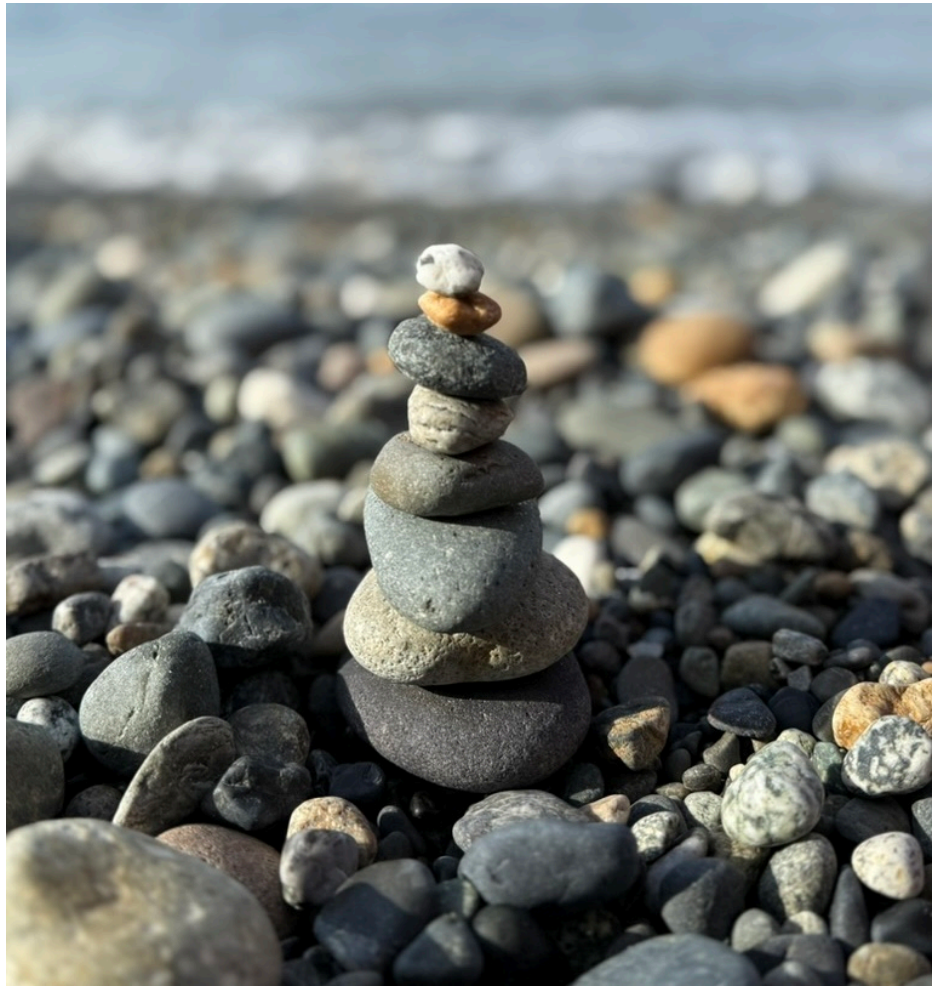
CPSC 330 Lecture 16: Recommendation systems

Varada Kolhatkar

Focus on the breath!



Announcements

- HW6 was due yesterday.
- HW7 has been released (a new homework on transfer learning)
- No classes or OH during the midterm break.
- Midterm 2 coming up next week!
- If you find the breathing exercises helpful, you're very welcome to join our weekly meditation sessions.
  - 🕑 When? **Every Tuesday at 1 PM**
  - 📍 Where? **ICCS 146**

iClicker question

What percentage of watch time on YouTube do you think comes from recommendations?

- a. 50%
- b. 70%
- c. 20%
- d. 90%

Today's big questions

1. How do we represent **ratings** when most are missing?
2. How can we **evaluate predictions** for unseen ratings?
3. How do **similar users** and **item features** help us recommend items?
4. Does lower prediction error mean **better recommendations**?

