COMP9727: Recommender Systems

Lecture 7: Sequential Recommendation

Wayne Wobcke

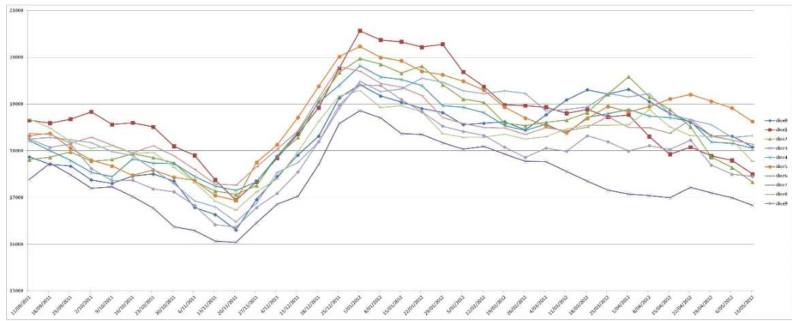
e-mail:w.wobcke@unsw.edu.au

This Lecture

Recommendation using temporal features, of/using temporal sequences

- Temporal Recommendation
  - Life Stages (Banking) and Life Cycles (Dating)
- Application Areas
  - Job Recommendation (Career Progression)
  - Session-Based Recommendation (E-Commerce, Video)
  - Temporal Aspects of Ratings (Movies)
- Techniques
  - (Hidden) Markov Models
  - Sequential rule/frequent pattern mining
  - Asymmetric SVD $\rightarrow$ timeSVD++

Seasonal Effects



Should recommender be different at different times of year?

Does user behaviour change depending on life cycle?

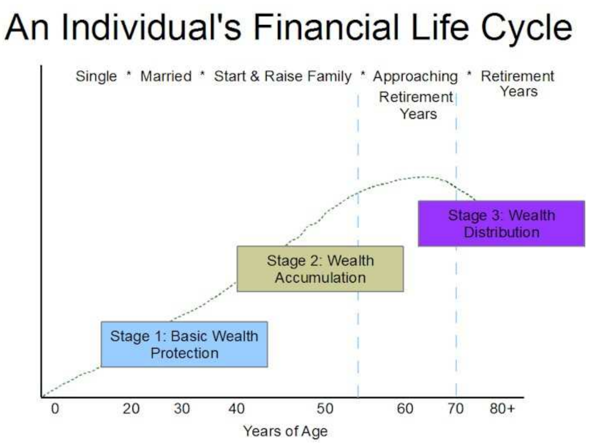
... in addition to temporal features such as “popularity”

Pre- and Post-Filtering

- Time – or time period – as context feature
  - e.g. season, time of day, special days (Halloween, Thanksgiving)
- Pre-filtering
  - Learn model using only context-relevant data (more sparse)
- Post-filtering
  - Generate candidates as normal, then filter/reweight by context
  - Multiply ratings by $P(u,i,C)$ found using content-based methods
  - Learn $P(u,i,C)$ from all users, e.g. $P(\text{comedy}|\text{weekend})$
  - ... or from just the similar users to u, depending on C

Life Stages: E-Commerce Example

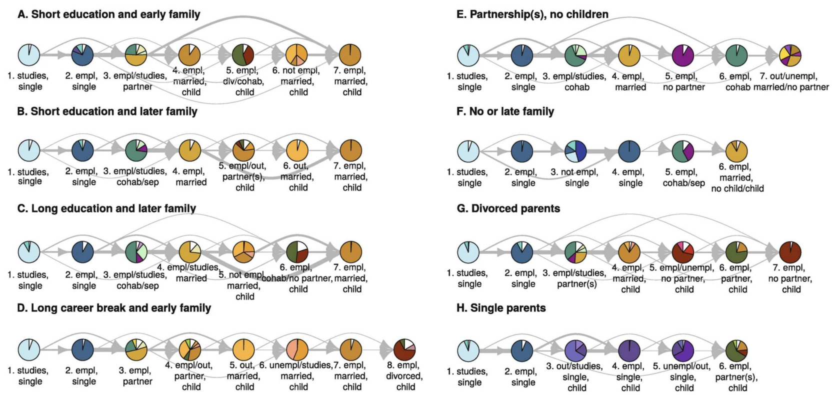
Financial Life Stages



Recommendation Based on Life Stages

Age	up to 17 years	18 - 26 years	27 - 35 years	36 - 45 years	46 - 54 years	55 - 64 years	65 years and more
Life Stage	Childhood	Career Start	Family Creation	Asset Building	Asset Protection	Late Career	Retirement
Special Events		Driving License	Stable Job	Rise in salary	Child Matures	Children start Career	Retirement
		Student	Marriage	Second child	Education for Child	Home Sales	Birth Grand-child
		First Income	Birth of a child	Divorce	Inherit Money	Mortgage paid	
Bank Focus		Car Loan	Insurance	Investment Program	Investment Program	Investment Program	
		Debit Card	Credit Card	Insurance for Children	Time Deposit	Time Deposit	
		First Saving Account	Personal Loan	Personal Loan	Real Estate	Real Estate	

