CSE 435: Data Mining

Recommendation Systems

Md Atikuzzaman

Lecturer

Department of Computer Science & Engineering

atik@cse.green.edu.bd

GREEN UNIVERSITY OF BANGLADESH

Outline

Foundations

- Recommendation goals
- Data and feedback
- User–item matrix
- Core strategies

Core Methods

- Content-based filtering
- Collaborative filtering
- Matrix factorization
- Implicit feedback

Evaluation

- RMSE and MAE
- Precision@K
- Recall@K
- NDCG@K
- Diversity and coverage

Modern Systems

- Hybrid recommenders
- Candidate retrieval
- Ranking and re-ranking
- Neural / sequential

Practical Issues

- Cold start
- Sparsity
- Popularity bias
- Fairness
- Privacy and drift

Application

- Course recommendation
- Hard constraints

What is a Recommendation System?

Definition

A **recommendation system** estimates and ranks items that are likely to be relevant or useful to a particular user in a given context.

Applications

- Movies and music
- E-commerce products
- News and social content
- Courses and learning resources
- Jobs
- Research papers

Objectives

- Relevance
- User satisfaction
- Discovery
- Diversity
- Retention
- Long-term value

Core Idea

A recommender is primarily a **ranking system**. Accurate rating prediction alone does not guarantee a useful recommendation list.

Recommendation Data

Recommendation systems normally combine several sources of information.

User Information

- User identity or profile
- Demographic/contextual features
- Historical preferences
- Long-term interests

Item Information

- Category / genre
- Price / difficulty
- Text and multimedia
- Availability and freshness

Interaction Information

- Ratings
- Clicks
- Views
- Purchases
- Watch / completion

Context

- Time
- Location
- Device
- Query / session

Explicit and Implicit Feedback

Feedback

Feedback provides evidence about a user's preference for an item.

Type	Examples	Characteristics
Explicit	Star ratings, likes, rankings, surveys	Direct preference signal, but usually sparse
Implicit	Clicks, views, purchases, skips, dwell time	Abundant, but noisy and exposure-dependent

Explicit Feedback

- Easier to interpret
- Strong preference evidence
- Users may not provide it frequently

Implicit Feedback

- Automatically collected
- Available at large scale
- Interaction does not always mean preference

Important

A missing interaction usually means **unknown preference**, not necessarily dislike.

User–Item Matrix and Core Strategies

$$R \in \mathbb{R}^{m\times n}, \qquad r_{ui} = \text{feedback from user } u \text{ on item } i$$

	Item A	Item B	Item C	Item D
User 1	5	?	3	?
User 2	4	2	?	1
User 3	?	5	4	?
User 4	1	?	?	5

Content-Based

Uses item attributes and individual user preferences.

Collaborative

Uses interaction patterns of similar users or items.

Hybrid

Combines content, interactions, and other evidence.

Sparsity

In real applications, most entries of the user–item matrix are missing.

Content-Based Filtering

Item Features $\rightarrow$ User Profile $\rightarrow$ Similarity $\rightarrow$ Ranking

- Represent item i using feature vector $\mathbf{x}_i$.
- Construct user profile $\mathbf{p}_u$ from preferred items.
- Compare candidate items with the user's profile.
- Rank unseen items by similarity.

Cosine Similarity

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

Strengths

- Works for new items
- Explainable
- Supports niche preferences

Limitations

- Overspecialization
- Requires useful metadata
- New-user problem

Worked Content-Based Example

Given Information

Suppose the movie features represent:

$$(\text{Action}, \text{Comedy})$$

The movie vectors are:

$$\mathbf{x}_A = (1, 0), \qquad \mathbf{x}_B = (0.8, 0.2)$$

$$\mathbf{x}_C = (0, 1)$$

The user strongly likes movie A. Therefore,

$$\boxed{\mathbf{p}_u = (1, 0)}$$

Cosine Similarity

$$\cos(\mathbf{p}_u, \mathbf{x}_i) = \frac{\mathbf{p}_u \cdot \mathbf{x}_i}{\|\mathbf{p}_u\|\|\mathbf{x}_i\|}$$

