

COMP9727: Recommender Systems

Lecture 1: Introduction

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au

This Lecture

- Application Areas for Recommender Systems
- Overview of Technical Approaches
- Recommender Systems Issues
- User Interface Design Issues for Recommender Systems
- Business Models for Recommender Systems
- Evaluation of Recommender Systems

About Me: Recommender Systems Work

- Personal Assistants
 - E-Mail Management Assistant (recommend folders)
 - Dialogue Assistant (tailor interaction to users)
 - Calendar Assistant (suggest task/activity times)
 - Multi-Agent Meeting Scheduling (suggest meeting times)
- Recommender Systems
 - Conversational recommender system for cars
 - People-to-people recommender system for online dating
 - Job recommender system based on career progression
- Event Extraction
 - Recommend events “of interest” from news, social media
 - Recommend events with high likelihood of violence

Application Areas

- News: Google, Facebook, Apple, etc.
- Social Media Feeds: Facebook, Instagram, WeChat, X, Weibo, etc.
- Music: Spotify, Apple Music, etc.
- Movies/TV/Videos: IMDb, Netflix, YouTube, TikTok, etc.
- Consumer Goods: Amazon, eBay, Taobao, JD.com, etc.
- Hotels: Booking.com, Expedia, TripAdvisor, Airbnb, etc.
- Restaurants: Yelp, Google, etc.
- Rideshare: Uber, Lyft, etc.
- People: Facebook, RSVP, OkCupid, eHarmony, etc.
- + Jobs, Meals, Video Games, Cars, Houses, Tourist Attractions, etc.

Broad Technical Approaches

- Content-Based Recommendation
 - ▶ News, Research Articles, Meals, Hotels, Restaurants, etc.
- Collaborative Filtering (CF)
 - ▶ Music, Movies/TV/Videos, Consumer Goods, etc.
- Knowledge-Based Recommender Systems
 - ▶ Cars, Houses/Apartments, Cameras, etc.
- Sequential Recommender Systems
 - ▶ e.g. recommend movies based on previous movies watched
- Context-Aware Recommender Systems
 - ▶ e.g. recommend music based on time of day, mood, emotion

Problems with Profiles

Example: News Articles

- Lack of novelty: recommended items tend to be very similar
- Echo chambers: user sees only articles with same point of view
- Lack of precision: profiles too coarse if categories are broad
- Updating profiles: user interests change over time
- Balance: short-term vs long-term interests
- Ambiguity: e.g. “Big Bang Theory”

Content-Based Recommendation

Useful when items **can** easily be categorized

- Categorize items using a predefined(?) ontology?
- Build **user profile** of interests using the same ontology
- Recommend items based on “similarity” between item and profile

Data sources: Explicit user interests, user–system interactions

Many ways to define “similarity”

Collaborative Filtering

Useful when items **cannot** easily be categorized

- Item-based: Recommend items “similar” to those consumed
 - ▶ Amazon: Items are “similar” if users bought them together
 - ▶ Users who bought **this** also bought **that**
- User-based: Recommend items consumed by “similar” users
 - ▶ Users are “similar” if they rated the same items similarly
 - ▶ You might like **this other item liked by those users**

Data sources: Explicit ratings, user–system interactions

Many ways to define “similarity” of users/items

Problems with (Movie) Ratings

- Meaning of scale changes over time: “uberization”
- Individual differences in rating interpretation
- Sparsity: users rate few movies so few ratings overall
- Biased towards popular movies: many ratings
- Biased against “bad” movies: users don’t rate them
- New movies rated more highly: novelty factor!
- New users rate movies more highly: inexperience?
- Biased towards frequent raters: e.g. teenagers?
- Biased towards English language movies: more users
- Astroturfing: Fake reviews and ratings

Knowledge-Based Recommender Systems

Useful when users don’t know exactly what they want

- Also known as “conversational” recommender systems
- Users cannot articulate their needs as a search query
- Suitable for casual and naive users, for infrequent purchases
- Start with a single example then provide “critiques”
- Helps users navigate a complex search space
- Helps users understand tradeoffs in a complex domain
- Users learn their preferences [during](#) search
- Functions as a decision-support tool
- Based on consumer buying behaviour models

