Recommender Systems
Recommender Systems (RSs)

- RSs are software tools providing suggestions for items to be of use to users, such as what items to buy, what music to listen to, or what online news to read.

- Primarily designed to evaluate the potentially overwhelming number of alternative items may offer:
  - The explosive growth & variety of information on the Web frequently lead users to make poor decisions.
  - Offer ranked lists of items by predicting the most suitable products or services based on the users’ preferences & constraints.
  - Often rely on recommendations provided by others in making routine, daily decisions, the collaborative-filtering technique.
  - Use various types of knowledge & data about users/items.
Recommender Systems

Content-based Recommender Systems:

- Try to recommend new items *similar* to those a given user has *liked* in the past
  - Identify the *common characteristics* of items being liked by user $u$ and recommend to $u$ new items that share these characteristics
  - An item $i$, which is a text document, can be represented as a feature vector $x_i$ that contains the TF-IDF weights of the most informative keywords
  - A profile of $u$, denoted profile vector $x_u$, can be obtained from the contents of items *rated* by $u$, denoted $\mathcal{R}_u$, and each item $i$ rated by $u$, denoted $r_{ui}$

\[ x_u = \sum_{i \in \mathcal{R}_u} r_{ui} x_i \]

which adds the weights of $x_i$ to $x_u$ a scalar value
Recommender Systems

- **Content-based Recommender Systems:**
  - Approach: analyze the descriptions of items previously rated by a user & build a **user profile** (to present user *interests/preferences*) based on the features of the items.
Content-Based Filtering

- Advantages:
  - User Independence: explores *solely ratings* provided by the user to build her own profile, but not other users’ (as in collaborative filtering)
  - Transparency: recommendations can be explained by explicitly listing *content features* that caused an item to be recommended
  - New Items: items *not yet rated* by any user can still be recommended, unlike collaborative recommenders which rely solely on other users’ rankings to make recommendations
Content-Based Filtering

- Shortcomings:

  - **Limited Content Analysis:** there is a natural limit in the number/types of features that can be associated with items which require *domain knowledge* (e.g., movies)

  - **Over-specialization:** tendency to produce recommendations with a limited degree of novelty, i.e., the *serendipity* problem, which restricts its usefulness in applications

  - **New User:** when few ratings are available (as for a new user), CBF cannot provide reliable recommendations
Recommender Systems

- Collaborative Filtering

Recommender Systems:

- Unlike content-based filtering approaches which use the content of items previously rated by users.
- Collaborative filtering (CF) approaches rely on the ratings of a user, and those of other users in the system.
- Intuitively, the rating of a user $u$ for a new item $i$ is likely similar to that of user $v$ if $u$ and $v$ have rated other items in a similar way.
- Likewise, $u$ is likely to rate two items $i$ and $j$ in a similar fashion, if other users have given similar ratings to $i$ & $j$.
- CF overcomes the missing content problem of the content-based filtering approach through the feedback, i.e., ratings, of other users.
Collaborative Filtering

Recommender Systems:

- Instead of relying on content, which may be a bad indicator, CF are based on the quality of items evaluated by peers.

- Unlike content-based systems, CF can recommend items with very different content, as long as other users have already shown interested for these different items.

- Goal: identify users whose preferences are similar to those a given user has liked in the past.

- Two general classes of CF methods:
  - Neighborhood-based methods
  - Model-based methods
Collaborative Filtering

- Neighborhood-based (or heuristic-based) Filtering:
  - User-item ratings stored in the system are directly used to predict ratings for new items, i.e., using either the user-based or item-based recommendation approach
    - **User-based**: evaluates the interest of a user $u$ for an item $i$ using the ratings for $i$ by other users, called neighbors, that have similar rating patterns
      - The neighbors of $u$ are typically users $v$ whose ratings on the items rated by both $u$ and $v$ are most correlated to those of $u$
    - **Item-based**: predicts the rating of $u$ for an item $i$ based on the ratings of $u$ for items similar to $i$
Neighborhood-Based Recommendation

- Example.

- Eric & Lucy have very similar tastes when it comes to movies, whereas Eric and Diane have different tastes.

