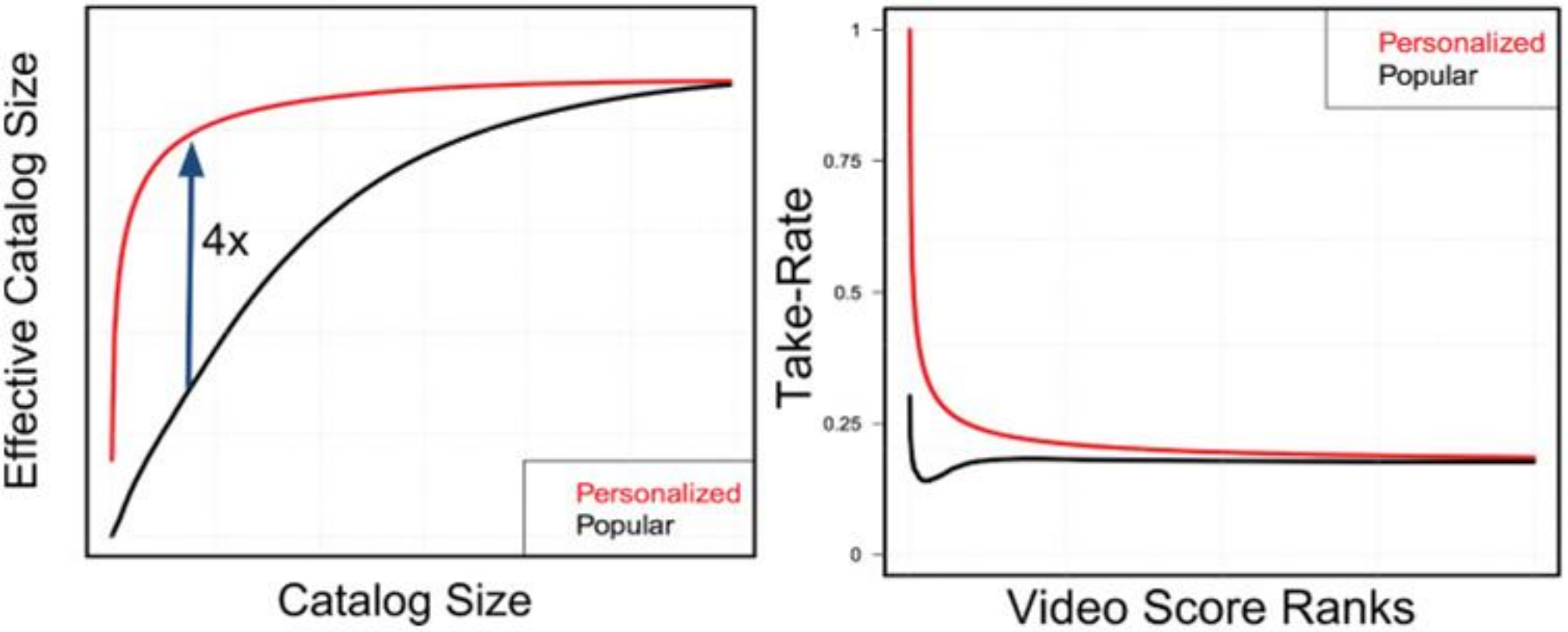
13 The Netix Recommender System Algorithms Business Value And Innovation Carlos A Gomezuribe And Neil Hunt


NETFLIX ORIGINAL
ORANGE is the new BLACK
Watch Season 3 on June 12
Piper Chapman doesn't deserve her prison sentence. Of course, every one of her fellow inmates thinks the same thing.
Popular on Netflix
sense8
Trending Now
HOUSE OF CARDS
HOMELAND
NETFLIX


NETFLIX ORIGINAL
ORANGE is the new BLACK
Watch Season 3 Now
Piper Chapman doesn't deserve her prison sentence. Of course, every one of her fellow inmates thinks the same thing.
Continue Watching
sense8
Recently Added
CAKE
ANTARCTICA
NETFLIX


usual
Titles related to The Usual Suspects
a b c d e f
g h i j k l
m n o p q r
s t u v w x
y z 1 2 3 4
5 6 7 8 9 0
dogs
PULP FICTION
ROUNDERS
HOUSE of CARDS
Back


fren
a b c d e f
g h i j k l
m n o p q r
s t u v w x
y z 1 2 3 4
5 6 7 8 9 0
French Stewart
Kate French
Edward Lewis French
Dawn French
Patricia French
Harold French
FRENEMIES
FRENCH
FRENCH II
Titles related to French Movies
Amélie
Young & Beautiful
Le Chef
Back


Effective Catalog Size
4x
Personalized
Popular
Catalog Size
Take-Rate
1
0.75
0.5
0.25
0
Personalized
Popular
Video Score Ranks


The Others
2001
While awaiting the return of her soldier husband from World War II, a devout Christian mother of two begins to suspect the family house is haunted.
Suspenseful Movies
THE OTHERS
THE QUIET AMERICAN
CARLOS
THE SIXTH SENSE
Because you watched Frontline: Secret State of North Korea
LOSING IRAQ
BATTLEGROUND
KORENGAL
NETFLIX


The Universe: Collection
2007
1 Season
In a flash, we went from nothing to everything. Discover what happened and experience the greatest story of all.
Top Picks for Carlos
THE UNIVERSE
JESSIE
Romantic Movies
NETFLIX

