IMAGINE THE POSSIBILITIES...
MLaaS Applications in Digital Video – Supplanting Disliked Content

Srilal Weera PhD
Principal Engineer
Charter Communications
MLaaS - machine learning as a service

• Machine Learning applications in the video delivery pipeline are ubiquitous (from ingest to delivery network and storage optimizing to data analytics)
• e.g. Manual scanning of TV ads at ingest can be automated with an ML classification engine*
• However, these applications are internal to the enterprise

MLaaS

• The technology is ripe for the next stage of ML revolution (similar to Saas/PaaS)
• The goal is to create AI/ML based revenue generating products

*paper presented at SCTE - CableTech - Expo 2019
Example – Enhanced Rating System for Movies

- Current TV/Movie ratings are confusing (TV-Y7-FV, PG-13, TV-14..)
- The restricted content definition is subjective.
- Parents would appreciate to know the type/placement of restricted content.
- Scan videos in the repository and tag restricted content with timestamps.
- Create API to access ‘Enhanced Ratings’ from the program guide.
Use case – Ad Revenue Loss Due to Channel Surfing

- TV viewers routinely encounter shows they dislike, but are unable to avoid seeing them. The end result is flipping the channel.
- Causes advertising revenue loss for the programmer
- IP streaming offers capability, but need an E2E solution. (Contractual obligations)

Supplanting unappealing content (ML Automation)
- The goal is to keep the viewer in the same channel
- Comparison with Recommender Systems – Dissuade vs. Persuade
- Could an RS model predict disliked content?
- Not having user ratings is a barrier for applying the RS model for ‘disliked content’.
Recommender Systems (RS)

- Types: Content-Based filtering (CB) and Collaborative-Filtering (CF) and hybrid.
- CB approach is based on the attributes of each item and the user’s affinity (rating) for similar items.
- CF recommendations are based on multiple users’ ratings.
- Similarity is determined by first forming vectors (rows of ratings matrix) per each user, and computing similarity measures.
- Users in the same ‘neighborhood’ would have Cosine of angle close to 1

\[
\text{similarity} = \cos(\theta) = \frac{u \cdot v}{||u|| ||v||} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}}
\]
Matrix factorization (MF)

- Matrix factorization (MF) is a well-known algorithmic approach to solve RS
- Assumption - latent factors determine the user ratings of items
- The latent factors are not measured directly
- But their impact is reflected in the user ratings (observed variable)
- The starting point is the user-item ‘rating matrix’ (utility matrix)

<table>
<thead>
<tr>
<th></th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie 3</th>
<th>. . . .</th>
<th>Movie n</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>5</td>
<td>N/A</td>
<td>2</td>
<td>. . . .</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>3</td>
<td>N/A</td>
<td>. . . .</td>
<td>N/A</td>
</tr>
<tr>
<td>User 3</td>
<td>N/A</td>
<td>4</td>
<td>1</td>
<td>. . . .</td>
<td>4</td>
</tr>
<tr>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
<tr>
<td>User m</td>
<td>N/A</td>
<td>4</td>
<td>1</td>
<td>. . . .</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Matrix factorization

- The rating matrix of has many empty cells (sparse).
- The goal of recommender system is to fill these gaps.
- The utility matrix is expressed in terms of two latent factor matrices (one for user and the other for item attributes).
- The dot product would determine user’s affinity for the movie.
Matrix Factorization – Example

Example –

- Bob is an action movie aficionado
  
  Bob’s profile = 60% Action + 30% Comedy + 10% Romance + 0% Historical
  
  Titanic’s profile = 30% Action + 0% Comedy + 60% Romance + 10% Historical

- Larger the dot product, the item is better suited for user’s taste (recommend)
- The missing values in the matrix are approximated via optimization
Matrix factorization – Example

- The rating matrix of has many empty cells (sparse).
- The goal of recommender system is to fill these gaps.
- The utility matrix is expressed in terms of two latent factor matrices (one for user and the other for item attributes).
- The dot product would determine user’s affinity for the movie.

Example – Bob is an action movie aficionado.

Bob’s profile = 60% Action + 30% Comedy + 10% Romance + 0% Historical
Titanic’s profile = 30% Action + 0% Comedy + 60% Romance + 10% Historical
Singular Value Decomposition (SVD)

\[ A = U \Sigma V^T \]

- \( A \) is the given data matrix \((m \times n)\) of rank \( r \)
- \( U \) is an orthonormal \( m \times r \) matrix
- \( \Sigma \) is a diagonal \( r \times r \) matrix
- \( V \) is an orthonormal \( n \times r \) matrix
Singular Value Decomposition – Element Form

\[ A \approx U_{mxr} \Sigma_{rxr} V_{rxn}^T \]

- SVD filters the dominant features from the data (Sigma matrix) and identify hidden correlations in the singular vectors \( U \) and \( V^T \).
- SVD helps find a lower rank matrix approximation to the original matrix.
- Each singular value defines the ‘strength’ of the concept.
- Pick only the top singular values from the sigma matrix and discard others.
Supplanting Unappealing TV Programs