Recommender Systems Issues

- Cold start problems: users and items
- Over-recommendation of popular items
- Users with obscure tastes
- The “long tail”: 80/20 rule
- Diversity of recommendations
- Serendipity: discovery of [genuinely](#) new items
- Transparency of recommendations(?)
- Explainability of recommendations(?)
- Legal regulations: e.g. financial, medical advice
- Polarization: users only see extremes of an argument

User Modelling and Personalization

- How to model users: “user profile”
- How to get data about users: clicks, likes, purchases, etc.
- How reliable is the data: e.g. watching vs rating a movie
- How to model changes in user interests: dynamics
- How to balance long-term and short-term interests
- Basis of recommendation: profile, session, etc.
- Can users be “clustered” into groups?
- Ethical/legal concern of data privacy

Let the Data Speak for Itself ...

Yum!

"When we've done analysis at Taco Bell, every time we shift sales to digital, we see an increase in frequency and check size. If we were to go from somebody just using our app casually to being a member of our loyalty program, we can control for other factors around the normal usage. We get richer data and an even better connection. This is why our new menu items, the only way to order is if you're on the app and a member of our loyalty program."

2024

CURRENT YUM BRANDS CEO DAVID GIBBS

MODERN MBA

Master of Business Administration

User Interface Design Issues

- How many recommendations to give at one time
- How and where on the screen to display them
- How to describe the recommendations: “suggestions”?
- How to get user feedback (non-intrusively)
- How to get useful feedback (especially dislikes)
- How to use user feedback (in real-time?)
- When to give (and change) recommendations
- How not to annoy the user so they will come back

But What is It Saying?

Yum!

"You can have all the data in the world but if you either don't have a hypothesis or you don't have any insight, then it is absolutely useless. And because everyone's pretty much got access to the same data, it comes back to who has the best instinct of what that data is telling them. That is what I fundamentally believe in."