House of Cards

House of Cards

70175658

70136120

70153404

70295061

80017537

80045789

80041601

70219642

70153382

70143642

70136120

80025384

70153404

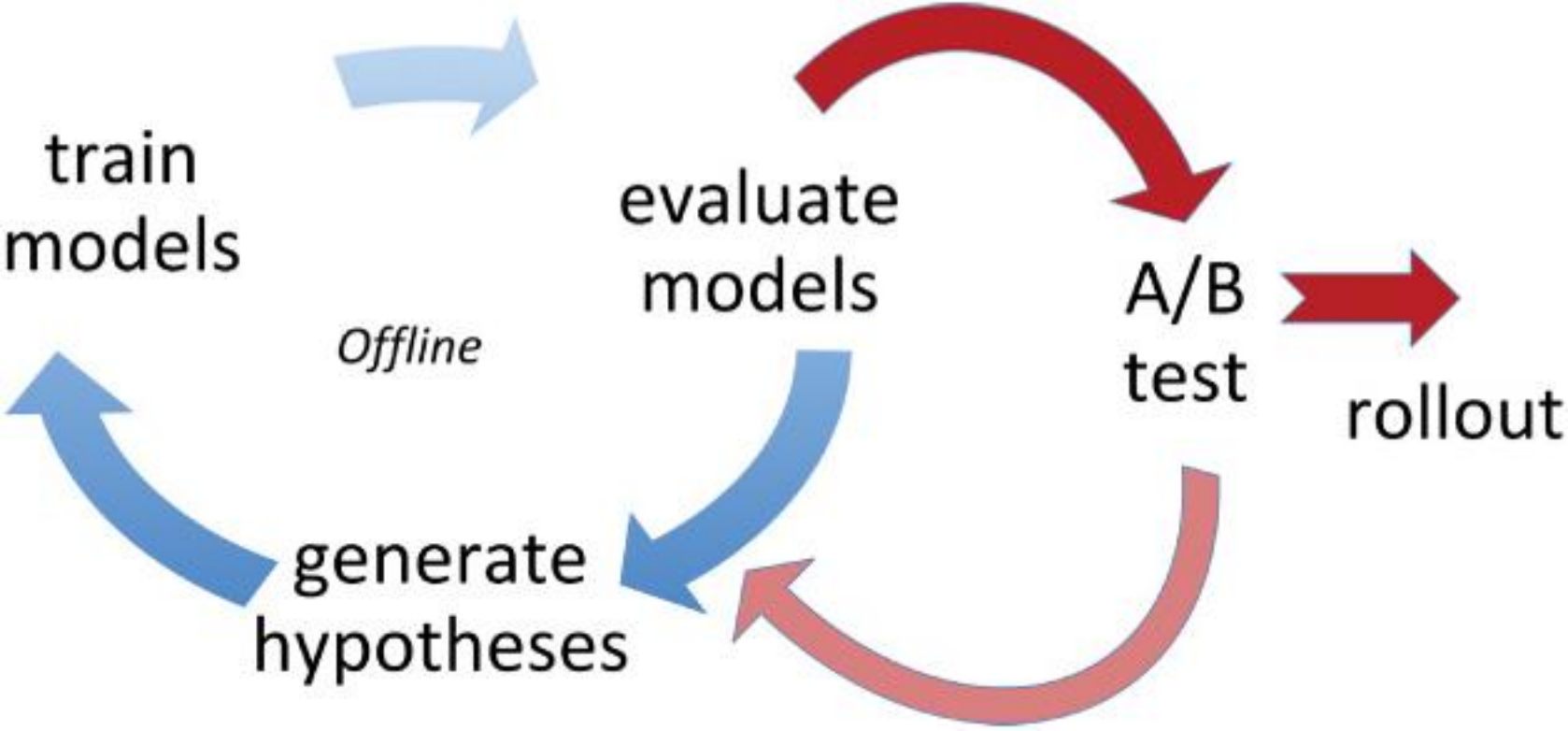
70136124

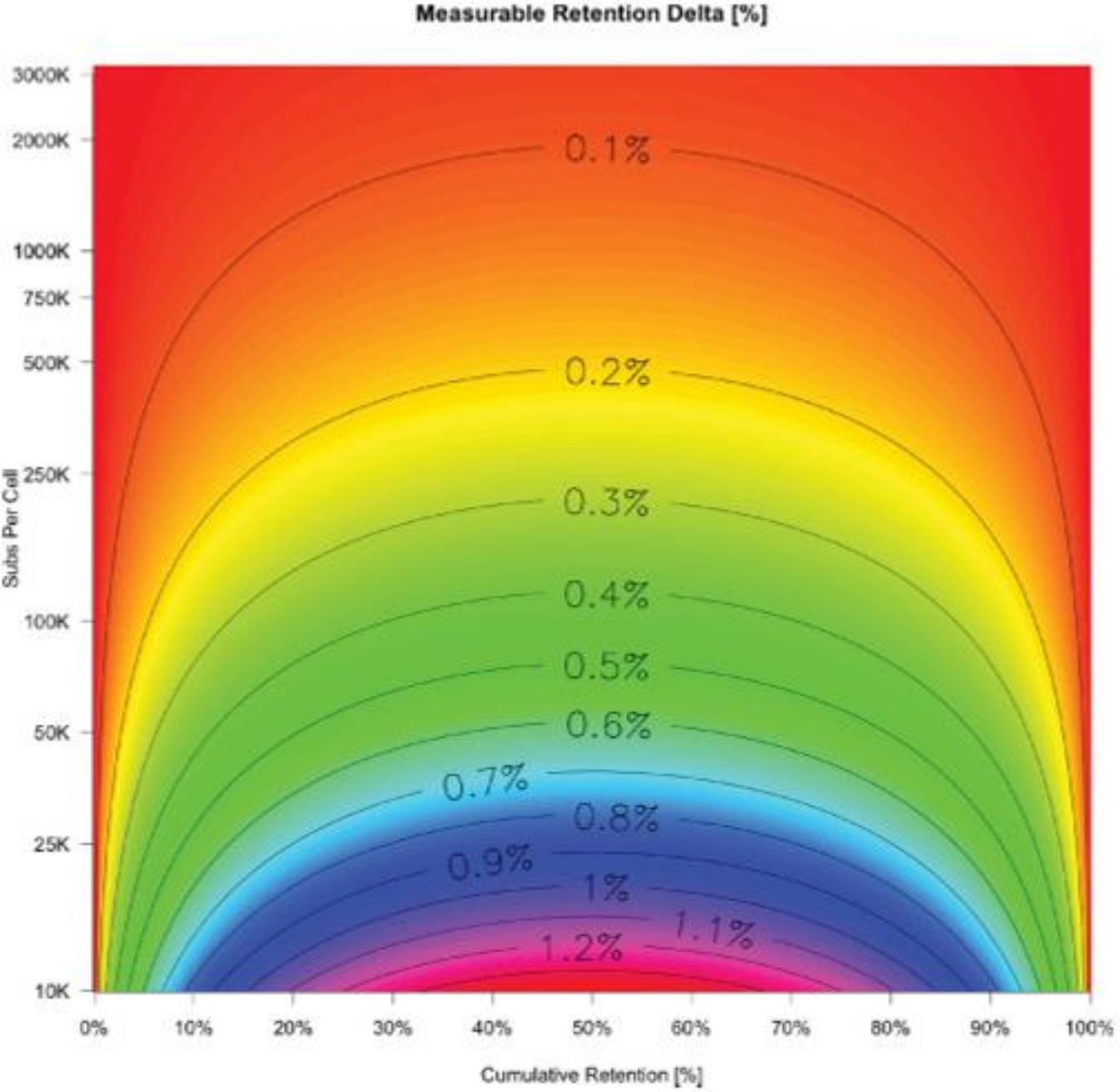
80017537

80018294

70140358

70136141


train models
evaluate models
A/B test
rollout
Offline
generate hypotheses


Measurable Retention Delta [%]
3000K
2000K
1000K
750K
500K
250K
100K
50K
25K
10K
Subs Per Cell
0.1%
0.2%
0.3%
0.4%
0.5%
0.6%
0.7%
0.8%
0.9%
1%
1.1%
1.2%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Cumulative Retention [%]

The Netflix Recommender System: Algorithms, Business Value, and Innovation

CARLOS A. GOMEZ-URIBE and NEIL HUNT, Netflix, Inc.

This article discusses the various algorithms that make up the Netflix recommender system, and describes its business purpose. We also describe the role of search and related algorithms, which for us turns into a recommendations problem as well. We explain the motivations behind and review the approach that we use to improve the recommendation algorithms, combining A/B testing focused on improving member retention and medium term engagement, as well as offline experimentation using historical member engagement data. We discuss some of the issues in designing and interpreting A/B tests. Finally, we describe some current areas of focused innovation, which include making our recommender system global and language aware.

Categories and Subject Descriptors: C.2.2 [**Recommender Systems**]: Machine Learning

General Terms: Algorithms, Recommender Systems, A/B Testing, Product Innovation

Additional Key Words and Phrases: Recommender systems

ACM Reference Format:
Carlos A. Gomez-Uribe and Neil Hunt. 2015. The Netflix recommender system: Algorithms, business value, and innovation. ACM Trans. Manage. Inf. Syst. 6, 4, Article 13 (December 2015), 19 pages.
DOI: http://dx.doi.org/10.1145/2843948

1. INTRODUCTION

Storytelling has always been at the core of human nature. Major technological breakthroughs that changed society in fundamental ways have also allowed for richer and more engaging stories to be told. It is not hard to imagine our ancestors gathering around a fire in a cave and enjoying stories that were made richer by supporting cave paintings. Writing, and later the printing press, led to more varied and richer stories that were distributed more widely than ever before. More recently, television led to an explosion in the use and distribution of video for storytelling. Today, all of us are lucky to be witnessing the changes brought about by the Internet. Like previous major technological breakthroughs, the Internet is also having a profound impact on storytelling.

Netflix lies at the intersection of the Internet and storytelling. We are inventing Internet television. Our main product and source of revenue is a subscription service that allows members to stream any video in our collection of movies and TV shows at any time on a wide range of Internet-connected devices. As of this writing, we have more than 65 million members who stream more than 100 million hours of movies and TV shows per day.

The Internet television space is young and competition is ripe, thus innovation is crucial. A key pillar of our product is the recommender system that helps our members find videos to watch in every session. Our recommender system is not one algorithm,

Authors' address: C. A. Gomez-Uribe and N. Hunt, 100 Winchester Cir, Los Gatos, CA 95032.

ACM 2158-656X/2015/12-ART13 $15.00
DOI: http://dx.doi.org/10.1145/2843948

but rather a collection of different algorithms serving different use cases that come together to create the complete Netflix experience. We give an overview of the various algorithms in our recommender system in Section 2, and discuss their business value in Section 3. We describe the process that we use to improve our algorithms in Section 4, review some of our key open problems in Section 5, and present our conclusions in Section 6.