Similarity with Movie B

$$\cos(\mathbf{p}_u, \mathbf{x}_B) = \frac{(1)(0.8) + (0)(0.2)}{\sqrt{1^2 + 0^2}\sqrt{0.8^2 + 0.2^2}}$$

$$= \frac{0.8}{\sqrt{0.68}} \approx \boxed{0.970}$$

Similarity with Movie C

$$\cos(\mathbf{p}_u, \mathbf{x}_C) = \frac{(1)(0) + (0)(1)}{\sqrt{1^2 + 0^2}\sqrt{0^2 + 1^2}}$$

$$\boxed{\cos(\mathbf{p}_u, \mathbf{x}_C) = 0}$$

Recommendation

$0.970 > 0$

Therefore,

$\boxed{\text{Movie } B \text{ should be ranked before Movie } C.}$

Collaborative Filtering

Main Assumption

Users with similar historical behaviour may prefer similar items. Similarly, items consumed by similar users may be related.

User-Based CF

1. Find users similar to target user.
2. Select nearest neighbours.
3. Collect their preferences.
4. Predict or rank unseen items.

Item-Based CF

1. Find items similar to target item.
2. Identify items already consumed.
3. Aggregate their similarities.
4. Rank candidate items.

Challenge

Collaborative filtering depends strongly on sufficient interaction overlap between users or items.

Cosine Similarity and Pearson Correlation

Cosine Similarity

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

Characteristics

- Simple and efficient
- Works with sparse vectors
- Measures vector orientation
- Does not automatically correct rating-scale differences

Pearson Correlation

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

Characteristics

- Mean-centers ratings
- Adjusts personal rating scale
- Uses co-rated items

Jaccard Similarity and Similarity Shrinkage

Jaccard Similarity

For two items with interacting-user sets U_i and U_j:

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

Useful for **binary interactions** such as clicks, purchases, tags, or views.

Example:

$$U_i = \{1,2,3,4\}, \qquad U_j = \{2,3,5\}$$

$$J(i,j) = \frac{|\{2,3\}|}{|\{1,2,3,4,5\}|} = \boxed{0.4}$$

Jaccard Similarity and Similarity Shrinkage

Similarity Shrinkage

When only a few common interactions exist:

$$sim' = \frac{n_{\text{common}}}{n_{\text{common}} + \lambda} sim$$

This reduces unreliable similarities based on very small overlap.

User-Based Rating Prediction

Prediction Formula

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

Given Information

$$\bar{r}_u = 3.5$$

$$sim(u,1) = 0.8, \qquad sim(u,2) = 0.5$$

$$r_{1i} = 5, \quad \bar{r}_1 = 4$$

$$r_{2i} = 3, \quad \bar{r}_2 = 2.5$$

Neighbour Deviations

$$r_{1i} - \bar{r}_1 = 5 - 4 = 1$$

$$r_{2i} - \bar{r}_2 = 3 - 2.5 = 0.5$$

Substitution

$$\hat{r}_{ui} = 3.5 + \frac{0.8(5-4) + 0.5(3-2.5)}{0.8 + 0.5}$$

$$= 3.5 + \frac{0.8(1) + 0.5(0.5)}{1.3}$$

$$= 3.5 + \frac{0.8 + 0.25}{1.3}$$

$$= 3.5 + \frac{1.05}{1.3}$$

$$= 3.5 + 0.808$$

$$\boxed{\hat{r}_{ui} \approx 4.31}$$

Interpretation
The predicted rating is high; therefore, item *i* is a **strong recommendation candidate**.

