

The Browsemaps: Collaborative Filtering at LinkedIn

Lili Wu
LinkedIn
lwu@linkedin.com

Sam Shah
LinkedIn
samshah@linkedin.com

Sean Choi
LinkedIn
schoi@linkedin.com

Mitul Tiwari
LinkedIn
mtiwari@linkedin.com

Christian Posse
Google
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. This paper presents LinkedIn's horizontal collaborative filtering infrastructure, known as *browsemaps*. The platform enables rapid development, deployment, and computation of collaborative filtering recommendations for almost any use case on LinkedIn. In addition, it provides centralized management of scaling, monitoring, and other operational tasks for online serving. We also present case studies on how LinkedIn uses this platform in various recommendation products, as well as lessons learned in the field over the several years this system has been in production.

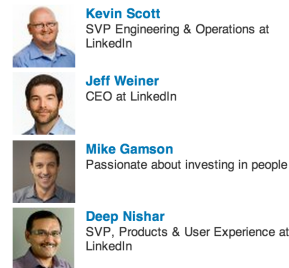
Keywords: collaborative filtering, social networks, recommender systems

1. INTRODUCTION

The proliferation of data and 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) [13]. At LinkedIn, the largest online professional social network, 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 entity type 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 piece of horizontal 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 inclusion of ICF-based features into other models and products. For example, "Companies You May Want to Follow" recommender system, which allows members to follow a company to receive its status

People Also Viewed



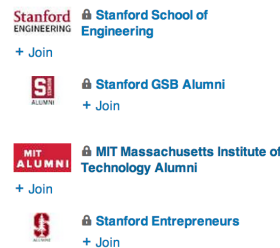
(a) Profiles

People also viewed



(b) Jobs

People Also Explored



(c) Groups

People Also Viewed



(d) Companies

Figure 1: Examples of browsemaps generated for various entities. Recommendations are computed by counting co-occurrence of views for these entities.

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 of across entity types 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 batches on Hadoop [16], 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 that is needed, the location of the input data, and 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 has been in production at LinkedIn for over four years and powers over two dozen use cases on the site.

The contributions of this paper are the following:

- The architecture of a large-scale collaborative filtering system at a top online property
- A description of the diverse set of applications that are powered through the availability of an easy collaborative filtering primitive
- 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 catalogs related work. Section 3 describes the Browsemap platform, with Section 4 showcasing applications that are powered with this infrastructure. Section 5 recounts lessons learned in deploying and running Browsemaps and finally, Section 6 concludes the paper.

2. RELATED WORK

Collaborative filtering is a commonly applied technique in commercial recommender systems. Amazon uses a neighborhood-based approach for its product recommendations [9]. YouTube combines covisitation statistics with a user’s personal activity on the site to show additional videos to watch [4]. Netflix employs matrix factorization methods for its movie recommendations [8]. eBay applies collaborative filtering as a component in their query suggestions [5]. Tivo uses correlation-based similarity to suggest shows to watch [2]. Yahoo! applies a factorization approach for song recommendations on its music property [7]. Google uses a linear combination of memory- and model-based collaborative filtering to showcase additional stories for the user to read in their news product [3].

In this work, we describe LinkedIn’s collaborative filtering solution, which, rather than applied to a specific vertical, is a horizontal recommendation infrastructure that powers several principal recommendation products and complements content-based features in other recommendation products. In particular, we describe the infrastructure as well as challenges and lessons learned in deploying and running a large-scale recommendation system.

3. ARCHITECTURE

The Browsemaps are an item-to-item collaborative filtering platform, where member browsing histories are used to build a latent graph of co-occurrences of 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, across the billions of weekly page views on LinkedIn.

Figure 2 illustrates the Browsemap system architecture. The Browsemap platform is a hybrid offline/online system. The offline system uses Hadoop [16] 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 [14], which provides low-latency queries.

3.1 Offline Batch Computation

LinkedIn’s frontend services emit activity events on every page view, either on LinkedIn’s website or through its mobile applications. These behavior events are transported to Hadoop via a distributed publish-subscribe messaging system for event collection [15].