2. THE NETFLIX RECOMMENDER SYSTEM

Internet TV is about choice: what to watch, when to watch, and where to watch, compared with linear broadcast and cable systems that offer whatever is now playing on perhaps 10 to 20 favorite channels. But humans are surprisingly bad at choosing between many options, quickly getting overwhelmed and choosing "none of the above" or making poor choices (e.g., see Schwartz [2015]). At the same time, a benefit of Internet TV is that it can carry videos from a broader catalog appealing to a wide range of demographics and tastes, and including niche titles of interest only to relatively small groups of users.

Consumer research suggests that a typical Netflix member loses interest after perhaps 60 to 90 seconds of choosing, having reviewed 10 to 20 titles (perhaps 3 in detail) on one or two screens. The user either finds something of interest or the risk of the user abandoning our service increases substantially. The recommender problem is to make sure that on those two screens each member in our diverse pool will find something compelling to view, and will understand why it might be of interest.

Historically, the Netflix recommendation problem has been thought of as equivalent to the problem of predicting the number of stars that a person would rate a video after watching it, on a scale from 1 to 5. We indeed relied on such an algorithm heavily when our main business was shipping DVDs by mail, partly because in that context, a star rating was the main feedback that we received that a member had actually watched the video. We even organized a competition aimed at improving the accuracy of the rating prediction, resulting in algorithms that we use in production to predict ratings to this day [Netflix Prize 2009].

But the days when stars and DVDs were the focus of recommendations at Netflix have long passed. Now, we stream the content, and have vast amounts of data that describe what each Netflix member watches, how each member watches (e.g., the device, time of day, day of week, intensity of watching), the place in our product in which each video was discovered, and even the recommendations that were shown but not played in each session. These data and our resulting experiences improving the Netflix product have taught us that there are much better ways to help people find videos to watch than focusing only on those with a high predicted star rating.

Now, our recommender system consists of a variety of algorithms that collectively define the Netflix experience, most of which come together on the Netflix homepage. This is the first page that a Netflix member sees upon logging onto one's Netflix profile on any device (TV, tablet, phone, or browser)—it is the main presentation of recommendations, where 2 of every 3 hours streamed on Netflix are discovered.

An example of our current TV homepage is shown in Figure 1. It has a matrix-like layout. Each entry in the matrix is a recommended video, and each row of videos contains recommendations with a similar "theme." Rows are labeled according to their theme to make the theme transparent and (we think) more intuitive to our members.

2.1. Personalized Video Ranker: PVR

There are typically about 40 rows on each homepage (depending on the capabilities of the device), and up to 75 videos per row; these numbers vary somewhat across devices because of hardware and user experience considerations. The videos in a given row

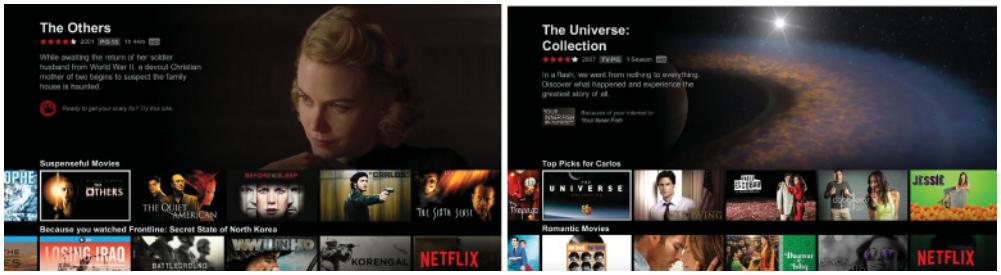


Fig. 1. (Left) An example of the page of recommendations, showing two of the roughly 40 rows of recommendations on that page. Suspenseful Movies is an example of a genre row driven by the PVR algorithm (Section 2.1). The second row is a Because You Watched row driven by the sims algorithm (Section 2.5). (Right) A homepage showing the Top Picks row driven by the Top N algorithm (Section 2.2). Romantic Movies is a genre row driven by PVR.

Fig. 2. (Left) Two more rows of recommendations on a homepage. The popularity-heavy Popular row and the Trending Now row (Section 2.3) focus on the latest viewing trends. (Right) A homepage for a Continue Watching session with a Continue Watching row (Section 2.4).

typically come from a single algorithm. Genre rows such as *Suspenseful Movies*, shown on the left of Figure 1, are driven by our *personalized video ranker* (PVR) algorithm. As its name suggests, this algorithm orders the entire catalog of videos (or subsets selected by genre or other filtering) for each member profile in a personalized way. The resulting ordering is used to select the order of the videos in genre and other rows, and is the reason why the same genre row shown to different members often has completely different videos. Because we use PVR so widely, it must be good at general-purpose relative rankings throughout the entire catalog; this limits how personalized it can actually be. Equivalently, PVR works better when we blend personalized signals with a pretty healthy dose of (unpersonalized) popularity, which we use to drive the recommendations in the Popular row shown on the left of Figure 2. See Amatriain and Basilico [2012] for more on personalized video ranking.

2.2. Top-N Video Ranker

We also have a *Top N* video ranker that produces the recommendations in the Top Picks row shown on the right of Figure 1. The goal of this algorithm is to find the best few personalized recommendations in the entire catalog for each member, that is, focusing only on the head of the ranking, a freedom that PVR does not have because it gets used to rank arbitrary subsets of the catalog. Accordingly, our Top N ranker is optimized and evaluated using metrics and algorithms that look only at the head of the catalog ranking that the algorithm produces, rather than at the ranking for the entire catalog (as is the case with PVR). Otherwise the Top N ranker and PVR share similar

attributes, for example, combining personalization with popularity, and identifying and incorporating viewing trends over different time windows ranging from a day to a year.

2.3. Trending Now

We have also found that shorter-term temporal trends, ranging from a few minutes to perhaps a few days, are powerful predictors of videos that our members will watch, especially when combined with the right dose of personalization, giving us a *trending* ranker [Padmanabhan et al. 2015] used to drive the Trending Now row (left of Figure 2). There are two types of trends that this ranker identifies nicely: (1) those that repeat every several months (e.g., yearly) yet have a short-term effect when they occur, such as the uptick of romantic video watching during Valentine's Day in North America, and (2) one-off, short-term events, for example, a big hurricane with an impending arrival to some densely populated area, being covered by many media outlets, driving increased short-term interest in documentaries and movies about hurricanes and other natural disasters.

2.4. Continue Watching

Given the importance of episodic content viewed over several sessions, as well as the freedom to view nonepisodic content in small bites, another important video ranking algorithm is the *continue watching ranker* that orders the videos in the Continue Watching row (see right of Figure 2). Most of our rankers sort unviewed titles on which we have only inferred information. In contrast, the continue watching ranker sorts the subset of recently viewed titles based on our best estimate of whether the member intends to resume watching or rewatch, or whether the member has abandoned something not as interesting as anticipated. The signals that we use include the time elapsed since viewing, the point of abandonment (mid-program vs. beginning or end), whether different titles have been viewed since, and the devices used. In general, our different video ranking algorithms use different mathematical and statistical models, different signals and data as input, and require different model trainings designed for the specific purpose each ranker serves.

2.5. Video-Video Similarity

Because You Watched (BYW) rows are another type of categorization. A BYW row anchors its recommendations to a single video watched by the member. The *video-video similarity* algorithm, which we refer to simply as "sims," drives the recommendations in these rows. An example row is shown on the left of Figure 1. The sims algorithm is an unpersonalized algorithm that computes a ranked list of videos—the similars—for every video in our catalog. Even though the sims ranking is not personalized, the choice of which BYW rows make it onto a homepage *is* personalized, and the subset of BYW videos recommended in a given BYW row benefits from personalization, depending on what subsets of the similar videos we estimate that the member would enjoy (or has already watched).

2.6. Page Generation: Row Selection and Ranking

The videos chosen for each row represent our estimate of the best choices of videos to put in front of a specific user. But most members have different moods from session to session, and many accounts are shared by more than one member of a household. By offering a diverse selection of rows, we hope to make it easy for a member to skip videos that would be good choices for a different time, occasion, or member of the household, and quickly identify something immediately relevant.

