The Browsemaps: Collaborative Filtering at LinkedIn

Lili Wu
LinkedIn Corporation
lwu@linkedin.com

Sam Shah
LinkedIn Corporation
samshah@linkedin.com

Sean Choi
LinkedIn Corporation
schoi@linkedin.com

Mitul Tiwari
LinkedIn Corporation
mtiwari@linkedin.com

Christian Posse
Google Inc
cposse@google.com

ABSTRACT

Many web properties make extensive use of item-based collaborative filtering, which showcases relationships between pairs of items based on the wisdom of the crowd, for navigational aids and recommendation systems. This paper presents LinkedIn's collaborative filtering infrastructure, known as Browsemaps. A key characteristic of our solution is that rapid development, deployment, and computation of collaborative filtering is possible for almost any use case through a simple domain specific language with scaling and other operational issues handled by the system. As part of this work, we also present case studies on how this platform is used at LinkedIn in various recommendation products, as well as lessons learned in the field over the several years this system has been in production.

1. INTRODUCTION

The proliferation of data and the information-rich user experiences have transformed data mining into a core production use case, especially in the consumer web space. A typical example is showcasing relationships between pairs of items based on the wisdom of the crowd, also known as item-to-item collaborative filtering (ICF) [10]. At LinkedIn, the largest online professional social network with over 300 million members, item-to-item collaborative filtering is used for people, job, company, group, and other entity recommendations and is a principal component of engagement. That is, for each type of entity on the site, there exists a navigational aid that allows members to browse and discover other content, as shown in Figure 1. We call each of these a *browsemap*.

Initially designed to showcase co-occurrence in views of other member's profiles (a profile browsemap or "People Who Viewed This Profile Also Viewed"), we grew the browsemap computation into a generic horizontal piece of relevance infrastructure that can support any entity with a simple configuration change. This infrastructure, the Browsemap platform, enables easy addition of other navigational content recommendations. Moreover, the availability of a scalable collaborative filtering primitive also permits easy plug-in of ICF-based features into other models and products. For example, the "Companies You May Want to Follow" recommendation product, which allows members to follow a company to receive its status updates, uses the Browsemap platform to compute collaborative filtering of company follows as part of its recommendation set. In essence, browsemaps form a latent graph of co-occurrences between any entity type on LinkedIn.

Browsemap is a managed platform with mostly shared components and some vertical-specific logic. LinkedIn's frontend framework emits activity events on every page view. A parameterized pipeline for each entity type uses these events to construct a co-occurrence matrix with some entity-specific tuning. Browsemaps are computed offline incrementally in batch on Hadoop [15], loaded into an online key-value store [14], and queried through an entity-agnostic online API. As Browsemap is a horizontal platform, it provides high leverage to each application developer through reuse of common components, centralized monitoring, and ease of scaling to the billions of weekly page views on LinkedIn. An application developer simply specifies the type of collaborative filtering they need, the location of the input data, and optionally changes any parameters if needed; the resulting browsemap is then available in Hadoop and via an online API in a straightforward manner.

The Browsemap platform is established at LinkedIn and powers over two dozen use cases on the site.

The contributions of this paper are the following:

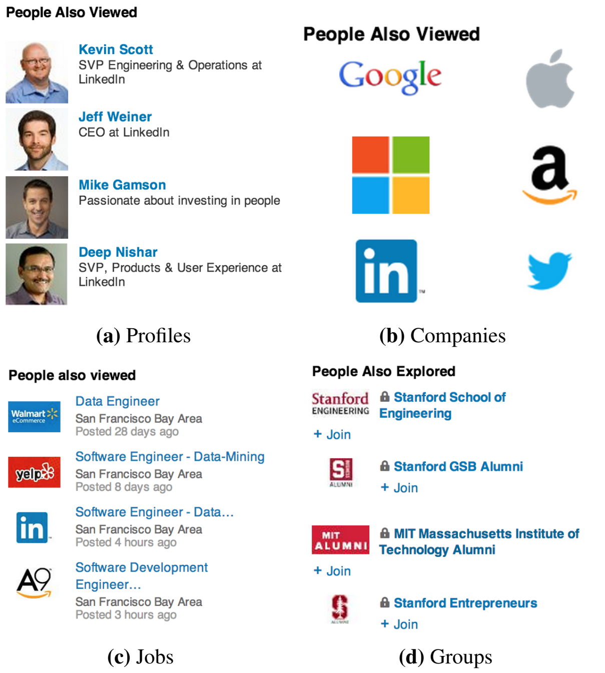
1. The architecture of a large-scale collaborative filtering system at a top online property;
2. A description of the diverse set of applications that are powered through the availability of an easy collaborative filtering primitive;
3. A collection of lessons learned in developing and deploying the Browsemap platform in the field.

The rest of the paper is organized as follows. Section 2 describes the Browsemap platform with Section 3 showcasing the applications that are powered with this infrastructure. Section 4 recounts lessons learned in deploying and running browsemaps. Section 5 catalogs related work and finally, Section 6 concludes.

2. ARCHITECTURE

Browsemap is an item-to-item collaborative filtering platform, where member browsing histories are used to build a latent graph of co-occurrences between the entities.

The platform has three properties. First, it supports all entity types on LinkedIn, such as member profiles, company pages, and job postings. Creating a browsemap for a new entity type requires minimal effort. Second, the platform is flexible to address each entity's own characteristics. For example, while member profiles do not expire, a job posting does expire after a certain date. The computation of the job browsemap needs to remove such expired jobs. Last, the platform is able to scale, through judicious use of incremental computation and pipelining, to efficiently scale across the billions of weekly page views on LinkedIn.