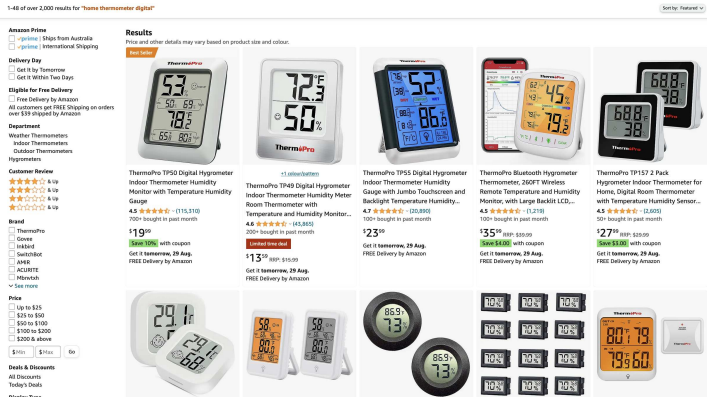
2021

FORMER YUM BRANDS & TACO BELL CEO GREG CREED

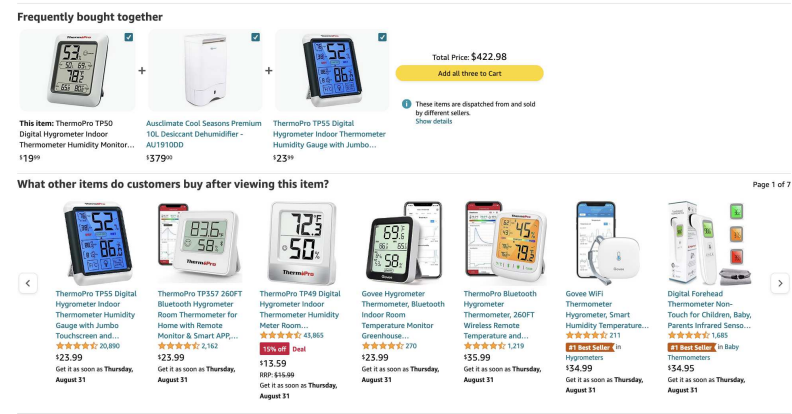
MODERN MBA

Master of Business Administration

Amazon Search Results



Amazon Recommendations

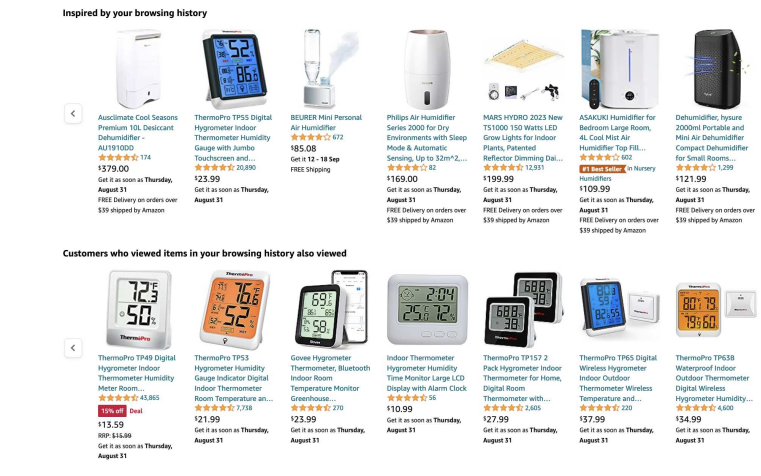


Business Models

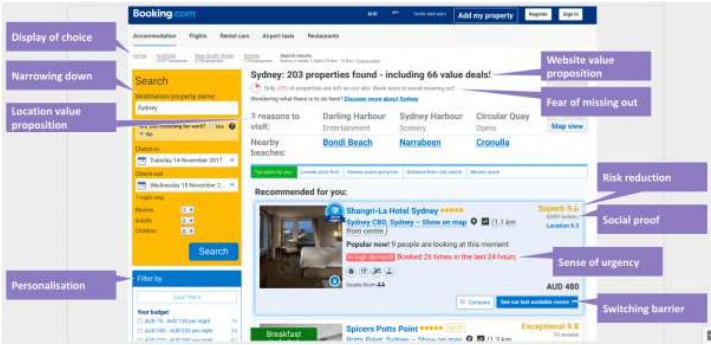
Examples: Booking.com, Expedia, TripAdvisor

- Booking.com (US\$112 Billion)
 - ▶ Commission model – Cut from actual bookings
- Expedia (US\$15 Billion)
 - ▶ Merchant model – Buy rooms in bulk and resell
- TripAdvisor (US\$2 Billion)
 - ▶ Click-through advertising – Paid per view (cents per mille)

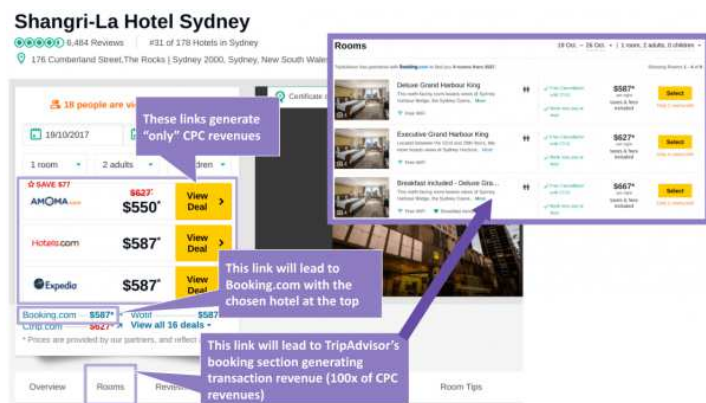
Amazon Recommendations



Booking.com

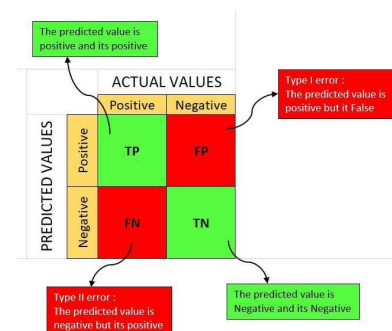


TripAdvisor



Metrics: Precision and Recall

When you have exactly **one** positive class (e.g. movies user likes)