Bias Baseline and Matrix Factorization

Bias Baseline

A simple baseline models general rating tendencies:

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

where:

$$\mu = \text{global mean}, \quad b_u = \text{user bias}, \quad b_i = \text{item bias}$$

Matrix Factorization

Approximate the sparse rating matrix using lower-dimensional factors:

$$R \approx PQ^\top$$

Prediction becomes:

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

where $\mathbf{p}_u$ and $\mathbf{q}_i$ represent latent user and item characteristics.

Worked Matrix-Factorization Prediction

Suppose:

$$\mu = 3.5, \qquad b_u = 0.2, \qquad b_i = -0.1$$

User and item latent vectors:

$$\mathbf{p}_u = (0.8, 0.3)$$

$$\mathbf{q}_i = (0.5, 0.4)$$

Compatibility:

$$\mathbf{p}_u^\top \mathbf{q}_i = (0.8)(0.5) + (0.3)(0.4) = 0.52$$

Therefore:

$$\hat{r}_{ui} = 3.5 + 0.2 - 0.1 + 0.52$$

$$\boxed{\hat{r}_{ui} = 4.12}$$

Interpretation

The prediction combines global behaviour, user/item bias, and latent user–item compatibility.

Implicit Feedback and Pairwise Ranking

For implicit interactions, define:

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0 \\ 0, & \text{otherwise} \end{cases}$$

and confidence:

$$c_{ui} = 1 + \alpha r_{ui}$$

Interpretation

Observed interaction gives evidence of preference. Missing interaction remains a low-confidence unknown.

Negative Sampling

- Positive samples: observed user–item interactions
- Negative samples: selected unobserved items

Bayesian Personalized Ranking (BPR)

For preferred item i and sampled item j:

$$\hat{x}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$$

The desired ranking is:

$$\boxed{\hat{r}_{ui} > \hat{r}_{uj}}$$

Rating and Top-K Evaluation Metrics

Rating Prediction

RMSE

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

MAE

$$MAE = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} |r_{ui} - \hat{r}_{ui}|$$

Top-K Ranking

Precision@K

$$Precision@K = \frac{|Rel_u \cap Rec_u^K|}{K}$$

Recall@K

$$Recall@K = \frac{|Rel_u \cap Rec_u^K|}{|Rel_u|}$$

Key Difference

RMSE/MAE evaluate numerical predictions. Precision and Recall evaluate recommendation lists.

NDCG: Position-Aware Ranking Quality

Precision and Recall do not distinguish whether relevant items appear near the top or bottom of the recommendation list.

Discounted Cumulative Gain

$$DCG@K = \sum_{j=1}^{K} \frac{2^{rel_j} - 1}{\log_2(j+1)}$$

Normalized DCG

$$\boxed{NDCG@K = \frac{DCG@K}{IDCG@K}}$$

where *IDCG@K* is the DCG of the ideal ranking.

Interpretation

- Rewards relevant items near the top
- Supports graded relevance
- Usually ranges from 0 to 1
- Higher values indicate better ranking

Worked Top-K Evaluation

Given Information
Recommended top five:

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

Relevant items:

$$\{A, C, F\}$$

Relevant recommended items:

$$\{A, C\}$$

Precision@5

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

Recall@5

$$Recall@5 = \frac{2}{3} = \boxed{0.667}$$

NDCG@5 Calculation
Relevant items occur at ranks 1 and 3:

$$DCG@5 = 1 + \frac{1}{\log_2(4)} = 1.5$$

Ideal relevant ranks are 1, 2, and 3:

$$IDCG@5 = 1 + \frac{1}{\log_2(3)} + \frac{1}{\log_2(4)}$$

$$IDCG@5 \approx 2.131$$

Therefore,

$$NDCG@5 = \frac{1.5}{2.131}$$

$$\boxed{NDCG@5 \approx 0.704}$$

Interpretation
The recommender retrieves 66.7% of the relevant items, while the NDCG score shows that the relevant items are ranked reasonably close to the top.

Recommendation Quality Beyond Accuracy

A useful recommendation system must consider more than predictive accuracy.