People Also Viewed

Kevin Scott
SVP Engineering & Operations at LinkedIn

Jeff Weiner
CEO at LinkedIn

Mike Gamson
Passionate about investing in people

Deep Nishar
SVP, Products & User Experience at LinkedIn

(**a**) Profiles

People Also Viewed

(**b**) Companies

People also viewed

Data Engineer
San Francisco Bay Area
Posted 28 days ago

Software Engineer - Data-Mining
San Francisco Bay Area
Posted 8 days ago

Software Engineer - Data...
San Francisco Bay Area
Posted 4 hours ago

Software Development Engineer...
San Francisco Bay Area
Posted 3 hours ago

(**c**) Jobs

People Also Explored

Stanford School of Engineering
+ Join

Stanford GSB Alumni
+ Join

MIT Massachusetts Institute of Technology Alumni
+ Join

Stanford Entrepreneurs
+ Join

(**d**) Groups

Figure 1: Examples of browsemaps that are generated for various entities. All of the recommendations are generated from co-occurrence of views between the items recommended.

Figure 2 illustrates the Browsemap system architecture. Browsemap platform is a hybrid offline/online system. The offline system uses Hadoop [15] for its batch computation engine because of its high throughput, fault tolerance, and horizontal scalability. Computed browsemaps are bulk loaded into a distributed key-value store, which permits low-latency queries.

2.1 Offline Batch Computation

LinkedIn's frontend services emit activity events on every page view either on LinkedIn's website or through our mobile application. These behavior events are transported to Hadoop by a low-latency distributed publish-subscribe system for event collection [14].

The Browsemap Engine uses the well-known technique of association rule mining or co-occurrence [1] to process the data and generate the latent browsemap graph. The system uses techniques to dampen entities that are overly popular. For example, President Barack Obama is an active member of the site and his profile is viewed several orders of magnitude more than most other members; this dampening prevents him from being overly correlated throughout the ecosystem. The system also includes a form of hysteresis so that newer views are weighted more heavily than older ones, creating a sense of dynamism.

The engine supports the diverse characteristics of the browsemaps on LinkedIn. First, there are many entities such as job and company, and each entity may have multiple types of activity events. For example, job entity has two types of events—*view* and *apply*; people can view and apply for jobs. Similarly, the company entity has *view* and *follow* activity events. Multiple event types can be combined together to generate one browsemap or they can each power a browsemap. For example, job browsemap combines the job-apply and job-view events with more emphasis on job-apply activity. Company entity, on the other hand, has company-view and company-follow browsemaps, each is built on an event type of company entity. Lastly, different browsemaps can share some common functionalities while each has its own requirements. For example, all browsemaps need to filter out activities by spam users, and job browsemap has an additional requirement to exclude expired job.

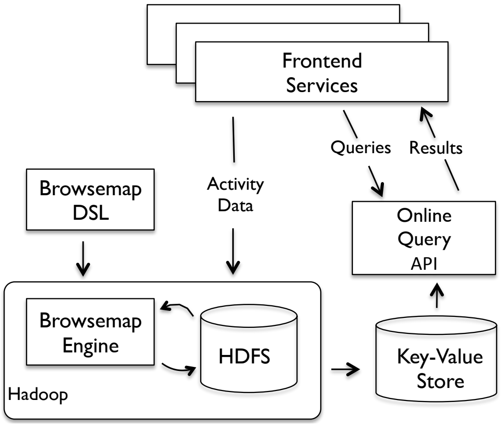


Figure 2: Browsemap architecture consists of offline computation on Hadoop to generate a set of browsemaps, and online query API which fetches the results from a key-value store.

To meet the different requirements of the various browsemaps, we developed an in-house Browsemap Domain-Specific Language (Browsemap DSL) that describes how to build a browsemap, and a collection of modules that can be chained together via the DSL. The *module collection* contains a set of *modules*; each one is a component performing a particular task. Some modules can be used by different browsemaps such as counting the co-occurrence and filtering out spam user activities, and some modules are specific to a browsemap, such as filtering out expired jobs.

A configuration file written in the DSL defines a browsemap workflow. First it describes the module dependency: which modules are to be used and how the modules are chained together to create the workflow. The input dataset and output location of a module are also specified in the configuration file.

In addition, the Browsemap DSL provides mechanisms to tune parameters for an entity-specific browsemap workflow. For example, the browsemap for job entity needs to be refreshed more frequently due to the ephemeral nature of job postings, but the browsemaps for company entity can be refreshed less frequently as it is more static.

The collection of modules promotes knowledge sharing and is a main contributing factor for the quick development of new browsemaps. While some modules are specific to a browsemap such as expired job filtering, many are common modules that can be shared among different browsemaps.

Internally, each module is implemented as a set of Hadoop jobs, where each job produces output that is the input for the subsequent job. The workflows are managed and executed by a workflow manager [14]. Certain modules are computed incrementally with Hourglass [3], an open-source library that operationalizes incremental computation of time series data.

The job entity has job-view and job-apply events. Computation of the job browsemap starts with combining activities from these two events. The aggregated dataset is the input to the next module that filters out expired job, a module that is only used by the job browsemap workflow. After filtering expired jobs, the remaining active jobs become the input of the subsequent module which is to filter out activities from spam users. After a few more steps, the co-occurrence-counting module is used to do the bulk work of generating the latent graph. Before the workflow finishes, some techniques to alleviate the cold-start problems are applied to increase the level of coverage.

Similar to the job entity, company entity also has two event types. A member can view and follow a company. Contrary to the job entity that combines the events, company entity has two browsemaps based on the two event types: company-view and company-follow browsemaps. The two workflow are very similar in that both use the same set of common modules such as spam user filtering, co-occurrence counting, and cold-start techniques. The difference is at some of the tuning parameters. For example, company-follow events are less frequent than the company-view events, and thus needs a longer session length when doing computation. This kind of tuning parameters are specified in the configuration file that generates the browsemap workflows.

As of writing, the Browsemap Engine processes hundreds of terabytes weekly, and has more than 130 Hadoop jobs to compute all entities.

