The MovieLens Datasets: History and Context

F. MAXWELL HARPER and JOSEPH A. KONSTAN, University of Minnesota

The MovieLens datasets are widely used in education, research, and industry. They are downloaded hundreds of thousands of times each year, reflecting their use in popular press programming books, traditional and online courses, and software. These datasets are a product of member activity in the MovieLens movie recommendation system, an active research platform that has hosted many experiments since its launch in 1997. This article documents the history of MovieLens and the MovieLens datasets. We include a discussion of lessons learned from running a long-standing, live research platform from the perspective of a research organization. We document best practices and limitations of using the MovieLens datasets in new research.

CCS Concepts: • **Information systems** → **Recommender systems**; • **Human-centered computing** → **Collaborative filtering; Collaborative and social computing systems and tools**; • **General and reference** → *Surveys and overviews;*

Additional Key Words and Phrases: Datasets, MovieLens, ratings, recommendations

ACM Reference Format:
F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens datasets: History and context. ACM Trans. Interact. Intell. Syst. 5, 4, Article 19 (December 2015), 19 pages.
DOI: http://dx.doi.org/10.1145/2827872

1. INTRODUCTION

The MovieLens datasets, first released in 1998, describe people's expressed preferences for movies. These preferences take the form of `<user, item, rating, timestamp>` tuples, each the result of a person expressing a preference (a 0–5 star rating) for a movie at a particular time. These preferences were entered by way of the MovieLens website[1]—a recommender system that asks its users to give movie ratings in order to receive personalized movie recommendations.

The MovieLens datasets are heavily downloaded (140,000+ downloads[2] in 2014) and referenced in the research literature (7,500+ references to "movielens" in Google Scholar). This popularity is, to a certain degree, a reflection of the incredible rate

[1]http://movielens.org.
[2]http://grouplens.org/datasets/movielens.

The reviewing of this article was managed by associate editor Barry Smyth.
This material is based on work supported by the National Science Foundation under grants DGE-9554517, IIS-9613960, IIS-9734442, IIS-9978717, EIA-9986042, IIS-0102229, IIS-0324851, IIS-0534420, IIS-0808692, IIS-0964695, IIS-0968483, IIS-1017697, and IIS-1210863. This project was also supported by the University of Minnesota's Undergraduate Research Opportunities Program and by grants and/or gifts from Net Perceptions, Inc., CFK Productions, and Google.
Authors' addresses: F. Maxwell Harper and Joseph A. Konstan, Department of Computer Science and Engineering, University of Minnesota, 4-192 Keller Hall, 200 Union Street SE, Minneapolis, MN 55455; emails: {harper, konstan}@cs.umn.edu.

DOI: http://dx.doi.org/10.1145/2827872

of growth of personalization and recommendation research, in which datasets such as these have substantial value in exploring and validating ideas. The popularity might also be attributed to the flexibility of ratings data, which is naturally suited not just to recommendation technology, but also to a more general family of data science techniques concerned with summarization, pattern identification, and visualization. Also, because movie preferences are highly subject to personal tastes, the movie domain is well suited to testing personalization technology. Finally, the popularity may reflect the perceived accessibility of movies as a content domain: movies are a common interest, making algorithmic output easy to discuss.

This article explores the history of the MovieLens system in order to document the factors that have shaped the resulting datasets. Along the way, we also share lessons learned from operating a long-running research platform, and we document best practices for conducting research using the datasets. We include two main sections. In Section 2, we share history and lessons from the MovieLens system; in Section 3, we share descriptions and guidelines concerning the datasets.

2. THE MOVIELENS SYSTEM

The MovieLens datasets are the result of users interacting with the MovieLens online recommender system over the course of years. As with most long-lived and dynamic online systems, MovieLens has undergone many changes—both in design and in functionality. These changes necessarily impact the generation of ratings: users only rate movies that appear on their screens; users' ratings are also influenced by the predicted ratings themselves [Cosley et al. 2003].

In this section, we present a history of MovieLens with an eye toward illustrating those changes most likely to have had an impact on the datasets' content.

2.1. Origins and Landmarks

University of Minnesota researchers began developing MovieLens in the summer of 1997, the result of Digital Equipment Corporation's (DEC) decision to close their own movie recommender, EachMovie. At this time, DEC contacted the recommender systems community looking for an organization to develop a replacement site to carry on the same mission, and perhaps serve the same users (but without using DEC's proprietary code); GroupLens volunteered. Legal issues blocked directly transferring user accounts, but DEC did transfer an anonymized dataset to GroupLens, who used this dataset to train the first version of the MovieLens recommender.

MovieLens launched in the fall of 1997. As much as possible, this first version was developed to look like the EachMovie interface that it was replacing (see Figure 5 for a screenshot). EachMovie used a proprietary recommendation algorithm; MovieLens instead incorporated user–user collaborative filtering (CF) as implemented in the GroupLens usenet news recommender system [Konstan et al. 1997].

Use of MovieLens increased significantly during late 1999 when it received attention from the mass media. These events—an article in *The New Yorker* magazine by Malcolm Gladwell [Gladwell 1999], an episode of ABC's *Nightline*, and a favorable mention by noted film critic Roger Ebert—each prominently featured MovieLens as an example of the future of online computing and personalization.

Since that time, the growth of the MovieLens system has been remarkably stable, especially considering the nearly total absence of marketing efforts. MovieLens has averaged 20 to 30 new user signups every day for a long time, which are largely the result of word-of-mouth or unsolicited press. See Figure 1 for a visualization of the growth of MovieLens since its launch, Figure 2 for a visualization of the number of monthly active users, Figure 3 for a visualization of the number of users in each calendar year logging in more than n times, and Figure 4 for a visualization of the distribution of ratings data over time.

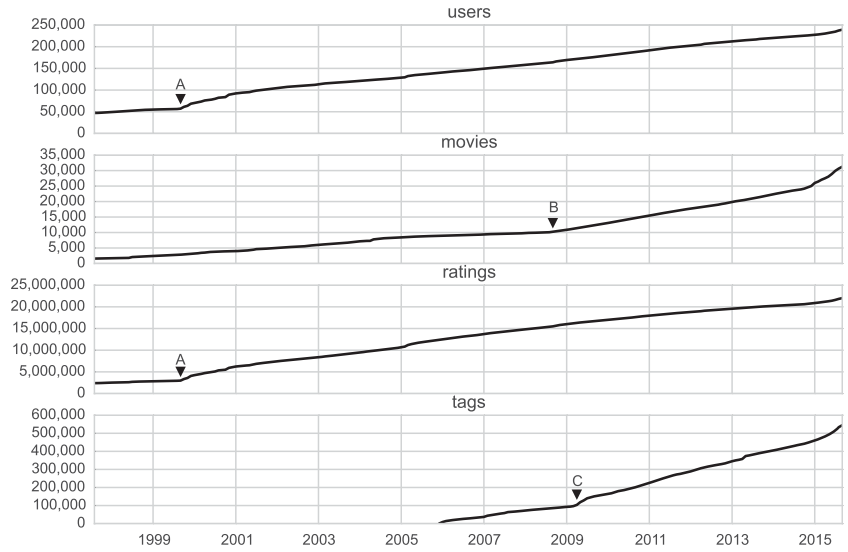

Fig. 1. A 17-year view of growth in movielens.org, annotated with events A, B, and C. User registration and rating activity show stable growth over this period, with an acceleration due to media coverage (A). The rate of movies added to MovieLens grew (B) when the process was opened to the community. There is an acceleration in tag applications with the release of the "tag expression" feature (C).