- A customer identifies a TV program/show/series that is unappealing to her
- Her choices can be supplied via a web portal/API or an icon added to the TV Guide.
- The Content Provider establishes Alternate content channels for selected regular channels.
- The Alternate content could be programs that aired earlier in the day and stored in the CDN.
- Whenever the consumer tunes to an unappealing program content, it will trigger content replacement automatically

<table>
<thead>
<tr>
<th>Supplanting a Program by Alt Content (Traveler Channel)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original Program</strong></td>
</tr>
<tr>
<td><em>Ghoul Adventures</em></td>
</tr>
</tbody>
</table>
Process Flow for Alt Content Usage in Program Replacement

1. Disliked program on-air
2. Stores programs and Alt. content
3. User tunes to a channel with Disliked program (on-air)
4. ‘Disliked’ data (ML analytics)
5. Check if the program is in Disliked list.
6. Retrieve binding data from ACS. Update manifest file accordingly
7. Obtain Alt cont. location (URL)s and modifies manifest
8. Retrieve Alt. cont. file segments from CDN and send to Display device
9. Alternate content served to the viewer

Notify MM of customer choice. Keep track of TV program schedule

- Encoder
- Packager
- Alternate Content Server (ACS)
- Manifest Manipulator
- TV Guide
- Alternate Content Gen. (ACG)
- Origin
- User Device
- CDN

© 2020 SCTE•ISBE, CableLabs & NCTA. All rights reserved. | scte.org • isbe.org
Challenges in Applying RS to Disliked content

- Disliked Content – No user ratings; Viewers simply change the channel
- Not having an explicit metric is a barrier for applying the Recommender Systems
- Most users never even tune to the hundreds of channels offered
- RS model uses similarity measures in the latent space to determine affinity
- A different metric is needed for feature vector creation
- Our analysis is applicable to channels with a mix of liked-content and disliked-content
- If there are no liked-programs at all (i.e. customer never tunes to the channel), then the issue of ad revenue loss is moot.
Implicit Identification of Disliked content

- **Method** – Collect viewership data over time and apply data analytics to identify disliked content indirectly.
- **A peculiarity of viewership data** – vast difference in the time scale
  - ‘Disliked content’, the channel surfing times are about one to two seconds
  - ‘Liked content’ (regular viewing) – durations vary from minutes to hours
- **The wide range is a barrier for applying machine learning algorithms (accuracy)**
- **Solution** – Change of scale adjustment
Disliked Content Algorithmic Impact

- Due to the large disparity in the time scales (few seconds for disliked content vs. thousands of seconds for liked content), the user/item vectors in the latent space will not be an accurate depiction.

- The impact of disliked content will be hard to quantify.

- Note - Simply inverting the scale (e.g., assigning 5 stars for disliked and 1 for liked), will not capture the nuances.

\[
similarity = \cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}}
\]
Disliked Content Algorithmic Impact - SVD

- In the Singular values matrix, the dominant values are at the top left and decreases down the diagonal.

  \[ \sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots \]

- The quantities of interest (disliked content with 1-2 seconds), are at the lower end of the time-scale

- Their impact is washed out by much larger terms in the matrix (thousands of seconds for liked content)

- Issue – The dominant \( \sigma \) values do not reflect the impact of disliked-content

- Necessary to rescale the data so that the disliked content reflect the dominant terms in the utility matrix.
Data Rescaling

- Channel change time could vary from 1 second to several hours
- Metric – Log reciprocal of channel surf time

\[
\text{Disliked Content Measure} = \log_{10} \left(\frac{10^4}{T}\right) = 4 - \log_{10} T
\]

- Rescale: 1 second to over 2 hours (original span) \(\rightarrow 0 - 4\) (after rescale)
- More granular than inversely assigning 5 stars to shorter times etc.

<table>
<thead>
<tr>
<th>Duration</th>
<th>Disliked Content</th>
<th>Liked Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sec</td>
<td>4</td>
<td>15 min</td>
</tr>
<tr>
<td>10 sec</td>
<td>3</td>
<td>2 hour</td>
</tr>
<tr>
<td>15 min</td>
<td></td>
<td>1.045</td>
</tr>
<tr>
<td>2 hour</td>
<td></td>
<td>0.143</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c|c|c|c|c}
\text{Duration} & \text{1 sec} & \text{10 sec} & \text{15 min} & \text{2 hour} \\
\hline
\log_{10} \left(\frac{10^4}{T}\right) & 4 & 3 & 1.045 & 0.143 \\
\end{array}
\]
Enhancements to Recommender Systems

- Apply RS method to create a disliked content matrix
- Supplies additional data about user preferences, to enhance the recommender system
- Useful since a major issue with RS is the sparsity of the input data matrix
MLaaS - Benefits to Service Providers and Programmers

- Machine learning enables identifying and auto-replacement of disliked content.
- Novel offering ("Don’t Like, Don’t Watch!") – new revenue opportunity
- Improved Recommender Systems with the additional data from disliked content.
- Targeted Ad Campaigns – Complement demographic data with viewership patterns

Summary

Cable-tech is ripe for disruption and transformation with MLaaS based technologies.
Thank You!

Srilal Weera PhD
Principal Engineer
Charter Communications