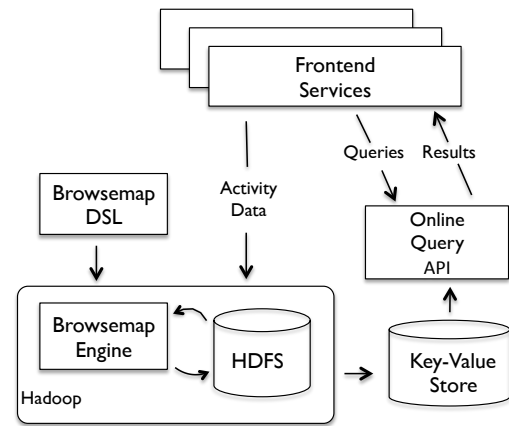


Figure 2: Browsemap architecture consists of offline computation on Hadoop to generate a set of browsemaps, and an online query API, which fetches the results from a key-value store.

The Browsemap Engine uses the well-known technique of co-occurrence or association rule mining [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 members, jobs, and companies, and each entity may have multiple types of activity events. For example, job entity has two types of events—*view* and *apply*, as people can view and apply for jobs. Similarly, the company entity has *view* and *follow* activity events. Multiple event types can be combined 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 separately. 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 or banned users, and job browsemap has an additional requirement to exclude expired jobs.

To meet the different requirements of the various browsemaps, we developed an in-house 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. Figure 3 demonstrates an example that defines the workflow for job and company-follow browsemaps. 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 removing spam user activities, and some modules are specific to a browsemap, such as removing expired jobs.

A configuration file written in the DSL defines a browsemap workflow. First it describes the module dependency: which modules to use 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 frequently due

```

module_collection:
  module: filter_spam_user
  module: filter_expired_job
  module: count_co_occurrence

#...some more modules...

---
application: job-view
refresh_rate: 6 hours
workflow:
  filter_expired_job:
    input: /data/job/views
    output: /data/job/expired_jobs_removed
  filter_spam_user:
    dependencies: filter_expired_job
    output: /data/job/spam_user_removed

#...some more steps...

count_co_occurrence:
  dependencies: ...
  output: /data/job/browsemap

---
application: company-follow
refresh_rate: 1 day
workflow:
  filter_spam_user:
    input: /data/company/follow
    output: /data/company-follow/spam_user_removed

#...some more steps...

count_co_occurrence:
  dependencies: ...
  output: /data/company-follow/browsemap

```

Figure 3: An example of defining job browsemap and company-follow browsemap workflows in Browsemap DSL. The *module_collection* contains a set of *modules*; each performs a particular task. Subsequently, each workflow is defined by chaining the modules together and providing parameters for each module. (The parameter values shown here are for demonstration purposes only.)

to the ephemeral nature of job postings, but the browsemaps for companies can be refreshed less frequently as they are more static.

The collection of modules promotes knowledge sharing and is a main contributing factor for the quick development of a new browsemap. While some modules are specific to a browsemap, 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 [15]. Certain modules are computed incrementally with Hourglass [6], an open source library that operationalizes incremental computation of time series data.

For example, the job entity has job-view events. This dataset is the input to the module that filters out expired jobs, 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 filters activities from spam users. After a few more steps, the co-occurrence counting module is used to do the bulk work of generating the browsemap.

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

3.2 Online Query API

The latent browsemap graph computed by the offline Browsemap Engine is bulk loaded into Voldemort [14], 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: in LinkedIn’s production data centers, more than 99% of requests are serviced within 10 milliseconds.

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

4. APPLICATIONS

The Browsemap platform powers many navigational aids on LinkedIn. They are well received by our members, and a sizable portion of LinkedIn’s traffic is directly attributed to them. Besides being a 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 these applications, in addition to content information such as member profiles and job descriptions. Inclusion of collaborative filtering-based features means simply plugging in the readily available browsemap datasets.

4.1 Navigational Aids

Each entity on LinkedIn has a navigational aid. Figure 1 illustrates a few examples. In Figure 1a, a navigational aid is displayed on a member’s profile that allows members to discover other related profiles. Similarly, the jobs page shows other jobs (Figure 1b) and the group page shows other groups (Figure 1c).