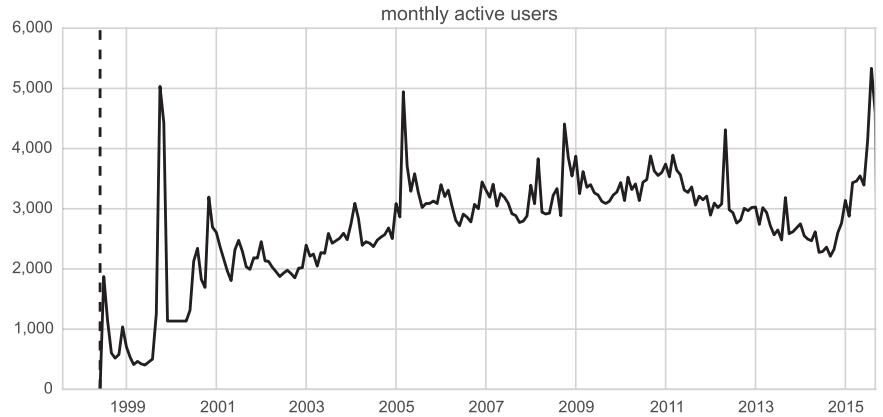

Fig. 2. Number of users that logged into MovieLens each month. Data before the dashed line (July, 1998) is missing.

Steady growth does not mean that MovieLens has not changed since the late 1990s. In fact, since its initial launch, MovieLens has seen five major releases (v0–v4), each release representing a full reimplementation of the server- and client-side code. We summarize the most significant changes in Table I, which includes dates and short summaries of some of the most significant changes to MovieLens. Also, for a visual frame of reference, we provide historical screenshots in Figures 5 through 8 and a screenshot of the current interface in Figure 9.

In the remainder of this section, we discuss key changes to MovieLens features that can most inform the shape and research use of the MovieLens datasets.

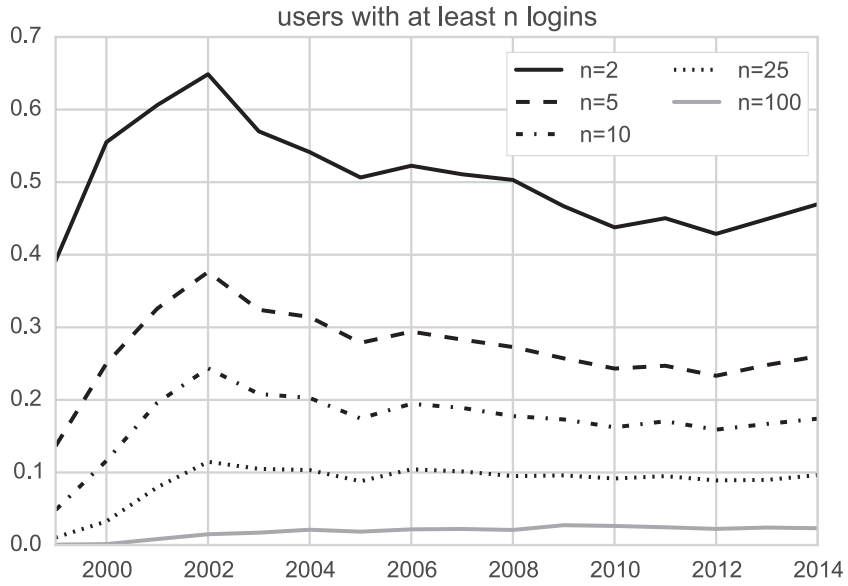

Fig. 3. Fraction of users that logged in to MovieLens at least n times during each full calendar year. For example, in 2014, 10% of the users who visited MovieLens logged in 25 or more times.

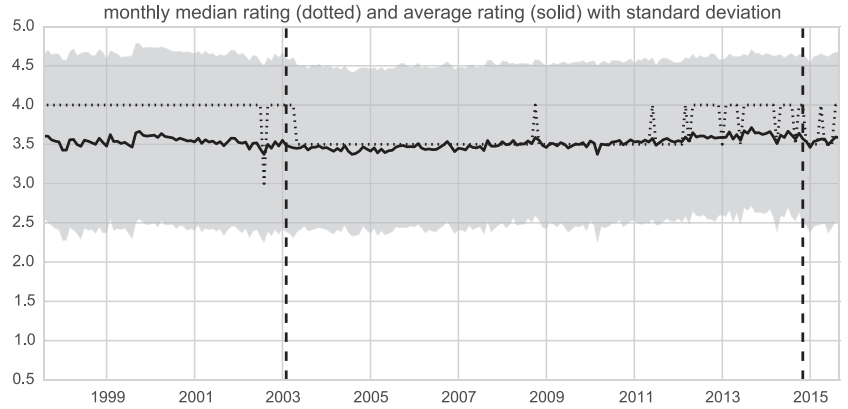

Fig. 4. Rating distribution, aggregated by month. The solid line represents each month's average rating. The gray area covers the average plus or minus the standard deviation of that month's ratings. The dotted line shows each month's median rating. The two vertical dashed lines represent the launch of v3 (with half-star ratings) and v4 (with a simpler star ratings widget).

2.2. Changes to Key Features

2.2.1. Exploring. The mechanisms for searching, filtering, and ordering movies affect which movies a user can rate, therefore have a fundamental influence on the shape of the MovieLens datasets. The core user experience in MovieLens revolves around a feedback loop between rating movies and viewing movie recommendations: the user views items in a list of recommendations and rates those items, which in turn alters the items shown on subsequent page views.

This rating/recommendation cycle is enabled by CF algorithms [Resnick et al. 1994]. MovieLens, until the release of v4, ordered lists of movies strictly based on user-personalized predicted rating values—the output of its underlying collaborative

Table I. MovieLens Landmarks

Date	Landmark
8/1997	v0 interface; EachMovie seed; GroupLens user–user recommender
9/1999	v1 interface
11/1999	Media exposure; Net Perceptions user–user recommender
2/2000	v2 interface; movie groups; additional external movie metadata (e.g., box office, DVD releases); reviews
2/2003	v3 interface; multilens item–item recommender; Ajax ratings widget
6/2005	Discussion forums
12/2005	Tagging
9/2008	Member-based movie adding
1/2012	Lenskit item–item recommender
11/2014	v4 interface; user-selectable recommender; external movie data

Note: MovieLens has changed constantly since its release in 1997; these are some of the most meaningful events in its history.

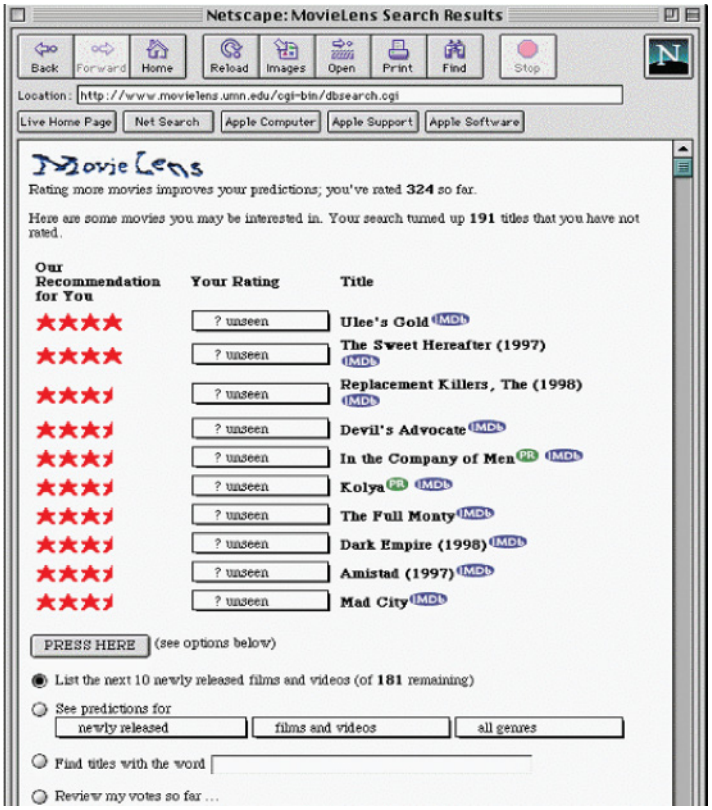

Fig. 5. Screenshot of MovieLens v0, circa 1998.

filtering algorithms. In other words, the movies that appear first in search results for a user are those movies that the algorithm predicts that user will rate the highest. v4 blends a popularity factor in with predicted rating to order the recommendation lists [Ekstrand et al. 2015; Harper et al. 2015].

