calculate similarity:

- **Jaccard Similarity**: Treat $\mathbf{r}^{(x)}$ and $\mathbf{r}^{(y)}$ as sets of items rated ($\mathbf{r}^{(x)} = \{0,3,4\}$ and $\mathbf{r}^{(y)} = \{0,2,3\}$, then we calculate the similarity with

$$\frac{\text{Intersection}(\mathbf{r}^{(x)}, \mathbf{r}^{(y)})}{\text{Union}(\mathbf{r}^{(x)}, \mathbf{r}^{(y)})}.$$

  (in this case $\frac{2}{4}$). This is also called **Intersection over Union**.

- **Cosine Similarity**: Treating $\mathbf{r}^{(x)}$ and $\mathbf{r}^{(y)}$ as vectors take the cosine similarity between them:[6]

$$\frac{\mathbf{r}^{(x)} \cdot \mathbf{r}^{(y)}}{||\mathbf{r}^{(x)}||\,||\mathbf{r}^{(y)}||}$$

  (in this case -0.41). Note, if we made the 1s 5 and the 2s 4, the cosine similarity is 0.46.

---

[5] Notice, the vectors are of fixed length, each position represents a consistent movie (e.g., $\vec{\text{Alice}}_4$ is a 5 because she rated Twilight 5, a shocking claim), and 0 represents an unrated movie. That is, $\mathbf{r}^{(x)} \in \mathbb{R}^n$ where n is the number of total movies in our dataset.

[6] Recall the magnitude of a vector $\mathbf{v}$ of length n is $||\mathbf{v}|| = \sqrt{v_0^2 + v_1^2 + \cdots + v_n^2}$.

Practice Problems

1. Calculate the Jaccard similarity between Alice and Keisha and Alice and Carol in Table **??**. Is Alice more similar to Keisha or Carol?
2. Calculate the Cosine similarities between Alice and Keisha and Alice and Carol
3. Which measure matches your intuition for this pair?

Predicting Ratings

To predict ratings for unseen films, we use a set of most similar users and aggregate over similar users ratings. There are two main approaches to this:[7]

$$r_i^{(x)} = \frac{1}{|N|} \sum_{y \in N} r_i^{(y)}$$

$$r_i^{(x)} = \frac{\sum_{y \in N} \text{sim}(\mathbf{r}^{(x)}, \mathbf{r}^{(y)}) r_i^{(y)}}{\sum_{y \in N} \text{sim}(\mathbf{r}^{(x)}, \mathbf{r}^{(y)})}$$

Question

Give the predicted rating for Fastest and Furiousest for Alice using both aggregation metrics. Consider the top two most similar users. For reference, Alice has a similarity of 0 with Jessica, a similarity of 0.4 for Keisha, and a similarity of 0.28 for Carol.

---

[7] For both equations, $\mathbf{r}$ is a rating vector, $r_i^{(x)}$ is the rating of item i by user x, N is a set of similar users, sim is a similarity function, and $|N|$ is the cardinality of this set (i.e., how many similar users are being aggregated over).

Item-based Collaborative Filtering

> **Scenario III**:
> We would like to determine what movies to recommend to Keisha. We notice that she like a certain set of movies. People who like those movies, like a few movies that Keisha hasn't seen yet. We predict she will like these movies too.

We need to establish at least three things to operationalize this scenario:

- Characterize an item
- Define the similarity between two items
- Predict ratings

We will use Table 3 as our running example.

users

movies

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
A	1		3			5			5		4	
B			5	4			4			2	1	3
C	2	4		1	2		3		4	3	5	
D		2	4		5			4			2	
E			4	3	4	2					2	5
F	1		3		3			2			4	

Table 3: Sample data for users rating movies

Characterizing an Item

We will represent an item, i as a vector, $\mathbf{s}^{(i)}$, of the ratings users have given to it. For example, $\mathbf{s}^{(A)}$ is [1, 0, 3, 0, 0, 5, 0, 0, 5, 0, 4, 0] and $\mathbf{s}^{(F)}$ is [1, 0, 3, 0, 3, 0, 0, 2, 0, 0, 4, 0].

Characterizing Similarity

This proceeds in the same way as characterizing the similarity between users in the preceding section. We might use the cosine similarity to compare two item vectors.

Predicting Ratings

To make a prediction for an item for some user, we gather the most similar items to our target item. Then, we aggregate the target users ratings for those similar items in order to predict their rating for the target item. We can use the same metrics we discussed with user-based collaborative filtering.

Suppose we want to predict the rating for movie A for user 5 in Table 3. The similarities between movie A and all other movies are:

$$\begin{bmatrix} 1 \\ -0.18 \\ 0.41 \\ -0.10 \\ -0.31 \\ 0.59 \end{bmatrix}$$

If we were using the two most similar movies, we would be looking at movie C and movie F. User 5 gave those a rating of 2 and 3 respectively. The predicted rating (using those ratings weighted by the similarity of the items) is 2.59.

Practice

Practice Problems

1. Using item-based collaborative filtering (and the top two most similar items), calculate the rating for movie A for user 3 based on Table 4.

Before Next Class

- Read and pre-class quiz
- Complete Codelet 2

users

movies

	1	**2**	**3**
A	5	3	0
B	4	0	2
C	0	2	5

Table 4: Sample data for users rating movies