What is a recommendation system?

A recommendation system suggests products, services, or content that a user is likely to consume or enjoy.

- Items could be movies, songs, videos, products, or news articles.
- The system uses past data to predict what a user may want next.
- It helps users choose from a large catalogue.

Why recommendation systems?

- They help users navigate **information overload**.
- Searching a large catalogue takes effort.
- Good recommendations help users find relevant choices.



The other side: filter bubbles

- Recommenders can reinforce existing interests by repeatedly showing similar content.
- This can contribute to filter bubbles, limiting exposure to different information or viewpoints.
- In news, politics, and science, narrowed exposure can affect what people learn and believe.

Recommendation quality includes the information people have an opportunity to encounter.

Explicit vs. implicit feedback

Explicit feedback

- The user directly reports a preference.
- Example: a 1-to-5 movie rating.

Implicit feedback

- We infer preferences from behaviour.
- Examples: clicks, watch time, purchases.

Behaviour needs interpretation: a purchase could be a gift.

The utility matrix

- **Rows:** users. **Columns:** items.
- Entry y_{ij} is user i's rating of item j.
- Blank entries represent **unknown ratings**.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	?	?	2	?	3
User 2	3	?	?	?	?
User 3	?	5	4	?	5
User 4	?	?	?	?	?
User 5	?	?	?	5	?
User 6	?	5	4	3	?

7-movie illustration: ratings data

```
toy_ratings = pd.read_csv(DATA_DIR / "toy-movie-ratings.csv")
users = np.unique(toy_ratings["user_id"])
movies = np.unique(toy_ratings["movie_id"])
N_toy, M_toy = len(users), len(movies)
```

user_id	movie_id	rating
Sam	Lion King	5
Sam	Toy Story	4
Sam	The Little Mermaid	5
Sam	Bambi	5
Sam	The Social Dilemma	1
Eva	Toy Story	1

First six rows shown. The full table has 14 ratings, 4 users, and 7 movies.

7-movie illustration: utility matrix

							
Eva	?	?	?	5.0	?	5.0	1.0
Jim	?	?	4.0	5.0	?	5.0	?
Pat	5.0	5.0	?	?	4.0	?	?
Sam	5.0	5.0	?	?	5.0	1.0	4.0

Four users × seven movies. A question mark means no observed rating.

Missing ratings and sparsity

Most users rate only a small fraction of the catalogue.

- **Low rating:** observed evidence of a low reported preference.
- **Missing rating:** the user may never have encountered the movie.
- Filling missing entries with zero changes their meaning.

Our task: estimate ratings we have not observed.

Recommendation approaches

Approach	Evidence used	This lecture's example
Collaborative filtering	Patterns in user–item interactions	Average ratings from similar users
Content-based filtering	Known item features and a user's ratings	Per-user regression on genres
Hybrid	Both sources	Combine candidate sources or scores

Our focus is **content-based rating prediction** using regression we already know.