The algorithm itself has changed over time, which has had mostly unstudied effects on user satisfaction and rates of contribution:

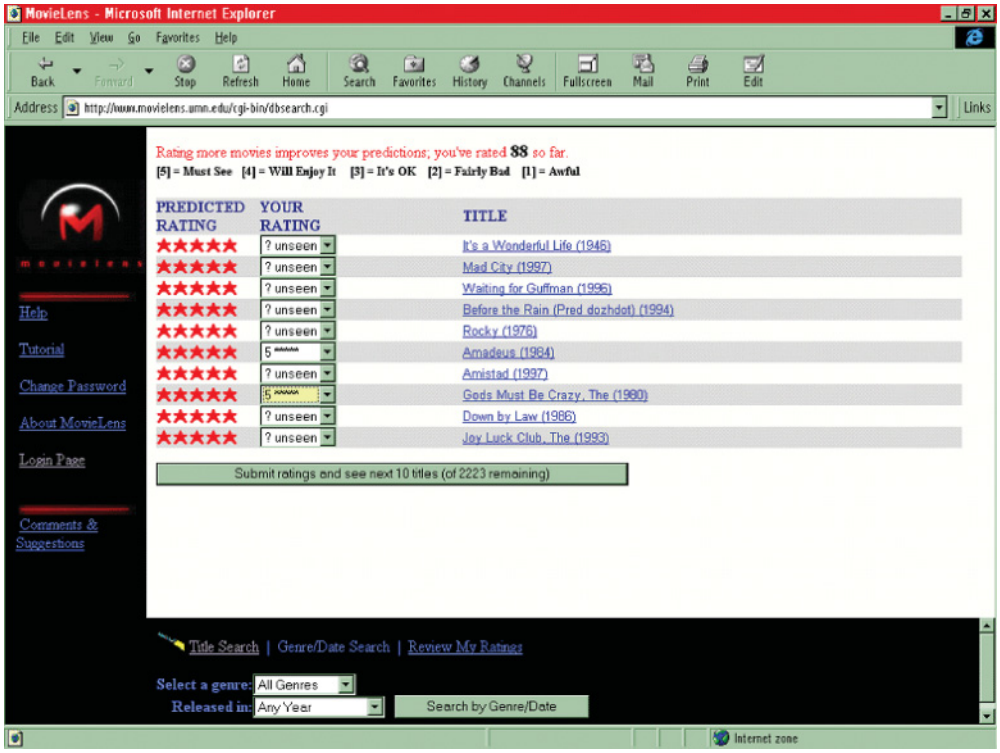

Fig. 6. Screenshot of MovieLens v1, circa 1999.

—1997: User–user CF via the GroupLens Usenet recommender [Konstan et al. 1997]
—1999: User–user CF via Net Perceptions[3]
—2003: Item–item CF via MultiLens [Miller 2003]
—2012: Item–item CF via LensKit [Ekstrand et al. 2011]
—2014: Users able to select [Ekstrand et al. 2015] from a nonpersonalized algorithm, a recommender that supports new users [Chang et al. 2015], item–item CF, and FunkSVD via LensKit

The *main page* of MovieLens is a focus of user attention, and a prime spot to show top-recommended movies to users. Early versions (v0, v1) simply displayed a list of top recommendations on this page. v2 and v3 were more complex, showing several prefiltered lists of top picks (recent movies in theaters, recent movies on DVD/VHS), along with a list of links out to different site features. v4 simply shows a list of lists, headed by overall "top picks" at the top of the page. Versions v2 through v4 all encourage the user to begin exploring a particular slice of the movie database, though the available search customizations grew over time.

The MovieLens interface has always supported searching and filtering operations that allow users to restrict their view to particular groups of movies. All versions have supported title search; v4 includes an autocomplete function. Starting with v0, the interface supported filtering by movie genre and release date. v3 expanded the filtering options by allowing users to filter movies by actor, director, and other attributes

[3]Net Perceptions was a recommender systems company cofounded in 1996 by GroupLens faculty and students.

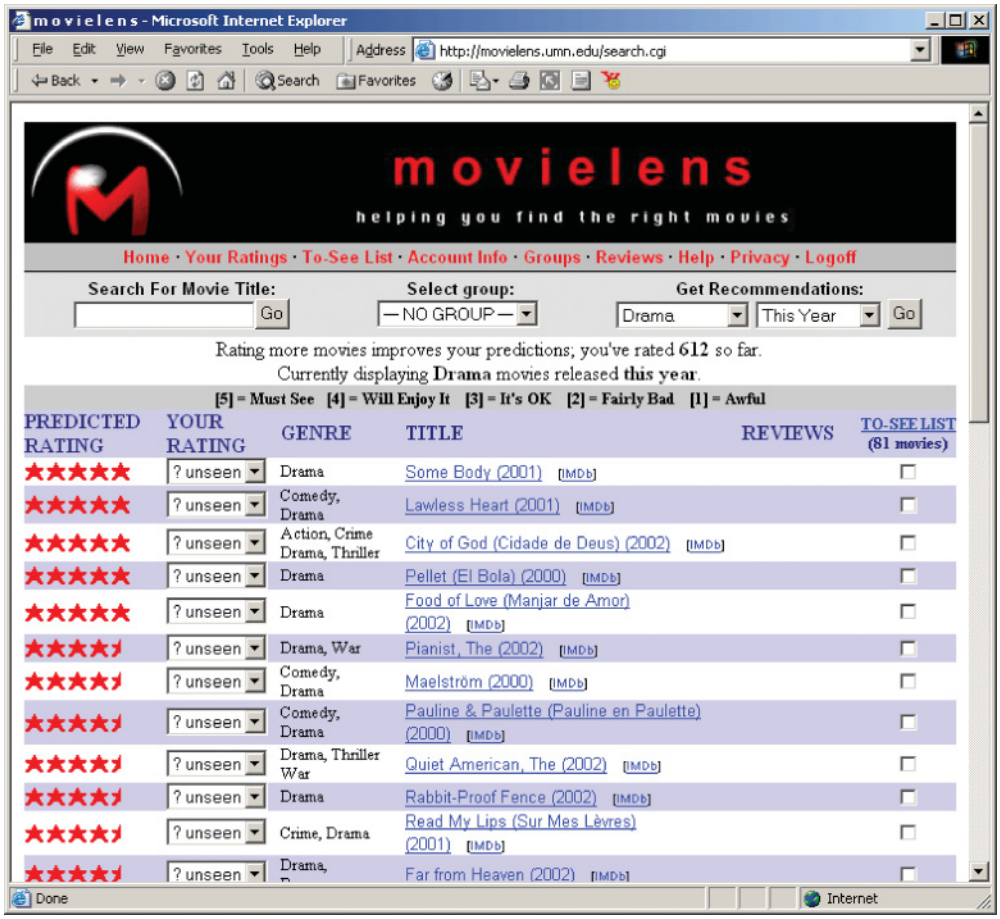

Fig. 7. Screenshot of MovieLens v2, circa 2000.

as well. In mid-2004, MovieLens added an "advanced search" page that brought an extremely expressive (and complicated) control panel for finding movies. v4 launched with a smaller set of search filters, but an expanded array of predefined searches with previews on the front page.