2.2 Online Query API

All of the browsemap dataset computed by the offline Browsemap Engine are bulk loaded into Voldemort [13], an open-source distributed key-value store, for the Browsemap online query API to access. Voldemort provides low latency, high throughput and high availability features that facilitate responding to user requests in a timely manner: 99% of requests are serviced within 10 milliseconds.

The online API is entity-agnostic; no change is needed when a new browsemap dataset is added into Voldemort. The store is a composite key of the entity type and identifier, with the value representing a set of recommendations. We can A/B test different models by shunting to different recommendation stores for a percentage of viewing traffic.

3. APPLICATIONS

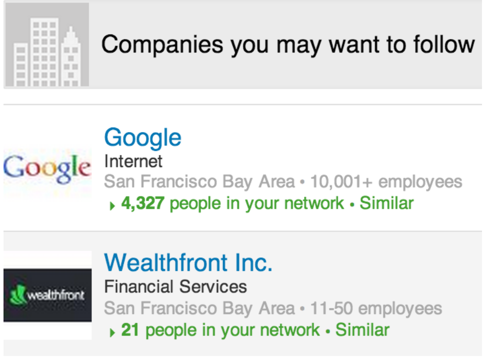
The Browsemap platform powers many navigational aids on LinkedIn. They are well received by our members and a substantial portion of LinkedIn's traffic is directly attributed to them. Besides being a principal component of engagement on LinkedIn, these browsemaps are used in several hybrid recommendation applications that use a combination of collaborative-filtering and content-based features. The aggregated behavior of a large number of users provides strong signals to the applications, in addition to content information such as member profiles and job description. Inclusion of collaborative-filtering-based features is to simply plug-in the readily available browsemap datasets.

3.1 Navigational Aids

LinkedIn has many entities and each entity has a navigational aid. Figure 1 illustrates a few examples. Shown in Figure 1a, a navigational aid is displayed on a member's profile that allows members to discover other related profiles, as shown in. Similarly, the group page has the navigational aid showcasing other groups, as illustrated by Figure 1d.

An entity can be associated with multiple types of activity events, as in the case of job and company entities. Job navigational aid, as illustrated by Figure 1c, is computed based on both of its *apply* and *view* events, with more emphasis given to the *apply* events. A job seeker's application to two jobs is a much stronger signal showing that these two jobs are related.

Company entity, on the other hand, has two navigational aids, one is powered by the company-follow browsemap and the other is powered by the company-view browsemap (shown in Figure 1b). The company-follow browsemap is for deep engagement with a company; following a company lets members to keep track of the status updates from this company. The company-view browsemap, however, is for cursory browsing and serendipitous discovery of content. The two navigational aids serve different needs for LinkedIn members.



Companies you may want to follow

Google
Internet
San Francisco Bay Area • 10,001+ employees
▸ 4,327 people in your network • Similar

Wealthfront Inc.
Financial Services
San Francisco Bay Area • 11-50 employees
▸ 21 people in your network • Similar

Figure 3: An illustration of *Companies You May Want To Follow*, a user-to-company recommendation system. The recommendations in this module are generated by combining signals from company-follow browsemap and other content based features.

Lastly, a particular member segment may want a more customized navigational experience. Recruiting is a main use case exercised by premium users of LinkedIn, and recruiters can use a customized navigational aid to discover profiles that are usually viewed together by other recruiters. Using the Browsemap platform, this is easily achieved by plug-in a member selection module that selects viewing events performed by the recruiting community.

3.2 Companies You May Want To Follow

Companies can establish presence on LinkedIn through Company Pages. Currently there are more than 3 million companies that have created Company Pages to showcase their business. "Companies You May Want To Follow", illustrated by Figure 3, is a product on LinkedIn that recommends companies to members using a combination of collaborative-filtering and content-based features. A member's previous *follow* action is a strong signal about interest in related companies, the information that company-follow browsemap can provide.

At a high level, the recommendation algorithm finds a set of possible companies, the *candidate set*, that the member may be interested in. Each company in the candidate set forms a (member, company) tuple with the member. The algorithm computes a propensity score for each tuple predicting the probability the member will follow this company. The companies with high propensity scores are returned as the recommendations for the member.

Figure 4 demonstrates the process that company-follow browsemap is used to generate the "related-companies" feature. This feature is used later to enrich the member profile by augmenting the textual content input by the member. The feature is generated by iterating through all of the companies that a member has already followed and retrieves the company-follow browsemap for each of them. Merging all of the browsemaps produces a list of related companies that the member may like.

Besides the company-follow browsemap, this recommendation system also uses content-based features. Member features such as industry, location, and experience are used. Company features include company name, industry, location, and description and so on.

The propensity score for a (member, company) tuple is computed by pairwise-matching the related features of the member and company entities. Figure 5 illustrates the matching process. It is broken down to a series of matching between member and company features.

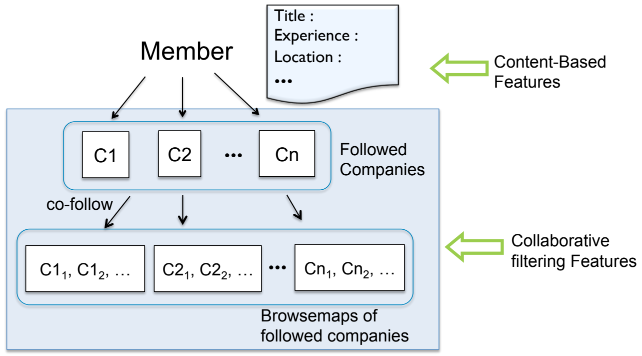


Figure 4: "Companies You May Want To Follow" augments member information with the company-follow browsemap. It iterates through all companies a member already follows, and aggregate the browsemaps of these companies.