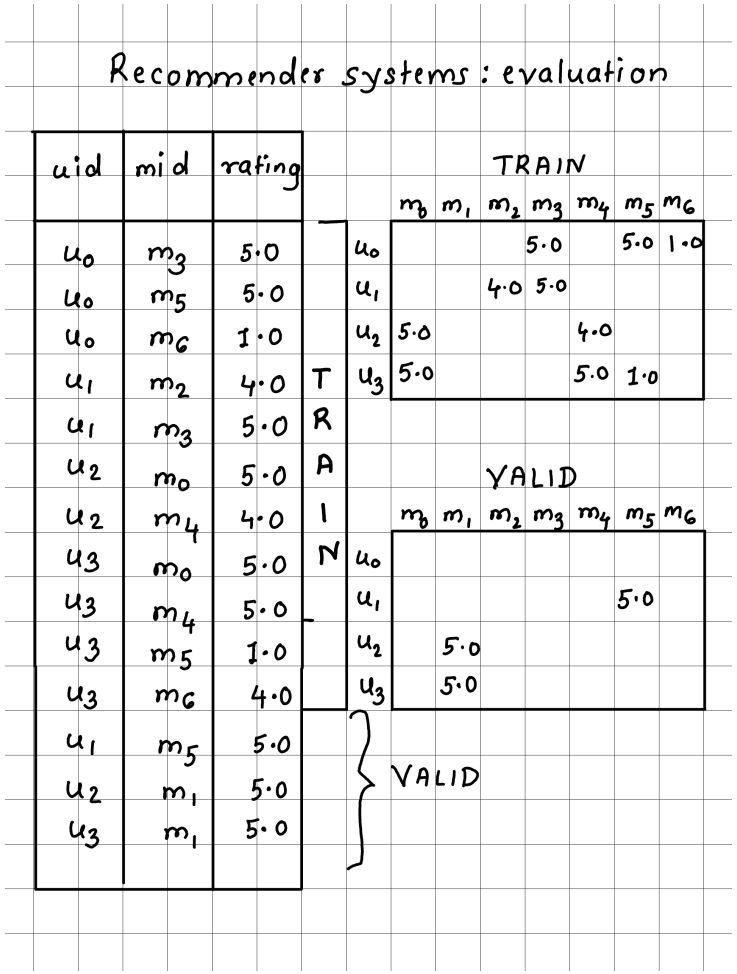
Hold out observed ratings

We cannot check a rating that the user has never provided.

1. Split the **observed ratings table** into training and validation rows.
2. Fit using **training ratings only**.
3. Compare predictions with **held-out ratings**.

The held-out ratings give us targets for evaluation.

Train and validation sets



- The train and validation utility matrices typically have the same shape: same users, same items.
- The difference is which entries are observed.
- Validation entries are masked during training and revealed only for evaluation.

RMSE on held-out ratings

List the n **held-out ratings** as $y_1, \dots, y_n$, with predictions $\hat{y}_1, \dots, \hat{y}_n$.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}.$$

- Lower RMSE means closer predictions, in **rating units**.
- Unknown ratings contribute **no terms**.
- Every evaluated entry needs a finite prediction.

Global-average baseline

Predict the mean of the **observed training ratings** for every pair:

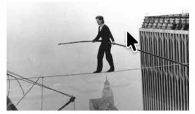
$$\hat{y}_{ij} = \bar{y}_{\text{train}}.$$

```
avg = np.nanmean(train_mat)
pred_g = np.full(train_mat.shape, avg)
```

A simple reference: does personalization improve held-out predictions?

Per-user and per-item baselines

7-movie illustration: arithmetic before splitting, using all 14 ratings.

								User averages
Eva	?	?	?	5.0	?	5.0	1.0	**11/3**
Jim	?	?	4.0	5.0	?	5.0	?	**14/3**
Pat	5.0	5.0	?	?	4.0	?	?	**14/3**
Sam	5.0	5.0	?	?	5.0	1.0	4.0	**20/5**
Item averages	**10/2**	**10/2**	**4.0**	**10/2**	**9/2**	**11/3**	**5/2**	

- **Per-user:** predict the user's mean observed rating.
- **Per-item:** predict the item's mean observed rating.
- **Combined:** average the user and item means.

KNN-style imputation: intuition

Jim has not rated **Toy Story**. How could we predict his rating?

							
Eva	?	?	?	5.0	?	5.0	1.0
Jim	?	?	4.0	5.0	?	5.0	?
Pat	5.0	5.0	?	?	4.0	?	?
Sam	5.0	5.0	?	?	5.0	1.0	4.0

Which users have overlapping ratings with Jim? Who has rated Toy Story?

KNN imputation and missing evidence

```
imputer = KNNImputer(n_neighbors=2, keep_empty_features=True)
pred_knn = imputer.fit_transform(train_mat)
```

- Compare users using **jointly observed ratings**, then average neighbours' ratings for the target movie.
- No movie ratings or no user history? Use a **fallback** (implemented in the demo).
- The imputer copies known ratings: **zero training RMSE is automatic**.