- Eric likely asks Lucy the opinion on the movie “Titanic” and discards the opinion of Diane.
Collaborative Filtering

- **User-based Rating Prediction:**
  
  - Predicts the rating $r_{ui}$ of a user $u$ for a new item $i$ using the ratings given to $i$ by users most similar to $u$, called nearest-neighbors.
  
  - Given the $k$-nearest-neighbor of $u$ who have rated item $i$, denoted $N_i(u)$, the rating of $r_{ui}$ can be estimated as:

    $$r_{ui} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

  - If the neighbors of $u$ can have different levels of similarity with respect to $u$, denoted $w_{uv}$, the predicted rating is:

    $$r_{ui} = \frac{\sum_{v \in N_i(u)} w_{uv} r_{vi}}{\sum_{v \in N_i(u)} |w_{uv}|}$$
Collaborative Filtering

- Item-based Rating Prediction:
  - While *user-based* methods rely on the opinion of like-minded users, i.e., similar users, to predict a rating, item-based approaches look at ratings given to similar items.
  - **Example.** Instead of consulting with his peers, Eric considers the ratings on the movies he (& others) has (have) seen.

- Let $N_u(i)$ be the set of items rated by user $u$ most similar to item $i$, the predicted rating of $u$ for $i$ is

$$r_{ui} = \frac{\sum_{j \in N_u(i)} w_{ij} r_{uj}}{\sum_{j \in N_u(i)} |w_{ij}|}$$
Collaborative Filtering

Advantages of Neighborhood-based Filtering:

- **Simplicity**: the methods are intuitive & relatively simple to implement (w/ only the no. of neighbors requires tuning)

- **Justifiability**: the methods provide a concise & intuitive justification for the computed predictions

- **Efficiency**: the methods require no costly training phases & storing nearest neighbors of a user requires very little memory. Thus, it is scalable to millions of users & items

- **Stability**: the methods are not significantly affected by the constant addition of users, items, and ratings in a large commercial applications & do not require retraining
Evaluating Recommendation Systems

- Recommendation systems have a variety of properties, such as accuracy and robustness, that may affect user experience.

Prediction Accuracy:

- Predict user opinions (e.g., ratings) over items or the probability of usage (e.g., purchasing).

- Basic assumption: a recommender system that provides more accurate predictions will be preferred by the user.

- Measuring prediction accuracy in a user study measures the accuracy given a recommendation.

- Commonly used rating accuracy measures: Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE).
Recommender Systems

Evaluating Recommendation Systems (Continued)

Prediction Accuracy:

- **Usage Prediction Measures** (recommend items for users to use): Precision, Recall (True Positive Rate), and False Positive Rate

- 4 possible outcomes for the recommended & hidden items

<table>
<thead>
<tr>
<th></th>
<th>Recommended</th>
<th>Not Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>True-Positive (TP)</td>
<td>False-Negative (FN)</td>
</tr>
<tr>
<td>Not Used</td>
<td>False-Positive (FP)</td>
<td>True-Negative (TN)</td>
</tr>
</tbody>
</table>

- Precision = \( \frac{\#TP}{\#TP + \#FP} \)
- Recall = \( \frac{\#TP}{\#TP + \#FN} \)
- False Positive Rate = \( \frac{\#FP}{\#FP + \#TN} \)
Recommender Systems

- **Novelty Recommendations**
  - Recommending items that the user did not know about, but is not surprising, e.g., a new movie by the same director

- **An offline evaluation strategy**
  - Split the data set on time by hiding all the user ratings that occurred after a specific point in time ($ST$)
  - Hide ratings that occurred prior to $ST$
  - When recommending, the system is rewarded for each item that was recommended & rated after $ST$, but is punished for each item that was recommended prior to $ST$
Recommender Systems

- **Serendipity Recommendations**
  - Is a measure of how *surprising* the successful recommendations are
  - E.g., following a successful movie recommendation, the user learns of a new actor that (s)he likes

- **Diversity Recommendations**
  - Is generally defined as the opposite of similarity in which suggesting similar items may *not* be useful to the user
  - The most explored method for measuring diversity uses item-item similarity, typically based on item content
  - In recommenders that assist in information search, more diverse recommendations will result in *shorter* search interactions