An entity can be associated with multiple types of activity events, such as the case for companies. One aid is powered by the company-follow browsemap and the other is powered by the company-view browsemap (shown in Figure 1d). The two navigational aids serve different needs for LinkedIn members. The company-follow browsemap is for deep engagement with a company; following a company helps members keep track of the status updates from this company. The company-view browsemap, however, is for cursory browsing and serendipitous discovery of content.

Lastly, a particular member segment may want a more customized navigational experience. Recruiting is a prominent use case exercised by premium members 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 plugging in a member selection module that selects viewing events performed by the recruiting community.

4.2 Companies You May Want To Follow

Companies can establish a 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 4, 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. Then, 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 fol-

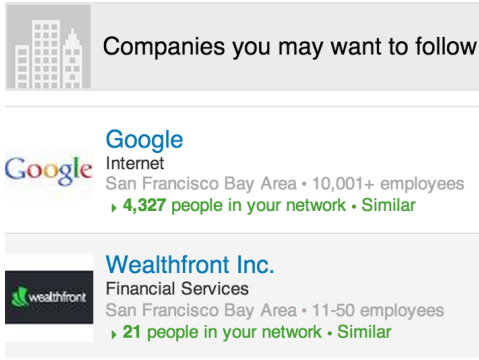


Figure 4: An illustration of “Companies You May Want To Follow”, a member-to-company recommender system. The recommendations are generated by combining signals from company-follow browsemap and other content-based features.

low this company. The companies with high propensity scores are returned as the recommendations for the member.

Figure 5 demonstrates the process where the company-follow browsemap is used to generate the “related-companies” feature. Later, this feature is used to enrich the member profile by augmenting the textual content entered by the member. The feature is generated by iterating through 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 recommender system also uses content-based features. Member features such as industry, location, and experience are used. Company features include company name, industry, location, description, and so on.

With these two types of features of the member and company entities, the propensity score for a (member, company) tuple is computed by first calculating the similarity scores of the related features, and then combining these individual scores together using a logistic regression model. Figure 6 gives some examples of related features: the member’s company-follow browsemaps is matched against the company identifier, the member’s industry is matched against the company’s industry, and the member’s experience is matched against the company’s description.

Explicitly, for member i and company j , there are a set of related features $X_1(i, j), X_2(i, j), \dots, X_n(i, j)$. Training samples are taken from historical data and are labelled with $Y_{ij} \in \{0,1\}$, where 0 means the member i did not follow a company and 1 means the member i followed a company. The logistic regression model can be represented as:

$$P(Y_{ij} = 1|X) = \sigma(b + \sum_{k=0}^n W_k \cdot X_k(i, j))$$

where W_k represents the learned weight for the k th feature, and b is a constant scalar.

The company-follow browsemap is important in this product because it surfaces the implicit 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. The addition of this feature in the training model helps boost the propensity scores of companies that are more related as perceived by our members.

4.3 Similar Companies

The previous product, “Companies You May Want To Follow”, is a member-to-company recommendation: suggesting companies

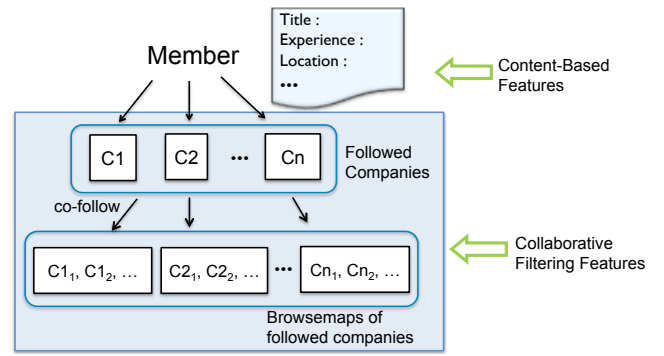


Figure 5: “Companies You May Want To Follow” augments member information with the company-follow browsemap. It iterates through all companies a member already follows, and aggregates the browsemaps of these companies.