Many recommendation interfaces offer a "more like this" function; MovieLens began offering a version of this in late 2009 with the introduction of the "movie tuner" [Vig et al. 2012]. This feature allowed users to navigate the information space differently, by jumping from movie to movie directly, outside the influence of the primary exploration interface. This feature, which persists in v4 with a different interface design, is unique in MovieLens for scoring similarities using a content-driven algorithm (using tags) rather than ratings-based techniques.

MovieLens users have typically had access to only a limited set of movie metadata in the site. Until mid-2005, movies only appeared in lists with very basic information such as release dates and actors. In mid-2005, MovieLens added "movie detail" pages into the site design to support linking with the discussion forum (see Section 2.2.6). The addition of movie tagging in late 2005 (see Section 2.2.4) provided both objective and subjective data for describing movies. In Spring 2009, MovieLens integrated with Netflix (Netflix stopped supporting this API in late 2014), incorporating poster art and

Fig. 8. Screenshot of MovieLens v3, circa 2003.

a plot synopsis into the movie details pages. v4 integrated MovieLens with The Movie Database (TMDb), bringing plot summaries, movie artwork, and trailers directly into the site.

2.2.2. Rating. Since MovieLens launched, ratings have been expressed as a "star" value, which is a standard user interface pattern for users to input preferences. The biggest change to the ratings interface came with the launch of v3 (February, 2003) when the interface shifted from "whole star" to "half star" ratings, the most-requested feature in a user survey, doubling the range of preference values from five (1–5) to ten (0.5–5.0). The release of v3 also upgraded the ratings widget to submit values asynchronously, without requiring the user to click a submit button.

Versions v0 through v3 employed two separate visual elements for the ratings widget. The *view* element was an image showing a certain number of stars, the color representing prediction or actual rating. The *input* element was an html `<select>` element allowing the user to choose the star value. With the release of v4, the user interface combined these two elements into a single five-star representation that accepts touch/click events.

Throughout the life of v1 through v3, a small legend describing different star values (see Figures 6 through 8) sat in the top of each screen. The labels describing the star values changed in v3. It is possible that these labels influenced rating behavior substantially, given the body of literature pointing to anchoring effects (e.g., Lynch et al. [1991]), but there has been no empirical study of this effect in MovieLens. At a coarse

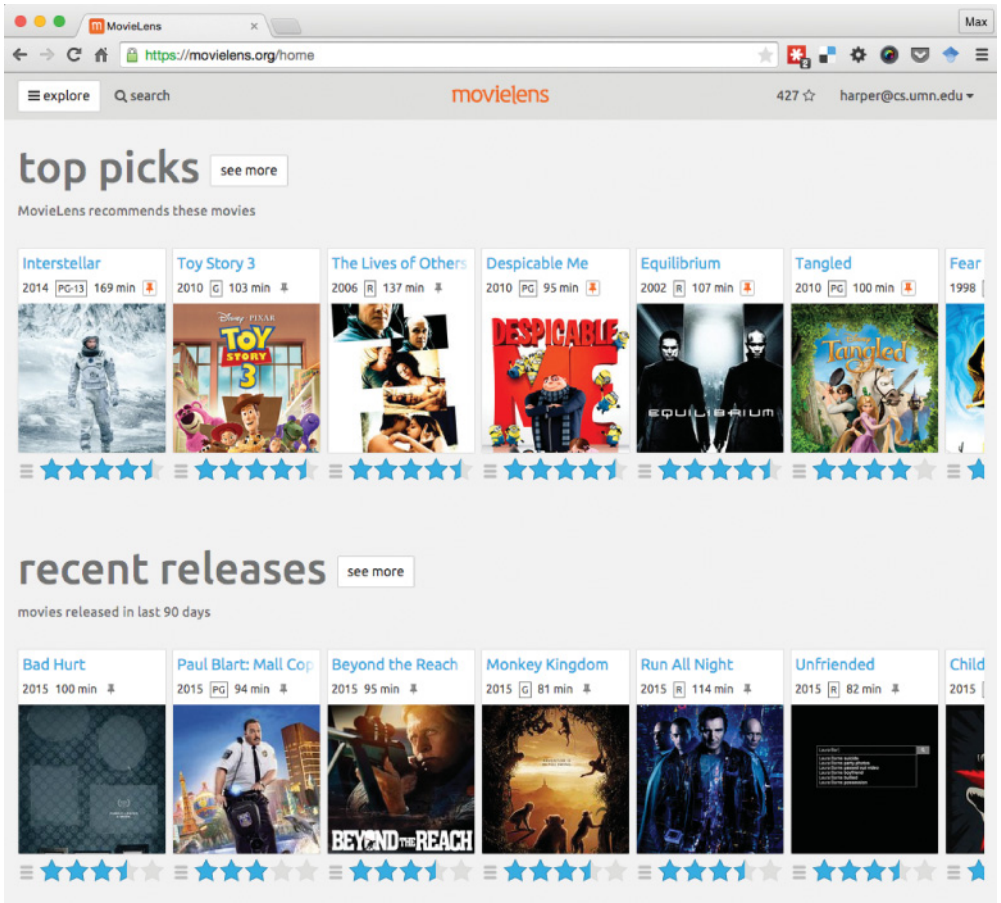

Fig. 9. Screenshot of MovieLens v4, 2015.

level of analysis, Figure 4 shows no visible global changes in ratings distributions resulting from these interface changes.

2.2.3. Bootstrapping. The ratings-based CF algorithms that MovieLens is built on do not work well for users until they have provided some initial ratings to the system [Rashid et al. 2002]. MovieLens, until the release of v4, required users to cross some ratings barrier [Drenner et al. 2006] before entering the main system.

Early versions, starting in v0, required users to rate 5 movies before entering the system. MovieLens showed users movies 10 at a time, where nine were randomly chosen from the database, and the tenth chosen from a hand-chosen list of highly recognizable titles [Rashid et al. 2002].

In v3, the process was changed to require 15 ratings from users. Users were again shown movies 10 at time, but now movies were selected for their popularity, excluding the most popular 50 to 150 movies. Other methods were briefly evaluated for experimental purposes [Rashid et al. 2008] but did not have a lasting effect.

Version v4 dropped the 15-rating requirement, instead developing a special new user recommender, based on a faster group-based personalization process [Chang et al. 2015]. We do not yet know the long-term benefits and costs of dropping this barrier.

2.2.4. Tagging. MovieLens introduced tagging in December, 2005; tags appear in the later MovieLens datasets (10M and 20M). This feature allows users to apply tags—words or short phrases—to movies. MovieLens displays tags next to movies. Tags are clickable to show a list of movies on which that tag has been applied.

The visibility and ordering of tags differed by user in early versions to provide A/B testing data [Sen et al. 2006], but by 2006 the interface was consolidated, sorting tags by the likelihood that the tag is "high value" according to a published metric [Sen et al. 2009]. In January, 2007, MovieLens launched a tag rating feature that put clickable thumbs-up-and-down icons next to tags [Sen et al. 2007].

In Spring 2009, the tagging interface was given a new feature called "tag expressions" [Vig et al. 2010] that dramatically influenced tagging behavior. This interface allowed users to retag movies much more easily, and caused an increase in the rate of tagging activity, as well as an increase in tag diversity.