The *page generation* algorithm uses the output of all the algorithms already described to construct every single page of recommendations, taking into account the relevance of

each row to the member as well as the diversity of the page. A typical member has tens of thousands of rows that could go on one's homepage, making it challenging to manage the computations required to evaluate them. For this reason, before 2015, we used a rule-based approach that would define what type of row (e.g., genre row, BYW row, Popular row) would go in each vertical position of the page. This page layout was used to construct all homepages for all members. Today, we have a fully personalized and mathematical algorithm that can select and order rows from a large pool of candidates to create an ordering optimized for relevance and diversity. Our current algorithm does not use a template, thus is freer to optimize the experience, for example, choosing not to have any BYW row for a given homepage and devoting half of the page to BYW rows for another homepage. A recent blogpost [Alvino and Basilico 2015] on this algorithm discusses it in more detail.

2.7. Evidence

Together, these algorithms make up the complete Netflix recommender system. But there are other algorithms, such as *evidence selection* ones, that work together with our recommendation algorithms to define the Netflix experience and help our members determine if a video is right for them. We think of evidence as all the information we show on the top left of the page, including the predicted star rating that was the focus on the Netflix prize; the synopsis; other facts displayed about the video, such as any awards, cast, or other metadata; and the images we use to support our recommendations in the rows and elsewhere in the UI. Evidence selection algorithms evaluate all the possible evidence items that we can display for every recommendation, to select the few that we think will be most helpful to the member viewing the recommendation. For example, evidence algorithms decide whether to show that a certain movie won an Oscar or instead show the member that the movie is similar to another video recently watched by that member; they also decide which image out of several versions use to best support a given recommendation.

2.8. Search

Our recommender system is used on most screens of the Netflix product beyond the homepage, and in total influences choice for about 80% of hours streamed at Netflix. The remaining 20% comes from search, which requires its own set of algorithms. Members frequently search for videos, actors, or genres in our catalog; we leverage information retrieval and related techniques to find the relevant videos and display them to our members. However, because members also often search for videos, actors, or genres that are not in our catalog (Figure 3, left) or for general concepts (Figure 3, right), even search turns into a recommendation problem. In such cases, search recommends videos for a given query as alternative results for a failed search. The extreme crudeness of text input on a TV screen means that interpreting partial queries of two or three letters in the context of what we know about the searching member's taste is also especially important for us.

The search experience is built around several algorithms. One algorithm attempts to find the videos that match a given query, for example, to retrieve Frenemies for the partial query "fren." Another algorithm predicts interest in a concept given a partial query, for example, identifying the concept French Movies for the query "fren." A third algorithm finds video recommendations for a given concept, for example, to populate the videos recommended under the concept French Movies. Our search algorithms combine play data, search data, and metadata to arrive at the results and recommendations that we offer.

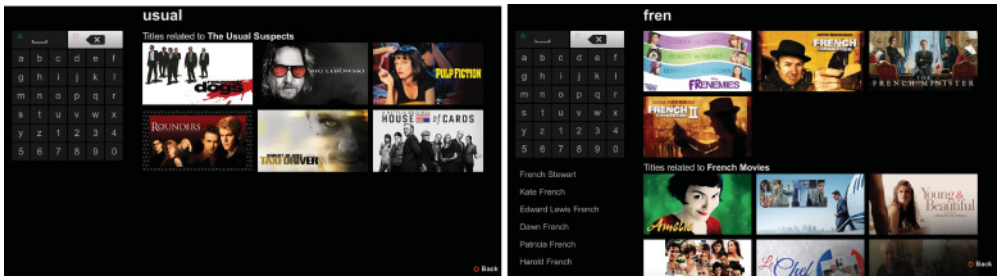


Fig. 3. (Left) Search experience for query "usual," presumably for the movie "The Usual Suspects" which was not available at Netflix at the time of the query. The results are instead recommendations based on the query entered. (Right) Search experience for the query "fren," showing standard search results at the top for videos with names that contain the substring "fren," people results on the lower left, and search recommendations based on the guess that the intent was searching for French Movies.

2.9. Related Work

Each of the algorithms in our recommender system relies on statistical and machine-learning techniques. This includes both supervised (classification, regression) and unsupervised approaches (dimensionality reduction through clustering or compression, e.g., through topic models)—Hastie et al. [2011] and Murphy [2012] provide good overviews of such techniques, and Blei et al. [2003] and Teh et al. [2006] are good examples of useful topic models—as well as adaptations that are specialized to the recommender systems domain, particularly around matrix factorization. A good introduction to factorization approaches is Koren et al. [2009], with more in-depth material found in Koren [2008]. Some useful generalizations of the more traditional factorization approaches include factorization machines [Rendle 2010], methods that reduce the number of parameters in the models (e.g., Paterek [2007]), and connections to probabilistic graphical models (e.g., Mnih and Salakhutdinov [2007]) that are easy to expand on to suit different problems.

3. BUSINESS VALUE

We seek to grow our business on an enormous scale, that is, becoming a producer and distributor of shows and movies with a fully global reach. We develop and use our recommender system because we believe that it is core to our business for a number of reasons. Our recommender system helps us win *moments of truth:* when a member starts a session and we help that member find something engaging within a few seconds, preventing abandonment of our service for an alternative entertainment option.

Personalization enables us to find an audience even for relatively niche videos that would not make sense for broadcast TV models because their audiences would be too small to support significant advertising revenue, or to occupy a broadcast or cable channel time slot. This is very evident in our data, which show that our recommender system spreads viewing across many more videos much more evenly than would an unpersonalized system. To make this more precise, we introduce a specific metric next.

The effective catalog size (ECS) is a metric that describes how spread viewing is across the items in our catalog. If most viewing comes from a single video, it will be close to 1. If all videos generate the same amount of viewing, it is close to the number of videos in the catalog. Otherwise it is somewhere in between. The ECS is described in more detail in Appendix A.

Without personalization, all our members would get the same videos recommended to them. The black line in left plot in Figure 4 shows how the ECS without personalization increases as the number of videos we include in our data increases, starting with the

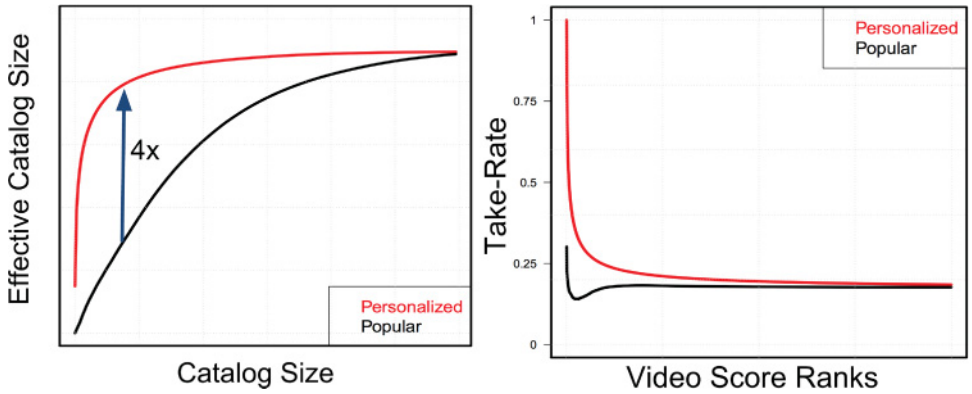


Fig. 4. (Left) The black line is the effective catalog size (ECS) plotted as a function of the number of most popular videos considered in the catalog, ranging from 1 through N (the number of videos in the catalog) on the x-axis. The red line is the effective catalog size for the first k PVR-ranked videos for each member. At a PVR rank corresponding to the median rank across all plays, the ECS in red is roughly 4 times that in black. The values in the x and y axis are not shown for competitive reasons. For more details, see Appendix A. (Right) The take-rate from the first k ranks, as a function of the video popularity rank in black, and as a function of the PVR rank in red. The y-values were normalized through division by a constant so that the maximum value shown equalled 1.

most popular video and adding the next popular video as we move to the right on the x-axis. The red line on the same plot, on the other hand, shows how the ECS grows not as a function of the videos that we include, but rather as a function of the number of PVR ranks that we include to capture personalization. Although the difference in the amount of catalog exploration with and without personalization is striking, it alone is not compelling enough. After all, perhaps we would spread viewing even more evenly by offering completely random recommendations for each session.