Class demo 1: ratings, split, and baselines

We now switch to the **12-movie demo**, which includes genre features.

The user names recur, but **these are different ratings** from the 7-movie illustration.

Open the recommender demo notebook

1. Inspect the new ratings table and utility matrix.
2. Check which entries move into validation.
3. Compare the global mean and KNN predictions.

Before running: What happens if a movie has no training ratings?

iClicker 16.1: evaluation and baselines

Select all statements that are true in our setup.

- a. Training and validation utility matrices have the same shape.
- b. Replacing missing ratings with zero preserves their meaning.
- c. Validation ratings must be excluded when calculating the global baseline.
- d. Zero training RMSE proves that KNN predicts unseen ratings perfectly.
- e. A global-mean fallback provides a prediction for a movie with no training ratings.

iClicker 16.2: a movie with no ratings

A **new movie** arrives. Pat has rated other movies, but **nobody has rated this one**: its utility-matrix column is all missing.

We use similar-user KNN to predict Pat's rating.

Which statement is correct? Select one.

- a. We can average the nearest users' ratings for the new movie.
- b. No users have rated it, so KNN needs a fallback rather than a neighbour-based prediction.
- c. The missing ratings show that everyone dislikes the movie.
- d. Increasing the number of neighbours guarantees a personalized prediction.

Content-based filtering

Content-based filtering intro

One way to address cold start is content-based filtering. The key idea is to recommend items similar to what the user liked before based on item features. Examples of item features:

- movie genres
- plot text
- actors
- release year
- budget

Content-based filtering example

							
Eva	?	?	?	5.0	?	5.0	1.0
Jim	?	?	4.0	5.0	?	5.0	?
Pat	5.0	5.0	?	?	4.0	?	?
Sam	5.0	5.0	?	?	5.0	1.0	4.0

Eva's profile

	Animation	Drama	Documentary	Rating
	0	0	1	5.0
	0	0	1	5.0
	1	1	0	1.0

Pat's profile

	Animation	Drama	Documentary	Rating
	1	1	0	5.0
	1	1	0	5.0
	0	0	1	1.0
...	...	...	...	...

Fit a separate Ridge model for each user

$$\hat{y}_{ij} = b_i + x_j^\top w_i.$$

```
pat_model = Ridge(alpha=1.0)
pat_model.fit(X_pat, y_pat)
```

- x_j: known genre features of movie j.
- w_i and b_i: learned preferences and intercept for user i.
- Ridge regularization helps when each user has **few ratings**.

Predictions to recommendations

For Pat:

1. Score movies from their features.
2. Exclude movies already rated **in training**.
3. Sort the remaining candidates by predicted rating.

Class demo 2: content-based regression

Continue the recommender demo notebook

1. Inspect Pat's X, y, and fitted coefficients.
2. Compare Pat's and Eva's predictions for the same movies.
3. Compare validation RMSE with the baselines and rank Pat's candidates.

Before running: Must content-based Ridge beat the global mean?

Comparing the models: 12-movie demo

Method	Training RMSE	Validation RMSE
Global mean	1.41	0.83
KNN imputation	0.00	0.66
Content-based Ridge	0.52	0.58

Same five held-out ratings. A small difference is weak evidence.

New items and new users

Situation	Similar-user KNN	Our content-based model
New movie, known features, no ratings	No movie ratings to average	Score using an existing user's fitted model
New user, no ratings	No personal history for similarity	No ratings to fit that user's model

For a new user: request **onboarding ratings** or use a **non-personalized fallback**.

Feature limit: for a given user, movies with identical genre vectors receive identical predictions.

iClicker 16.3: content-based filtering

Select all statements that are true for our separate user models.

