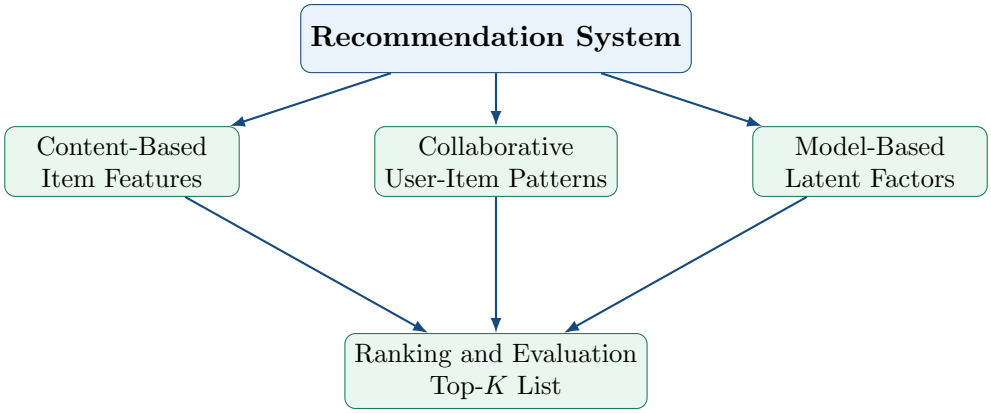
Recommendation Systems

CSE 435: Data Mining

Central Idea

A recommendation system estimates the relevance of candidate items for a user and produces a ranked list. The system may use item content, user-item interaction patterns, latent factors, context, or a combination of these signals. A good recommender must balance accuracy with diversity, coverage, novelty, fairness, and practical constraints.



1 Foundations of Recommendation Systems

Definition

Let $U = \{u_1, \ldots, u_m\}$ be a set of users and $I = \{i_1, \ldots, i_n\}$ a set of items. A recommender estimates a utility or preference score

$$\widehat{r}_{ui} = f(u, i, \text{context})$$

and recommends the top-K valid unseen items:

$$\text{Rec}_K(u) = \underset{i \in I_u^{\text{valid}}}{\arg \operatorname{topK}} \widehat{r}_{ui}.$$

The final goal is ranking useful items, not merely predicting numerical ratings.

1.1 Objectives and applications

Goal	Meaning	Example
Relevance	Recommend items matching the user's current need	Suitable course or movie
Discovery	Help the user find useful unknown items	New research paper
Diversity	Avoid nearly identical recommendations	Several genres or topics
Retention	Improve long-term engagement	Continued platform usage
Long-term value	Avoid optimizing only an immediate click	Sustainable learning path

Table 1: Common recommendation objectives.

1.2 Recommendation data and feedback

Source	Examples	Main issue
Explicit feedback	Ratings, likes, rankings, surveys	Strong but usually sparse
Implicit feedback	Clicks, views, purchases, skips, dwell time	Abundant but noisy and exposure-dependent
User information	Profile, interests, demographics	Privacy and fairness concerns
Item information	Category, text, price, difficulty	Metadata quality and maintenance
Context	Time, location, device, session	Preference may change with situation

Table 2: Sources of recommendation evidence.

Note

A missing rating means that preference is **unknown**. It does not automatically mean that the user dislikes the item.

1.3 User-item matrix and sparsity

Key Formula

The rating matrix is

$$R = [r_{ui}] \in \mathbb{R}^{m \times n},$$

where r_{ui} is observed feedback and a missing entry is unknown. If $|\Omega|$ ratings are observed, matrix sparsity is

$$\text{Sparsity} = 1 - \frac{|\Omega|}{mn}.$$

Worked Example 1: Constructing a User-Item Matrix

Five users rate six items on a 1–5 scale. A question mark represents an unknown rating.

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	5	4	?	1	?	3
U_2	4	?	5	?	2	?
U_3	?	5	4	?	3	?
U_4	1	?	2	5	?	4
U_5	?	3	?	4	5	?

Step 1: Count all possible entries.

$$mn = 5 \times 6 = 30.$$

Step 2: Count observed ratings. The row counts are $4, 3, 3, 4, 3$, so

$$|\Omega| = 4 + 3 + 3 + 4 + 3 = 17.$$

Step 3: Calculate sparsity.

$$\text{Sparsity} = 1 - \frac{17}{30} = \frac{13}{30} = \boxed{0.4333 \text{ or } 43.33\%}.$$

Step 4: Convert U_1 to binary implicit feedback. If every observed rating is treated as an interaction,

$$\mathbf{p}_{U_1} = (1, 1, 0, 1, 0, 1).$$

The zeros mean unobserved, not necessarily disliked.

2 Content-Based Filtering

Definition

Content-based filtering represents item i by a feature vector $\mathbf{x}_i$ and represents user u by a preference profile $\mathbf{p}_u$. Candidate items are ranked by their similarity with the user profile.

$$\text{Item features} \longrightarrow \text{User profile} \longrightarrow \text{Similarity} \longrightarrow \text{Ranking}.$$

Key Formula

A weighted user profile from preferred items L_u is

$$\mathbf{p}_u = \frac{\sum_{i \in L_u} w_{ui} \mathbf{x}_i}{\sum_{i \in L_u} w_{ui}}.$$

Cosine similarity is

$$s(u,i) = \cos(\mathbf{p}_u, \mathbf{x}_i) = \frac{\mathbf{p}_u^\top \mathbf{x}_i}{\|\mathbf{p}_u\|_2 \|\mathbf{x}_i\|_2}.$$

Worked Example 2: Content-Based Ranking with Six Movies

The three feature dimensions are Action, Comedy, and Drama. The user has rated movies A and E highly; the remaining movies are candidates.

