Recommender Systems: Advanced Concepts in Research and Practice

Joseph A. Konstan
University of Minnesota

konstan@cs.umn.edu
http://www.grouplens.org

A Bit of History

- Ants, Cavemen, and Early Recommender Systems
  - The emergence of critics
- Information Filtering and User Modeling
- Manual Collaborative Filtering
- Automated Collaborative Filtering
  - Social Navigation and other approaches
- The Commercial Era

Historical Challenges

- Collecting Opinion and Experience Data
- Finding the Relevant Data for a Purpose
- Presenting the Data in a Useful Way

Introductions

- Me
- You
- This tutorial

About Me

- Professor of Computer Science
  - University of Minnesota
- Background: Human-Computer Interaction
- Recommender Systems Experience
  - Started on GroupLens project in late 1994
  - Co-founded Net Perceptions
  - Still actively working on RS research

About You

- Name
- What you do
- Who you work for / where you study
- Briefly
  - Your experience with recommender systems
  - One key thing you want to get out of this tutorial
About This Tutorial

Background
- I have taught an introductory tutorial on RS about a dozen times (often with John Riedl, Anthony Jameson)

The idea:
- A venue to go beyond the introductory material to explore newer knowledge and current problems
- Inherently biased by what I find to be interesting (for better or for worse)
- Assumes basic understanding of RS (e.g., k-nearest-neighbor collaborative filtering – if not, let’s fix that!)

Goals of this Tutorial

- To understand the state of research and practice in recommender systems:
  - Algorithms
  - Interface Design
  - Evaluation
- To explore the future of user-centered recommender system design
- To have fun while doing so!

Where do Recommenders Fail?

Your experiences
- Where have recommenders really missed the mark?
- Where have they looked dumb?
- And why????

My Examples

- Recommending without enough data (weak confidence)
- Recommending items ignorant of context
- Ignoring overall interest or balance
- Just plain wrong!

Amazon Explanation

Amazon.com Tolkien
More Generally …

- Recommenders fail when …
  - they lack awareness of their own knowledge
  - they don’t consider the context of recommendation
  - they don’t consider the user's goals and needs
How do we Evaluate Recommenders -- today?

- Industry outcome
  - Add-on sales
  - Click-through rates

Real-world Experience

- Large international catalog retailer
  - 17% hit rate, 23% acceptance rate in call center
- Medium European outbound call center
  - 17% hit rate, 6.7% acceptance rate from an outbound telemarketing call
  - $350.00 price of average item sold
  - Items were in an electronics over-stocked category and were sold-out within 3 weeks
- Medium American online toy store (e-mail campaign)
  - 19% click-thru rate vs. 10% industry average
  - 14.3% conversion to sale vs. 2.5% industry average

How do we Evaluate Recommenders -- today?

- Industry outcome
  - Add-on sales
  - Click-through rates
- Research measures
  - User satisfaction
- Metrics
  - To anticipate the above beforehand (offline)

Evaluating Recommendations

- Prediction Accuracy
  - MAE, MSE,
- Decision-Support Accuracy
  - Reversals, ROC
- Recommendation Quality
  - Top-n measures (e.g., Breese score)
- Item-Set Coverage

What’s Wrong with This Approach?

- What is the purpose of recommenders?
  - to help people find things they don’t already know – and that they’ll like/value/use
  - to serve as a useful advisor
- What are we measuring, mostly?
  - how well the recommenders perform at finding things the users already know
  - performance on individual recommendations

There are Alternatives!

- The “easy” alternative
  - test on real users, real situation
  - have them consume and evaluate
- The “hard” alternative
  - extend our knowledge and understanding about metrics
Extending our Knowledge …

From Items to Lists

- Do users really experience recommendations in isolation?

