EFFECTIVE UTILIZATION OF M-ABR (MULTICAST ASSISTED ABR) USING BIG DATA AND REAL-TIME ANALYTICS

Sridhar Kunisetty, Jeffrey Tyre, and Robert Myers
ARRIS
Why ABR (Adaptive Bit Rate) in TV eco-system?
Multi-Screen Lifestyle is Here Today
Multi-Screen Drives New Video Delivery Technology

- Multi-screen consumer devices use web-based technologies
- Video is delivered over an IP network using HTTP / TCP
- Content Delivery Networks (CDN) are based on HTTP
- Adaptive Bit-Rate (ABR) video streaming is used to address “best-effort” delivery characteristics of IP networks and provides a better Quality of Experience (QoE).

Pay TV Operators are currently offering “Second-screen” Services
Why Multicast in the Context of ABR?
ABR Through Unicast

- ABR requires one unicast stream per device, meaning high bandwidth than traditional TV broadcast or multicast network.
- Delivering HD & UHD to every screen with ABR breaks the bandwidth bank.
- Scalability would be an issue due to high bandwidth requirements and associated network cost.

Bandwidth Savings with Multicast

- Capacity Needed for 50K Subs
- Bandwidth Savings with Multicast
- Moderate Video Resolution
- High Video Resolution (4K)
M-ABR (Multicast Assisted ABR) Solution
M-ABR Solution
Components in the M-ABR Solution

Multicast Server
- Ingest Unicast ABR, wrap in multicast
- Output multicast over NORM protocol, for added reliability

Multicast Controller
- Distributes configuration/multicast map to multicast servers and EMCs
- Statically and/or dynamically manage multicast map updates
- Gathers events from edge devices on channel change data

Embedded Multicast Client (EMC)
- Join multicasts and cache ABR files
- Forward all other unicast ABR requests to CDN for fulfillment
- Act as transparent HTTP proxy for the home - either respond with content from the cache or pass the unicast request through
- Provide usage info back to Multicast Controller for dynamic multicast map updates
Big Data and Real-time Analytics and their Application to M-ABR
Characteristics of Big Data

Volume
Refers to the sheer amount of data that is created continuously.

Velocity
The speed with which this data can be processed and analyzed for a meaningful use.

Variety
The different types and formats of data that is collected.
What Does a Typical Big Data System Do?

• Collect a variety of large amounts of data
• Use parallel processing to process and analyze that information to derive meaningful insight
• Provide access to the derived information through dashboards, reports, and APIs
• Achieve all this with
  ➢ Linear scalability - as the data size increases, one can throw more cheap commodity hardware at it and see processing complete in almost the same amount of time
  ➢ Fault tolerant to failures on any of the computer nodes processing that data
Big Data Related Technologies

• Some of the commonly used ones are: Hadoop, HDFS, MapReduce, YARN, SpringXD, Flume, Pig, Hive, Spark, Storm, etc.

• Batch oriented vs. real-time processing
  ➢ Hadoop is used for batch processing and storage
  ➢ Spark is mainly used for real-time processing
  ➢ In many cases, both Hadoop and Spark coexist
Applicability to M-ABR

Volume
The devices in the millions of subscriber homes generate a large volume of data

Velocity
Millions of data events are produced each minute. Data could collected at a segment level (i.e., every 2 secs) from each EMC

Variety
Different types of events are produced by EMC and also by server side components

For effective detection and fixing of operational issues and to take advantage of user behavior, all this data needs to be processed in real-time
Analytics System Overview

Data Ingest Layer
- Facilitates collection of data from a variety of data sources in real-time.
- Parses data and pre-processes as needed before passing to the next layer.

Analysis & Processing Layer
- Real-time processing and analytics happen here.
- Data is processed by different processes running asynchronously in parallel.
- One set of processes operate on the raw data, process them and stores in a data store.
- The other set of processes perform the next level of processing by doing summarizations, correlations, deeper analysis, etc.