Movie	Action	Comedy	Drama	User rating
A	1.0	0.0	0.0	5
B	0.8	0.2	0.0	?
C	0.1	0.8	0.1	?
D	0.0	0.2	0.8	?
E	0.7	0.0	0.3	4
F	0.2	0.4	0.4	?

Step 1: Build the weighted user profile from A and E.

$$\begin{aligned}\mathbf{p}_u &= \frac{5(1,0,0) + 4(0.7,0,0.3)}{5+4} \\ &= \frac{(7.8,0,1.2)}{9} = \left(\frac{13}{15}, 0, \frac{2}{15}\right) \\ &\approx (0.8667, 0, 0.1333).\end{aligned}$$

Its norm is

$$\|\mathbf{p}_u\| = \sqrt{0.8667^2 + 0^2 + 0.1333^2} \approx 0.8769.$$

Step 2: Calculate one candidate similarity in full. For movie B,

$$\begin{aligned}\mathbf{p}_u^\top \mathbf{x}_B &= (0.8667)(0.8) + (0)(0.2) + (0.1333)(0) \\ &\approx 0.6933, \\ \|\mathbf{x}_B\| &= \sqrt{0.8^2 + 0.2^2} = 0.8246, \\ s(u,B) &= \frac{0.6933}{(0.8769)(0.8246)} \approx 0.9589.\end{aligned}$$

Step 3: Repeat for all unseen movies.

Candidate	Dot product	Item norm	Cosine similarity
B	0.6933	0.8246	0.9589
C	0.1000	0.8124	0.1404
D	0.1067	0.8246	0.1475
F	0.2267	0.6000	0.4308

Step 4: Rank the candidates.

$$B\ (0.9589) > F\ (0.4308) > D\ (0.1475) > C\ (0.1404).$$

Content-Based Result

The recommendation order is

$$\boxed{B > F > D > C}.$$

Movie B is recommended first because its feature direction is most similar to the user's preference profile.

Note

Content-based filtering supports new items and explanations, but it may overspecialize and it cannot form a reliable profile for a completely new user without initial preference information.

3 Collaborative Filtering

Definition

Collaborative filtering uses interaction patterns rather than item descriptions.

- **User-based CF:** similar users provide evidence for an unseen item.
- **Item-based CF:** items similar to those already rated by the user provide evidence.

Both methods require sufficient overlap in the user-item matrix.

3.1 Cosine similarity and Pearson correlation

Key Formula

For rating vectors $\mathbf{r}_u$ and $\mathbf{r}_v$,

$$\mathrm{sim}_{\cos}(u,v) = \frac{\mathbf{r}_u^\top \mathbf{r}_v}{\|\mathbf{r}_u\|_2 \|\mathbf{r}_v\|_2}.$$

Pearson correlation over co-rated items I_{uv} is

$$\mathrm{sim}_P(u,v) = \frac{\sum_{i\in I_{uv}} (r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}} (r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{uv}} (r_{vi}-\bar{r}_v)^2}}.$$

Pearson correlation adjusts for users who apply different personal rating scales.

3.2 User-based collaborative filtering

Key Formula

The mean-centered user-based prediction is

$$\widehat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_k(u,i)} \operatorname{sim}(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in N_k(u,i)} |\operatorname{sim}(u,v)|}.$$

Worked Example 3: User-Based CF with Five Users

Predict the missing rating r_{U_1,I_4} using the two most similar positively correlated users.

	I_1	I_2	I_3	I_4	I_5	Row mean
U_1	5	3	4	?	2	3.5
U_2	4	2	5	5	1	3.4
U_3	5	3	4	4	2	3.6
U_4	1	5	2	1	5	2.8
U_5	2	4	1	2	4	2.6

The co-rated items with U_1 are I_1, I_2, I_3, I_5.

Step 1: Calculate $\operatorname{sim}(U_1, U_2)$**.** The centered vectors on the co-rated items are

$$U_1 : (1.5, -0.5, 0.5, -1.5),$$

$$U_2 : (0.6, -1.4, 1.6, -2.4).$$

Therefore,

$$\begin{aligned} \text{numerator} &= (1.5)(0.6) + (-0.5)(-1.4) + (0.5)(1.6) + (-1.5)(-2.4) \\ &= 0.9 + 0.7 + 0.8 + 3.6 = 6.0, \\ \text{denominator} &= \sqrt{1.5^2 + (-0.5)^2 + 0.5^2 + (-1.5)^2} \\ &\quad \times \sqrt{0.6^2 + (-1.4)^2 + 1.6^2 + (-2.4)^2} \\ &= \sqrt{5}\sqrt{10.64} \approx 7.2938. \end{aligned}$$

Hence,

$$\operatorname{sim}(U_1, U_2) = \frac{6}{7.2938} \approx 0.8226.$$

Step 2: Calculate the remaining similarities.

