CS276B

Web Search and Mining
Winter 2005

Lecture 6

Recap: Recommendation Systems

- What they are and what they do?
- A couple of algorithms
  - Classical *Collaborative Filtering* (CF): Nearest neighbor-based approaches
- Going beyond simple behavior: context
- How do you measure their quality?

Implementation

- We worked in terms of matrices, but
- Don't really want to maintain this gigantic (and sparse) vector space
  - Dimension reduction
  - Fast nearest neighbors
- Incremental versions
  - update as new transactions arrive
  - typically done in batch mode
  - incremental dimension reduction etc.

Plan for Today

- Issues related to last time
  - Extensions
  - Privacy
- Model-based RS approaches
    - Learn model from database, and make predictions from model rather than iterating over users each time
  - Utility formulation
    - Matrix reconstruction for low-rank matrices
  - Model-based probabilistic formulations
- Evaluation and a modified NN formulation

Extensions

- Amazon - "Why was I recommended this"
  - See where the "evidence" came from
- Clickstreams - do sequences matter?
- HMMs (next IE lecture) can be used to infer user type from browse sequence
  - E.g., how likely is the user to make a purchase?
  - Meager improvement in using sequence relative to looking only at last page

Privacy

- What info does a recommendation leak?
  - E.g., you're looking for illicit content and it shows me as an expert
- What about compositions of recommendations?
- "These films are popular among your colleagues"
- "People who bought this book in your dept also bought … "
  - "Aggregates" are not good enough
- Poorly understood

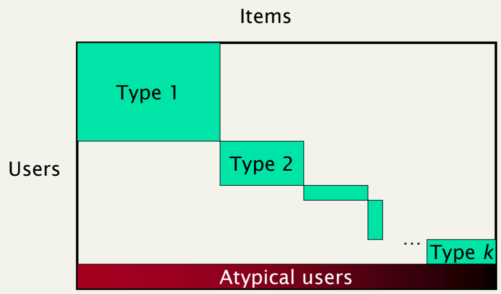
Utility formulation of RS

- Microeconomic view
- Assume that each user has a real-valued *utility* for each item
- $m \times n$ matrix U of utilities for each of m users for each of n items
  - not all utilities known in advance
- Predict which (unseen) utilities are highest for each user

User types

- If users are arbitrary, all bets are off
  - typically, assume matrix U is of low rank
  - say, a constant k independent of m,n
  - some perturbation is allowable
- I.e., users belong to k well-separated types
  - (almost)
  - Most users' utility vectors are close to one of k well-separated vectors

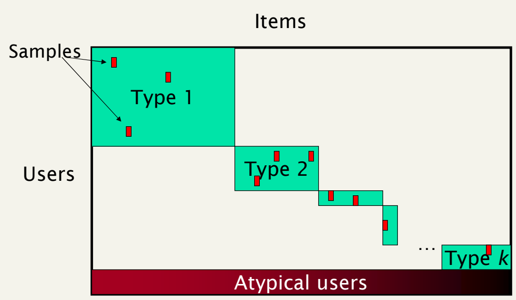
Intuitive picture (exaggerated)



Matrix reconstruction

- Given some utilities from the matrix
- Reconstruct missing entries
  - Suffices to predict biggest missing entries for each user
  - Suffices to predict (close to) the biggest
  - For most users
    - Not the atypical ones

Intuitive picture



Matrix reconstruction: Achlioptas/McSherry

- Let $\hat{U}$ be obtained from U by the following sampling: for each i,j
  - $\hat{U}_{ij} = U_{ij}$, with probability $1/s$,
  - $\hat{U}_{ij} = 0$ with probability $1-1/s$.
- The sampling parameter s has some technical conditions, but think of it as a constant like 100.
- Interpretation: $\hat{U}$ is the sample of user utilities that we've managed to get our hands on
  - From past transactions
  - (that's a lot of samples)

How do we reconstruct U from $\hat{U}$?

- First the "succinct" way
  - then the (equivalent) intuition
- Find the best rank k approximation to $s\hat{U}$
  - Use SVD (best by what measure?)
  - Call this $\hat{U}_k$
- Output $\hat{U}_k$ as the reconstruction of U
  - Pick off top elements of each row as recommendations, etc.

Achlioptas/McSherry theorem

- With high probability, reconstruction error is small
  - see paper for detailed statement
- What's high probability?
  - Over the *samples*
  - not the matrix entries
- What's error – how do you measure it?