More important, personalization allows us to significantly increase our chances of success when offering recommendations. One metric that gets at this is the take-rate—the fraction of recommendations offered resulting in a play. The two lines in the right plot in Figure 4 show the take-rate, one as a function of a video's popularity, and the other as a function of a video's PVR rank. The lift in take-rate that we get from recommendations is substantial. But, most important, when produced and used correctly, recommendations lead to meaningful increases in overall engagement with the product (e.g., streaming hours) and lower subscription cancellations rates.

Our subscriber monthly churn is in the low single-digits, and much of that is due to payment failure, rather than an explicit subscriber choice to cancel service. Over years of development of personalization and recommendations, we have reduced churn by several percentage points. Reduction of monthly churn both increases the lifetime value of an existing subscriber, and reduces the number of new subscribers we need to acquire to replace cancelled members. We think the combined effect of personalization and recommendations save us more than $1B per year.

4. IMPROVING OUR ALGORITHMS

Good businesses pay attention to what their customers have to say. But what customers ask for (as much choice as possible, comprehensive search and navigation tools, and more) and what actually works (a few compelling choices simply presented) are very different.

Using our own intuition, even collective intuition, to choose the best variant of a recommendation algorithm also often yields the wrong answer, and is frequently simply

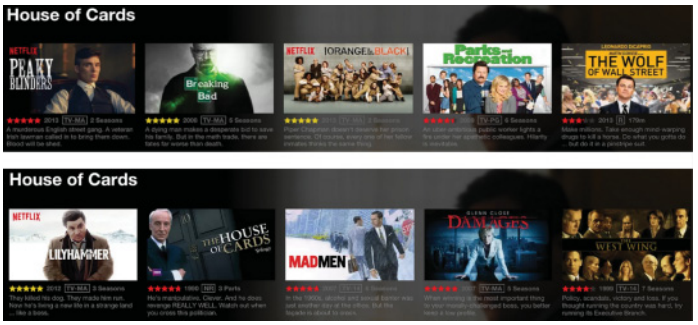


Fig. 5. Two sets of video similars for "House of Cards." The bottom ones seem more relevant, but turn out to be worse than the ones shown on top that have a stronger popularity influence.

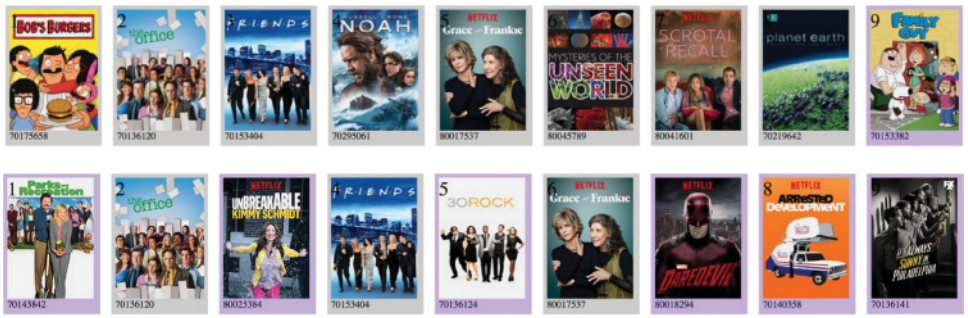


Fig. 6. The 9 highest ranking videos in the catalog according to two variants of the PVR algorithm, evaluated for one of the authors.

impossible, particularly when trying to tell good from great recommendations apart. For an example of intuition failure, Figure 5 shows two sets of videos similar to "House of Cards." People often intuitively think the bottom ones are better because they seem more relevant, for example, they include the original version of "House of Cards." Yet the other set of similars shown at the top turn out to be better according to A/B testing (see Section 4.1).

Another example, Figure 6, shows the highest-ranked PVR videos in the catalog for one of the authors, and even that author has no intuition based on these rankings about which one offers better choices for him. Assessing the ranking for other people is even harder. So, how do we know when an algorithm variant is better or worse than another?

4.1. Choosing Metrics For A/B Testing

Our subscription business model suggests a framework to find the answer. Because our revenue comes exclusively from the monthly subscription fee that our current members pay, and we make it very simple to cancel the subscription at any time, we think that maximizing revenue through product changes is fairly equivalent to maximizing the value that our members derive from our service. Revenue is proportional to the number of members, and three processes directly affect this number: the acquisition rate of new members, member cancellation rates, and the rate at which former members rejoin.

If we create a more compelling service by offering better personalized recommendations, we induce members who were on the fence to stay longer, and improve retention. In addition, all members with an improved experience (not just those on the fence) may be more enthusiastic when describing Netflix to their friends, strongly influencing new subscriber acquisition through word-of-mouth effects. Both recall of a better

experience and stronger word-of-mouth may influence former members to rejoin more quickly. While we can measure retention directly (and changes in retention through A/B testing), we have no reliable way to measure word-of-mouth for different algorithm variants because its effect, by definition, goes beyond those people who experienced a variant of Netflix.

Changes to the product directly impact only current members; thus, the main measurement target of changes to our recommendation algorithms is improved member retention. That said, our retention rates are already high enough that it takes a very meaningful improvement to make a retention difference of even 0.1% (10 basis points). However, we have observed that improving engagement—the time that our members spend viewing Netflix content—is strongly correlated with improving retention. Accordingly, we design randomized, controlled experiments, often called A/B tests, to compare the medium-term engagement with Netflix along with member cancellation rates across algorithm variants. Algorithms that improve these A/B test metrics are considered better. Equivalently, we build algorithms toward the goal of maximizing medium-term engagement with Netflix and member retention rates.

Specifically, our A/B tests randomly assign different members to different experiences that we refer to as *cells*. For example, each cell in an A/B test could map to a different video similars algorithm, one of which reflects the default (often called "production") algorithm to serve as the *control cell* in the experiment—other cells in the test are the *test cells*. We then let the members in each cell interact with the product over a period of months, typically 2 to 6 months. Finally, we analyze the resulting data to answer several questions about member behavior from a statistical perspective, including:

—Are members finding the part of the product that was changed relative to the control more useful? For example, are they finding more videos to watch from the video similars algorithm than in the control?
—Are members in a test cell streaming more on Netflix than in the control? For example, is the median or other percentile of hours streamed per member for the duration of the test higher in a test cell than in the control?[1]
—Are members in a test cell retaining their Netflix subscription more than members in the control?

When a test cell is a clear improvement over the current experience, we see members engaging more with the part of the product that was changed (a local engagement metrics win), more with the Netflix product overall (an overall engagement win), and higher retention rates (a clear overall win). While we have found multiple clear wins per year every year, we see more overall engagement wins that are not large enough to affect retention rates, and even more local engagement wins that do not change overall streaming or retention rates (e.g., because they simply cannibalize streaming from other parts of the product, or because they increase overall engagement or retention rates by too small of an amount for us to detect with any reasonable statistical confidence given the test's sample size).

We design our A/B tests to give a consistent product experience to each member in the test for its duration. A more conventional alternative would be to randomly choose for each Netflix session which algorithmic experience to offer, a design with better statistical performance for local metrics (e.g., see Chapelle et al. [2012]) but without the possibility of measuring changes to the overall engagement with the entire product or retention rates over many sessions.

[1]The number of hours streamed on Netflix during an A/B test is a nonnegative real number for every member. We collect them in every cell to create an empirical probability distribution of streaming hours for every cell in the test. We then compare these distributions across cells.

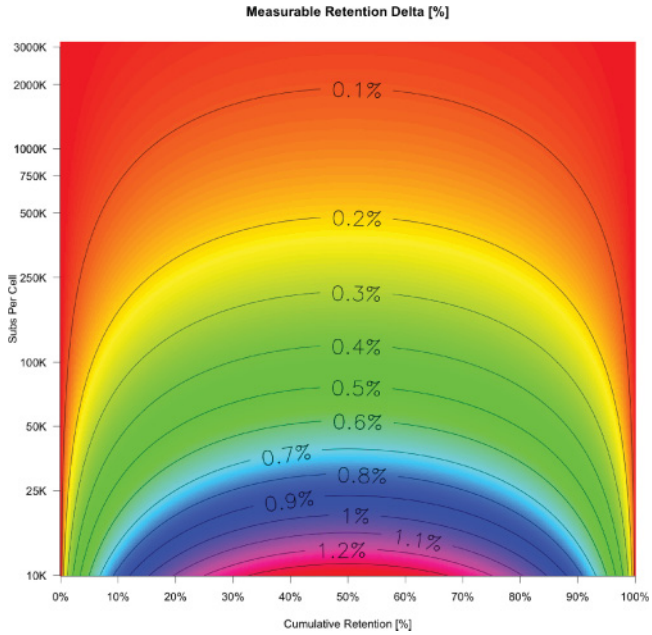


Fig. 7. A plot for the minimum retention delta that can be measured with statistical confidence, as a function of the average retention per cell and the cell size.