Neighbor	Pearson numerator	Pearson denominator	Similarity
U_2	6.0	7.2938	0.8226
U_3	5.0	5.0000	1.0000
U_4	−7.5	8.2341	−0.9108
U_5	−4.5	5.8481	−0.7695

Step 3: Select the top two positive neighbors.

$$N_2(U_1, I_4) = \{U_3, U_2\}.$$

Step 4: Predict the missing rating.

$$\begin{aligned}\widehat{r}_{U_1,I_4} &= 3.5 + \frac{(1.0000)(4-3.6)+(0.8226)(5-3.4)}{1.0000+0.8226}\\ &= 3.5 + \frac{0.4+1.3162}{1.8226}\\ &= 3.5 + 0.9416\\ &= \boxed{4.4416}.\end{aligned}$$

User-Based CF Result

The predicted rating is approximately 4.44 out of 5. Therefore, I_4 is a strong recommendation candidate for U_1.

3.3 Item-based collaborative filtering

Key Formula

For items i and j, compute similarity using users who rated both items. A simple weighted prediction is

$$\widehat{r}_{ui} = \frac{\sum_{j\in N_k(i;u)} \operatorname{sim}(i,j) r_{uj}}{\sum_{j\in N_k(i;u)} |\operatorname{sim}(i,j)|}.$$

Worked Example 4: Item-Based CF Using the Same Full Table

Again predict r_{U_1,I_4}. Since U_1 did not rate I_4, calculate item similarities over users U_2, U_3, U_4, U_5.

	I_1	I_2	I_3	I_4	I_5
U_1	5	3	4	?	2
U_2	4	2	5	5	1
U_3	5	3	4	4	2
U_4	1	5	2	1	5
U_5	2	4	1	2	4

The co-rating vectors are

$$I_4 = (5,4,1,2), \quad I_1 = (4,5,1,2),$$
$$I_2 = (2,3,5,4), \quad I_3 = (5,4,2,1), \quad I_5 = (1,2,5,4).$$

Step 1: Calculate one cosine similarity.

$$\begin{aligned}\operatorname{sim}(I_4,I_1) &= \frac{(5)(4)+(4)(5)+(1)(1)+(2)(2)}{\sqrt{5^2+4^2+1^2+2^2}\sqrt{4^2+5^2+1^2+2^2}}\\ &= \frac{45}{\sqrt{46}\sqrt{46}} = \frac{45}{46} = 0.9783.\end{aligned}$$

Step 2: Calculate all candidate item similarities.

Item pair	Dot product	Norm product	Cosine similarity
(I_4, I_1)	45	46.000	0.9783
(I_4, I_2)	35	49.840	0.7023
(I_4, I_3)	45	46.000	0.9783
(I_4, I_5)	26	46.000	0.5652

Step 3: Select the two most similar items.

$$N_2(I_4; U_1) = \{I_1, I_3\}.$$

Step 4: Calculate the weighted prediction.

$$\begin{aligned}\widehat{r}_{U_1,I_4} &= \frac{(0.9783)(5) + (0.9783)(4)}{0.9783 + 0.9783} \\ &= \frac{8.8047}{1.9566} = \boxed{4.50}.\end{aligned}$$

Item-Based CF Result

The predicted rating is 4.50. Items I_1 and I_3 provide the strongest evidence because their rating patterns are most similar to I_4.

4 Jaccard Similarity and Similarity Shrinkage

Key Formula

For binary interaction sets U_i and U_j,

$$J(i,j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|}.$$

With n_{common} common observations, shrinkage gives

$$\text{sim}'(i,j) = \frac{n_{\text{common}}}{n_{\text{common}} + \lambda}\,\text{sim}(i,j).$$

Shrinkage reduces apparently strong similarities supported by very little overlap.

Worked Example 5: Jaccard Similarity from a Full Binary Table

The table records six users' interactions with five items.

	A	B	C	D	E
U_1	1	0	1	0	1
U_2	1	1	1	0	0
U_3	1	1	0	1	0
U_4	1	0	0	1	1
U_5	0	1	1	1	0
U_6	0	0	1	0	1

Step 1: Form the interacting-user sets for A and B.

$$U_A = \{U_1, U_2, U_3, U_4\}, \qquad U_B = \{U_2, U_3, U_5\}.$$

Step 2: Find intersection and union.

$$U_A \cap U_B = \{U_2, U_3\}, \qquad |U_A \cap U_B| = 2,$$

$$U_A \cup U_B = \{U_1, U_2, U_3, U_4, U_5\}, \qquad |U_A \cup U_B| = 5.$$

Step 3: Calculate Jaccard similarity.

$$J(A,B) = \frac{2}{5} = \boxed{0.40}.$$

Step 4: Apply similarity shrinkage with $\lambda = 3$.

$$J'(A,B) = \frac{2}{2+3}(0.40) = 0.4(0.40) = \boxed{0.16}.$$

The reduced score reflects the fact that only two users interacted with both items.