2.2.5. Movie Adding and Editing. The presence or absence of movies shapes rating and tagging behavior, as does the accuracy and completeness of the associated metadata.

Early versions of MovieLens relied on a small group of administrators and a content expert to curate the movie database. During this phase, MovieLens represented a relatively narrow universe of movies: those with a widespread United States theatrical release.

Starting in mid-2004, member control of the database increased. Soon after the release of v3, MovieLens added a link that allowed members to suggest titles; in mid-2004, MovieLens began to allow members to edit movie details directly [Cosley et al. 2005]. After early experiments (circa 2004) allowing members to add movies, an interface for movie adding became a permanent feature of the system in late 2008. In v4, control of the metadata shifted from member input to an external source—http://themoviedb.org.

2.2.6. Interacting with Other Users. The MovieLens interface has always emphasized interacting with movies over interacting with other users. However, through the history of MovieLens, features have come and gone that draw users' attention toward more social uses of the system.

MovieLens offered two iterations of a feature designed for groups of people to receive joint recommendations. v2 included a feature called "groups" [O'Connor et al. 2001], which were persistent named collections of users that could receive jointly optimized recommendations. This structure was redesigned in v3, when it (now called "buddies") was reoriented to behave more like a social network. This redesign recognized that group creation was too hard and groups were too inflexible—users could have pairwise relationships and decide which buddies to include in a request for recommendations. The buddies feature was used by approximately 9,000 users to connect with a median of one other MovieLens user (avg=1.8).

MovieLens offered three iterations of a feature designed to support conversations around movies. The first of these, launched with v2, was a simple threaded discussion forum. This version was not deeply connected with the main site, and was barely used. It was discontinued when v3 was released.

The second iteration, released in mid-2005, was a threaded discussion forum designed to integrate more deeply with the movie database. The custom forum software automatically detected movie titles in the text, and placed prediction/rating widgets for each referenced movie next to it in a sidebar. These widgets linked to the movie details pages about each movie, and the movie detail pages linked to the five most recent posts referencing the movie (or similar movies) [Drenner et al. 2006].

The third iteration, released in 2009, replaced the forum interface with one that supported question-asking. This redesign diminished user interest in the feature, because

the question-asking-and-answering format did not fit as naturally with the community's desired uses.

The v3 discussion forums hosted a moderate amount of activity, most of the content from a small group of users. Over the span of 4.5 years, about 900 users posted about 5.5 times each day, creating about 10,000 posts and 11,000 links between the movies interface and the discussion forum. The question-oriented design had less use; over the span of 5 years, about 900 users posted about 1.5 times each day, creating about 2,000 posts and 1,500 links between the movies interface and the question-asking area.

Several experiments—affecting thousands of users over the life of the discussion forums—created short-lived features that engaged users socially. The first of these, which ran from early 2005 until early 2006, added personalized messages to the home-page inviting users to participate in the discussion forums [Harper et al. 2007a]. Following this work, from late 2006 until late 2007, was a large-scale intervention that added additional social features, including individual profile pages, group profile pages, and home-page recommendations. This design was the result of a large-scale field study of the differential influences in an online community of building interpersonal friendships or group identities [Ren et al. 2012; Harper et al. 2007b].

2.3. Lessons Learned

During the more than 17 years that our research lab has operated MovieLens, we have learned several lessons about the costs and benefits of operating a "real" system in support of academic research.

The primary benefit of running a live research system is clear: we are able to run online field experiments (e.g., Ekstrand et al. [2015], Ren et al. [2012], Harper et al. [2007a], Sen et al. [2006], Drenner et al. [2006], Cosley et al. [2005], and Rashid et al. [2002]), gaining an ecological validity and a larger number of subjects that can be very difficult to achieve in a lab setting. Because we control the operation of the system, we have complete flexibility in designing experiments, from modifications to the user interface to the backing algorithms. This flexibility makes it easy to ask for user judgments and opinions, often a key element in determining if an experimental change is successful. Even better, we have discovered that MovieLens users are open to experiments on a broad range of features outside of the core feature set of the system, enabling our group's work on tagging, online communities, and user motivation. Because of this openness, we are also able to email MovieLens users to participate in one-off user studies [Shani and Gunawardana 2011] to test features that are not ready to scale or integrate into the main user interface.

The primary cost in running a live system is the time required by researchers on nonresearch tasks such as software development, hardware maintenance, user communication, database maintenance, and marketing. There are many hidden tasks that consume the time of researchers, such as restarting a server after a power failure, responding to angry users' emails when their favorite feature is changed, or maintaining secure backups of user data. However, some of these tasks carry indirect benefits for researchers: MovieLens developers gain large-scale, team-based software development experience. Moreover, interacting with and managing a community of users often reveals new research ideas.

There are several lessons that we have learned over the course of running MovieLens that may generalize to other research organizations hoping to build active research platforms.

—*Lessons from the technology startup community apply.* It is difficult to build a system that is popular with users because the world is full of high-quality, free services that compete for their time. Our research lab has launched several other systems, ranging

Table II. Quantitative Summary of the MovieLens Ratings Datasets

Name	Date Range	Rating Scale	Users	Movies	Ratings	Tag Apps	Density
ML 100K	9/1997–4/1998	1–5, stars	943	1,682	100,000	0	6.30%
ML 1M	4/2000–2/2003	1–5, stars	6,040	3,706	1,000,209	0	4.47%
ML 10M	1/1995–1/2009	0.5–5, half-stars[a]	69,878	10,681	10,000,054	95,580	1.34%
ML 20M	1/1995–3/2015	0.5–5, half-stars[a]	138,493	27,278	20,000,263	465,564	0.54%

Note: The sole computed column, *Density*, represents the percentage of cells in the full user-item matrix that contain rating values.
[a]MovieLens changed from a 1- to 5-star scale to a 0.5- to 5.0-half-star scale on February 18, 2003.

from Cyclopath, which was successful enough to run several small experiments (e.g., Priedhorsky et al. [2010]) to GopherAnswers, which failed to attract enough users to advance our research goals. Thus, we have learned that many of the same ideas from the world of startup technology companies (e.g., Ries [2011]) also apply in this context: launch quickly, and fail fast—most of the time spent getting a system up and running is not productive from a research perspective.

—*Running a live site implies continual work, not one-time effort.* User expectations advance quickly in response to technological norms, which creates pressure for the research site to keep up or lose users. For example, innovations by search engine companies have shifted the norms of the behavior of search boxes; in response, MovieLens has seen significant changes to its implementation of search since its launch. While keeping up with these technological changes may be important to retain a healthy user base, it is also important in attracting graduate students or other researchers, who may be uninterested in working with code that is several years out of date.

—*Encourage good experimental code through code reuse and social coding conventions.* The code for any given experiment can harm a system over the long run, for example, by introducing bugs or by worsening code quality. A natural response to this problem is to build a research framework into the code base to contain and manage experiments. However, we have found that experiments take such varied forms that it is difficult to capture all of the requirements in any framework. We have found better success through two efforts. First, we provide tools to make common research tasks—for example, logging and condition assignment—easier and more consistent. Second, we have established social coding conventions, including code review, to ensure that new experiments follow established norms and minimize impact to the core code.

—*Invest in tools that allow the community of users to help.* In many cases, it is possible to build features that allow users to perform actions without intervention from an administrator. For example, while researchers in our lab have little interest in maintaining a movie database, our users are passionate about the subject.