Making Good Lists

- Individually good recommendations do not equal a good recommendation list
- Other factors are important
  - Diversity
  - Affirmation
  - Appropriateness
- Called the “Portfolio Effect”
  [Ali and van Stam, 2004]

Topic Diversification

- Re-order results in a rec list
- Add item with least similarity to all items already on list
- Weight with a ‘diversification factor’
- Ran experiments to test effects

Experimental Design

- Books from BookCrossing.com
- Algorithms
  - Item-based CF
  - User-based CF
- Experiments
  - On-line user surveys
  - 2125 users each saw one list of 10 recommendations

Online Results
Diversity is Important

• User satisfaction more complicated than only accuracy
• List makeup is important to users
• 30% change enough to alter user opinion
• Change not equal across algorithms

Human-Recommender Interaction

• Three premises:
  - Users perceive recommendation quality in context; users evaluate lists
  - Users develop opinions of recommenders based on interactions over time
  - Users have an information need and come to a recommender as a part of their information seeking behavior

HRI

• A language for communicating user expectations and system behavior
• A process model for customizing recommenders to user needs
• An analytic theory to help designers focus on user needs

HRI Pillars and Aspects

Recommendation Dialog

• The individual recommender interaction
• Historical Aspects
  - Correctness, Quantity, Spread
• New Aspects
  - Transparency
  - Saliency
  - Serendipity
  - Usefulness
  - Usability
**Recommendation Personality**

- Experience over repeated interactions
- Nature of recommendations
  - Personalization, Boldness, Freshness, Risk
- Progression over time
  - Adaptability, Pigeonholing
- Relationship
  - Affirmation, Trust

**Information-Seeking Task**

- One of the current limits of HRI
- Concreteness
- Compromise
- Appropriateness of Recommender
- Role of Recommender
- Expectation of Usefulness

**HRI Process Model**

- Makes HRI Constructive
  - Links Users/Tasks to Algorithms
- But, Needs New Metrics

**Developing New Metrics**

- Identify candidate metrics
- Benchmark a variety of algorithms
  - and datasets?
  - establish that metric can distinguish algorithms
- Establish link to HRI aspects
  - definitional links; user studies
- Detailed Examples:
  - Ratability, Boldness, Adaptability

**Metric Experimental Design**

- ACM DL Dataset
  - Thanks to ACM!
  - 24,000 papers
  - Have citations, titles, authors, & abstracts
  - High quality
- Algorithms
  - User-based CF
  - Item-based CF
  - Naive Bayes Classifier
  - TF/IDF Content-based
  - Co-citation
  - Local Graph Search
  - Hybrid variants

**Ratability**

- Probability a user will rate a given item
  - “Obviousness”
  - Based on current user model
  - Independent of liking the item
  - Many possible implementations
  - Naive Bayes Classifier
**Ratability Results**

- **Ratability**
  - Mean Ratability
  - Bars representing Local Graph, Bayes, Item, 50 nbrs, TFIDF, User, 50 nbrs.

**Boldness**

- **Measure of “Extreme Predictions”**
  - Only defined on explicit rating scale
  - Choose “extreme values”
  - Count appearance of “extremes” and normalize
- **For example, MovieLens movie recommender**
  - 0.5 to 5.0 star scale, half-star increments
  - Choose 0.5 and 5.0 as “extreme”

**Boldness Results**

- **Boldness**
  - Ratio to Expected
  - Bars representing Item, 50 nbrs, User, 30 nbrs.