- a. Pat’s model requires rating-history similarity to Eva.
- b. Each training example contains movie features and that user’s rating.
- c. A new movie can be scored if its features and a fitted user model are available.
- d. Genre features alone tell us a completely new user’s preferences.
- e. A user with no validation ratings prevents evaluating all other users.

Your turn: lower error, better top choice?

Pat's held-out ratings and two illustrative models:

Movie	Pat's rating	Model A	Model B
A	5	4.0	5.0
B	4	4.1	3.0
C	1	1.0	2.0

1. Which model has lower RMSE?
2. Which recommends Pat's highest-rated movie if we show **one** suggestion?
3. Why do these answers differ?

Beyond prediction error

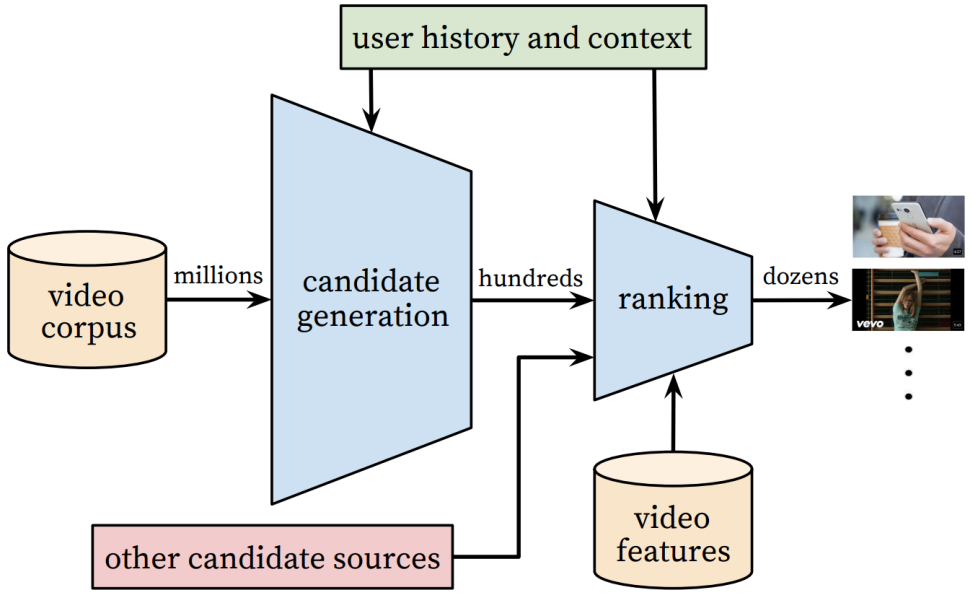
Lower RMSE does not guarantee better recommendations.

Users see a **ranked list**, not a table of predicted ratings.

- **Relevance:** does the list match what the user wants now?
- **Diversity:** does it offer useful variety?
- **Freshness:** can recently released items be discovered?
- **Long-term satisfaction:** does it remain useful over time?

These objectives can conflict: a familiar favourite may feel safer than an unfamiliar choice.