4.2. Test Cell Sizes for Statistical Validity

We use statistics as a guide to whether we have enough data to conclude that there is a difference in an A/B test metric across cells. As an example, suppose that we find that after two months, a fraction p_c and p_t of members in the control and test cell of an A/B test with 2 cells are still Netflix members, with $\Delta = p_t - p_c > 0$. Intuitively, we should trust the observed delta more the more members we have in the test. But how many members are enough to trust the test result?

The standard approach toward an answer posits a probability model that is supposed to have generated the data used to compute the metric in question, and then uses this model to estimate how much we would expect our metric to vary if we were to repeat the experiment (using the same sample size) a large number of times. For example, the fewer the percentage of repeated hypothetical experiments that the probability model thinks would yield a negative Δ, the more confidence we can have that our test cell indeed increased retention. See Appendix B for an example of such a probability model, or Siroker and Koomen [2013], Deng et al. [2013], and Pekelis et al. [2015] for more on the statistics of A/B testing.

The probability model can also be used to determine the sample size needed to measure an increase or decrease of a given magnitude with enough confidence. As an example, Figure 7 shows the size of the measurable retention delta across two test cells with the same number of members, as a function of both the average retention rates across the two cells (x-axis) and the number of members (y-axis) in each cell for the simple associated probability model in Appendix B. For example, if we find that 50% of the members in the test have retained when we compute our retention metric, then we need roughly 2 million members per cell to measure a retention delta of 50.05% to 49.95%=0.1% with statistical confidence.

4.3. Nuances of A/B Testing

A/B test results are our most important source of information for making product decisions. Most times, our tests are extremely informative. Yet, despite the statistical sophistication that goes into their design and analysis, interpreting A/B tests remains partly art. For example, we sometimes see retention wins that pass the statistical tests, but that are not supported by increases in overall or local engagement metrics. In such cases, we tend to assume a random variation not driven by our test experiences. Our common practice is to then rerun such A/B tests. We usually find that the retention wins do not repeat, unlike clearer wins supported by local and overall engagement metrics increases.

Other times, we see overall engagement increases without local metrics increases. We are similarly skeptical of those, and often repeat them as well, finding that the positive results do not repeat. The number of tests with seemingly confusing results can be decreased through more sophisticated experiment design and analysis, for example, using so-called variance reduction techniques such as stratified sampling (e.g., see Deng et al. [2013]) to make the cells in a test even more comparable to each other, for instance, in terms of attributes that are likely to correlate highly with streaming and retention rates, such as the method of payment or the device of sign-up.

4.4. Alternative Metrics

There are many other possible metrics that we could use, such as time to first play, sessions without a play, days with a play, number of abandoned plays, and more. Each of these changes, perhaps quite sensitively, with variations in algorithms, but we are unable to judge which changes are for the better. For example, reducing time to first play could be associated with presenting better choices to members; however, presenting more representative supporting evidence might cause members to skip choices that they might otherwise have played, resulting in a better eventual choice and more satisfaction, but associated with a *longer* time to first play.

4.5. Test Audience

We typically test algorithm changes on two groups of members: existing members and new members. The advantage of testing on existing members is that the sample size can be larger because we have many of them. But existing members have experienced a different version of the product in the past; suddenly changing their experience to reflect that of a test cell can yield behaviors that are influenced by their previous experience. Often, such tests actually measure the impact of the immediate change in the product, rather than the impact of the new experience itself over the medium term: if existing members have to learn a different way to accomplish a goal than they are already used to, for example, how to search for actors, change often measures negatively; if the change is manifest only as different choices, the novelty often results in exposing previously undiscovered titles, leading to a positive measurement not representative of better choices in the medium and long term.

We prefer to test on new members because they have not experienced a different version of the product before; thus, their responses tend to be indicative of the effectiveness of the alternative versions of the algorithm rather than the change from old to new, yielding cleaner measurements. A disadvantage is that we have fewer new members, only as many signups as we get during the time period when we allocate new members into a test. Another disadvantage is that we offer new members a one-month-free trial, so we see few cancellations before this free month expires and cannot measure accurate retention rates until one month after the last new member in the test joined Netflix.

4.6. Faster Innovation Through Offline Experiments

The time scale of our A/B tests might seem long, especially compared to those used by many other companies to optimize metrics, such as click-through rates. This is partly addressed by testing multiple variants against a control in each test; thus, rather than having two variants, A and B, we typically include 5 to 10 algorithm variants in each test, for example, using the same new model but different signal subsets and/or parameters and/or model trainings. This is still slow, however, too slow to help us find the best parameter values for a model with many parameters, for example. For new members, more test cells also means more days to allocate new signups into the test to have the same sample size in each cell.

Another option to speed up testing is to execute many different A/B tests at once on the same member population. As long as the variations in test experience are compatible with each other, and we judge them not to combine in a nonlinear way on the experience, we might allocate each new member into several different tests at once – for example, a similars test, a PVR algorithm test, and a search test. Accordingly, a single member might get similars algorithm version B, PVR algorithm version D, and search results version F. Over perhaps 30 sessions during the test period, the member's experience is accumulated into metrics for each of the three different tests.

But to really speed up innovation, we also rely on a different type of experimentation based on analyzing historical data. This *offline experimentation* changes from algorithm to algorithm, but it always consists of computing a metric for every algorithm variant tested that describes how well the algorithm variants fit previous user engagement.

For example, for PVR, we might have 100 different variants that differ only in the parameter values used, and that relied on data up to two days ago in their training. We then use each algorithm variant to rank the catalog for a sample of members using data up to two days ago, then find the ranks of the videos played by the members in the sample in the last two days. These ranks are then used to compute metrics for each user across variants—for example, the mean reciprocal rank, precision, and recall—that are then averaged across the members in the sample, possibly with some normalization. For a different and detailed offline metric example, used for our page construction algorithm, see Alvino and Basilico [2015]. Offline experiments allow us to iterate quickly on algorithm prototypes, and to prune the candidate variants that we use in actual A/B experiments. The typical innovation flow is shown in Figure 8.

As appealing as offline experiments are, they have a major drawback: they assume that members would have behaved the same way, for example, playing the same videos, if the new algorithm being evaluated had been used to generate the recommendations. Thus, for instance, a new algorithm that results in very different recommendations from the production algorithm is unlikely to find that its recommendations have been played more than the corresponding recommendations from the production algorithm that actually served the recommendations to our members. This suggests that offline experiments need to be interpreted in the context of how different the algorithms being tested are from the production algorithm. However, it is unclear what distance metric across algorithms can lead to better offline experiment interpretations that will correlate better with A/B test outcomes, since the latter is what we are after. Thus, while we do rely on offline experiments heavily, for lack of a better option, to decide when to A/B test a new algorithm and which new algorithms to test, we do not find them to be as highly predictive of A/B test outcomes as we would like.