Life Stage Clusters



Job Recommendation

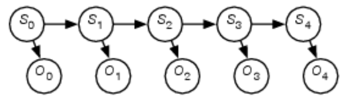
Current Job Title	Job Transition Title
software developer	senior, programmer, analyst, analytics, developer
software	engineer, developer, senior, technical, specialist
developer	engineer, senior, software, specialist, ios

- Recommend jobs with "similar" titles to job transition terms
  - e.g. software developer ⇒ senior analyst
- Also look at terms in advertised job descriptions
  - e.g. business analyst ⇒ project manager

Markov Chain

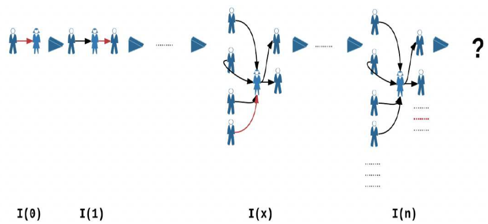
- Bayesian network
  - $P(S_0)$ specifies initial conditions
  - $P(S_{i+1}|S_i)$ specifies dynamics (stationary if same for each i)
- Independence assumptions
  - $P(S_{i+1}|S_0,\cdots,S_i) = P(S_{i+1}|S_i)$
  - Transition probabilities dependent only on current state S_i – independent of history to reach that state $S_0,\cdots,S_{i-1}$
  - The future is independent of the past, given the present

Hidden Markov Models



- Bayesian network
  - $P(S_0)$ specifies initial conditions
  - $P(S_{i+1}|S_i)$ specifies dynamics
  - $P(O_i|S_i)$ specifies "observations"
- Independence Assumptions
  - $P(S_{i+1}|S_0,\cdots,S_i) = P(S_{i+1}|S_i)$ (Markov Chain)
  - $P(O_i|S_0,\cdots,S_{i-1},S_i,O_0,\cdots,O_{i-1}) = P(O_i|S_i)$
  - Observations (actions) depend only on current state

Hidden Markov Model for Online Dating

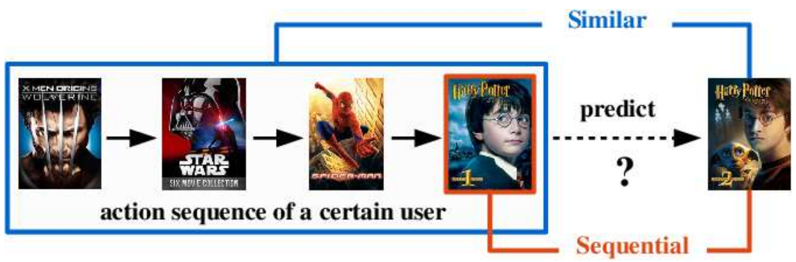


- Each output (message) has user and derived features
- Markov model is left-right topology
- Model states are n previous messages of recipient
- Choose number of states empirically (e.g. 5 or 6)
- Transitions learnt using Baum-Welch algorithm

Session-Based Recommendation

- Users interact with site in a "session"
  - Page views, favourite, add to cart, purchase, etc.
  - Task is to predict next (useful) interaction
  - Assuming temporal ordering is significant
- Use sequential rule/frequent pattern mining
  - Extends item-based CF and association rule mining
  - Requires a lot of data to get useful recommendations
- Use neighbourhood CF methods with similarity over sequences
  - Jaccard or cosine similarity over action sets in the sequence
  - Or vectors of actions weighted according to their recency

Similarity vs Sequential Recommendation



Association Rule Mining

Given a list of transactions ("market baskets")

- Itemset: Any set of items
- Frequent itemset: Itemset occurs in many baskets (fixed threshold)
- Association Rule: If $I \subseteq$ itemset then $J \subseteq$ itemset
- Support(I): Fraction of all transactions containing itemset I
- Confidence: $\text{Support}(I \cup J)/\text{Support}(I) \approx P(J|I)$

Note: Rule can have high confidence, but low support

Sequential Pattern Mining (Simplified)