3. THE MOVIELENS DATASETS

There have been four MovieLens datasets released, known as `100k`, `1m`, `10m`, and `20m`, reflecting the approximate number of ratings in each dataset. These datasets, first released in 1998, and with a major release every 5 to 6 years since, have grown in size along with the MovieLens system. Along with the release of the `20m` dataset, GroupLens began hosting additional, nonarchival datasets (an unabridged version for completeness along with a small version for speed) that are refreshed periodically to include up-to-date movies: `latest` and `latest-small`.

Summary statistics for the MovieLens datasets are shown in Table II.

All of the currently released MovieLens datasets share several characteristics. They each represent rating tuples of the form `<user, item, rating, timestamp>`. Ratings

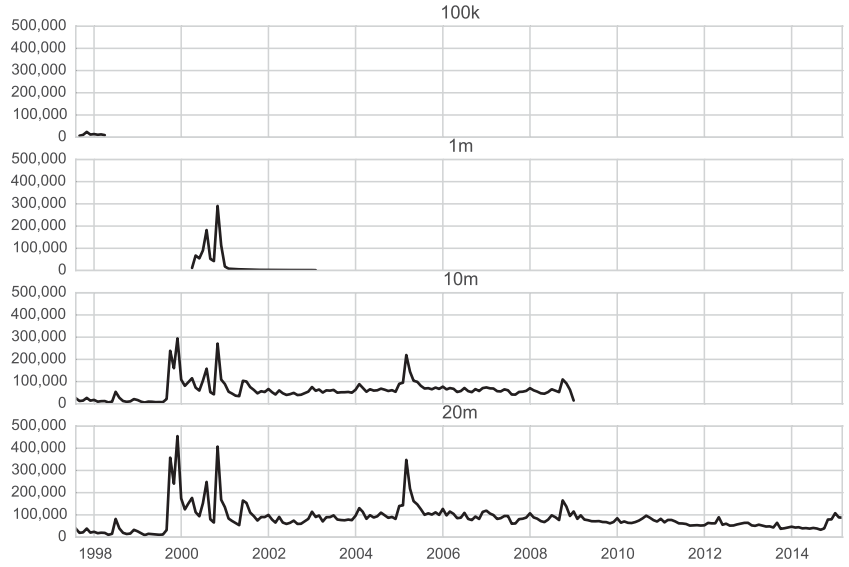

Fig. 10. Number of ratings across time in the four MovieLens datasets, aggregated by month.

and other data are attributed to anonymized user IDs (these IDs do not map across datasets). Movies are listed along with their titles in MovieLens, along with zero or more genres (movie IDs do map across datasets). Only users with at least 20 ratings are included.

These four datasets differ in their sampling methods. Though all four required a minimum of 20 ratings to include a user, `100k` also required complete demographic data. `1m` sampled users who joined the system in the year 2000, but the sample was collected in early 2003, leading to a sparse set of ratings from 2001 and 2002. `10m` and `20m` selected random users across the entire history of the system. These collection methods created datasets that reflect the bursts of ratings activity that happened in MovieLens. See Figure 10 for a visualization of the density of ratings within the four datasets across time.

Several notable changes have affected the structure of the datasets between releases. The `100k` and `1m` datasets include users' demographic data (age, gender, occupation, zip code), while the `10m` and `20m` datasets do not include any demographic information (it stopped being collected in the site). Only the `10m` and `20m` datasets include tag applications (the earlier datasets were released before the tagging feature existed). `20m` includes a table mapping MovieLens movie IDs to movie IDs in two external sites, to allow dataset users to build more complete content-based representations of the items.

As noted earlier, these datasets are the result of actual people interacting with MovieLens, a steadily evolving system. Some of the design changes, for example, changing to half-star ratings, have a directly measurable effect on the contents of the datasets. Other changes, for example, adding a minimum of 15 ratings to the sign-up process, may have a more subtle influence. Changes such as these may have led different users to reach the 20 ratings minimum, or to rate different types of movies. These subtle differences are a necessary artifact of generating datasets from a long-running and changing system.

3.1. Impact

As has become increasingly well recognized by both researchers and funders of research, shared datasets can have a substantial positive impact on research, education, and practice. By publishing datasets covering more than 17 years of usage of a single evolving system, the MovieLens datasets have been able to make a distinctive contribution in the field of recommender systems as well as related fields.

3.1.1. Contributions to Research

—In Spring 2015, a search for "movielens" yields 2,750 results in Google Books and 7,580 results in Google Scholar.
—We are particularly proud that the MovieLens datasets were used to develop and test many of the core algorithmic advances in recommender systems, including item–item collaborative filtering [Sarwar et al. 2001; Karypis 2001; Deshpande and Karypis 2004], dimensionality reduction collaborative filtering [Sarwar et al. 2000], trust-based recommenders [O'Donovan and Smyth 2005; Massa and Avesani 2007], recommenders on rapidly changing sets of items [Das et al. 2007], and cold-start algorithms [Schein et al. 2002].
—The datasets continue to be actively used today (1,112 search results in Google Scholar are dated 2014), and we hope that the recent release of our `20m` rating dataset will help many other researchers.

3.1.2. Contributions to Education and Practice

—In 2014, the datasets were downloaded more than 140,000 times from the GroupLens website (http://grouplens.org/datasets/movielens). While some of these downloads reflect research use, we believe the vast majority of them are for educational purposes. We hear from faculty who use the datasets in their courses (in some cases seeking our permission to make custom-formatted versions for their students), from students using the datasets in assignments or self-study, and from textbook authors and others who use the datasets in their materials. From the beginning, we have made MovieLens datasets free for any noncommercial use (including internal research and education by commercial entities).
—The datasets have found a niche as test data for sample code in popular-press programming and data science books. For example, the datasets have been used to demonstrate building an item–item collaborative filtering algorithm in Python [Segaran 2007], as well as to demonstrate loading, exploring, and visualizing data in a high-performance distributed computing platform [Pentreath 2015].
—We have disseminated the MovieLens datasets ourselves through a massive online open course (MOOC) on recommender systems [Konstan et al. 2014] and through automated build tool integration with the LensKit recommender system toolkit (http://lenskit.org) [Ekstrand et al. 2011].
—We also are aware of fairly extensive use by commercial entities. We have been contacted by companies using the MovieLens datasets to test their own recommender algorithms and systems, to benchmark external systems, and for training and demonstration purposes.

3.2. Limitations

The extensive changes to the MovieLens user experience inevitably have led to less "clean" ratings data than if we had kept the interface constant. As discussed earlier, changes to the searching tools, recommendation algorithms, and available features all change the content that is shown to users, which in turn affects the resulting ratings data. Further, our extensive experimentation on the MovieLens platform has further

changed user rating behavior in a variety of ways. However, MovieLens is not unique in this problem—recommendation sites such as Amazon and Netflix also have introduced substantial interface changes and perturbations through A/B testing.

The MovieLens datasets include data only from users with at least 20 ratings, therefore are inherently biased towards "successful" users. That is, the users who are less interested in rating movies, were unable to find enough ratable content, or did not enjoy their initial experience in the system are not included in the datasets. It is possible that these users are fundamentally different from the users in the datasets.

The datasets associate timestamps with each rating, but these timestamps do not represent the date of consumption. Ratings in MovieLens can happen any time, possibly many years after watching a movie. Often, users will enter a large number of ratings in a single session, backfilling their ratings history for personal satisfaction or in the hopes of getting more personalized recommendations. As discussed earlier, these backfilled ratings are often prompted by the recommendations that we display, further reducing the value of the timestamps as even an estimate of when the user thought about the movie.

3.3. Recommendations for Usage

In this section, we offer some experience-based guidelines for making more effective use of the MovieLens datasets, to further our aim of making them generally valuable to researchers, students, and practitioners.