5 Bias Baseline and Matrix Factorization

Definition

A bias baseline models systematic rating tendencies. Matrix factorization additionally learns low-dimensional latent characteristics for every user and item.

$$R \approx PQ^\top.$$

Key Formula

The bias baseline is

$$\widehat{r}_{ui} = \mu + b_u + b_i.$$

With latent factors $\mathbf{p}_u, \mathbf{q}_i \in \mathbb{R}^f$,

$$\widehat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i.$$

A regularized training objective is

$$\min_{P,Q,b} \sum_{(u,i)\in\Omega} \left(r_{ui} - \mu - b_u - b_i - \mathbf{p}_u^\top \mathbf{q}_i\right)^2 + \lambda \left(\|P\|_F^2 + \|Q\|_F^2 + \|b\|_2^2\right).$$

Worked Example 6A: Bias Baseline from the Full Rating Table

Reuse the five-user rating matrix from collaborative filtering. There are 24 observed entries whose sum is 76.

Step 1: Calculate the global mean.

$$\mu = \frac{76}{24} = 3.1667.$$

Step 2: Calculate the target user's simple bias. U_1 has observed mean 3.5:

$$b_{U_1} = 3.5 - 3.1667 = 0.3333.$$

Step 3: Calculate the target item's simple bias. I_4 has observed ratings $5, 4, 1, 2$, so its mean is 3:

$$b_{I_4} = 3.0 - 3.1667 = -0.1667.$$

Step 4: Predict with the baseline.

$$\widehat{r}_{U_1,I_4} = 3.1667 + 0.3333 - 0.1667 = \boxed{3.3333}.$$

These unregularized sample biases are used only to demonstrate the calculation. Production systems normally estimate them jointly with regularization.

Worked Example 6B: Matrix-Factorization Prediction

Suppose the trained model has the following two-dimensional latent vectors.

User	p_{u1}	p_{u2}	Item	q_{i1}	q_{i2}
U_1	0.8	0.3	I_1	0.7	0.2
U_2	0.7	0.4	I_2	0.2	0.8
U_3	0.9	0.2	I_3	0.8	0.1
U_4	0.1	0.8	I_4	0.5	0.4
U_5	0.2	0.7	I_5	0.1	0.9

For U_1 and I_4,

$$\mathbf{p}_{U_1} = (0.8, 0.3), \qquad \mathbf{q}_{I_4} = (0.5, 0.4).$$

Step 1: Calculate latent compatibility.

$$\mathbf{p}_{U_1}^{\top}\mathbf{q}_{I_4} = (0.8)(0.5) + (0.3)(0.4) = 0.40 + 0.12 = 0.52.$$

Step 2: Add the global mean and biases.

$$\begin{aligned}\widehat{r}_{U_1,I_4} &= \mu + b_{U_1} + b_{I_4} + \mathbf{p}_{U_1}^{\top}\mathbf{q}_{I_4} \\ &= 3.1667 + 0.3333 - 0.1667 + 0.52 \\ &= \boxed{3.8533}.\end{aligned}$$

The latent term raises the baseline prediction from 3.3333 to 3.8533 because the learned user and item factors are compatible.

Worked Example 6C: One Stochastic-Gradient Update

For one observed rating $r_{ui} = 5$, suppose

$$\mu = 3.1667, \quad b_u = 0.1, \quad b_i = -0.1,$$

$$\mathbf{p}_u = (0.2, 0.1), \quad \mathbf{q}_i = (0.3, 0.4), \quad \gamma = 0.05, \quad \lambda = 0.02.$$

Step 1: Prediction and error.

$$\widehat{r}_{ui} = 3.1667 + 0.1 - 0.1 + (0.2)(0.3) + (0.1)(0.4) = 3.2667,$$

$$e_{ui} = r_{ui} - \widehat{r}_{ui} = 5 - 3.2667 = 1.7333.$$

Step 2: Update biases.

$$\begin{aligned}b_u' &= b_u + \gamma(e_{ui} - \lambda b_u) = 0.1 + 0.05(1.7333 - 0.002) = 0.1866, \\ b_i' &= b_i + \gamma(e_{ui} - \lambda b_i) = -0.1 + 0.05(1.7333 + 0.002) = -0.0132.\end{aligned}$$

Step 3: Update factors using the old vectors.

$$\begin{aligned}\mathbf{p}_u' &= \mathbf{p}_u + \gamma(e_{ui}\mathbf{q}_i - \lambda\mathbf{p}_u) \approx (0.2258, 0.1346), \\ \mathbf{q}_i' &= \mathbf{q}_i + \gamma(e_{ui}\mathbf{p}_u - \lambda\mathbf{q}_i) \approx (0.3170, 0.4083).\end{aligned}$$

The update increases compatibility because the original prediction was below the observed rating.

6 Implicit Feedback and Pairwise Ranking

Definition

Implicit feedback records actions such as clicks, views, or purchases. Observed interaction is evidence of preference, while an unobserved pair remains a low-confidence unknown.

Key Formula

For interaction count r_{ui},

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0, \\ 0, & r_{ui} = 0, \end{cases} \qquad c_{ui} = 1 + \alpha r_{ui}.$$

Weighted implicit matrix factorization minimizes

$$\sum_{u,i} c_{ui}(p_{ui} - \widehat{r}_{ui})^2 + \lambda(\|P\|_F^2 + \|Q\|_F^2).$$

For Bayesian Personalized Ranking (BPR), if user u prefers observed item i over sampled unobserved item j,

$$\widehat{x}_{uij} = \widehat{r}_{ui} - \widehat{r}_{uj}, \qquad \mathcal{L}_{\text{BPR}} = -\ln \sigma(\widehat{x}_{uij}).$$

Worked Example 7A: Preference and Confidence Tables

The entries are interaction counts for five users and six items.