$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}) = \text{TP}/(\text{Predicted Positive})$$

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) = \text{TP}/(\text{Actual Positive})$$

Evaluation of Recommender Systems

- Historical data analysis
 - ▶ Often this is all we have ... until deployment
 - ▶ Predict user actions (but **without** the recommender)
- User studies
 - ▶ Informal: 5–10 people for feedback (but often just on the UI)
 - ▶ Formal: A/B testing to compare different versions of the system

Metrics: Precision and Recall

Still with exactly **one** positive class

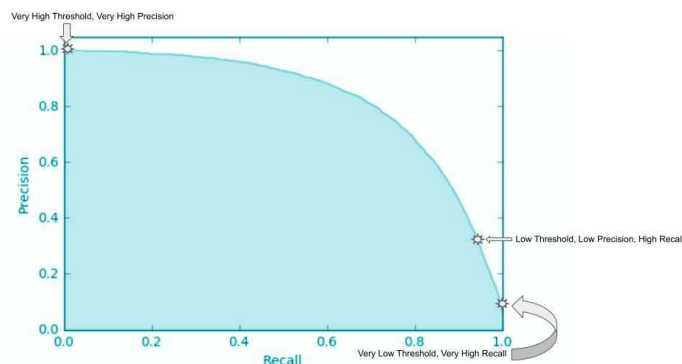
- Precision (P) = $\text{TP}/(\text{TP}+\text{FP})$ – you want what you get
- Recall (R) = $\text{TP}/(\text{TP}+\text{FN})$ – you get what you want
- $F1 = 2\text{PR}/(\text{P}+\text{R})$ – harmonic mean of precision and recall
- Accuracy = $(\text{TP}+\text{TN})/(\text{Whole Dataset})$

Suppose recommendations are ranked and look only at top-N

- Precision@N = $(\text{TP in top-N})/N$ – precision over top-N
- Recall@N = $(\text{TP in top-N})/(\text{Actual Positive})$ – this will be small

Precision-Recall Curve

Still with exactly **one** positive class



Tradeoff between precision and recall; suitable when classes **imbalanced**

Multiple Classes: Per-Class Metrics

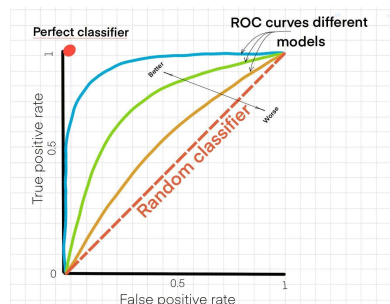
$n \times n$ Confusion Matrix (each instance in one class)

	Predicted c_1	Predicted c_2	...
Class c_1	c_{11}	c_{12}	c_{13}
Class c_2	c_{21}	c_{22}	c_{23}
...	c_{31}	c_{32}	c_{33}

- Precision (class c_i) = $c_{ii} / \sum_j c_{ji}$ – add down column
 - Proportion of items predicted as c_i correctly classified (as c_i)
- Recall (class c_i) = $c_{ii} / \sum_j c_{ij}$ – add along row
 - Proportion of items in class c_i predicted correctly (as c_i)
- Accuracy = $\sum_i c_{ii} / \sum_i \sum_j c_{ij}$ – add all items in table

Receiver Operating Characteristic Curve

Still with exactly **one** positive class



True Positive Rate = Recall
False Positive Rate = FP/(Negatives)
AUC = Area Under the Curve

Tradeoff between TP and FN; suitable when classes **balanced**

Multiple Classes: Micro/Macro-Averaging

n (one per class) 2×2 Contingency Tables

- Micro-average = Aggregated measure over all classes
 - micro-precision = $\sum_c TP_c / \sum_c (TP_c + FP_c)$
 - micro-recall = $\sum_c TP_c / \sum_c (TP_c + FN_c)$
 - Same when each instance has and is given one and only one label
 - Dominated by larger classes
- Macro-average = Average of per-class measures
 - macro-precision = $\frac{1}{n} \sum_c TP_c / (TP_c + FP_c)$
 - macro-recall = $\frac{1}{n} \sum_c TP_c / (TP_c + FN_c)$
 - Dominated by smaller classes
 - Useful for imbalanced classes, e.g. sentiment analysis