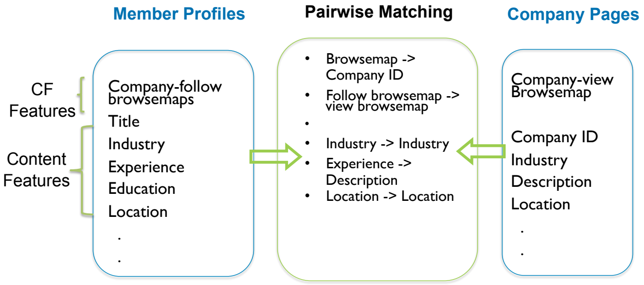


Figure 5: "Companies You May Want To Follow" has two types of features for a membe: collaborative-filtering features extracted from the company-follow and company-view browsemaps, and the content-based features extracted from the member profile. The algorithm pairwise-matches the corresponding fields from member and company entities.

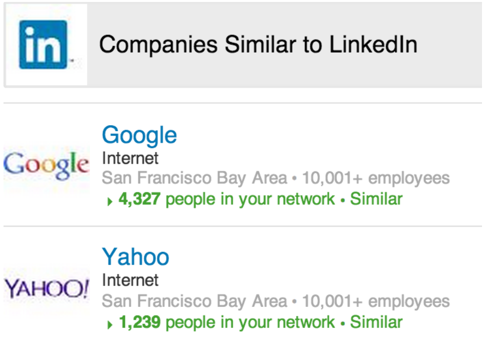
For example, the member's related-companies feature is matched against the company name from the company entity. The member's industry is matched against the company's industry, and the member's experience is compared with the company's description. Each pairwise-matching produces a score based on cosine similarity between vector space representations of the corresponding features. A binary classification model that optimizes for click-through-rate is learned with historical data. The weights learned by the model are used to combine the individual scores to get an overall propensity score for this (member, company) tuple.

The company-follow browsemap is important in this product because it captures a notion of connection between companies that is driven by members' preference. It creates a latent graph of the companies that is not visible by studying the content alone.

3.3 Similar Companies

The previous product "Companies You May Want To Follow" is a member-to-company recommendation, suggesting companies based on matching member and company information. "Similar Companies", shown by Figure 6, is a different recommendation product on LinkedIn that suggests companies based on matching company and company information.

Collaborative-filtering-based and content-based features are extracted from the company entities. The collaborative filtering features include three browsemaps: company-follow, company-view, and company-occupation browsemaps. They are pairwise-matched

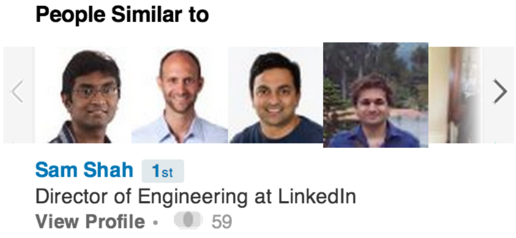


Companies Similar to LinkedIn

Google
Internet
San Francisco Bay Area • 10,001+ employees
4,327 people in your network • Similar

Yahoo
Internet
San Francisco Bay Area • 10,001+ employees
1,239 people in your network • Similar

Figure 6: An illustration of *Similar Companies*, a company-to-company recommendation system. The recommendations in this module are generated by combining signals from multiple browsemaps and other content based features.



People Similar to

Sam Shah 1st
Director of Engineering at LinkedIn
View Profile • 59

Figure 7: "Similar Profiles" is a hybrid recommendation system for member-to-member suggestions. It uses both company-view browsemap and profile browsemap to enrich the profile information.

to the target company. The perception is that if the target company is contained in one or more of the browsemaps, it indicates some degree of similarity because of the aggregated member behavior.

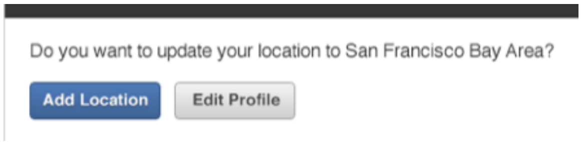
3.4 Similar Profiles

Helping recruiters and hiring managers to find highly qualified candidates is an important service LinkedIn provides. Through a product called "Similar Profiles", hiring professionals can discover other similar quality talent on LinkedIn.

Company-view browsemap and profile browsemap, along with several content features from profiles, are used for this recommendation system. The algorithm for "Similar Profiles" follows the same design pattern of the previous two recommendation systems.

The company-view browsemap is used to expand the source member's current company to a set of companies. The expanded set is pairwise-matched against the target company. This expansion significantly increases the recall of the model. Although this enhancement is done with minimal effort due to the availability of the browsemap dataset, it is one of the most powerful signals in the model — just leveraging the company-view browsemap alone increased this product's contribution to profile view by more than 30%.

The previous examples demonstrate how the browsemap is used directly as additional features. It is also possible to use browsemap as a level of indirection, as exhibited by how the profile browsemap is used to extend the member content information. We can augment member profiles with more content from other affiliated profiles profiles. For example, a member's skill information can be augmented by skills from people he is affiliated with. We call this the "virtual profile" [8] of the member. In "Similar Profiles", profile browsemap is used to find the affiliated members. The perception



Do you want to update your location to San Francisco Bay Area?

Figure 8: An illustration of “Suggested profile updates”. Suggested location update is a module that recommends a user to update their current location based on profile browsemap and user’s connections.

is that LinkedIn’s members are more likely to be viewed with other members who are similar in professional aspects, such as titles, skills, employment history, and education background. Aggregating the member information from all of the member’s profile browsemap essentially extends the member’s profile to a much richer profile. A/B testing proved the perception—we observed that “Similar Profiles” generates 15% more profile views with the addition of the virtual profile.

3.5 Suggested Profile Updates

LinkedIn has always encouraged its members to complete as many sections of their member profile as possible. When a user has more detailed information such as work experience, education, and location, LinkedIn is able to provide better service to her with a richer user experience and more personalized recommendations on the website.

To make it easier for a member updating a profile, LinkedIn predicts certain attributes that she has not yet included, such as company and location. The prediction is shown to the member, and upon approval, the information is saved to her member profile. Figure 8 shows the suggested location update for a member.