	A	B	C	D	E	F
U_1	3	0	1	0	2	0
U_2	0	2	1	0	0	1
U_3	1	0	0	4	0	2
U_4	0	1	0	3	2	0
U_5	2	0	1	0	0	3

For U_1,

$$\mathbf{r}_{U_1} = (3, 0, 1, 0, 2, 0).$$

Step 1: Convert counts to preferences.

$$\mathbf{p}_{U_1} = (1, 0, 1, 0, 1, 0).$$

Step 2: Calculate confidence with $\alpha = 2$.

$$\mathbf{c}_{U_1} = 1 + 2\mathbf{r}_{U_1} = (7, 1, 3, 1, 5, 1).$$

Thus the three interactions with item A give more confidence than the one interaction with C.

Worked Example 7B: BPR Pairwise Loss

For U_1, choose observed item A as the positive item and unobserved item B as a sampled negative item. Suppose

$$\mathbf{p}_{U_1} = (0.6, 0.2), \quad \mathbf{q}_A = (0.8, 0.1), \quad \mathbf{q}_B = (0.2, 0.7).$$

Step 1: Calculate the two scores.

$$\widehat{r}_{U_1,A} = (0.6)(0.8) + (0.2)(0.1) = 0.50,$$

$$\widehat{r}_{U_1,B} = (0.6)(0.2) + (0.2)(0.7) = 0.26.$$

Step 2: Calculate the pairwise margin.

$$\widehat{x}_{U_1AB} = 0.50 - 0.26 = 0.24.$$

Step 3: Convert the margin to a probability.

$$\sigma(0.24) = \frac{1}{1 + e^{-0.24}} \approx 0.5597.$$

Step 4: Calculate BPR loss.

$$\mathcal{L}_{\text{BPR}} = -\ln(0.5597) \approx \boxed{0.5803}.$$

Training attempts to increase $\widehat{r}_{U_1,A} - \widehat{r}_{U_1,B}$ and therefore reduce this loss.

7 Evaluation of Recommendation Results

Definition

Rating metrics evaluate numerical prediction. Top-K metrics evaluate the recommended list. Beyond-accuracy metrics evaluate qualities such as diversity and catalogue coverage. The evaluation split should respect time whenever the system predicts future behavior.

7.1 RMSE and MAE

Key Formula

For test set Ω_{test},

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega_{\text{test}}|} \sum_{(u,i)\in\Omega_{\text{test}}} (r_{ui} - \widehat{r}_{ui})^2},$$

$$\text{MAE} = \frac{1}{|\Omega_{\text{test}}|} \sum_{(u,i)\in\Omega_{\text{test}}} |r_{ui} - \widehat{r}_{ui}|.$$

RMSE penalizes large errors more strongly than MAE.

Worked Example 8A: RMSE and MAE with Six Predictions

Case	Actual r	Predicted $\widehat{r}$	$\lvert r - \widehat{r}\rvert$	$(r - \widehat{r})^2$
1	5	4.5	0.5	0.25
2	4	3.5	0.5	0.25
3	3	2.5	0.5	0.25
4	2	2.0	0.0	0.00
5	1	1.5	0.5	0.25
6	4	3.0	1.0	1.00
		Column sum	3.0	2.00

$$\text{MAE} = \frac{3.0}{6} = \boxed{0.5000},$$

$$\text{RMSE} = \sqrt{\frac{2.0}{6}} = \sqrt{0.3333} = \boxed{0.5774}.$$

7.2 Precision@K and Recall@K

Key Formula

Let Rel_u be the set of relevant items and Rec_u^K the top-K recommendation set:

$$\text{Precision@K} = \frac{|\,\text{Rel}_u \cap \text{Rec}_u^K\,|}{K}, \qquad \text{Recall@K} = \frac{|\,\text{Rel}_u \cap \text{Rec}_u^K\,|}{|\,\text{Rel}_u\,|}.$$

Worked Example 8B: Top-5 Precision and Recall

Recommended ranking:

$$[A, B, C, D, E].$$

Relevant items:

$$\text{Rel}_u = \{A, C, F\}.$$

Rank	Recommended item	Relevant?	Cumulative relevant
1	A	Yes	1
2	B	No	1
3	C	Yes	2
4	D	No	2
5	E	No	2

Two of the five recommended items are relevant:

$$\text{Precision@5} = \frac{2}{5} = \boxed{0.40}.$$

Two of the three relevant items were retrieved:

$$\text{Recall@5} = \frac{2}{3} = \boxed{0.6667}.$$

7.3 NDCG@K

Key Formula

For graded relevance rel_j at rank j,

$$\text{DCG@K} = \sum_{j=1}^{K} \frac{2^{\text{rel}_j} - 1}{\log_2(j+1)}, \qquad \text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}}.$$

IDCG@K is the DCG of the ideal ordering.

Worked Example 8C: NDCG@5 for the Same Ranking

Use binary relevance. Relevant items occur at ranks 1 and 3.

