Behind the Curtain: Data & Algorithms that power the Netflix User Experience

January, 2014

Xavier Amatriain
Director - Algorithms Engineering - Netflix
What we were interested in:

- High quality *recommendations*

Proxy question:

- Accuracy in predicted rating
- Improve by 10% = $1million!

\[
RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (y_j - \hat{y}_j)^2}
\]
From the Netflix Prize
to today

2006

2013
Big Data @ Netflix

- > 40M subscribers
- Ratings: ~5M/day
- Searches: >3M/day
- Plays: >50M/day
- Streamed hours: 5B hours in Q3 2013

Member Behavior
- Time
- Geo-information
- Impressions
- Device Info
- Metadata
- Social
- Ratings
- Demographics
Smart Models

- Regression models (Logistic, Linear, Elastic nets)
- SVD & other MF models
- Factorization Machines
- Restricted Boltzmann Machines
- Markov Chains & other graph models
- Clustering (from k-means to HDP)
- Deep ANN
- LDA
- Association Rules
- GBDT/RF
- …
Behind the curtain: Netflix Algorithms
“In a simple Netflix-style item recommender, we would simply apply some form of matrix factorization (i.e. NMF)”
Rating Prediction

When his wife is sent to jail on murder charges she fervently denies, a college professor hatches a meticulous plan for the ultimate prison escape.

Starring: Russell Crowe, Elizabeth Banks
Director: Paul Haggis

Based on your interest in: Iron Man 2, John Q and X-Men Origins: Wolverine

Our best guess for Xavier:

Not Interested

In Instant Queue
2007 Progress Prize

- Top 2 algorithms
  - SVD - Prize RMSE: 0.8914
  - RBM - Prize RMSE: 0.8990

- Linear blend Prize RMSE: 0.88

- Currently in use as part of Netflix’ rating prediction component

- Limitations
  - Designed for 100M ratings, we have 5B ratings
  - Not adaptable as users add ratings
  - Performance issues
**SVD - Definition**

\[
A_{[n \times m]} = U_{[n \times r]} \Lambda_{[r \times r]} (V_{[m \times r]})^T
\]

- **A**: \( n \times m \) matrix (e.g., \( n \) documents, \( m \) terms)
- **U**: \( n \times r \) matrix (\( n \) documents, \( r \) concepts)
- **Λ**: \( r \times r \) diagonal matrix (strength of each ‘concept’) (\( r \): rank of the matrix)
- **V**: \( m \times r \) matrix (\( m \) terms, \( r \) concepts)
SVD - Properties

- ‘spectral decomposition’ of the matrix:

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 \\
5 & 5 & 5 & 0 \\
0 & 0 & 0 & 2 \\
0 & 0 & 0 & 3 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
= \begin{bmatrix}
u_1 & u_2 \\
\end{bmatrix}
\times \begin{bmatrix}
\lambda_1 & \emptyset \\
\emptyset & \lambda_2 \\
\end{bmatrix}
\times \begin{bmatrix}
v_1 \\
v_2 \\
\end{bmatrix}
\]

- ‘documents’, ‘terms’ and ‘concepts’:

- **U**: document-to-concept similarity matrix
- **V**: term-to-concept similarity matrix
- **Λ**: its diagonal elements: ‘strength’ of each concept
SVD for Rating Prediction

- User factor vectors \( p_u \in \mathbb{R}^f \) and item-factors vectors \( q_v \in \mathbb{R}^f \)
- Baseline (bias) \( b_{uv} = \mu + b_u + b_v \) (user & item deviation from average)
- Predict rating as \( r_{uv}' = b_{uv} + p_u^T q_v \)
- SVD++ (Koren et. al) asymmetric variation w. implicit feedback

\[
    r_{uv}' = b_{uv} + q_y^T \left( |R(u)|^{\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j + |N(u)|^{\frac{1}{2}} \sum_{j \in N(u)} y_j \right)
\]

- Where
  - \( q_v, x_v, y_v \in \mathbb{R}^f \) are three item factor vectors
  - Users are not parametrized, but rather represented by:
    - \( R(u) \): items rated by user \( u \) & \( N(u) \): items for which the user has given implicit preference (e.g. rated/not rated)
Restricted Boltzmann Machines

- Restrict the connectivity in ANN to make learning easier.
- Only one layer of hidden units.
- Although multiple layers are possible
- No connections between hidden units.
- Hidden units are independent given the visible states.
- RBMs can be stacked to form Deep Belief Networks (DBN) – 4th generation of ANNs
What about the final prize ensembles?