Social graphs of a member can provide strong location clues. There are two types of graphs: the latent graph provided by the profile browsemap, and the explicit connection graph the member has established on LinkedIn. The perception of using profile browsemap is that a member is usually viewed together with the people they interact with in the real world.

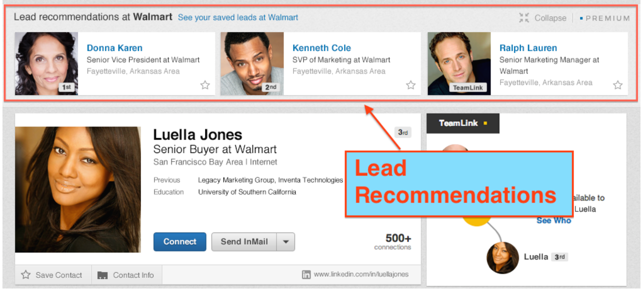
The algorithm’s goal is to find the possible locations for a member. The problem is formulated to find the likelihood that a member resides in a particular location. That is, with a collection of (member, location) tuples, find the probability of each tuple. The member’s most probable location can be predicted by performing a top-1 operation on these probabilities.

Each tuple is associated with a feature vector that is extracted from both graphs: the number of related profiles who indicated on their member profiles that they reside in the given location. The (member, location) tuple’s probability is computed based on a binary classification model. Aggregating through all (member, location) tuples, the location with the highest probability score is used as the predicted location.

3.6 Lead recommendations

“Lead Recommendation” is a product that helps sales professionals discover more leads at their client companies. Figure 9 illustrates how it is presented to sales professionals. On a key prospective client’s profile page, a list of recommended members is shown, suggesting some decision-makers and influencers critical to a successful sale at the same company.

The product is based on the insight that a prospect’s colleagues who work closely with the prospect and have similar title seniority levels as the prospect can potentially influence the prospect. The algorithm is split into two steps, both leveraging the prospect’s profile browsemap: discovering the prospect’s colleagues in his company who he works closely, and identifying the colleagues who have similar seniority level as the prospect.



Lead recommendations at Walmart See your saved leads at Walmart

Donna Karen
Senior Vice President at Walmart
Fayetteville, Arkansas Area

Kenneth Cole
SVP of Marketing at Walmart
Fayetteville, Arkansas Area

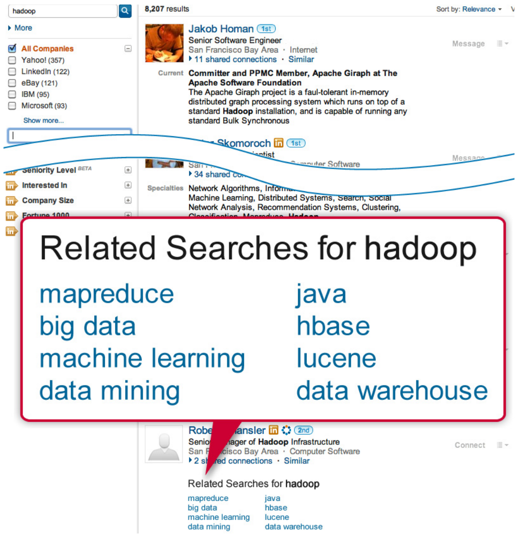
Ralph Lauren
Senior Marketing Manager at Walmart
Fayetteville, Arkansas Area

Luella Jones
Senior Buyer at Walmart
San Francisco Bay Area | Internet
Previous Legacy Marketing Group, Inventa Technologies
Education University of Southern California

500+ connections

www.linkedin.com/in/luellajones

Figure 9: An illustration of “Lead Recommendation”, a product that allows sale professionals to discover of new leads at their client companies.



hadoop

8,207 results

Jakob Homan 1st
Senior Software Engineer
San Francisco Bay Area · Internet
11 shared connections · Similar
Current Committer and PPMC Member, Apache Giraph at The Apache Software Foundation
The Apache Giraph project is a faul-tolerant in-memory distributed graph processing system which runs on top of a standard **Hadoop** installation, and is capable of running any standard Bulk Synchronous

Specialties Network Algorithms, Inform... Machine Learning, Distributed Systems, Search, Social Network Analysis, Recommendation Systems, Clustering,

Related Searches for hadoop

mapreduce
big data
machine learning
data mining
java
hbase
lucene
data warehouse

Senior ...ager of **Hadoop** Infrastructure
San F...cisco Bay Area · Computer Software
2 sh...ed connections · Similar

Related Searches for **hadoop**

mapreduce
big data
machine learning
data mining
java
hbase
lucene
data warehouse

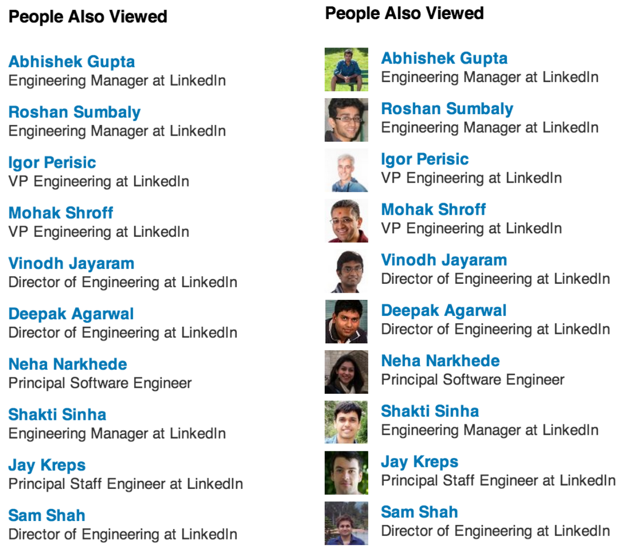
Figure 10: A screenshot of related searches in the context of a search for the query “Hadoop”. It uses search query browsemap as a signal for generating related searches.