Data Output Layer
- Processed data is exposed through multiple methods – dashboards and charts that provide a visual view into the different operational and usage aspects of M-ABR.
- REST APIs that can be programmatically consumed by various services to act on the analyzed data.
High-Level Goals of Real-Time Analysis

• Understanding usage of M-ABR components, services, and network
• Understanding network and system limitations with respect to scalability, capacity planning, and bandwidth management
• Creating dashboards to provide business intelligence, and APIs for downstream usage
• Analyzing performance of multicast over DOCSIS, including the telemetry of packet loss
• Evaluating the workload within the EMC to perform the functions of the “transparent proxy cache”
• Examining the performance of the CMTS/CCAP for multicast delivery and implications related to its support for IGMP
Real-Time Analysis Helps Answer Questions

Some examples include:

- How often does a file segment experience any packet loss?
- Is the packet loss characterized by factors such as time of day, downstream capacity, CMTS configuration, etc.?
- How does channel surfing affect the IGMP behavior of joins and leaves and when does it become more of a burden and less of a benefit?
- How quickly can initial tuning to a new channel result in moving from unicast video segments to multicast (cached) video segments?
- How long should the EMC remain on a multicast stream before it determines there is no benefit to caching stream segments?
- What is the timing of a cache segment to client request?
Segment Data and Related Analysis

For each multicast video segment received on the EMC, this data event is produced and helps in analyzing the following:

- Segment packet loss details and whether the packet loss was sequential or multiple packet-loss across the segment
- Segment timing details such as, when it is received with and without error and how soon after the client requested the video segment
- Segment details about how long it is maintained in the EMC cache
Multicast Session Data and Related Analysis

For each IGMP session on the EMC this data event is produced to track the lifecycle of the multicast stream and helps in analyzing the following:

- IGMP behavior on the EMC and the success of each multicast video segment received
- Estimate of the IGMP load on the CMTS as a subscriber transitions from channel to channel
- Details on the multicast session, such as when it is initiated, when it receives data, when the first segment starts, how many segments had an error, etc.
Client Session Data and Related Analysis

For each client HLS session associated to a multicast service, this data event is generated and it helps in analyzing the following:

- Whether the video segment was provided from the cache or required the EMC to pass the request through unicast
- Estimate the initial unicast load for the client buffer fill as well as how often the client requests a segment that results in a cache miss and goes unicast
- The experience of a client session using specific multicast channel
Example Insight – Channel Utilization

• Shows the number of client sessions that exist on a per multicast channel (stream) basis over a specified time period (popularity of a channel)
• This information is helpful in environments where the number of channels that can be simultaneously multicast is limited due to bandwidth capacity limitations. It can help to determine which channels are more popular and when

![Graph showing channel utilization over time](image)
Example Insight – Missed Segments

• Shows the peak and average number of segments that could not be delivered from the EMC cache to a client device (in an optimal network this should be zero)
• Missed segments but can be impacted by issues in such as network errors or multicast server failures
Example Insight – Network Latency

- Shows the peak and the average time that it took for the EMCs to receive a segment from the multicast stream
- If this value becomes greater than the segment duration it could indicate an issue somewhere in the network or in the multicast server
Summary

• M-ABR is a promising solution for operators as they roll out IP Video as it helps in optimizing bandwidth while still supporting ABR streaming.

• Understanding the operational issues becomes essential for this new technology so that we can address those issues to improve efficiency, reduce cost and provide better user experience.

• Actionable insight derived from collecting, processing, and analyzing data events provides useful information in understanding and driving operational efficiencies.
Smart Bandwidth Sharing for Multi-Program Content Distribution

Sean McCarthy, Ph.D., ARRIS
Boston, MA • May 18, 2016
Intro

How can we know that we are delivering great viewer experiences consistently in a world of More of Everything

- programs
- displays types
- distribution protocols
- compression technologies
Our Proposal

“Video Quality Stress” is a no-reference metric we have derived from previous experience in statistical multiplexing and expanded to OTT, IPTV, CBR, and ABR use cases.

How often will a program’s video quality be too low?

What bitrates should I allocate to my ABR, OTT, and multi-program services to match video quality?