4.7. Estimating Word-of-Mouth Effects

As described earlier, improving the experience for members might be expected to generate stronger word-of-mouth; this, by definition, has influence beyond the boundaries

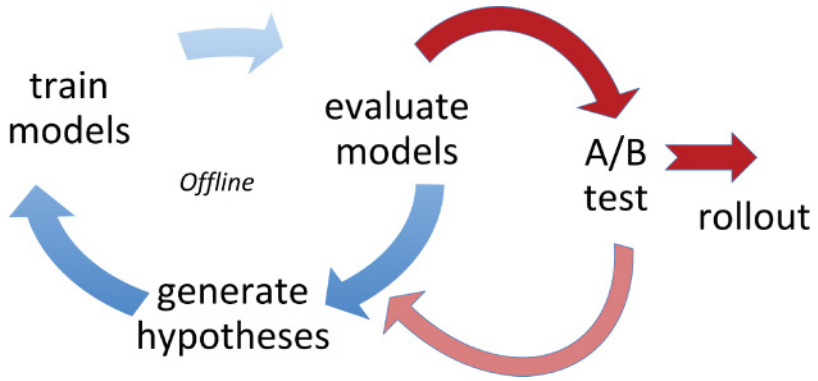


Fig. 8. We iterate quickly to prototype an algorithm through offline experimentation by analyzing historical data to quantify how well a new algorithm can predict previous positive member engagement, such as plays. The key underlying assumption, which is not always true, is that members would have engaged with our product in exactly the same way, for example, playing the same videos, had the new algorithm been used to generate recommendations. Once we see encouraging-enough results in offline experiments, we build an A/B test to use the new algorithm to generate recommendations for members. If the A/B test succeeds, we change our product to use that new algorithm by default. If the A/B test is flat or negative, we either abandon the research direction or go back to the offline experimentation world to try to make the new algorithm even better for a possible future A/B test.

of an A/B test cell, thus is hard to measure. By taking advantage of some natural experiments, in which we have been able to explore long-term changes in the experience limited to one country but not another, we can extrapolate from deviations in acquisition rate between the pairs of countries to put approximate boundaries on the magnitude of the word-of-mouth impact of such changes. While the estimates are subject to many assumptions and are quite unreliable, we conclude that, if a change might lead to retaining more existing members in a period of time, it might be expected to generate enhanced word-of-mouth that could stimulate a comparable magnitude of new members. (Presumably, such estimates would not apply at very low or very high penetration rates in a given population.)

5. KEY OPEN PROBLEMS

Although Netflix has been investing in developing our recommender system for over a decade, we still believe that our recommendations can be significantly better than they are today. Some of our main current open problems revolve around A/B testing, others around the recommendation algorithms themselves.

5.1. Better Experimentation Protocols

We want to have a better alternative to offline experimentation that allows us to iterate just as quickly, but that is more predictive of A/B test outcomes. One possibility that we are exploring is interleaving-based A/B tests focused on local algorithm metrics, such as click-through rates. It remains to be seen whether we can determine the circumstances under which the outcomes in these tests correlate well with overall streaming and retention wins in our standard A/B tests. Another possibility is developing new offline experiment metrics that are more predictive of A/B test outcomes. We are also interested in general improvements to our A/B testing, for example, effective variance reduction methods to conduct experiments with higher resolution and fewer noisy results, or new A/B engagement metrics that are even more highly correlated with retention rates.

A related challenge with engagement metrics is to determine the proper way to balance long- and short-form content. Since we carry both movies (typically 90–120 minutes of viewing) and multiseason TV shows (sometimes 60 hour-long episodes), a single discovery event might engage a customer for one night or for several weeks of viewing. Simply counting hours of streaming gives far too much credit to multiseason shows; counting "novel plays" (distinct titles discovered) perhaps overcorrects in favor of one-session movies.

5.2. Global Algorithms

We intend to offer Netflix globally before the end of 2016. Our industry relies on content licenses, which are often exclusive, and regional or even country-specific. This results in different Netflix video catalogs in different countries. Today, we group countries into regions that share very similar catalogs, yet have a big enough member base that generates enough data to fit all the necessary models. We then run copies of all of our algorithms isolated within each region. Rather than scaling this approach as we offer our service around the world, we are developing a single global recommender system that shares data across countries. The data shared include not only the relevant engagement data, such as plays, but also what the catalog of videos is in each country. Our goal is to improve the recommendations for smaller countries without affecting larger ones. We are thus interested in approaches that generalize many of the standard mathematical tools and techniques used for recommendations to reflect that different members have access to different catalogs, for example, relying on ideas from the statistical community on handling missing data [Schafer 1997].

We are also interested in models that take into account how the languages available for the audio and subtitles of each video match the languages that each member across the world is likely to be comfortable with when generating the recommendations, for example, if a member is only comfortable (based on explicit and implicit data) with Thai and we think would love to watch "House of Cards," but we do not have Thai audio or subtitles for it, then perhaps we should not recommend "House of Cards" to that member, or if we do have "House of Cards" in Thai, we should highlight this language option to the member when recommending "House of Cards."

Part of our mission is to commission original content across the world, license local content from all over the world, and bring this global content to the rest of the world. We would like to showcase the best French drama in Asia, the best Japanese anime in Europe, and so on. It will be too laborious and expensive to cross-translate every title into every other language, thus we need to learn what languages each member understands and reads from the pattern of content that they have watched, and how they have watched it (original audio vs. dub, with or without subtitles), so that we can suggest the proper subset of titles to members based on what they will enjoy.

5.3. Controlling for Presentation Bias

We have a system with a strong positive feedback loop, in which videos that members engage highly with are recommended to many members, leading to high engagement with those videos, and so on. Yet, most of our statistical models, as well as the standard mathematical techniques used to generate recommendations, do not take this feedback loop into account. In our opinion, it is very likely that better algorithms explicitly accounting for the videos that were actually recommended to our members, in addition to the outcome of each recommendation, will remove the potential negative effects of such a feedback loop and result in better recommendations. For example, a problem in this area is finding clusters of members that respond similarly to different recommendations; another is finding effective ways to introduce randomness into the recommendations and learn better models.

5.4. Page Construction

Page construction is a relatively new and unexplored area for us. It took us a couple of years to find a fully personalized algorithm to construct a page of recommendations that A/B tested better than a page based on a template (itself optimized through years of A/B testing). We think that there are endless possibilities for improving this algorithm. We have not seen the page construction problem being a main focus of the academic recommender systems community yet, but we think that many recommendations problems have similar properties of needing to address diverse moods, needs, contexts, or situations in a way that is orthogonal to the general problem of ranking the items in the catalog for each individual in a personalized way.

5.5. Member Coldstarting

We know that our recommender system does a satisfactory job helping members with a large Netflix history, but not so for new members, about whom we know little. For example, our PVR algorithm tends to rank videos discovered by our members much more highly before they are played for existing members than for newer members. Because new members get a one-month-free trial, cancellations rates are highest among them and decrease quickly after that. This is not surprising, since new members need to decide whether they want to pay for Netflix at all, while longer-tenured members have already payed for Netflix in previous months and only need to decide whether to pay for another month. Thus, we are always interested in finding better models and signals to improve the recommendations for new members, to increase their engagement and their retention rates. Today, our member coldstart approach has evolved into a survey given during the sign-up process, during which we ask new members to select videos from an algorithmically populated set that we use as input into all of our algorithms.

5.6. Account Sharing

We market Netflix subscriptions to families; in many cases, several individuals with different tastes share a single account. We allow our members to create up to 5 different profiles for every account, and we personalize the experience for each profile. However, a large percentage of profiles are still used by multiple people in the household. Our recommender system has, by necessity, evolved through years of A/B testing to deliver a mix (union) of suggestions necessary to provide good suggestions to whichever member of the household may be viewing (owner, spouse, children) at any time, but such amalgamated views are not as effective as separated views.

We have lots of research and exploration left to understand how to automatically credit viewing to the proper profile, to share viewing data when more than one person is viewing in a session, and to provide simple tools to create recommendations for the intersection of two or more individuals' tastes instead of the union, as we do today.

Children's viewing presents a particular problem in shared profiles, since kid videos tend to be shorter, and because young children have a predilection to view the same movie or episode many times, which is not a behavior typical of adults, and which can lead to very strange biases to the recommendations generated from that data. As children age, their taste changes much more quickly than adults (a year in a five-year-old's life is 20% of their experience, but only 2% of a 50-year-old's life). We have much research left in learning and modeling the aging-up process more effectively.

5.7. Choosing the Best Evidence to Support Each Recommendation

We have several images, synopsis, and other evidence that we can use to present each recommendation. These can be chosen to highlight different aspects of a video, such as an actor or director involved in it, awards won, setting, genre, and so on. The

area of evidence selection for us involves finding the best evidence to present for each recommendation; we are now investigating how much to personalize these choices.