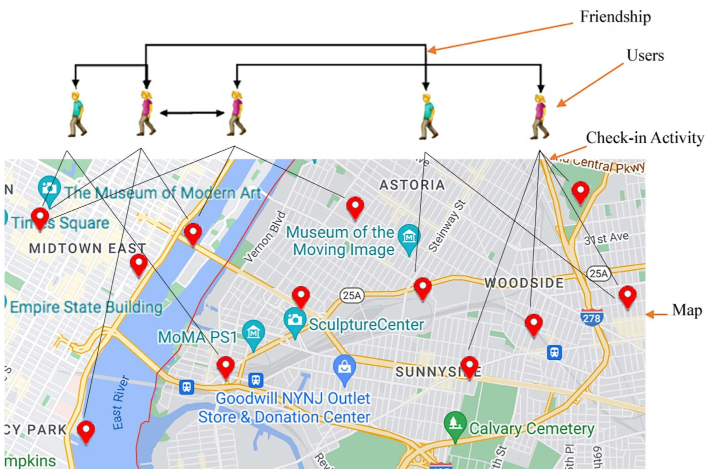
Given a list of ordered transactions ("market baskets")

- Itemseq: Any sequence of items
- Frequent itemseq: Itemseq occurs in many baskets (may be gaps)
- Association Rule: If $I \sqsubseteq$ itemseq then $J \sqsubseteq$ itemseq
- Support(I): Fraction of all transactions containing itemseq I
- Confidence: $\text{Support}(I \cup J)/\text{Support}(I) \approx P(J|I)$

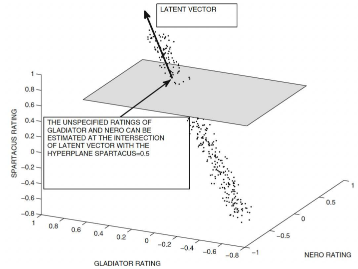
Note: Original definition allows sequences of itemsets (not items)

Point-of-Interest Recommendation

Point-of-Interest CF Recommendation



Matrix Factorization (MF)



- Ratings of (some) movies are highly correlated
- Ratings matrix $m \times n$ is approximately rank $k \ll n$
- Approximate the matrix R using k latent factors
  - $R = UV^T$ where U is $m \times k$, V is $n \times k$

Latent Factors

R

		NERO	JULIUS CAESAR	CLEOPATRA	SLEEPLESS IN SEATTLE	PRETTY WOMAN	CASABLANCA
HISTORY	1	1	1	1	0	0	0
	2	1	1	1	0	0	0
	3	1	1	1	0	0	0
BOTH	4	1	1	1	1	1	1
ROMANCE	5	-1	-1	-1	1	1	1
	6	-1	-1	1	1	1	1
	7	-1	-1	-1	1	1	1

$\approx$

U

	HISTORY	ROMANCE
1	1	0
2	1	0
3	1	0
4	1	1
5	-1	1
6	-1	1
7	-1	1

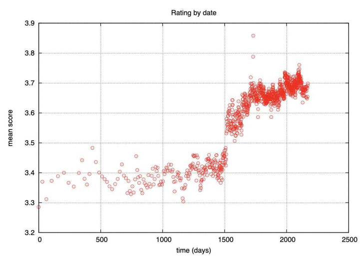
X

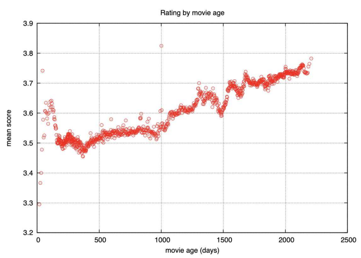
V^T

	NERO	JULIUS CAESAR	CLEOPATRA	SLEEPLESS IN SEATTLE	PRETTY WOMAN	CASABLANCA
HISTORY	1	1	1	0	0	0
ROMANCE	0	0	1	1	1	1

- u_{ik} is how much u's tastes are like k; v_{jk} is how much v_j is like k
- $\hat{r}_{ij} = u_i \cdot v_j$ = sum of $u_{ik}.v_{jk}$ over all the factors k
- Factors are not always interpretable like this!

Temporal Dynamics of Movie Ratings





- Sudden jump in average movie rating in early 2004 (day 1500)
- Ratings tend to increase with movie age at time of rating
- Ratings stable over short periods (around 10 weeks)

Asymmetric Matrix Factorization

- Normally ratings matrix $R = UV^T$ where U is $m \times k$, V is $n \times k$
- Incorporate "implicit feedback" matrix F
  - Each row of F non-zero only if item rated by user, normalized
  - Force decomposition to look only at each user's rated items
  - Factorize $R = [FY]V^T$ where F is $m \times n$, Y is $n \times k$
  - User factors are linear combinations of implicit item factors

$$\underbrace{\begin{pmatrix} 1 & -1 & 1 & ? & 1 & 2 \\ ? & ? & -2 & ? & -1 & ? \\ 0 & ? & ? & ? & ? & ? \\ -1 & 2 & -2 & ? & ? & ? \end{pmatrix}}_{R} \Rightarrow \underbrace{\begin{pmatrix} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 0 & 1/\sqrt{5} & 1/\sqrt{5} \\ 0 & 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 1/\sqrt{1} & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 & 0 & 0 \end{pmatrix}}_{F}$$