**Adaptability**

- **Measure of how algorithm changes in response to changes in user model**
  - How do users grow in the system?
- **Perturb a user model with a model from another random user**
  - 50% each
  - See quality of new recommendation lists

**Adaptability Results**

- **Adaptability, Even-Split**
  - Mean % adaptable
  - Bars representing Item, 50 nbrs, TFIDF, User, 50 nbrs.
Adaptability Results

Adaptability, Even-Split

More Generally …

Recommender Algorithms

Collaborative Filtering

E-Commerce Scale

- Millions of Products
- Millions of Customers
- Thousands of Clicks per Second
  - Scalability!
Collaborative Filtering Algorithms

- Non-Personalized Summary Statistics
  - K-Nearest Neighbor
    - user-user
    - item-item
  - Dimensionality Reduction
    - LSI
    - PLSI
    - Factor Analysis

- Content + Collaborative Filtering
  - Burke’s Survey of Hybrids
  - Graph Techniques
  - Clustering
  - Classifier Learning
    - Naïve Bayes
    - Bayesian Belief Networks
    - Rule-induction

Item-Item Collaborative Filtering

- Many customers have no relationship
- Many products have no relationship
- Synonymy
  - Similar products treated differently
  - Increases sparsity, loss of transitivity
  - Results in poor quality

Item-Item Collaborative Filtering

Item-based algorithm:
Similarity measure

Item-Item Collaborative Filtering

Item-Item Matrix Formulation

Target item

5 closest neighbors

Raw scores for prediction generation
Approximation based on linear regression

Item similarities

Used for similarity computation

MAE

Adjusted cosine
Pure cosine
Correlation
Neighborhood Size

Incremental Item-Item Algorithm
- Model Building
  - Compute similarity between items
  - Record $p$ most similar items for each item
    - $p$ is model size
- Prediction Generation
  - $p'$ is subset of $p$ rated by $u$
  - Use $\min(p',k)$ items for prediction
    - $k$ is neighborhood size

Model size sensitivity

Model Size vs. Throughput

Item-Item Discussion
- Good quality, in sparse situations
- Promising for incremental model building
  - Small quality degradation
  - Big performance gain

Collaborative Filtering Algorithms
- Non-Personalized Summary Statistics
  - K-Nearest Neighbor
    - user-user
    - item-item
- Dimensionality Reduction
  - LSI
  - PLSI
  - Factor Analysis
- Content + Collaborative Filtering
  - Burke's Survey of Hybrids
- Graph Techniques
  - Horting
- Clustering
- Classifier Learning
  - Naïve Bayes
  - Bayesian Belief Networks
  - Rule-induction
**Dimensionality Reduction**

- Latent Semantic Indexing
  - Used by the IR community
  - Worked well with the vector space model
  - Used Singular Value Decomposition (SVD)
- Main Idea
  - Term-document matching in feature space
  - Captures latent association
  - Reduced space is less-noisy

**SVD: Mathematical Background**

- Use Singular Value Decomposition (SVD)
  - Main Idea
    - Term-document matching in feature space
    - Captures latent association
    - Reduced space is less-noisy

\[ \mathbf{R_k} = \mathbf{U_kS_kV_k^T} \]

The reconstructed matrix \( \mathbf{R_k} = \mathbf{U_kS_kV_k^T} \) is the closest rank-\( k \) matrix to the original matrix \( \mathbf{R} \).

**SVD for Collaborative Filtering**

1. Low dimensional representation
   - \( \mathbf{O(m+n)} \) storage requirement
2. Direct Prediction

\[ \mathbf{R_{m \times n}} \quad \mathbf{U_{m \times k}} \quad \mathbf{S_{k \times k}} \quad \mathbf{V_{k \times n}} \]

**Experimental Setup**

- MovieLens Data (www.movielens.umn.edu)
  - Size 943 x 1,682; 100,000 ratings entry
  - Ratings are from 1-5
  - Used for Prediction and Neighborhood experiments
- E-Commerce Data
  - Size 6,502 x 23,554; 97,045 purchase entry
  - Purchase entries are dollar amounts
  - Used for Neighborhood experiment
- Training and Test Portions
  - Percentage of Training data, \( x \)
  - 10x cross-validation

**Experimental Setup**

- Benchmark Systems
  - CF-Predict
  - CF-Recommend
- Evaluation Metrics
  - Prediction
    - Mean Absolute Error (MAE)
  - Top-N Recommendation
    - Recall and Precision
    - Combined score F1