For those performing algorithmic research, much of the power of these datasets lies in the ability to compare results with prior research papers or with other algorithms. We strongly suggest using the same dataset used in a published study when comparing against those published results. Alternatively, in many cases it is possible to use tools such as LensKit to replicate prior results and apply them—and your new algorithm—to the new dataset.

For those not comparing with historical published results, we strongly recommend using the `20m` rating dataset because of the greater number of tags and the links to external metadata sources. Publishing results based on this dataset will make it easier for others to compare your results with the results of approaches that make greater use of metadata and content-based techniques.

Those who are interested in combining ratings or tagging data with rich movie content data should consider using the `20m` dataset in combination with external resources. The `20m` dataset includes a map of MovieLens IDs to IMDb[4] ("Internet Movie Database") IDs. Though IMDb does not provide APIs or encourage access to their data, their IDs are recognized across many different movie sites, including several with API access to interesting sources of metadata. Since MovieLens movie IDs are stable across datasets, the `20m` linking file can be used in combination with any of the earlier released datasets.

Educators should consider using the MovieLens `latest-small` dataset. Its size on disk makes it quick to download, parse, and process. It contains recent movies, which students may enjoy (as compared with the traditional `100k` dataset, that contains no movies past 1998). The dataset is also formatted as a `.csv` file with a header row, which is consistent with modern dataset conventions.

3.4. Alternative Datasets

There are several other datasets with explicit ratings that have been frequently used with (or instead of) the MovieLens datasets. These alternatives differ from MovieLens in terms of their size and shape, their domain, and their context of user interactions. Researchers might consider alternative datasets when they are a better match with

[4]http://imdb.com.

Table III. Quantitative Summary of the Prominent Alternative Datasets that Include Explicit Ratings Data

Name	Date Range	Domain	Rating Scale	Ratings	Density
Book-Crossing	2001–2004	books	0–10, 11 discrete values	1.1m	0.003%
EachMovie[a]	1995–1997	movies	0–14, 26 discrete values[b]	2.7m	2.872%
Jester (dataset1)	1999–2003	jokes	−10–10, continuous	4.1m	57.463%
Amazon	1996–2014	many	1–5, 5 discrete values	82.8m	<0.001%
Netflix Prize[a]	1998–2005	movies	1–5, 5 discrete values	100.5m	1.178%
Yahoo Music (C15)	1999–2009	music[c]	0–100, 101 discrete values[d]	262.8m	0.042%

Note: The sole computed column, *Density*, represents the percentage of cells in the full user-item matrix that contain rating values.
[a]No longer available.
[b]The result of using multiple ratings scales, rescaled for the final dataset.
[c]Includes ratings for songs, albums, artists, and genres.
[d]98% of rating values are multiples of 10.

the online system that they wish to simulate [Shani and Gunawardana 2011], or if they wish to evaluate an approach across multiple datasets. There are too many alternatives to mention; in this section, we discuss several of the most frequently cited or currently prominent alternatives. See Table III for an overview, ordered by number of ratings.

Book-Crossing. The Book-Crossing dataset[5] [Ziegler et al. 2005] is a snapshot of an online book-rating community. The dataset contains 100,000 users, 340,000 books, and 1.1 million ratings, collected in 2004. It has low density ($3.2x10^{-5}$). The ratings take 11 possible values, on a 0 to 10 scale. The snapshot was taken in 2004, but the ratings are not associated with timestamps. However, the dataset reveals users' locations and ages.

EachMovie. The EachMovie dataset was heavily used in research in the early 2000s before it was made unavailable by DEC due to legal concerns about the potential re-identifiability of the users in the dataset. This dataset contains 2.7 million ratings from 59,000 users across 1,500 movies, collected from 1995 through 1997. The ratings take 26 possible values in the range of 0 to 14, reflecting the use and subsequent rescaling of several approaches to collecting ratings. As mentioned earlier, this dataset was the original seed data for the MovieLens recommender, but this ratings data is not included in any of the MovieLens datasets.

Jester. The Jester datasets[6] [Goldberg et al. 2001] contain continuous, explicit ratings of jokes on a −10 to 10 scale—users rate jokes using a slider widget. There are several versions currently available; the largest ("Dataset 1") contains 4.1 million ratings from 73,421 users, collected from 1999 to 2003. Because there are only 100 jokes, this dataset is very dense relative to other options. This dataset does not contain rating timestamps.

Amazon. Amazon review datasets[7] [McAuley et al. 2015a, 2015b] are notable for several reasons. The ratings are associated with textual reviews, span 18 years, and encompass a wide variety of products, from instant videos to baby clothes. The full ("aggressively deduplicated") dataset is also very large and sparse: 82 million ratings from 21 million users across nearly 10 million items (density $3.9x10^{-7}$). The ratings take 5 possible values, 1 to 5, and are associated with timestamps.

Netflix. The Netflix Prize dataset was made available in 2006 as part of the Netflix Prize to improve the accuracy of predictions[8]. It was taken down in 2009 for legal reasons. The training dataset contains 480,000 users, 17,000 items, and 100 million

[5]http://www2.informatik.uni-freiburg.de/~cziegler/BX/.
[6]http://eigentaste.berkeley.edu/dataset/.
[7]http://jmcauley.ucsd.edu/data/amazon/.
[8]http://www.netflixprize.com/community/viewtopic.php?id=68.

ratings; it has density comparable to the MovieLens 10M dataset. The ratings take 5 possible values, 1 to 5, and are associated with timestamps.

Yahoo Music. Yahoo! Labs provides a number of music datasets[9] extracted from their Music product. Their "C15" dataset, released for the KDD Cup [Dror et al. 2012], provides user ratings on songs, albums, artists, and music genres on a 101-point (0–100) rating scale (though 98% of the ratings are multiples of 10). The dataset contains 1 million users, 620,000 music items (the item types listed earlier are mixed together), and 262 million ratings. The dataset contains partial timestamps that disclose time but obfuscate the absolute date.

4. CONCLUSION

In this article, we provide a historical view of the MovieLens system and datasets, which have had a substantial impact in education, research, and industry. The datasets are a product of the MovieLens system, which has changed substantially in the 17 years since its first release. We discuss the key features of the system in terms of their impact on users, their impact on the research literature through field studies, and their downstream impact on the datasets.

We recently released a new version of MovieLens (v4) and a new benchmark dataset (`20m`). We hope these releases continue to encourage the development of high-quality educational materials, software systems, startup companies, and academic research.

ACKNOWLEDGMENTS

Many have dedicated themselves to building and improving MovieLens and the MovieLens datasets. In particular, we would like to thank John Riedl for his instrumental contributions and leadership. Other key contributors include Istvan Albert, Al Borchers, Dan Cosley, Brent J. Dahlen, Rich Davies, Michael Ekstrand, Dan Frankowski, Nathaniel Good, Jon Herlocker, Daniel Kluver, Shyong (Tony) Lam, Michael Ludwig, Sean McNee, Chad Salvatore, Shilad Sen, and Loren Terveen. We also gratefully acknowledge the MovieLens members, who make this project possible.

REFERENCES

Shuo Chang, F. Maxwell Harper, and Loren Terveen. 2015. Using groups of items for preference elicitation in recommender systems. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW'15)*. ACM, New York, NY, 1258–1269. DOI: http://dx.doi.org/10.1145/2675133.2675210

Dan Cosley, Dan Frankowski, Sara Kiesler, Loren Terveen, and John Riedl. 2005. How oversight improves member-maintained communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*. ACM, New York, NY, 11–20. DOI: http://dx.doi.org/10.1145/1054972.1054975

Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. 2003. Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'03)*. ACM, New York, NY, 585–592. DOI: http://dx.doi.org/10.1145/642611.642713

Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM, New York, NY, 271–280. DOI: http://dx.doi.org/10.1145/1242572.1242610

Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems* 22, 1, 143–177. DOI: http://dx.doi.org/10.1145/963770.963776

Sara Drenner, Max Harper, Dan Frankowski, John Riedl, and Loren Terveen. 2006. Insert movie reference here: A system to bridge conversation and item-oriented web sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*. ACM, New York, NY, 951–954. DOI: http://dx.doi.org/10.1145/1124772.1124914

[9]http://webscope.sandbox.yahoo.com/catalog.php?datatype=c and http://webscope.sandbox.yahoo.com/catalog.php?datatype=r.