Nearly impossible to cover all permutations with subjective testing, PSNR, SSIM, etc.
VIDEO-QUALITY STRESS
Constant Quality Variable Bitrate Encoding

Constant Video Quality

Variable Bitrate

Constant Video Quality
Constant Bitrate Encoding

Encoder Rate Control

Variable Bitrate

Constant Video Quality

Constant Bitrate
Defining “Video Quality Stress”

Definition of “Video Quality Stress”

Constant Quality Variable Bitrate divided by the Actual Bitrate

In the CBR Scenario, “Video Quality Stress” is proportional to the Constant Quality Variable Bitrate

More stress
Worse video quality

Less stress
Better video quality
Generalized Video Quality Stress

Need Parameter (metadata) Proportional to bit rate needed for constant video quality

Need Parameter (metadata)

Cloud App

ABR Program Segments

Adaptive Bitrate (ABR)

Bitrate (metadata)

Video Quality Stress: ABR

More stress Worse video quality

Less stress Better video quality
STATISTICAL ANALYSIS OF VIDEO QUALITY STRESS
Video Quality Stress as a Probability

**A**
Video-Quality Stress Time Series
For a Sample of Programs and Times

**B**
Ensemble Cumulative Probability
This curve tells us how often video quality is better or worse than a benchmark video quality.
Video Quality Stress as a Probability

**Danger of Poor Video Quality?**

This point indicates that
~10% of the time
the actual bitrate allocated to programs is
~30% too low
to meet the
benchmark video quality
(Video Quality Stress)

**Possible Bandwidth Inefficiency?**

This point indicates that
~20% of the time
the actual bitrate allocated to programs is
~33% more than needed
to meet the
benchmark video quality
(Video Quality Stress = 1)
Quantitative Tuning of Video Quality

Define a video-quality performance target:
“Video quality should be better than or equal to a benchmark video quality a certain percentage of the time”

1. Example: “Video quality should be better than or equal to the benchmark 90% of the time.”

2. Determine corresponding Video Quality Stress Factor before tuning

3. New Tuned Bitrate = Video Quality Stress Factor x Original Bitrate

Diagram:
- Tuning Video Quality Performance
- Cumulative Probability
- Video Quality Stress
QUANTIFYING THE EFFICIENCY OF BANDWIDTH SHARING
Multi-Program Performance Tuning

Multi-Program Pool

Aggregate Video-Quality Stress

Cumulative probability

Video-quality stress

Multi-Program Pool

Aggregate Video-Quality Stress

Cumulative probability

Video-quality stress
Making Bandwidth Sharing Decisions

**Relative Efficiency of Bandwidth Sharing between N programs compared to M programs**

$$\frac{\Delta B_M - \Delta B_N}{1 + \Delta B_M}$$

$\Delta B_M$ is the percent additional bandwidth required to achieve equal to or better than the benchmark video quality

<table>
<thead>
<tr>
<th>Number of Programs</th>
<th>Percentage of time that tuned video quality is equal to or better than the benchmark video quality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90%</td>
</tr>
<tr>
<td>1</td>
<td>32%</td>
</tr>
<tr>
<td>3</td>
<td>19%</td>
</tr>
<tr>
<td>10</td>
<td>11%</td>
</tr>
<tr>
<td>30</td>
<td>6%</td>
</tr>
<tr>
<td>100</td>
<td>3%</td>
</tr>
<tr>
<td>300</td>
<td>2%</td>
</tr>
</tbody>
</table>

Example

\[ \Delta \text{BM} - \Delta \text{BN} \]

\[ 1 + \Delta \text{BM} \]

Example

\[ \text{Relative Efficiency of Bandwidth Sharing Compared to CBR} \]

<table>
<thead>
<tr>
<th>Number of Programs</th>
<th>Percentage of time that tuned video quality is equal to or better than the benchmark video quality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90%</td>
</tr>
<tr>
<td>1</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>10%</td>
</tr>
<tr>
<td>10</td>
<td>16%</td>
</tr>
<tr>
<td>30</td>
<td>20%</td>
</tr>
<tr>
<td>100</td>
<td>22%</td>
</tr>
<tr>
<td>300</td>
<td>23%</td>
</tr>
</tbody>
</table>
“Video Quality Stress” is a no-reference metric for monitoring & optimizing a wide range of video distribution models.

Statistical Analysis of Video Quality Stress could help us deliver great viewer experiences consistently in an increasingly diverse video distribution ecosystem.
Thank you

sean.mccarthy@arris.net
Bringing the power of Analytics to improve end-user Quality of Experience

Sangeeta Ramakrishnan
Principal Engineer
Cisco Systems

Co-authors:
Xiaoqing Zhu, Frank Chan, Bhanu Krishnamurthy, Cindy Chan, Zheng Lu, Kashyap Kambhatla
Cisco Systems
Adaptive Bit Rate Overview
Video QoE Analytics: Why?