**Dimension Sensitivity**

![Graph showing sensitivity of no. of dimensions](image)

- Sensitivity of No. of Dimensions
- Ideal value of \( k \)
- Number of Dimensions, \( k \)
- Sensitivity graph with data points indicating the ideal value of \( k \).
**SVD Prediction Results**

SVD as Prediction Generator

(k is fixed at 14 for SVD)

<table>
<thead>
<tr>
<th>x (train/test ratio)</th>
<th>Pure-CF</th>
<th>SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.715</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.735</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.755</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.775</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.795</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0.815</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.835</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.855</td>
<td></td>
</tr>
</tbody>
</table>

**Discussion**

- **Pros:**
  - SVD addresses sparsity
  - Comparable quality with only 14 features
  - Storage space: $O(mn)$ vs. $O(m+n)$
- **Cons:**
  - Problem with dynamic database
  - SVD computation is expensive

**SVD Performance Issues**

<table>
<thead>
<tr>
<th>Model Building</th>
<th>Traditional CF</th>
<th>SVD-based CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline performance</td>
<td>Relatively Fast</td>
<td>Slow</td>
</tr>
<tr>
<td>Prediction Generation</td>
<td>Slow</td>
<td>Very Fast</td>
</tr>
</tbody>
</table>

**SVD Folding-in**

A simple projection technique

\[
U \rightarrow S \rightarrow V
\]

**Singular Value Decomposition**

Reduce dimensionality of problem
- Results in small, fast model
- Richer Neighbor Network

Incremental Update
- Folding in
- Model Update

**Collaborative Filtering Algorithms**

- Non-Personalized Summary Statistics
- K-Nearest Neighbor
  - user-user
  - item-item
- Dimensionality Reduction
  - LSI
  - PLSI
  - Factor Analysis
- Content + Collaborative Filtering
  - Burke's Survey of Hybrids
  - Graph Techniques
  - Horting
  - Clustering
  - Classifier Learning
    - Naive Bayes
    - Bayesian Belief Networks
    - Rule-induction
Algorithm Exercise

Recommender Application (choose one)
- Personalized newspaper
- Music streaming application
- Dentist recommender

The exercise
- Identify sources of data for recommendation
  - content and/or ratings
- Identify 2 user situations
- Explore recommender algorithms for application

Thinking About User Experience

Recommender Application Space

- Dimensions of Analysis
  - Domain
  - Purpose
  - Whose Opinion
  - Personalization Level
  - Privacy and Trustworthiness
  - Interfaces
  - "Algorithms Inside"

Domains of Recommendation

- Content to Commerce
  - News, information, “text”
  - Products, vendors, bundles

Google: Content Example
**Purposes of Recommendation**

- The recommendations themselves
  - Sales
  - Information
- Education of user/customer
- Build a community of users/customers around products or content

**Whose Opinion?**

- “Experts”
- Ordinary “phoaks”
- People like you

**Personalization Level**

- Generic
  - Everyone receives same recommendations
- Demographic
  - Matches a target group
- Ephemeral
  - Matches current activity
- Persistent
  - Matches long-term interests
Privacy and Trustworthiness

- Who knows what about me?
  - Personal information revealed
  - Identity
  - Deniability of preferences
- Is the recommendation honest?
  - Biases built-in by operator
    - “business rules”
  - Vulnerability to external manipulation

Interfaces

- Types of Output
  - Predictions
  - Recommendations
  - Filtering
  - Organic vs. explicit presentation
- Types of Input
  - Explicit
  - Implicit

Launching Organic Interfaces

- Launch.yahoo.com – a truly personal radio station
  - Observes play limits
  - Mixes different inputs, different recommenders
  - Kill a song – once and forever
  - Nice information on why a song is playing
Application Critiques

- Consider the following recommender-powered application
  - What’s good?
  - What’s bad?
  - Why?