From a catalogue to a short list



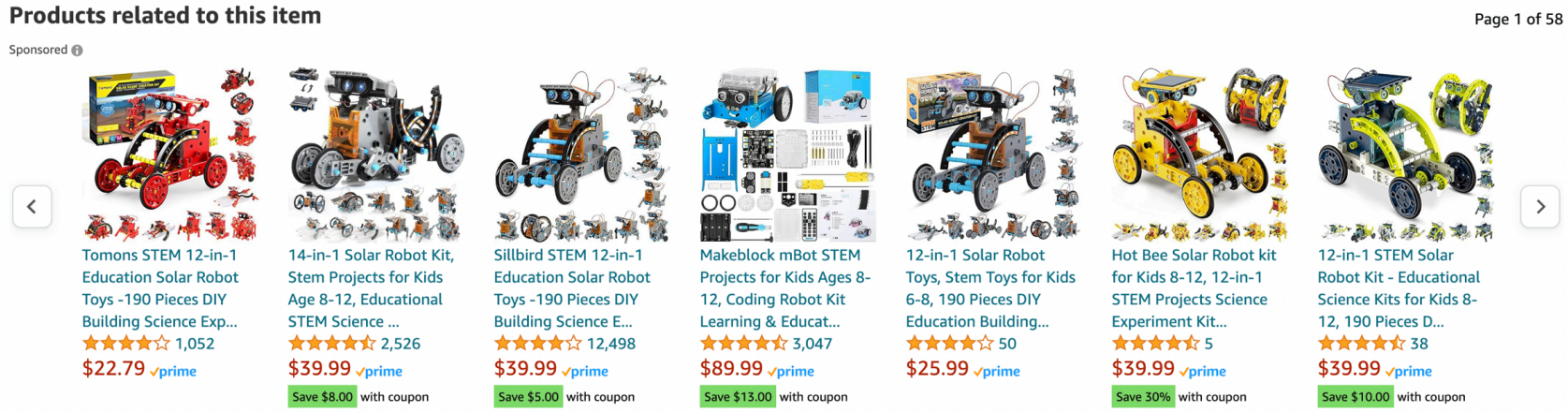
YouTube example; counts are illustrative. Covington et al. (2016), Fig. 2.

Ranking cannot recover an item excluded from the candidates.

Where could we introduce **new releases** or **more variety**?

Useful variety depends on context

You are considering a **solar robot toy**. Are these useful alternatives?



Amazon screenshot: sponsored related products.

What changes once you have already bought the toy?

Diversity and timing

Diversity describes how different the items in a recommendation list are from one another.

- **Before purchase:** similar toys help compare alternatives.
- **After purchase:** a compatible accessory or a different project may be more useful.
- **As interests change:** update recommendations using recent feedback.

Too little variety can narrow discovery. More variety is useful only when the items remain relevant.

Freshness, novelty, and exploration

Idea	Meaning	Movie example
Freshness	Recently released	A movie released this week
Novelty	New to this user	A classic they have never seen
Exploration	Try an uncertain choice to learn	A genre they have not rated

A fresh item can match familiar tastes. An older item can support exploration.

Familiar choices and exploration

You have one recommendation slot left. Which would you choose?

Choice	Possible benefit	What could we learn?
Familiar favourite (exploitation)	Strong evidence of relevance	Response to familiar content
Uncertain documentary (exploration)	A chance to discover a new interest	Response to a less-tested genre

What is the cost if the documentary misses the mark?

What we recommend affects what feedback we can collect

Trust and user feedback

A useful explanation should reflect the evidence the system actually used.

> “Recommended because you rated documentary movies highly.”

- Explanations **can support trust** when they are faithful and useful.
- Controls such as **“not interested”** let users correct recommendations.
- Recent feedback can help the system adapt when interests change.

What evidence would show improvement?

Question	Evidence to inspect
Are the top choices relevant?	User judgements of the top few items
Does the list offer useful variety?	Differences among items, plus user feedback
Can new items be discovered?	Exposure of recent releases and responses to them
Does it stay useful over time?	Repeated feedback about satisfaction and usefulness

Compare on a shared task and context. More clicks alone do not establish success.

Today's takeaways

- **Represent ratings:** missing entries are unknown preferences.
- **Evaluate honestly:** hold out observed ratings and fit using training data only.
- **Compare evidence sources:** collaborative methods use rating patterns; content-based models use item features.
- **Personalize with regression:** learn one user's model from their rated movies.
- **Handle missing history:** new items and new users pose different challenges.
- **Judge the final list:** relevance, variety, freshness, trust, and future feedback matter beyond RMSE.

Exit question

A recommender has **lower validation RMSE**, but users keep seeing the same kinds of movies and rarely discover anything new.

What would you investigate or change next, and why?

Explain both **what RMSE tells you** and **what it leaves unanswered**.