The prospect’s profile browsemap and explicit connection graphs are used to identify his close co-workers. This set of members is the candidate set, that is, the member pool that the recommendations are generated from. Similar to the “Suggest Profile Update” product, with a collection of (prospect, candidate) tuples, the problem can be formulated as calculating the probability, or the score, of each tuple. By aggregating the tuples for a prospect, the algorithm returns a top-n list based on the scores.

Profile browsemap is further used to extract seniority features from the the member’s current title. Each title in LinkedIn’s database is associated with a seniority score, representing the number of years of experience for the average member to achieve that position. The higher the seniority of a position, the more years it requires to attain the position. Employees with a similar level of seniority in a company usually have a similar seniority score and are usually viewed together. Based on this premise, we introduced several features utilizing seniority information such as the scores of prospect and candidate, and the average scores of their profile browsemap and explicit connection graph.

3.7 Related Searches

Related searches [9] is a search tool that suggests other queries that are related to the user queries. As shown by an example in



People Also Viewed

Abhishek Gupta
Engineering Manager at LinkedIn

Roshan Sumbaly
Engineering Manager at LinkedIn

Igor Perisic
VP Engineering at LinkedIn

Mohak Shroff
VP Engineering at LinkedIn

Vinodh Jayaram
Director of Engineering at LinkedIn

Deepak Agarwal
Director of Engineering at LinkedIn

Neha Narkhede
Principal Software Engineer

Shakti Sinha
Engineering Manager at LinkedIn

Jay Kreps
Principal Staff Engineer at LinkedIn

Sam Shah
Director of Engineering at LinkedIn

(a) Profile browsemap without member profile images

People Also Viewed

Abhishek Gupta
Engineering Manager at LinkedIn

Roshan Sumbaly
Engineering Manager at LinkedIn

Igor Perisic
VP Engineering at LinkedIn

Mohak Shroff
VP Engineering at LinkedIn

Vinodh Jayaram
Director of Engineering at LinkedIn

Deepak Agarwal
Director of Engineering at LinkedIn

Neha Narkhede
Principal Software Engineer

Shakti Sinha
Engineering Manager at LinkedIn

Jay Kreps
Principal Staff Engineer at LinkedIn

Sam Shah
Director of Engineering at LinkedIn

(b) Profile browsemap with member profile images

Figure 11: An example of UI enhancement without any changes in the items recommended. Showing profile images resulted in dramatic increase in CTR.

Figure 10, related searches enables users to refine and explore by providing alternate related queries, and improves members' search experience to find relevant results.

There are four main signals used to capture various dimensions of relatedness among search queries and to come up with a unified set of related search suggestions. The first signal is based on collaborative filtering and is generated by Browsemap platform. The collaborative-filtering-based signal uses temporal locality between queries for relating search queries, that is, searches correlated by time are considered related. The other three signals are: queries correlated by result clicks, queries with overlapping terms, and queries that are correlated by clicks on related search suggestions. We apply a step-wise union based approach to combine the search suggestions generated by each of these signals, where results from collaborative filtering are given the highest preference since suggestions from this signal have highest click-through rate. We evaluated each of these signals and unified search suggestions both offline in terms of precision-recall metrics, and online through A/B tests. In both of these evaluations the collaborative-filtering-based signal generated from Browsemap platform performs significantly better than any other technique [9].

4. LESSONS LEARNED

The Browsemap platform has been in production at LinkedIn for over four years. During that time, we learned some valuable lessons during development and rollout of the system and the products it supports.

Tall oaks grow from little acorns.

Initially, we developed a profile browsemap that quickly received traction, which we rolled out to other entity types through a parameterized pipeline. However, we noticed other applications wishing for collaborative filtering, but struggling with scaling and incrementalizing computation to handle LinkedIn's data volume. Rather than have each team reinvent the wheel, we embarked on creating the Browsemaps platform.

The availability of this platform allows any developer to quickly bootstrap a new browsemap and put it into production, typically in just a day or two. Their application can then query the generic online API. Most of the developer's time is spent in understanding the nature of the product, input data preprocessing, and any vertical-specific requirements.

Browsemap is almost always used as the first recommendation product for any new entity or any new action type on the site. For example, LinkedIn recently introduced a feature that allows members to showcase their portfolio of work on their profile page. A natural extension has been to show a content browsemap. As another example, LinkedIn added the ability to follow influential members on the site to receive their updates and long-form posts. On initial launch, a browsemap was introduced as part of the sidebar of each article to show "wisdom of the crowd" recommendations on other articles. Further, once a member follows an influencer, we know they're in "following mode" and can display another browsemap of co-follows of that influencer in the flow to further increase conversions.

These recommender systems can then be augmented as needed with more sophisticated similarity rankers using browsemap data elements as latent features: co-views, co-follows, co-likes, co-comment, and co-search browsemaps in the influencer case.

A picture is worth a thousand words.

Our observation, which has been reiterated through many examples, is that the context and presentation of browsemaps or any recommendation is paramount for a truly relevant user experience. That is, design and presentation represents the largest ROI, with data engineering being second, and algorithms last. One must first understand the user intent, then optimize the flow, and set the right expectations.