- Our offline studies showed they were too computationally intensive to scale
- Expected improvement not worth the engineering effort
- Plus, focus had already shifted to other issues that had more impact than rating prediction...
Ranking

- Ranking = **Scoring + Sorting + Filtering** bags of movies for presentation to a user
- Key algorithm, sorts titles in most contexts
- **Goal:** Find the best possible ordering of a set of videos for a user within a specific context in real-time
- **Objective:** maximize consumption & “enjoyment”

**Factors**
- Accuracy
- Novelty
- Diversity
- Freshness
- Scalability
- …
Example: Two features, linear model

Linear Model:

\[ f_{\text{rank}}(u,v) = w_1 p(v) + w_2 r(u,v) + b \]
Example: Two features, linear model
Ranking

- Popularity
- + Ratings
- + More Features & Optimized Models
Learning to Rank

- Ranking is a very important problem in many contexts (search, advertising, recommendations)

- Quality of ranking is measured using ranking metrics such as NDCG, MRR, MAP, FPC...

- It is hard to optimize machine-learned models directly on these measures
  - They are not differentiable

- We would like to use the same measure for optimizing and for the final evaluation
Learning to Rank Approaches

■ ML problem: construct ranking model from training data

1. **Pointwise** (Ordinal regression, Logistic regression, SVM, GBDT, ...)
   ■ Loss function defined on individual relevance judgment

2. **Pairwise** (RankSVM, RankBoost, RankNet, FRank...)
   ■ Loss function defined on pair-wise preferences
   ■ Goal: minimize number of inversions in ranking

3. **Listwise**
   ■ Indirect Loss Function (RankCosine, ListNet...)
   ■ Directly optimize IR measures (NDCG, MRR, FCP...)
     ■ Genetic Programming or Simulated Annealing
     ■ Use boosting to optimize NDCG (Adarank)
     ■ Gradient descent on smoothed version (CLiMF, TFMAP, GAPfm @cikm13)
     ■ Iterative Coordinate Ascent (Direct Rank @kdd13)
Similarity
What is similarity?

- Similarity can refer to different dimensions
  - Similar in metadata/tags
  - Similar in user play behavior
  - Similar in user rating behavior
  - ...

- You can learn a model for each of them and finally learn an ensemble
Graph-based similarities

- "The Fighter" to "Mad Men" (0.8)
- "Rango" to "Mad Men" (0.3)
- "How I Met Your Mother" to "Mad Men" (0.7)
- "The Fighter" to "Rango" (0.4)
- "The Fighter" to "How I Met Your Mother" (0.2)
- "Rango" to "How I Met Your Mother" (0.3)
Example of graph-based similarity: SimRank

- SimRank (Jeh & Widom, 02): “two objects are similar if they are referenced by similar objects.”

\[
s(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{\mid I(a)\mid} \sum_{j=1}^{\mid I(b)\mid} s(I_i(a), I_j(b))
\]

Figure 1: A small Web graph \( G \) and simplified node-pairs graph \( G^2 \). SimRank scores using parameter \( C = 0.8 \) are shown for nodes in \( G^2 \).
Final similarity ensemble

- Come up with a score of play similarity, rating similarity, tag-based similarity...
- Combine them using an ensemble
  - Weights are learned using regression over existing response
- The final concept of “similarity” responds to what users vote as similar
Row Selection Inputs

- Visitor data
  - Video history
  - Queue adds
  - Ratings
  - Taste preferences
  - Predicted ratings
  - PVR ranking
  - etc.