Metrics for Ranked Results

Suppose the ground truth data is **ranked** (e.g. user's rated movies) and each item i has a **relevance** $rel(i)$ – e.g. the rating

Let the recommender return the **ranked** itemlist r_1, r_2, \dots

- Cumulative Gain $CG@N = \sum_{i=1}^N rel(r_i)$
- Discounted Cumulative Gain $DCG@N = \sum_{i=1}^N rel(r_i) / \log_2(i+1)$
- Normalized DCG $nDCG@N = DCG@N / (\text{max possible } DCG@N)$
- Metrics for the most relevant item (hit rate, mean reciprocal rank)

Measures of closeness of ranking to ground truth relevance over top-N

Metrics for Regression and Estimation

Suppose the ground truth data is **numeric** (e.g. user's rated movies)

Let the recommender return an **estimate** \hat{r} for each item r (N items)

- Mean Absolute Error (MAE) $= \frac{1}{N} \sum |\hat{r} - r|$
- Mean Squared Error (MSE) $= \frac{1}{N} \sum (\hat{r} - r)^2$
- Root Mean Squared Error (RMSE) $= \sqrt{\frac{1}{N} \sum (\hat{r} - r)^2}$

Measures closeness of scores to ground truth scores

Metrics for Ranked Results

Suppose the ground truth data is **fully ranked** (e.g. user's rated movies) and each item i has a **relevance** $rel(i)$ – e.g. the rating

Let the recommender return the **fully ranked** itemlist r_1, r_2, \dots

- Item pair (i, j) is **concordant** if $rel(i) > rel(j)$
- Item pair (i, j) is **disconcordant** if $rel(i) < rel(j)$
- With N items, there are $N*(N-1)/2$ pairs
- Kendall $\tau_\beta = (\text{Concordant} - \text{Discordant}) / (\text{Number Item Pairs})$
- Similar metrics for when rankings include “ties”

Measures of correlation (–1 to 1) of ranking to ground truth ranking

Metrics for Recommender Systems

- Click-through rate
- Conversion rate
- User engagement
- Sales and revenue
- Average order value
- Customer lifetime value
- Customer retention
- Churn rate
- Coverage of item set
- Diversity: average dissimilarity of items in result sets

Problems with Metrics

- Averages don't capture variation between users
- Metrics dominated by subgroup of highly active users
- Recommender needs to work for **all** users
- Need to focus on top-N metrics, but for which N?
- Need tradeoffs between multiple metrics
- Metrics tend to ignore infrequently rated long tail
- Metrics unduly influenced by outliers (some more so)
- Balance short-term revenue and long-term user experience

Course Topics

1	Introduction to Recommender Systems
2	Content-Based Recommender Systems
3	Collaborative Filtering
4	Knowledge-Based Recommender Systems
5	Social Recommender Systems
6	Social Network Recommendation
7	Sequential Recommender Systems
8	Contextual Recommender Systems
9	LLM for Recommender Systems

Evaluation Pitfalls

- Historical data analysis insufficient
 - ▶ Recommender changes behaviour: users **will** click differently
 - ▶ Model could be learning how the old recommender system works
 - ▶ User tastes unclear as no explicit feedback (especially negative)
 - ▶ Dynamics of users and items difficult to capture
 - ▶ Need post-deployment evaluation to confirm analysis
- Commercial feasibility
 - ▶ Gap between research and deployment environments
 - ▶ Integration with commercial systems

Assessment

- Assignment
 - ▶ Week 4: Content-Based Recommender System (30%)
- Project
 - ▶ Project Pitch and Design
 - Week 5: Project Pitch (No marks)
 - Week 5: Project Design Document (20%)
 - ▶ Project Implementation and Evaluation
 - Week 10: Team Project Demonstration (30%)
 - Week 11: Individual Project Report (20%)