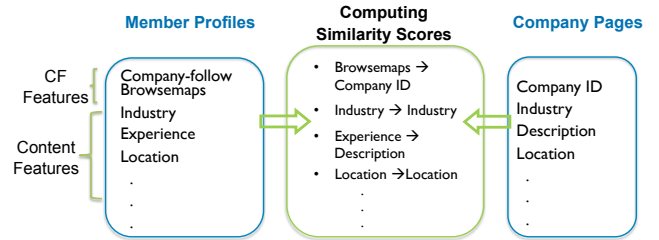


Figure 6: “Companies You May Want To Follow” has two types of features for members: collaborative filtering features extracted from the company-follow browsemaps, and the content-based features extracted from the member profile. The algorithm calculates the similarity scores from the matching features of member and company entities, and computes an overall propensity score from the individual scores.

based on matching member and company information. “Similar Companies”, shown in Figure 7, is a company-to-company recommendation product that suggests to LinkedIn members a set of companies based on matching company information.

Similar to “Companies You May Want To Follow”, the system extracts both collaborative filtering and content-based features from the company entities. The collaborative filtering features include co-occurrence browsemaps built from the *follow*, *view*, and *employed-at* activities on LinkedIn. The *employed-at* browsemap captures the LinkedIn members job transition, that is, “People who worked at company X also worked at company Y ”. High similarity scores of browsemap features indicate strong similarity because of aggregated member behavior.

4.4 Similar Profiles

Helping recruiters and hiring managers find highly qualified candidates is an important service LinkedIn provides. Shown in Figure 8, “Similar Profiles” enables hiring professionals to discover similarly qualified candidates on LinkedIn.

Company-view browsemap and profile browsemap, along with several content features from profiles, are used for this recommender system. The algorithm for “Similar Profiles” follows the same design pattern of the previous two recommender systems. For a set of (source member, target member) tuples, where the “source member” is the member who needs recommendation, and the “target member” is the potential recommended member, our goal is to find a propensity score for each tuple. Target members with high propen-

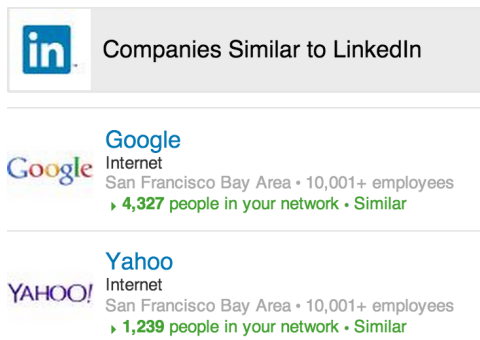


Figure 7: An illustration of “Similar Companies”, a company-to-company recommender system. The recommendations are generated by combining signals from multiple browsmaps and other content-based features.

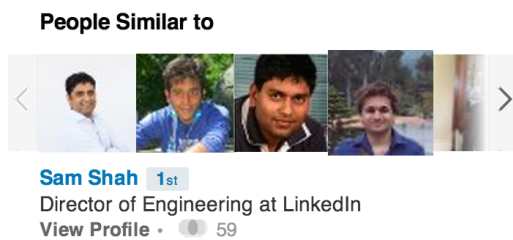


Figure 8: “Similar Profiles” is a hybrid recommender system for member-to-member suggestions. It uses both company-view browsemap and profile browsemap to enrich the profile information.

sity scores are returned as the recommendation for the source member.

The company-view browsemap is used to expand the member’s current company to a set of companies. This expansion significantly increases the recall of the model. Although this enhancement is done with minimal effort thanks to the availability of the browsemap dataset, it is one of the most powerful signals in the model: an A/B split test showed that leveraging the company-view browsemap alone increased profile views by more than 30%.

It is possible to use browsemap to infer content as well, 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 associated profiles. For example, a member’s skill information can be augmented by skills from people he is associated with. We call this the “virtual profile” of the member [10]. In “Similar Profiles”, the profile browsemap is used to find the associated members. The perception 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. An A/B split test shows a 15% lift in profile views with the addition of virtual profiles.

4.5 Suggested Profile Updates

When a member has more detailed information such as work experience, education, and location, LinkedIn can 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 her profile, LinkedIn predicts certain attributes that she has not yet included, such as

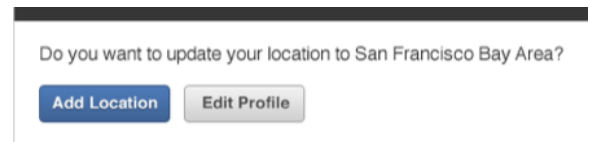


Figure 9: “Suggested Location Update” predicts a member’s location based on profile browsemap and member’s connections.