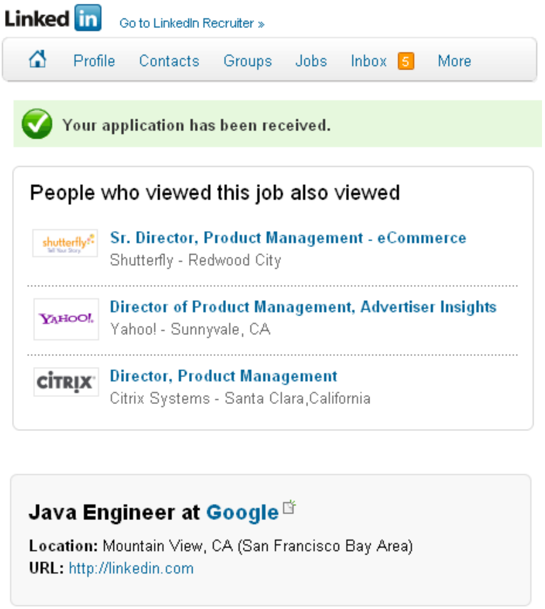
To elucidate this, consider Figure 11, which showcases the profile browsemaps that appear on a member's profile page. The recommendations provide a nice pivot when someone is in profile viewing mode, and the right expectations are set through explainability of their origins ("People Who Viewed This Profile Also Viewed.") On the left, these browsemaps show only the recommended member's name and title. On the right, the module also shows a member's photo, which makes the recommendations more pleasing and prominent (there were some engineering challenges to keep page load times constant.) The resulting 50% lift in click-through rate was one of the largest lifts in recommendation performance, and surpassed any algorithmic improvements by a sizable margin.

Besides changing the visual appearance, the context is also important. As an example, consider the jobs ecosystem at LinkedIn, where a member can naturally apply for a position after viewing a job page on the site. After they submit their application, the member is landed on a confirmation page, as shown in Figure 12. Up to this point the member is in the context of job searching and thus would very likely want to explore other related jobs, which is a great vehicle for the job browsemap. An A/B test of displaying the job browsemap at the end of the application process versus not indicates an order of magnitude lift in the job application rate.

One hand washes the other.

Further understanding and experimentation of user intent with recommendations has led us to the intuition that collaborative filtering-based and content-based recommendations serve different needs of members.

The job entity page, as shown in Figure 13, shows job browsemap recommendations. On the same page, it also shows "similar jobs", which performs content-based matching of job postings based on title, description, required skills, and location similarity. We per-



Your application has been received.

People who viewed this job also viewed

Sr. Director, Product Management - eCommerce
Shutterfly - Redwood City

Director of Product Management, Advertiser Insights
Yahoo! - Sunnyvale, CA

Director, Product Management
Citrix Systems - Santa Clara,California

Java Engineer at Google
Location: Mountain View, CA (San Francisco Bay Area)
URL: http://linkedin.com

Figure 12: An illustration of job browsemap to guide users to view related jobs after applying for a particular job.

formed a true multivariate test showing both recommendations, showing only one, adjusting locations and the number of recommendations, and found that these recommendation types can coexist without cannibalization of engagement. In fact, they actually amplify conversions as each module's conversion rate is almost independent of the other, as they independently show different facets. That is, collaborative filtering fulfills the members' curiosity to learn from other people, and content-based recommendation allows user to take a lead role in discovering new content. We repeated this test across other entity pages and found the same result.

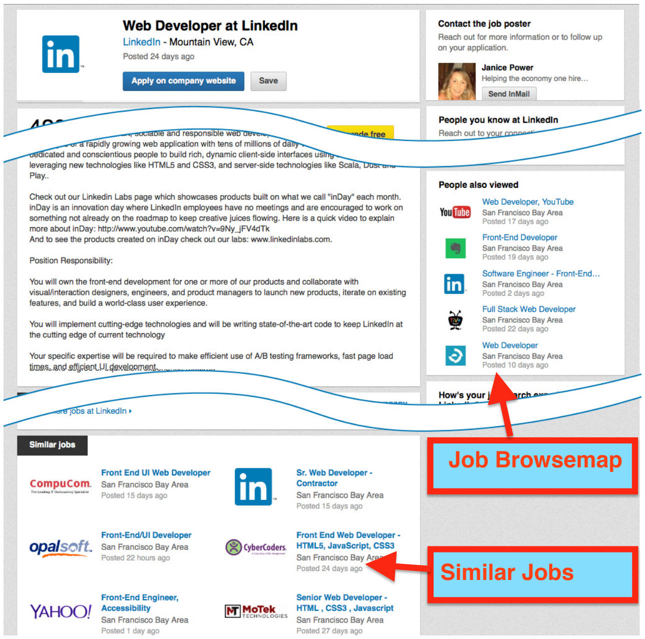
You can't get blood out of a stone.

A common problem inherent with collaborative filtering is cold start [11]. When a new job is posted or a new member registers, there is no activity on these new entities. Or for infrequently viewed items, there is sparsity in activity. Desparsification is vertical-specific and the platform provides techniques that can leverage the social graph or latent properties from other entities [12]. We've also commoditized another technique as part of the Browsemap platform that we found works reasonably well for our use cases: using a member's browsing history to personalize a backfill of any sparse entity recommendations.

Consider a member who has viewed several jobs, but then lands on a newly posted job with only minimal activity and thus a sparse browsemap. To combat this, the online system surfaces the browsemaps from the jobs he's previously viewed merged through a reduction function. Split testing has found that this technique can provide high coverage with virtually the same recommendation quality as measured by the click-through rate. However, the use of this technique is entity and context-specific. For example, we need to take care if the browsemap is used inside another recommender.

A chain is only as strong as its weakest link.

Browsemap computation, as any collaborative filtering recommendation, relies solely on user activities and is thus extremely sensitive to the quality and quantity of input data. Due to the many numbers and diverse nature of browsemaps that are computed, we initially faced significant consternation at the quality of input data:



Web Developer at LinkedIn
LinkedIn - Mountain View, CA
Posted 24 days ago

sociable and responsible web deve
of a rapidly growing web application with tens of millions of daily
dedicated and conscientious people to build rich, dynamic client-side interfaces using
leveraging new technologies like HTML5 and CSS3, and server-side technologies like Scala, Dust and Play..

Check out our Linkedin Labs page which showcases products built on what we call "inDay" each month. inDay is an innovation day where LinkedIn employees have no meetings and are encouraged to work on something not already on the roadmap to keep creative juices flowing. Here is a quick video to explain more about inDay: http://www.youtube.com/watch?v=9Ny_jFV4dTk
And to see the products created on inDay check out our labs: www.linkedinlabs.com.