Gideon Dror, Yahoo Labs, Noam Koenigstein, Yehuda Koren, and Markus Weimer. 2012. The Yahoo! music dataset and KDDCup11. In *Journal of Machine Learning Research Workshop and Conference Proceedings: Proceedings of KDD Cup 2011*. 3–18.

Michael D. Ekstrand, Daniel Kluver, F. Maxwell Harper, and Joseph A. Konstan. 2015. Letting users choose recommender algorithms: An experimental study. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys'15)*. ACM, New York, NY, 11–18. DOI: http://dx.doi.org/10.1145/2792838.2800195

Michael D. Ekstrand, Michael Ludwig, Joseph A. Konstan, and John T. Riedl. 2011. Rethinking the recommender research ecosystem: Reproducibility, openness, and lenskit. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, NY, 133–140. DOI: http://dx.doi.org/10.1145/2043932.2043958

Malcolm Gladwell. 1999. The science of the sleeper. *The New Yorker*. Retrieved November 13, 2015 from http://gladwell.com/the-science-of-the-sleeper/.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval* 4, 2, 133–151. DOI: http://dx.doi.org/10.1023/A:1011419012209

F. Maxwell Harper, Dan Frankowski, Sara Drenner, Yuqing Ren, Sara Kiesler, Loren Terveen, Robert Kraut, and John Riedl. 2007a. Talk amongst yourselves: Inviting users to participate in online conversations. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI'07)*. ACM, New York, NY, 62–71. DOI: http://dx.doi.org/10.1145/1216295.1216313

F. Maxwell Harper, Shilad Sen, and Dan Frankowski. 2007b. Supporting social recommendations with activity-balanced clustering. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys'07)*. ACM, New York, NY, 165–168. DOI: http://dx.doi.org/10.1145/1297231.1297262

F. Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen. 2015. Putting users in control of their recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys'15)*. ACM, New York, NY, 3–10. DOI: http://dx.doi.org/10.1145/2792838.2800179

George Karypis. 2001. Evaluation of item-based top-N recommendation algorithms. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM'01)*. ACM, New York, NY, 247–254. DOI: http://dx.doi.org/10.1145/502585.502627

Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM* 40, 3, 77–87. DOI: http://dx.doi.org/10.1145/245108.245126

Joseph A. Konstan, J. D. Walker, D. Christopher Brooks, Keith Brown, and Michael D. Ekstrand. 2014. Teaching recommender systems at large scale: Evaluation and lessons learned from a hybrid MOOC. In *Proceedings of the 1st ACM Conference on Learning @ Scale Conference (L@S'14)*. ACM, New York, NY, 61–70. DOI: http://dx.doi.org/10.1145/2556325.2566244

John G. Lynch, Jr., Dipankar Chakravarti, and Anusree Mitra. 1991. Contrast effects in consumer judgments: Changes in mental representations or in the anchoring of rating scales? *Journal of Consumer Research* 18, 3, 284–297.

Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys'07)*. ACM, New York, NY, 17–24. DOI: http://dx.doi.org/10.1145/1297231.1297235

Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015a. Inferring networks of substitutable and complementary products. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, New York, NY, 785–794. DOI: http://dx.doi.org/10.1145/2783258.2783381

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015b. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. ACM, New York, NY, 43–52. DOI: http://dx.doi.org/10.1145/2766462.2767755

Bradley Norman Miller. 2003. *Toward a Personal Recommender System*. Ph.D. dissertation. University of Minnesota, Minneapolis, MN. Retrieved from http://search.proquest.com/dissertations/docview/305324342/abstract/A46BCC87FC4D4DD4PQ/1?accountid=14586.

Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. PolyLens: A recommender system for groups of users. In *Proceedings of the 7th Conference on European Conference on Computer Supported Cooperative Work (ECSCW'01)*. Kluwer Academic Publishers, Norwell, MA, 199–218.

John O'Donovan and Barry Smyth. 2005. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI'05)*. ACM, New York, NY, 167–174. DOI: http://dx.doi.org/10.1145/1040830.1040870

Nick Pentreath. 2015. *Machine Learning with Spark*. Packt Publishing Ltd, Birmingham, UK.

Reid Priedhorsky, Mikhil Masli, and Loren Terveen. 2010. Eliciting and focusing geographic volunteer work. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW'10)*. ACM, New York, NY, 61–70. DOI: http://dx.doi.org/10.1145/1718918.1718931

Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. 2002. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI'02)*. ACM, New York, NY, 127–134. DOI: http://dx.doi.org/10.1145/502716.502737

Al Mamunur Rashid, George Karypis, and John Riedl. 2008. Learning preferences of new users in recommender systems: An information theoretic approach. *ACM SIGKDD Explorations Newsletter* 10, 2, 90–100. DOI: http://dx.doi.org/10.1145/1540276.1540302

Yuqing Ren, F. Harper, Sara Drenner, Loren Terveen, Sara Kiesler, John Riedl, and Robert Kraut. 2012. Building member attachment in online communities: Applying theories of group identity and interpersonal bonds. *Management Information Systems Quarterly* 36, 3 (Sept. 2012), 841–864.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An open architecture for collaborative filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94)*. ACM, New York, NY, 175–186. DOI: http://dx.doi.org/10.1145/192844.192905

Eric Ries. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, New York, NY.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. *Application of Dimensionality Reduction in Recommender System—A Case Study*. Technical Report. DTIC Document. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA439541.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW'01)*. ACM, New York, NY, 285–295. DOI: http://dx.doi.org/10.1145/371920.372071

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*. ACM, New York, NY, 253–260. DOI: http://dx.doi.org/10.1145/564376.564421

Toby Segaran. 2007. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly Media, Inc., Sebastopol, CA.

Shilad Sen, F. Maxwell Harper, Adam LaPitz, and John Riedl. 2007. The quest for quality tags. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work (GROUP'07)*. ACM, New York, NY, 361–370. DOI: http://dx.doi.org/10.1145/1316624.1316678

Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. 2006. Tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW'06)*. ACM, New York, NY, 181–190. DOI: http://dx.doi.org/10.1145/1180875.1180904

Shilad Sen, Jesse Vig, and John Riedl. 2009. Learning to recognize valuable tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI'09)*. ACM, New York, NY, 87–96. DOI: http://dx.doi.org/10.1145/1502650.1502666

Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, New York, NY, 257–297. http://link.springer.com/chapter/10.1007/978-0-387-85820-3_8

Jesse Vig, Shilad Sen, and John Riedl. 2012. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems* 2, 3, 13:1–13:44. DOI: http://dx.doi.org/10.1145/2362394.2362395

Jesse Vig, Matthew Soukup, Shilad Sen, and John Riedl. 2010. Tag expression: Tagging with feeling. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST'10)*. ACM, New York, NY, 323–332. DOI: http://dx.doi.org/10.1145/1866029.1866079

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*. ACM, New York, NY, 22–32. DOI: http://dx.doi.org/10.1145/1060745.1060754

Received July 2015; revised October 2015; accepted October 2015