Rank j	Item	rel_j	DCG contribution
1	A	1	$1/\log_2(2) = 1$
2	B	0	0
3	C	1	$1/\log_2(4) = 0.5$
4	D	0	0
5	E	0	0

Therefore,

$$\text{DCG@5} = 1 + 0.5 = 1.5.$$

There are three relevant items, so the ideal ranking places them at ranks $1, 2, 3$:

$$\begin{aligned}\text{IDCG@5} &= 1 + \frac{1}{\log_2(3)} + \frac{1}{\log_2(4)} \\ &\approx 1 + 0.6309 + 0.5 = 2.1309.\end{aligned}$$

Thus,

$$\text{NDCG@5} = \frac{1.5}{2.1309} = \boxed{0.7039}.$$

7.4 Diversity and coverage

Key Formula

For recommendation list L of size K,

$$\text{ILD}(L) = \frac{2}{K(K-1)} \sum_{i<j} (1 - \text{sim}(i,j)) .$$

Catalogue coverage across all users is

$$\text{Coverage} = \frac{|\bigcup_u \text{Rec}_K(u)|}{|I|}.$$

Worked Example 8D: Intra-List Diversity with Five Items

Suppose the recommendation list is $L = [A, B, C, D, E]$ and the item-similarity matrix is:

	A	B	C	D	E
A	1.0	0.8	0.2	0.1	0.3
B	0.8	1.0	0.3	0.2	0.4
C	0.2	0.3	1.0	0.7	0.6
D	0.1	0.2	0.7	1.0	0.5
E	0.3	0.4	0.6	0.5	1.0

The ten pairwise dissimilarities are

$$0.2, 0.8, 0.9, 0.7, 0.7, 0.8, 0.6, 0.3, 0.4, 0.5.$$

Their sum is 5.9. Therefore,

$$\text{ILD}(L) = \frac{2}{5(4)}(5.9) = 0.1(5.9) = \boxed{0.59}.$$

If the catalogue contains 20 items and recommendation lists across all users contain 12 unique items,

$$\text{Coverage} = \frac{12}{20} = \boxed{0.60 \text{ or } 60\%}.$$

7.5 Novelty and online measures

Key Formula

If $\text{pop}(i)$ is the number of users who interacted with item i, a self-information novelty score is

$$\text{Novelty}(L) = \frac{1}{|L|} \sum_{i \in L} -\log_2 \left(\frac{\text{pop}(i)}{|U|} \right).$$

Higher values indicate less familiar items. Novelty should be balanced with relevance.

Worked Example 8E: Novelty of Five Recommended Items

Assume a system has 1,000 users and recommends five items.

Item	Interacting users	Popularity probability	$-\log_2(p_i)$
A	500	0.500	1.0000
B	250	0.250	2.0000
C	100	0.100	3.3219
D	50	0.050	4.3219
E	25	0.025	5.3219

Therefore,

$$\text{Novelty}(L) = \frac{1 + 2 + 3.3219 + 4.3219 + 5.3219}{5} = \boxed{3.1931 \text{ bits}}.$$

Online measure	Formula or observation	Interpretation
Click-through rate	Clicks / recommendation impressions	Immediate interest
Conversion rate	Desired actions / impressions or clicks	Business or task completion
Completion time	Watched, read, or completed portion	Depth of engagement
Retention	Users returning after a defined period	Long-term usefulness
User satisfaction	Survey or explicit feedback	Perceived recommendation quality

Table 3: Online recommendation measures.

8 Hybrid Recommendation

Definition

Hybrid systems combine complementary signals. Common strategies include weighted combination, switching, cascade, feature augmentation, and joint modelling.

Key Formula

A weighted hybrid score is

$$s(u,i) = \alpha s_{\mathrm{CF}}(u,i) + (1-\alpha)s_{\mathrm{content}}(u,i), \qquad 0 \le \alpha \le 1.$$

Scores should be calibrated to comparable scales before they are combined.

Worked Example 9: Weighted Hybrid Ranking with Six Candidates

Use $\alpha = 0.6$, so collaborative filtering receives 60% weight and content-based filtering receives 40%. For M_1,

$$s(u, M_1) = 0.6(0.90) + 0.4(0.70) = 0.54 + 0.28 = 0.82.$$

Item	s_{CF}	s_{content}	Hybrid calculation	Final score
M_1	0.90	0.70	$0.6(0.90) + 0.4(0.70)$	0.82
M_2	0.60	0.95	$0.6(0.60) + 0.4(0.95)$	0.74
M_3	0.80	0.60	$0.6(0.80) + 0.4(0.60)$	0.72
M_4	0.40	0.85	$0.6(0.40) + 0.4(0.85)$	0.58
M_5	0.70	0.50	$0.6(0.70) + 0.4(0.50)$	0.62
M_6	0.55	0.65	$0.6(0.55) + 0.4(0.65)$	0.59

The final ranking is

$$\boxed{M_1 > M_2 > M_3 > M_5 > M_6 > M_4}.$$

The hybrid allows M_2 to rank highly because its strong content score compensates for its moderate collaborative score.

9 Two-Stage Recommendation Architecture

- **Candidate retrieval** searches a huge catalogue quickly and emphasizes recall.
- **Ranking** applies richer features to a smaller candidate set.
- **Re-ranking** applies diversity, freshness, safety, eligibility, and business constraints.

Worked Example 10: Retrieval, Ranking, and Re-Ranking

Suppose retrieval produces the following eight candidate courses. A final list may contain at most two courses from the same category.