Position Responsibility:

You will own the front-end development for one or more of our products and collaborate with visual/interaction designers, engineers, and product managers to launch new products, iterate on existing features, and build a world-class user experience.

You will implement cutting-edge technologies and will be writing state-of-the-art code to keep LinkedIn at the cutting edge of current technology

Your specific expertise will be required to make efficient use of A/B testing frameworks, fast page load times, and efficient UI development.

jobs at LinkedIn ›

Contact the job poster
Reach out for more information or to follow up on your application.
Janice Power
Helping the economy one hire…

People you know at LinkedIn
Reach out to your connect

People also viewed
Web Developer, YouTube
San Francisco Bay Area
Posted 17 days ago
Front-End Developer
San Francisco Bay Area
Posted 19 days ago
Software Engineer - Front-End…
San Francisco Bay Area
Posted 2 days ago
Full Stack Web Developer
San Francisco Bay Area
Posted 22 days ago
Web Developer
San Francisco Bay Area
Posted 10 days ago

How's your j arch

CompuCom Front End UI Web Developer San Francisco Bay Area Posted 15 days ago	Sr. Web Developer - Contractor San Francisco Bay Area Posted 15 days ago
opalsoft Front-End/UI Developer San Francisco Bay Area Posted 22 hours ago	CyberCoders Front End Web Developer - HTML5, JavaScript, CSS3 San Francisco Bay Area Posted 24 days ago
YAHOO! Front-End Engineer, Accessibility San Francisco Bay Area Posted 1 day ago	MoTek TECHNOLOGIES Senior Web Developer - HTML , CSS3 , Javascript San Francisco Bay Area Posted 27 days ago

Figure 13: Job description page has both collaborative filtering and content-based recommendations. The two recommendation types can coexist on the same page without cannibalization of engagement.

browsemaps are beholden to instrumentation on frontend services and the robustness of LinkedIn's data pipeline. The result was broken or incomplete browsemaps due to some upstream problem, which was often time-consuming to diagnose. For example, there could be a regression when emitting a activity event, which is hard to catch because it doesn't break business logic, only later downstream analysis.

In the last few years, LinkedIn has transformed its data pipeline from a batch-oriented file aggregation mechanism to a real-time publish-subscribe system [14]. We added robust auditing to ensure the per-hop reliable data transfer, from the frontend all the way to our relevance systems correctly. Browsemap platform also includes auditing as part of its run to compare input and output coverage and offline metrics, alerting if there's significant deviation. Further, we've added code-driven test automation for tracking events, so most regressions are caught as part of our continuous integration process, not after release. Data quality has vastly improved since these systems were put into place.

5. RELATED WORK

Collaborative filtering is a very generic term for a family of algorithms that share the similar goal of suggest new items or to predict the utility of a certain item for a particular user, based on the user's previous actions and the actions of the other like-minded users. Given a number of different algorithms, the family collaborative filtering algorithms can largely be divided into two families; memory-based and model-based. Memory-based mechanism is one of the earliest mechanisms of collaborative filtering. It's easy to implement and is effective. Some applied examples of this system are LinkedIn's Browsemap and Amazon's [7] item recommendation. Model-based mechanism involve developing models using various data mining and machine learning algorithms. Such algorithms include singular value decomposition, clustering models and many others. Some applied examples of this system are Netflix's [5] video recommendations and YouTube's [2] recommendation engine. Given the different approaches of collaborative filtering [6], memory-based approaches are often favored in the industry due to its simplicity with comparable performance are more amenable to explaining the reasoning behind prediction [4].

Most literature focuses on the recommendation algorithms while very few discussed about the creating a system that can serve millions of users into production. At the time of their writing, Amazon needed to handle 29 million items and several million catalog items. YouTube had millions of users with tens of millions of activity events. Amazon and YouTube also decoupled their offline computation from online serving to scale their systems. Browsemap is done in the similar fashion. In addition, Amazon and YouTube both built system targeted to a specific vertical; book recommendations for Amazon and movie recommendations for YouTube. LinkedIn's browsemap, on the other hand, is a solution that can support the development and deployment of many products horizontally and rapidly. It powers several principal recommendation products on LinkedIn. Browsemap dataset can also be leveraged as a complement to content-based features in other recommendation products on a professional social network website.

6. CONCLUSION

In this paper, we presented Browsemap, the item-based collaborative filtering platform at LinkedIn. A hybrid of offline/online system, Browsemap batch processes the computation-intensive task of correlating similarity among items, while serves results to users with low-latency. The ease of Browsemap's usability and quick onboarding procedure have enabled many behavior-based recommendation products on LinkedIn in the past few years. The various dataset it produces are also valuable to other content-based recommendations.

References

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[2] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The YouTube video recommendation system. In *RecSys*, pages 293–296, 2010.

[3] M. Hayes and S. Shah. Hourglass: A library for incremental processing on Hadoop. In *BigData Conference*, pages 742–752, 2013.

[4] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[5] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[6] Y. Koren and R. M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. 2011.

[7] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.

[8] H. Liu, M. Amin, B. Yan, and A. Bhasin. Generating supplemental content information using virtual profiles. In *RecSys*, pages 295–302, New York, NY, USA, 2013. ACM.

[9] A. Reda, Y. Park, M. Tiwari, C. Posse, and S. Shah. Metaphor: a system for related search recommendations. In *CIKM*, pages 664–673, 2012.

[10] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, New York, NY, USA, 2001. ACM.

[11] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260, New York, NY, USA, 2002. ACM.

[12] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, Jan. 2009.

[13] R. Sumbaly, J. Kreps, L. Gao, A. Feinberg, C. Soman, and S. Shah. Serving Large-scale Batch Computed Data with Project Voldemort. In *FAST*, 2012.

[14] R. Sumbaly, J. Kreps, and S. Shah. The "Big Data" ecosystem at LinkedIn. In *SIGMOD*, pages 1125–1134, New York, NY, USA, 2013. ACM.

[15] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, 2010.