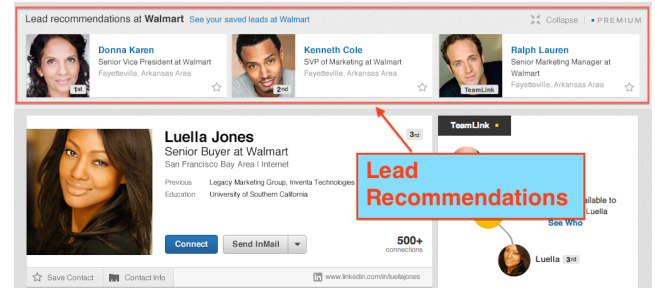


Figure 10: “Lead Recommendations” allows sale professionals to discover new leads at their client companies. It is a hybrid recommender system, combining profile browsemap and content-based features.

company and location. The prediction is shown to the member, and upon approval, the information is saved to her member profile. Figure 9 shows the suggested location update for a member.

Social graphs of a member can provide strong location clues. For predicting user locations, two types of social graphs are used: the latent graph provided by the profile browsemap, and the explicit connection graph the member has established on LinkedIn. The basis for 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 members 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.

4.6 Lead Recommendations

“Lead Recommendations” is a product that helps sales professionals discover more leads at their client companies, as shown in Figure 10. 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 close coworkers who have similar seniority levels as him 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 with whom he works closely, and identifying the colleagues who have similar seniority level as the prospect.

The prospect’s profile browsemap and explicit connection graphs are used to identify his close coworkers. This set of members is the candidate set; that is, the member pool that the recommendations

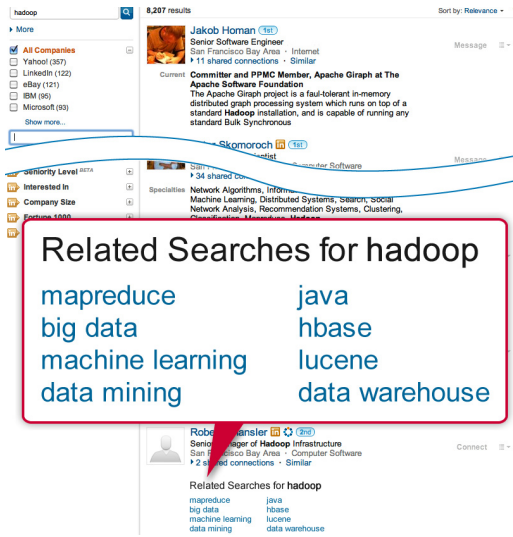


Figure 11: A screenshot of related searches in the context of a search for the query “Hadoop”. It uses search query browsemap as a signal for generating related searches.

are generated from. Similar to “Suggest Profile Updates”, with a collection of (prospect, candidate) tuples, the problem can be formulated as calculating the probability of each tuple. By aggregating the tuples for a prospect, the algorithm returns a top-n list based on the scores.

The profile browsemap is further used to extract seniority features from 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, “lead recommendations” uses several features utilizing seniority information such as the seniority scores of the prospect and the candidate, and the average seniority scores of their profile browsemap and explicit connection graph.

4.7 Related Searches











“Related Searches” is a search tool that suggests other queries that are related to the user queries [11]. As shown by the example in Figure 11, “Related Searches” enables users to refine and explore searches 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 similarity among search queries and to determine a unified set of related search suggestions. The first signal is based on collaborative filtering and is generated by the 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 correlated by clicks on related search suggestions. The system combines the search suggestions generated by each of these signals, where results from collaborative filtering are given the highest preference because suggestions from this signal have the highest click-through rate. In Reda et al. [11], we evaluated each of these signals and unified search suggestions, both offline in terms of precision-recall metrics, and online through A/B split tests. In both of these evaluations, the collaborative filtering-based signal performs significantly better than any other technique.