Course	Retrieval score	Ranking score	Category	Valid?
C_1	0.96	0.78	AI	Yes
C_2	0.92	0.91	AI	Yes
C_3	0.90	0.88	Data	Yes
C_4	0.87	0.95	AI	Yes
C_5	0.84	0.82	Systems	Yes
C_6	0.80	0.86	Data	No: prerequisite
C_7	0.78	0.80	Security	Yes
C_8	0.75	0.84	Web	Yes

Step 1: Candidate retrieval. Take the six highest retrieval scores:

$$[C_1, C_2, C_3, C_4, C_5, C_6].$$

Step 2: Remove invalid candidates. C_6 is removed because its prerequisite is missing:

$$[C_1, C_2, C_3, C_4, C_5].$$

Step 3: Rank by the richer ranking score.

$$[C_4(0.95), C_2(0.91), C_3(0.88), C_5(0.82), C_1(0.78)].$$

Step 4: Re-rank for category diversity. The first two positions are already AI courses. Therefore, defer the third AI course C_1 and retain the diverse top four:

$$\boxed{[C_4, C_2, C_3, C_5]}.$$

10 Modern Recommendation Models

Model	Mathematical idea	Suitable use
Factorization machines	$\widehat{y} = w_0 + \sum_j w_j x_j + \sum_{j<\ell} \langle \mathbf{v}_j, \mathbf{v}_\ell \rangle x_j x_\ell$	Sparse user, item, and context features
Neural CF	$\widehat{y}_{ui} = \sigma(\mathbf{h}^\top \phi(\mathbf{p}_u, \mathbf{q}_i))$	Nonlinear interaction patterns
Sequential model	$P(i_{t+1} \mid i_1, \ldots, i_t)$	Next-item recommendation
Transformer	$\text{Attention}(Q, K, V) = \text{softmax}(QK^\top/\sqrt{d})V$	Long interaction histories
Graph model	$\mathbf{e}^{(\ell+1)} = \mathcal{A}\mathbf{e}^{(\ell)} W_\ell$	Higher-order user-item relations
Context-aware model	$\widehat{r} = f(u, i, \text{time, device, location, session})$	Situation-dependent preferences

Table 4: Mathematical summary of modern recommendation models.

Worked Example 11: Scaled Dot-Product Attention over Five Interactions

A sequential recommender examines five previous interactions. Suppose the query-key dot products, after division by $\sqrt{d}$, are

$$\mathbf{z} = (1.2, 0.3, 1.0, -0.2, 0.7).$$

The attention weights are

$$a_j = \frac{e^{z_j}}{\sum_{t=1}^{5} e^{z_t}}.$$

Position	Scaled score z_j	e^{z_j}	Attention a_j	Value v_j
1	1.2	3.3201	0.3248	0.9
2	0.3	1.3499	0.1321	0.2
3	1.0	2.7183	0.2660	0.8
4	−0.2	0.8187	0.0801	0.1
5	0.7	2.0138	0.1970	0.6

First compute the softmax denominator:

$$Z = 3.3201 + 1.3499 + 2.7183 + 0.8187 + 2.0138 = 10.2208.$$

The resulting weights are

$$(0.3248, 0.1321, 0.2660, 0.0801, 0.1970),$$

which sum to 1.0000. The attention output is

$$\begin{aligned} h &= \sum_{j=1}^{5} a_j v_j \\ &= (0.3248)(0.9) + (0.1321)(0.2) + (0.2660)(0.8) \\ &\quad + (0.0801)(0.1) + (0.1970)(0.6) \\ &\approx \boxed{0.6577}. \end{aligned}$$

The first and third interactions receive the greatest attention.

11 Practical Challenges and Responsible Recommendation

Challenge	Why it occurs	Typical response
New user	No interaction history	Popular-diverse list, survey, context
New item	No interaction evidence	Metadata and content similarity
Sparsity	Few observed ratings	Regularization, latent factors, side information
Popularity bias	Popular items receive more exposure and interactions	Re-ranking, calibration, exploration
Provider fairness	Exposure may concentrate on a few providers	Exposure constraints and audits
Privacy	Personal behavior is sensitive	Data minimization, access control, privacy methods
Preference drift	User interests change over time	Time-aware features and monitoring
Feedback loop	Exposure changes future training data	Exploration and counterfactual evaluation

Table 5: Major practical challenges.

Exposure Fairness

If $\mathrm{pos}(i, u)$ is the displayed rank of item i for user u, discounted exposure can be measured as

$$\mathrm{Exposure}(i) = \sum_{u} \frac{\mathbb{I}[i \in \mathrm{Rec}_K(u)]}{\log_2(\mathrm{pos}(i, u) + 1)}.$$

Group-level exposure can then be compared across item providers or protected groups.

12 Case Study: University Course Recommendation

Definition

A course recommender first removes academically invalid courses and only then ranks the remaining candidates. A high predicted preference cannot override a prerequisite, timetable, semester-offering, or credit-limit constraint.

Worked Example 12: Constraint-Aware Course Recommendation

Assume the student has already completed Data Structures and may take only offered courses with satisfied prerequisites and no timetable conflict.

ID	Course	Score	Prerequisite	Offered	Conflict	Completed
C_1	Data Structures	4.8	Yes	Yes	No	Yes
C_2	Machine Learning	4.7	Yes	Yes	No	No
C_3	Compiler Design	4.5	No	Yes	No	No
C_4	Web Programming	4.3	Yes	Yes	Yes	No
C_5	Data Mining	4.6	Yes	Yes	No	No
C_6	Artificial Intelligence	4.4	Yes	No	No	No
C_7	Cybersecurity	4.2	Yes	Yes	No	No

Step 1: Apply hard constraints.