• Capacity Planning
• Troubleshooting
• QoE
Optimization/Network Optimization
Capacity Planning

![Capacity Planning Diagram](image-url)
Troubleshooting

Significant OpEx $$$
SDN-based Architecture for QoE Analytics

Video QoE Application

- Video Quality Analytics
- Client Info
- Streamer Info
- Network Info

Open Daylight Controller

- COPS
- CLI
- MIBs

Core

CMTS

VoD Streamer

Streamer Info

Client Info

Video QoE Application

2016 Spring Technical Forum
CABLELABS • NCTA • SCTE
Video QoE Aware WiFi SON

Video QoE Application

Business Policies

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info

Video Quality Analytics

CMTS VoD

Streamer

Video QoE Application

QoE API

WiFi RRM

RRM read/writes

Open Daylight Controller

COPS

CLI

MIBs

Network Info

Core

VoD Streamer

Streamer info
Fairness across ABR streams
QoE Optimization: Exploiting rate-quality trade-offs

More complex titles

More complex segments
QoE Optimization over DOCSIS

- Significant increase in stream packing efficiency
- Cooperative streaming – network and video working together
Summary

• Video QoE Analytics is critical to improving end-user QoE
• Analytics can be used for multiple purposes
• Significant improvement in QoE/network efficiency possible by leveraging analytics
• Harness advances in big data, machine learning, and visualization
ASPERA HIGH-SPEED TRANSFER
Moving the world’s data at maximum speed

Charles Shiflett    Bear @ Asperasoft.com
FASP™: HIGH-PERFORMANCE TRANSPORT

Distance degrades conditions on all networks
- Latency (or Round Trip Times) increase
- Packet losses increase
- Fast networks just as prone to degradation

TCP performance degrades with distance
- Throughput bottleneck becomes more severe with increased latency and packet loss

TCP does not scale with bandwidth
- TCP designed for low bandwidth
- Adding more bandwidth does not improve throughput

Alternative Technologies
- TCP-based - Network latency and packet loss must be low
- UDP traffic blasters - Inefficient and waste bandwidth
- Data caching - Inappropriate for many large file transfer workflows
- Modified TCP - Improves on TCP performance but insufficient for fast networks
- Data compression - Time consuming and impractical for certain file types
- CDNs & co-lo build outs - High overhead and expensive to scale
FASP™: HIGH-PERFORMANCE TRANSPORT

Maximum transfer speed
- Optimal end-to-end throughput efficiency
- Transfer performance scales with bandwidth independent of transfer distance and resilient to packet loss

Congestion Avoidance and Policy Control
- Automatic, full utilization of available bandwidth
- On-the-fly prioritization and bandwidth allocation

Uncompromising security and reliability
- Secure, user/endpoint authentication
- AES-128 cryptography in transit and at-rest

Scalable management, monitoring and control
- Real-time progress, performance and bandwidth utilization
- Detailed transfer history, logging, and manifest

Low Overhead
- Less than 0.1% overhead on 30% packet loss
- High performance with large files or large sets of small files

Resulting in
- Transfers up to thousands of times faster than FTP
- Precise and predictable transfer times
- Extreme scalability (concurrency and throughput)
FASP 4 TRANSFER BETWEEN TACC AND NCSA

- Performance of in Development FASP 4 Protocol
  - 65 Gbit/s Wire Rate transfer, TACC to NCSA
  - 61 Gbit/s Effective rate, over 90% of available bandwidth utilized for transfer

- Eliminate traditional bottlenecks which impede the efficient transmission of data

- Single Stream Single Node Transport Solution

- 91 Gbit/s effective throughput within LAN environment with single Mellanox ConnectX®-4
  - 1 PB of data transferred every day
  - 675 GB per minute
SC14 DEMO CONFIGURATION

2x Intel® Xeon® E5-2697 v3

5x Intel® DC P3700 NVMe SSD

2x Intel® XL710 40 GbE Ethernet QSFP+
Data Transfer Node Specs:

- Intel® Xeon® processor E5-2687W v3
- 4x Intel P3608 NVMe SSD
- 3x 40 Gbit Intel XL710 NIC
## ASPERA MOBILE APIs

| Android SDK | Aspera Android SDK provides a Java API to transfer files using FASP-AIR™. |
| iPhone SDK | Aspera iPhone SDK with Objective C API to transfer files using FASP-AIR. |