timeSVD++

Not really Singular Value Decomposition

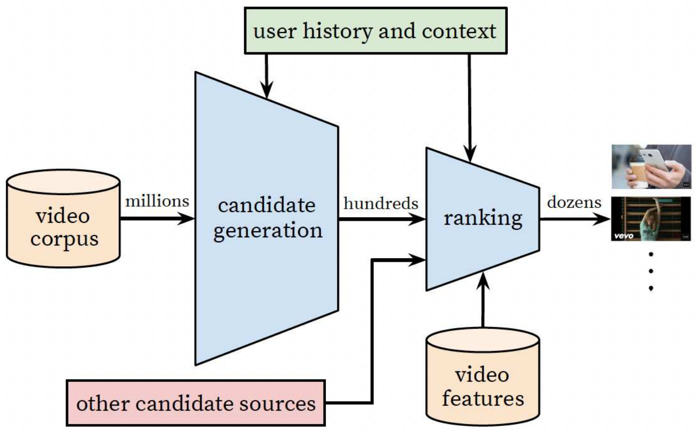
1. Centre R on global mean, estimate r_{ij} as $o_i + p_j + u_i \cdot v_j$
   - where o_i is user bias, p_j is item bias
2. $R = (U + FY)V^T$ where U is $m \times k$, F is $m \times n$, Y is $n \times k$, V is $n \times k$
   - FY adjusts U to add the "feedback": but now more parameters
   - Estimate $\hat{r}_{ij}$ as $o_i + p_j + \sum_k (u_{ik} + \sum_{h \in I_i} \frac{y_{hk}}{\sqrt{|I_i|}}) \cdot v_{jk}$
     - where I_i is the set of items rated by user i
3. Make o_i, p_j, u_{ik} functions of time t
   - Estimate $\hat{r}_{ij}(t)$ as $o_i(t) + p_j(t) + \sum_k (u_{ik}(t) + \sum_{h \in I_i} \frac{y_{hk}}{\sqrt{|I_i|}}) \cdot v_{jk}$

timeSVD++ Optimization Problem

Estimate $\hat{r}_{ij}(t)$ as $o_i(t) + p_j(t) + \sum_k (u_{ik}(t) + \sum_{h \in I_i} \frac{y_{hk}}{\sqrt{|I_i|}}) \cdot v_{jk}$

- Item bias $p_j(t)$
  - Time bins around 10 weeks: $p_j(t) = C_j + \text{offset}_{j,Bin(t)}$
- User bias $o_i(t)$ with mean rating time ν_i
  - $o_i(t) = K_i + \alpha_i.\text{dev}_i(t) + e'_{it}$ where $\text{dev}_i(t) = sign(t - \nu_i).|t - \nu_i|^\beta$ where $\text{dev}_i(t)$ is the deviation of t from ν_i and β is learnt (≈ 0.4)
- User factors $y_{ik}(t)$
  - $y_{ik}(t) = K'_{ik} + \alpha'_{ik}.\text{dev}_i(t) + e'_{ikt}$ similarly
- Solve by Stochastic Gradient Descent

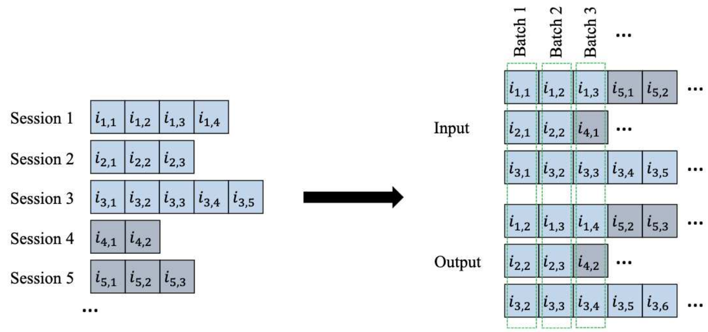
YouTube Recommender (2016)



YouTube Recommender (2016)

- Candidate generation
  - Neural network for averaged “video watches” embeddings
  - Neural network for averaged “search token” embeddings
  - Trained similarly to word2vec, with additional context features
  - Approximate nearest neighbour for candidates
- Candidate ranking
  - Embedding of sequences of watched videos/impressions
  - Extra features (time since last watch, previous impressions)
  - Considerable hand-crafting of features

Sequence Training in Neural Networks



Summary

- Many aspects to temporal/sequential recommendation
  - Time and time period as context – seasonal effects
  - Time sensitivity of recommendations, life stages
  - Life cycle of user interactions with a site
  - Relevance of temporal order of inputs to model
  - Changes in ratings over time due to novelty decay
  - Changes in meaning of ratings over time (concept drift)
  - Evolution of user’s taste over time
- Effectiveness of methods depends heavily on availability of data
- Overlap with next lecture on contextual recommendation