- Remove C_1 because it is already completed.
- Remove C_3 because its prerequisite is missing.
- Remove C_4 because of a timetable conflict.
- Remove C_6 because it is not offered.

The valid set is

$$I_u^{\text{valid}} = \{C_2, C_5, C_7\}.$$

Step 2: Rank only the valid courses.

$$C_2(4.7) > C_5(4.6) > C_7(4.2).$$

Step 3: Produce an explanation.

> Machine Learning is ranked first because it has the highest predicted preference among courses that are offered, conflict-free, incomplete, and academically eligible.

Therefore, the final recommendation is

$$\boxed{[\text{Machine Learning},\ \text{Data Mining},\ \text{Cybersecurity}]}.$$

13 Comparison of Recommendation Methods

Method	Main evidence	Prediction mechanism	Main strength	Main limitation
Content-based	Item metadata and user profile	Feature-vector similarity	New-item support and explanation	Overspecialization; new-user problem
User-based CF	Similar users' ratings	Neighbor mean deviations	Intuitive personalization	Weak under sparse overlap
Item-based CF	Similar item rating patterns	Weighted ratings of related items	Often stable and explainable	New-item and sparsity problems
Matrix factorization	Sparse interaction matrix	User/item latent dot product	Strong compact collaborative model	Latent factors can be hard to explain
Implicit/BPR	Clicks, views, purchases	Confidence weighting or pairwise ranking	Matches ranking objective	Negative sampling and exposure bias
Hybrid	Multiple signals	Score or model combination	Robustness and cold-start reduction	Calibration and system complexity
Sequential/ Transformer	Ordered history	Conditional next-item model and attention	Captures changing short-term intent	Higher data and computation needs

Table 6: Method-selection summary.

Practical Model-Selection Rules

- Start with popularity and simple content or neighborhood baselines.
- Use content-based filtering when item metadata is strong or new items are frequent.
- Use collaborative filtering when interaction overlap is sufficient.
- Use matrix factorization for large sparse explicit-feedback data.
- Use confidence weighting or BPR for implicit-feedback ranking.
- Use hybrid and two-stage systems when catalogue scale and cold start both matter.
- Evaluate ranking, diversity, coverage, fairness, latency, and user outcomes, not only RMSE.

14 Short Practice Problems

1. A 10×20 user-item matrix contains 50 observed ratings. Calculate its sparsity.
2. A user profile is $(0.8, 0.6)$ and an item vector is $(0.6, 0.8)$. Calculate cosine similarity.
3. Two item-interaction sets are $U_A = \{1, 2, 3, 4, 5\}$ and $U_B = \{2, 4, 6\}$. Calculate Jaccard similarity.
4. If $\bar{r}_u = 3.2$, one neighbor has similarity 0.8, mean rating 3.5, and rating 5 for the target item, calculate the one-neighbor mean-centered prediction.
5. For actual ratings $(5, 3, 2, 4, 1)$ and predictions $(4, 3, 3, 3, 2)$, calculate MAE.
6. A top-5 list contains three relevant items, while the user has six relevant items in total. Calculate Precision@5 and Recall@5.
7. With $\alpha = 0.7$, $s_{\text{CF}} = 0.8$, and $s_{\text{content}} = 0.5$, calculate the weighted hybrid score.
8. If $\widehat{r}_{ui} = 1.2$ and $\widehat{r}_{uj} = 0.5$, calculate the BPR margin $\widehat{x}_{uij}$.

Answer Check

1. $1 - 50/(10 \times 20) = 0.75$ or 75%.
2. $(0.8 \cdot 0.6 + 0.6 \cdot 0.8)/(1 \cdot 1) = 0.96$.
3. $|\{2,4\}|/|\{1,2,3,4,5,6\}| = 2/6 = 0.3333$.
4. $3.2 + [0.8(5 - 3.5)]/0.8 = 4.7$.
5. Absolute errors are $(1,0,1,1,1)$, so MAE $= 4/5 = 0.8$.
6. Precision@5 $= 3/5 = 0.60$ and Recall@5 $= 3/6 = 0.50$.
7. $0.7(0.8) + 0.3(0.5) = 0.71$.
8. $\widehat{x}_{uij} = 1.2 - 0.5 = 0.7$.

15 Key Takeaways

1. Recommendation is primarily a ranking problem supported by explicit, implicit, item, user, and contextual evidence.
2. Content-based methods compare item features with a user profile; collaborative methods exploit user-item interaction patterns.
3. Neighborhood predictions depend on similarity quality and sufficient overlap.
4. Matrix factorization combines global effects, user/item biases, and latent compatibility.
5. Implicit feedback requires confidence modelling or ranking-aware objectives such as BPR.
6. RMSE and MAE do not replace top-K evaluation. Precision, recall, NDCG, diversity, and coverage answer different questions.
7. Real systems normally use retrieval, ranking, and constraint-aware re-ranking while monitoring cold start, bias, privacy, drift, and feedback loops.

Prepared By
Md. Atikuzzaman
Lecturer, Department of Computer Science and Engineering
Green University of Bangladesh
Email: atik@cse.green.edu.bd