Norms of matrices

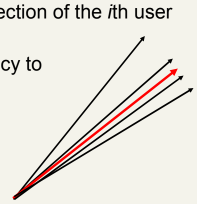
- Frobenius norm of a matrix M:
  - $|M|_F^2$ = sum of the square of the entries of M
- Let M_k be the rank k approximation computed by the SVD
- Then for any other rank k matrix X, we know
  - $|M- M_k|_F \leq |M-X|_F$
- Thus, the SVD gives the best rank k approximation for each k

Norms of matrices

- The L_2 norm is defined as
  - $|M|_2$ = max $|Mx|$, taken over all unit vectors x
- Then for any other rank k matrix X, we know
  - $|M- M_k|_2 \leq |M-X|_2$
- Thus, the SVD also gives the best rank k approximation by the L_2 norm
- What is it doing in the process?
  - Will avoid using the language of eigenvectors and eigenvalues

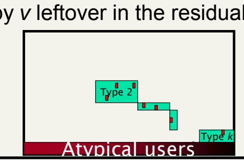
What is the SVD doing?

- Consider the vector v defining the L_2 norm of U:
  - $|U|_2 = |Uv|$
- Then v measures the "dominant vector direction" amongst the rows of U (i.e., users)
  - ith coordinate of Uv is the projection of the ith user onto v
  - $|U|_2 = |Uv|$ captures the tendency to align with v



What is the SVD doing, contd.

- U_1 (the rank 1 approximation to U) is given by Uvv^T
- If all rows of U are collinear, i.e., rank(U)=1, then $U= U_1$;
  - the error of approximating U by U_1 is zero
- In general of course there are still user types not captured by v leftover in the residual matrix $U-U_1$:



Iterating to get other user types

- Now repeat the above process with the residual matrix U-U_1
- Find the dominant user type in U-U_1 etc.
  - Gives us a second user type etc.
- Iterating, get successive approximations U_2, U_3, … U_k

Achlioptas/McSherry again

- SVD of $\hat{U}$: the uniformly sampled version of U
- Find the rank k SVD of $\hat{U}$
- The result $\hat{U}_k$ is close to the best rank k approximation to U
- Is it reasonable to sample uniformly?
  - Probably not
  - E.g., unlikely to know much about your fragrance preferences if you're a sports fan

Probabilistic Model-based RS

Breese et al. *UAI* 1998

- Similar to Achlioptas/McSherry but probabilistic:
  - Assume a latent set of k classes, never observed
  - These generate observed votes as a Naïve Bayes model (recall cs276a)
  - Learn a best model using the EM algorithm
- Bayesian Network model
  - Learn probabilistic decision trees for predicting liking each item based on liking other items
- They concluded that in many (but not all!) circumstances, Bayesian DT model works best

McLaughlin & Herlocker 2004

- Argues that current well-known algorithms give poor user experience
- Nearest neighbor algorithms are the most frequently cited and the most widely implemented CF algorithms, consistently are rated the top performing algorithms in a variety of publications
- But many of their top recommendations are terrible
- These algorithms perform poorly where it matters most in user recommendations
- Concealed because past evaluation mainly on offline datasets not real users

Novelty versus Trust

- There is a trade-off
  - High confidence recommendations
    - Recommendations are obvious
    - Low utility for user
    - However, they build trust
      - Users like to see some recommendations that they know are right
  - Recommendations with high prediction yet lower confidence
    - Higher variability of error
    - Higher novelty → higher utility for user
      - McLaughlin and Herlocker argue that "very obscure" recommendations are often bad (e.g., hard to obtain)

Common Prediction Accuracy Metric

- Mean absolute error (MAE)

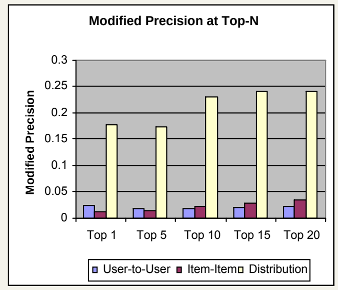
$$|\overline{E}| = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N}$$

- Most common metric
- Characteristics
  - Assumes errors at all levels in the ranking have equal weight
  - Sensitive to small changes
  - Good for "Annotate in Context" task
  - Seems not appropriate for "Find Good Items" task

McLaughlin & Herlocker 2004

- Limitations of the MAE metric have concealed the flaws of previous algorithms (it looks at all predictions not just top predictions)
- Precision of top *k* has wrongly been done on top *k* rated movies.
  - Instead, treat not-rated as disliked (underestimate)
    - Captures that people pre-filter movies
- They propose a NN algorithm where each user gives a movie a rating distribution, not a single rating, which is smoothed with a uniform rating
  - Movie recommendation must have enough evidence to overcome uniform rating

Rsults from SIGIR 2004 Paper



- Much better predicts top movies
- Cost is that it tends to often predict blockbuster movies
- A serendipity/ trust trade-off

Resources

- Achlioptas McSherry *STOC* 2001
  - http://portal.acm.org/citation.cfm?id=380858
- Breese et al. *UAI* 1998
  - http://research.microsoft.com/users/breese/cfalgs.html
- McLaughlin and Herlocker, SIGIR 2004
  - **http://portal.acm.org/citation.cfm?doid=1009050**