List Quality

- **Coverage**: catalogue/users served
- **Diversity**: variation within a list
- **Novelty**: unfamiliar items
- **Serendipity**: useful surprises
- **Freshness**: recent useful items

Online Measures

- Click-through rate
- Conversion
- Watch / completion time
- Retention
- User satisfaction

Intra-List Diversity

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

Evaluation Rule

Use chronological splits when the application predicts future behaviour.

Hybrid Recommendation

Purpose

Hybrid recommenders combine complementary signals from multiple methods.

Common Strategies

- Weighted combination
- Switching
- Cascade
- Feature augmentation
- Joint modelling

Why Combine Models?

- Reduce cold-start problems
- Improve robustness
- Use content and interaction signals
- Balance different objectives

Weighted Hybrid

$$\boxed{s(u,i) = \alpha s_{CF}(u,i) + (1-\alpha) s_{content}(u,i)}$$

Two-Stage Recommendation Architecture

Millions of Items → Candidate Retrieval → Hundreds of Items → Ranking → Top-K → Re-ranking

Candidate Retrieval

- Search huge catalogue
- Very fast
- High recall
- Produce hundreds of candidates

Ranking and Re-ranking

- Use richer features
- Improve relevance
- Add diversity / freshness
- Apply safety and business rules

Modern Recommendation Models

Method	Main Idea	Suitable For
Factorization Machines	Learn low-rank interactions among sparse features	User, item, and contextual features
Neural Collaborative Filtering	Learn nonlinear user–item interactions	Complex preference patterns
Sequential Models	Model ordered interaction history	Next-item recommendation
Transformers	Apply attention over previous interactions	Long behavioural sequences
Graph Models	Propagate signals through user–item graphs	Higher-order collaborative relations
Context-Aware Models	Include time, location, device, and session	Situation-dependent preference

Important

Higher model complexity does not automatically guarantee better recommendations. Strong baselines should always be evaluated.

Practical Challenges in Recommendation

Cold Start

New User

- Popular / diverse items
- Initial preference survey
- Contextual information

New Item

- Metadata
- Content similarity

Sparsity & Bias

- Few user interactions
- Weak neighbour similarity
- Popular items dominate
- Niche items receive less exposure

Responses

- Matrix factorization
- Side information
- Regularization
- Exploration

ment

- Fair user experience
- Fair provider exposure
- Protect user privacy
- Faithful explanations
- Monitor preference drift

Feedback Loop

Exposure $\rightarrow$ Interaction $\rightarrow$ Training Data $\rightarrow$ More Exposure

Case Study: University Course Recommendation

Inputs

- Completed courses
- Grades
- Student interests
- Career goals
- Program requirements
- Semester availability

Hard Constraints

Before ranking, remove courses that:

- have already been completed;
- require missing prerequisites;
- are not offered in the semester;
- conflict with the student's timetable;
- violate program or credit limits.

Expected Outputs

- Eligible courses
- Personalized ranking
- Relevant explanation
- Balanced workload
- Diverse academic choices

Key Lesson

Recommendation models rank **valid candidates**. A high predicted preference cannot override academic eligibility.

References

- F. Ricci, L. Rokach, and B. Shapira, eds.,
  Recommender Systems Handbook,
  Springer.
- C. C. Aggarwal,
  Recommender Systems: The Textbook,
  Springer, 2016.
- Y. Koren, R. Bell, and C. Volinsky,
  "Matrix Factorization Techniques for Recommender Systems,"
  Computer, vol. 42, no. 8, pp. 30–37, 2009.
- Y. Hu, Y. Koren, and C. Volinsky,
  "Collaborative Filtering for Implicit Feedback Datasets,"
  ICDM, 2008.
- S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme,
  "BPR: Bayesian Personalized Ranking from Implicit Feedback,"
  UAI, 2009.
- X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua,
  "Neural Collaborative Filtering,"
  WWW, 2017.