- Global data
  - Metadata
  - Popularity
  - etc.
Row Selection Core Algorithm

Candidate data → Groupify → Score/Select → Group selection

- Selection constraints
- Partial selection
Selection Constraints

• Many business rules and heuristics
  – Relative position of groups
  – Presence/absence of groups
  – Deduping rules
  – Number of groups of a given type
• Determined by past A/B tests
• Evolved over time from a simple template
Groupification

- Ranking of titles according to ranking algo
- Filtering
- Deduping
- Formatting
- Selection of evidence
- ...

Candidate data

First $n$ selected groups

Group for position $n+1$
Scoring and Selection

• Features considered
  – Quality of items
  – Diversity of tags
  – Quality of evidence
  – Freshness
  – Recency
  – ...

  – ARO scorer
  – Greedy algorithm
  – Framework supports other methods
    ■ Backtracking
    ■ Approximate integer programming
    ■ etc.
Search Recommendations

Related to Italy

- Miraculous: Canals of Venice
- The Medici
- Insiders: The Vatican
- Videocracy
- Borgia

Ancient Mysteries: Canals of Venice
- Year: 2005
- Rating: 3/5
- Duration: 46m

Known for its distinctive man-made canals and unparalleled aura of romance, the Italian city of Venice is like no other place on Earth.

TV Shows, Documentaries
Search Recommendation

Combination of PAS+PRS model

- **Play-After-Search** and **Play-Related-Search**
- **PAS**: Transition on Play after Query
- **PRS**: Similarity between user’s (query, play)
Unavailable Title Recommendations

Schindler's List is unavailable to stream

After searching for this title, many members stream:
Gamification Algorithms
Rating Game

Mood Selection Algorithm
1. Editorially selected moods
2. Sort them according to users consumption
3. Take into account freshness, diversity…
More data or better models?
More data or better models?

Anand Rajaraman: Former Stanford Prof. & Senior VP at Walmart
More data or better models?

Sometimes, it’s not about more data

---

**ABSTRACT**

The Netflix Prize (NP) competition gave much attention to collaborative filtering (CF) approaches. Matrix factorization (MF) based CF approaches assign low dimensional feature vectors to users and items. We link CF and content-based filtering (CBF) by finding a linear transformation that transforms user or item descriptions so that they are as close as possible to the feature vectors generated by MF for CF.

We propose methods for explicit feedback that are able to handle 140,000 features when feature vectors are very sparse. With movie metadata collected for the NP movies we show that the prediction performance of the methods is comparable to that of CF, and can be used to predict user preferences on new movies.

We also investigate the value of movie metadata compared to movie ratings in regards of predictive power. We compare

---

**1. INTRODUCTION**

The goal of recommender systems is to give personalized recommendation on items to users. Typically the recommendation is based on the former and current activity of the users, and metadata about users and items, if available.

There are two basic strategies that can be applied when generating recommendations. Collaborative filtering (CF) methods are based only on the activity of users, while content-based filtering (CBF) methods use only metadata. In this paper we propose hybrid methods, which try to benefit from both information sources.

The two most important families of CF methods are matrix factorization (MF) and neighbor-based approaches. Usually, the goal of MF is to find a low dimensional representation for both users and movies, i.e., each user and movie is associated with a feature vector. Movie metadata (which
More data or better models?

Norvig: “Google does not have better Algorithms, only more Data”

The Unreasonable Effectiveness of Data

Many features/low-bias models

Figure 1. Learning Curves for Confusion Set Disambiguation

[Banko and Brill, 2001]
More data or better models?

Sometimes, it’s not about more data.
“Data without a sound approach = noise”
Conclusion
More data +
Smarter models +
More accurate metrics +
Better system architectures

Lots of room for improvement!
Thanks!

We’re hiring!

Xavier Amatriain (@xamat)
xavier@netflix.com