## ASPERA BROWSER APIs

| Connect –JavaScript API | JavaScript API exposed by Aspera Connect for integration of FASP based file transfers into web applications for a complete in-browser experience |

## ASPERA APPLICATION APIs

| Shares API | Full programmatic control over browsing Shares, transfer authorization, and upload / download. |
| faspex™ Web API | A set of services that enables users to create and receive digital deliveries via a Web interface, while taking advantage of FASP high-speed transfer technology. |
| Console API | Full programmatic management of transfer sessions including initiation, queuing, management and control through a RESTful API. |

## ASPERA TRANSFER APIs

| Aspera Web Services | A SOAP based web service API that allows initiation, monitoring and controlling of FASP based file transfers. |
| FASP Manager | A class library that allows initiations, monitoring and controlling of FASP based file transfers. |
| Aspera Multicast SDK | A Java class library for initiation and management of IP multicast based data transmissions using Aspera FASP-MC™. |
QUANTIFYING USER LIKING AND WATCHING BEHAVIOURS

Sashikumar Venkataraman, Craig Carmichael
Rovi Corporation
Our Goal

• Determine a notion of “likeness”
  – Did user really like a movie/show?
  – Or watched due to habit/popular/availability?

• Quantify meta-content attributes using usage
  – How much is an asset “comedy” or “sci-fi”

• Introduce “discovery” aspect for long-tail
Overall Approach

• Unified collaborative filtering model
  – consumption data + user-ratings + meta-data

• Convert implicit watch to explicit ratings
  – Implicit similarity in $D_1 \rightarrow$ explicit similarity in $D_2$

• Users and Assets mapped to vectors
  – Meta-content similarity $\leftrightarrow$ Usage similarity
Merge into Knowledge Graph

• Augment meta-data
  – Keywords/genres/deep-descriptors
  – Dynamic relevance/popularity

• Co-relate assets from multiple sources
  – Helps when no explicit rating or sparse usage data
Twitter, Facebook, Google Trends, Tv.com, Nielsens, top40-charts.com, etc.

Reuters, CBS, NBC, BBC, ABC, The Guardian, NYTimes, CNN, etc.

ESPN, NFL, NBA, NCAA, MLB, IPL, UEFA, etc.

Rotten Tomatoes, Metacritic, etc.

ESPN, NFL, NBA, NCAA, MLB, IPL, UEFA, etc.

YouTube, Amazon, HBO, Hulu, Vudu.

Wikipedia, Wiktionary, Freebase, TMDB, etc.

Over 50K channels in 25K different lineups
Create meta-content factors

• Meta content -> k-dimensional vector (w2v)
• Each keyword/genre becomes a vector
  – Form clusters based on closeness
  – Ignore clusters with low weight
• Provides cold-start even without usage
Fuse in consumption data

- Item-vectors get closer based on usage

$$\text{Sim}(i, j, d) = \frac{\sum_{u \in \{i, j\}}^{U(d)} \left( r^X_{u}(d) - \bar{r}^X_{i}(d) \right) \left( r^X_{u}(d) - \bar{r}^X_{j}(d) \right)}{\sqrt{\sum_{u \in \{i, j\}}^{U(d)} \left( r^X_{u}(d) - \bar{r}^X_{i}(d) \right)^2} \sqrt{\sum_{u \in \{i, j\}}^{U(d)} \left( r^X_{u}(d) - \bar{r}^X_{j}(d) \right)^2}}$$

- Relate implicit watching and explicit ratings

$$\text{Error} = \sum_{d=1}^{D_{\text{impl}}} \sum_{i=1}^{N} \sum_{j=1}^{N} (X\text{Sim}(i, j) - I\text{Sim}(i, j, d))^2$$
Error and Back-propagate

Similarity Layer (Pearson, cosine similarity, ...)