6. CONCLUSIONS

We have described the different algorithms that make up the Netflix recommender system, the process that we use to improve it, and some of our open problems. Humans are facing an increasing number of choices in every aspect of their lives—certainly around media such as videos, music, and books, other taste-based questions such as vacation rentals, restaurants, and so on, but more importantly, around areas such as health insurance plans and treatments and tests, job searches, education and learning, dating and finding life partners, and many other areas in which choice matters significantly. We are convinced that the field of recommender systems will continue to play a pivotal role in using the wealth of data now available to make these choices manageable, effectively guiding people to the truly best few options for them to be evaluated, resulting in better decisions.

We also believe that recommender systems can democratize access to long-tail products, services, and information, because machines have a much better ability to learn from vastly bigger data pools than expert humans, thus can make useful predictions for areas in which human capacity simply is not adequate to have enough experience to generalize usefully at the tail.

APPENDIXES

A. THE EFFECTIVE CATALOG SIZE

Assume that we have N items in the catalog, ordered from the most popular in terms of hours streamed to the least popular and denoted by $v_1, \ldots, v_N$. Let the vector $\mathbf{p} = [p_1, \ldots, p_N]$ denote the probability mass function (p.m.f.) corresponding to the share of hours streamed from the popularity-ordered videos in the catalog, that is, p_i is the share of all hours streamed that came from video v_i which was the i-th most streamed video. Note that $p_i \geq p_{i+1}$ for $i = 1, \ldots, N-1$ and $\sum_{i=1}^{N} p_i = 1$. We seek a metric that is a function with $\mathbf{p}$ as its argument and a number in the range $[1, N]$ as its output, that in some sense tells us how many videos are required to account for a typical hour streamed. This metric should return a value slightly higher than 1 if the most popular video v_1 accounted for most all hours streamed, and a value of N if all videos in the catalog drove the same amount of streaming. One such metric is the *effective catalog size* (ECS), which we define as:

$$ECS(\mathbf{p}) = 2\left(\sum_{i=1}^{N} p_i i\right) - 1. \tag{1}$$

Equation (1) simply computes the average of the video index under the p.m.f. $\mathbf{p}$, and rescales it to lie in the appropriate range. It is easy to check that the ECS has a minimum value of 1 when $p_1 = 1$, and a maximum value of N when $p_i = 1/N$ for all i.

The ECS can be applied to any p.m.f. We start by computing a reference, the ECS for the p.m.f. that considers the hours of the most popular k videos only as we increase k from 1 to N. Specifically, we define $\mathbf{p}(k) = \alpha[p_1, \ldots, p_k]$, where $\alpha = 1/(\sum_{i=1}^{k} p_i)$ is a normalization constant, and plot the $ECS(\mathbf{p}(k))$ as we vary k, to get the black line in the left plot of Figure 4. This line lies below the identity line (not shown) because not all videos are equally popular. The red line in the same plot is the result of applying the ECS equation to a different p.m.f. $\mathbf{q}(k)$ as we vary k from 1 through N. The p.m.f. $\mathbf{q}(k)$ is the share of hours from each PVR rank of k or better out of all the streamed hours

that came from the top k PVR ranks. To form $\mathbf{q}(k)$, we take the k highest ranked PVR videos for each of our members, find all the streaming hours that these member–video pairs generated, and define its i-th entry to be the share of these streaming hours that came from PVR rank i. Note that, although for each member $\mathbf{q}(k)$ only includes k videos just as $\mathbf{p}(k)$ did, across a sample of members more videos, possibly all N, will appear, precisely because PVR is personalized. At the PVR rank corresponding to the median rank across all plays, the effective catalog size is roughly 4 times the size of the corresponding unpersonalized effective catalog size.

B. AN EXAMPLE OF A/B TEST STATISTICS

A simple standard model for retention assumes that each member in the control cell flips a coin that lands heads with probability μ_c, in which case the member will continue the subscription, independently of other members. Each member in the test cell similarly flips a coin to retain, but with a probability μ_t. We want to estimate the difference in the retention rates $\Delta = \mu_t - \mu_c$. Applying maximum likelihood to the retention data for each cell results in the estimates $p_c = \sum_{u=1}^{n_c} X_{uc}/n_c$ and $p_t = \sum_{u=1}^{n_t} X_{ut}/n_t$ for μ_c and μ_t, respectively, where X_{uc} is a Bernoulli random variable set to 1 if member u in the control cell c retained, and set to 0 otherwise, X_{ut} similarly describes the retention outcome of member u in the test cell, and n_c and n_t are the number of members in the control and test cells. We then estimate Δ by $\hat{\Delta} = p_t - p_c$. Then, the variance in our estimate for p_c is simply $\mu_c(1-\mu_c)/n_c \approx p_c(1-p_c)/n_c$, and a similar equation gives the variance of our estimate of p_t. Finally, the variance of $\hat{\Delta}$ is simply the sum of the variances of our estimates for p_c and p_t, that is, $\sigma^2 = p_c(1-p_c)/n_c + p_t(1-p_t)/n_t$. If the standard deviation σ is much smaller than $\hat{\Delta}$, then we have more confidence that the higher retention rates in the test cell are not due to having a finite and/or small sample of members in each cell. Loosely, a standard approach assumes Δ to follow a Gaussian distribution with mean $\hat{\Delta}$ and variance σ^2, and declares the test cell positive with respect to retention if $\hat{\Delta} \geq 1.96\sigma$. We show a plot of 1.96σ the decision boundary, as a function of the cell size and retention rate, when the two cells have equal sizes and have roughly comparable retention rates in Figure 7. This type of plot can be used as a guide to choose the sample size for the cells in a test, for example, detecting a retention delta of 0.2% requires the sample size traced by the black line labeled 0.2%, which changes as a function of the average retention rate when the experiment stops, being maximum (south of 500k members per cell) when the retention rate is 50%.

Different probability models would yield different results. For example, we could use prior test results to build different prior distributions for the various parameters, such as μ_c and μ_t, or we could model each member as having one's own probability of retaining, which could itself be a sample from a beta distribution, and aim to estimate the parameters of this underlying beta distribution for each cell, or we can account for stratified sampling if it was used when constructing the test cells, and so forth.

ACKNOWLEDGMENTS

We thank the many algorithms teams at Netflix for their work to run and improve our recommender system. We are also grateful to Angadh Singh for the effective catalog size and take-rate plots, and to Justin Basilico for helpful suggestions on this manuscript.

REFERENCES

Chris Alvino and Justin Basilico. 2015. Learning a Personalized Homepage. Retrieved December 6, 2015 from http://techblog.netflix.com/2015/04/learning-personalized-homepage.html.

Xavier Amatriain and Justin Basilico. 2012. Netflix Recommendations: Beyond the 5 stars (Part 2). Retrieved December 6, 2015 from http://techblog.netflix.com/2012/06/netflix-recommendations-beyond-5-stars.html

David M Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *The Journal of Machine Learning Research* 3, 993–1022.

Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems* 30, 1. DOI:http://dx.doi.org/10.1145/2094072.2094078

Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *WSDM*.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2011. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd. ed.). Springer.

Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY)*.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8, 30–37.

Andriy Mnih and Ruslan Salakhutdinov. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. 1257–1264.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective.* MIT Press, Cambridge MA.

Prasanna Padmanabhan, Kedar Sadekar, and Gopal Krishnan. 2015. What's trending on Netflix. Retrieved December 6, 2015 from http://techblog.netflix.com/2015/02/whats-trending-on-netflix.html.

Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop.* 5–8.

Leo Pekelis, David Walsh, and Ramesh Johari. 2015. The New Stats Engine. Internet. Retrieved December 6, 2015 from http://pages.optimizely.com/rs/optimizely/images/stats_engine_technical_paper.pdf.

Netflix Prize. 2009. The Netflix Prize. Retrieved December 6, 2015 from http://www.netflixprize.com/.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE 10th International Conference on Data Mining (ICDM).* IEEE, 995–1000.

Joseph L. Schafer. 1997. *Analysis of Incomplete Multivariate Data*. CRC Press, Boca Raton, FL.

Barry Schwartz. 2015. *The Paradox of Choice: Why More Is Less.* Harper Perennial, New York, NY.

Bryan Gumm. 2013. Appendix 2: Metrics and the Statistics Behind A/B Testing. In *A/B Testing: The Most Powerful Way to Turn Clicks into Customers*, Dan Siroker and Pete Koomen (Eds.). Wiley, Hoboken, NJ.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101, 476.

Received July 2015; revised September 2015; accepted November 2015