Prediction layer (Linear estimator with sigmoid)

Raw usage information (duration, # episodes, price user paid, etc.)
Co-relate Meta-content and Usage

• Meta-content similarity - $m_{ij}$
  – Using keywords/genres (unknown weights)
  – Possible hidden factors

• Usage based similarity - $s_{ij}$
  – implicit and explicit usage information

• Back-propagate error
  \[ E = \sum_{ij} (s_{ij} - m_{ij})^2 \]
Baseline Results

• Compare recall with different number of meta-content factors

<table>
<thead>
<tr>
<th>Model</th>
<th>Recall@K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K=5</td>
</tr>
<tr>
<td>Probsim</td>
<td>0.192</td>
</tr>
<tr>
<td>WVCF, F=20</td>
<td>0.163</td>
</tr>
<tr>
<td>WVCF, F=50</td>
<td>0.163</td>
</tr>
<tr>
<td>WVCF, F=100</td>
<td>0.186</td>
</tr>
<tr>
<td>WVCF, F=300</td>
<td>0.206</td>
</tr>
</tbody>
</table>
Discovery of long-tail programs

• Hidden-gems - usually never recommended
• Though popularity is low, (+ve : -ve) is high

<table>
<thead>
<tr>
<th>Title(i)</th>
<th>N(i)</th>
<th>Rating Avg(i)</th>
<th>G(i) +ve rating</th>
<th>B(i) -ve rating</th>
<th>P(i) = G(i)/N(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intimate Strangers</td>
<td>126</td>
<td>3.655</td>
<td>117</td>
<td>2</td>
<td>0.9832</td>
</tr>
<tr>
<td>Follow the Fleet</td>
<td>118</td>
<td>3.712</td>
<td>111</td>
<td>2</td>
<td>0.9823</td>
</tr>
<tr>
<td>Ax, The (couperet, I.e)</td>
<td>114</td>
<td>3.807</td>
<td>107</td>
<td>2</td>
<td>0.9817</td>
</tr>
<tr>
<td>One Man Hand</td>
<td>105</td>
<td>3.895</td>
<td>98</td>
<td>2</td>
<td>0.9800</td>
</tr>
<tr>
<td>Angels' Share</td>
<td>105</td>
<td>3.710</td>
<td>96</td>
<td>2</td>
<td>0.9796</td>
</tr>
<tr>
<td>Queen of Versailles, The</td>
<td>144</td>
<td>3.628</td>
<td>137</td>
<td>3</td>
<td>0.9786</td>
</tr>
<tr>
<td>Yes Men Fix the World, The</td>
<td>194</td>
<td>3.843</td>
<td>179</td>
<td>4</td>
<td>0.9781</td>
</tr>
</tbody>
</table>
Popularity buckets of user profiles

• Typical recs are Q1
• Introduce “loss” in scoring metric

\[
\text{LOSS}(u) = 1 - \alpha \sum_{b=1}^{\text{Nbins}} |P(u, b) - p(u, b)|
\]

• More hidden gems get now recommended
Comparing Watch/Like Models

![Graphs comparing Watch/Like models.](image-url)
Finding hidden emotions/moods

<table>
<thead>
<tr>
<th>Rank</th>
<th>Overall</th>
<th>Popular</th>
<th>Gems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>contrary</td>
<td>lazy</td>
<td>honest</td>
</tr>
<tr>
<td>2</td>
<td>angry</td>
<td>inconsiderate</td>
<td>contrary</td>
</tr>
<tr>
<td>3</td>
<td>brilliant</td>
<td>pity</td>
<td>cordial</td>
</tr>
<tr>
<td>4</td>
<td>wonder</td>
<td>obnoxious</td>
<td>powerful</td>
</tr>
<tr>
<td>5</td>
<td>bored</td>
<td>annoyed</td>
<td>lucky</td>
</tr>
<tr>
<td>6</td>
<td>insightful</td>
<td>woeful</td>
<td>wise</td>
</tr>
<tr>
<td>7</td>
<td>rough</td>
<td>haughty</td>
<td>aggressive</td>
</tr>
<tr>
<td>8</td>
<td>affectionate</td>
<td>uncertain</td>
<td>sad</td>
</tr>
<tr>
<td>9</td>
<td>weird</td>
<td>alive</td>
<td>nasty</td>
</tr>
<tr>
<td>10</td>
<td>folksy</td>
<td>silly</td>
<td>folksy</td>
</tr>
</tbody>
</table>
Future Directions

• Using latent factors and nearest neighbors to close gap between implicit and explicit ratings
• Explore hidden factors in meta-content
• Formalize “discovery” notion in recall metrics
Thank you