People Also Viewed

- Curtis Wang**
Applied Research Engineer at LinkedIn
- Roshan Sumbaly**
Engineering Manager at LinkedIn
- Igor Perisic**
VP Engineering at LinkedIn
- Sam Shah**
Director of Engineering at LinkedIn
- Mitul Tiwari**
Staff Research Engineer and Engineering Manager at LinkedIn
- Deepak Agarwal**
Director of Engineering at LinkedIn
- Lili Wu**
Staff Software Engineer at LinkedIn
- Sean Choi**
Software Engineer at LinkedIn
- Jay Kreps**
Principal Staff Engineer at LinkedIn
- Matthew Hayes**
Engineering Manager at LinkedIn

People Also Viewed

-  **Curtis Wang**
Applied Research Engineer at LinkedIn
-  **Roshan Sumbaly**
Engineering Manager at LinkedIn
-  **Igor Perisic**
VP Engineering at LinkedIn
-  **Sam Shah**
Director of Engineering at LinkedIn
-  **Mitul Tiwari**
Staff Research Engineer and Engineering Manager at LinkedIn
-  **Deepak Agarwal**
Director of Engineering at LinkedIn
-  **Lili Wu**
Staff Software Engineer at LinkedIn
-  **Sean Choi**
Software Engineer at LinkedIn
-  **Jay Kreps**
Principal Staff Engineer at LinkedIn
-  **Matthew Hayes**
Engineering Manager at LinkedIn

(a) Profile browsemap without member profile images (b) Profile browsemap with member profile images

Figure 12: An example of UI enhancement without any changes in the items recommended. Showing profile images resulted in dramatic increase in CTR.

5. LESSONS LEARNED

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

Tall oaks grow from little acorns.

With the expansion of data and content on web properties, there is an ever increasing need for recommendation products. The availability of a generic horizontal recommender system that supports different types of recommendation becomes crucial to quickly meet these product requirements.

Initially, we developed a profile browsemap that quickly received traction, which we rolled out to other entity types through a parameterized pipeline. However, we had other applications that would benefit from collaborative filtering, but were struggling with scaling and incrementalizing computation to handle LinkedIn’s data volume. Rather than having each team reinvent the wheel, we embarked on creating the Browsemap 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. The developer’s time is spent mostly in understanding the nature of the product, input data preprocessing, and any vertical-specific requirements.

Browsemaps are frequently 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 are in “following mode” and can display another browsemap of co-follows of that influencer in the flow to further increase conversions.

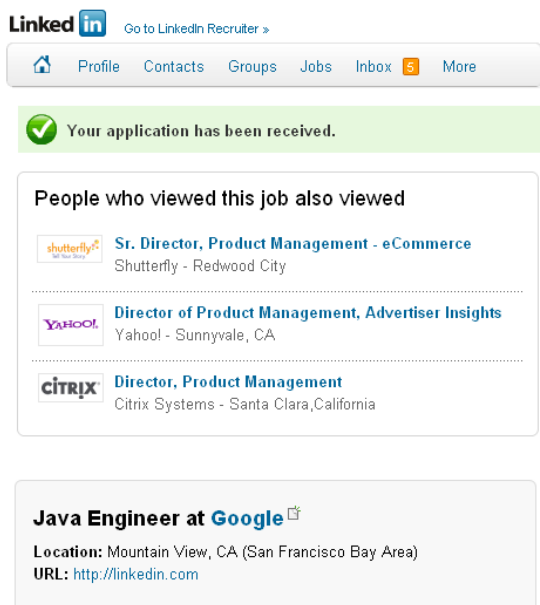


Figure 13: An illustration of job browsemap to guide users to view related jobs after applying for a particular job.

These recommender systems can be augmented as needed with more sophisticated similarity rankers using browsemap data elements as latent features: co-occurrences of views, follows, likes, comments, searches, and so on.

A picture is worth a thousand words.

Our observation, which was 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 then set the right expectations.

To clarify this, consider Figure 12, 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 the explanation 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. The resulting 50% lift in click-through rate was one of the largest lifts in recommendation performance for this product, 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 sent to a confirmation page, as shown in Figure 13. Up to this point the member is in the context of job searching and thus would likely want to explore other related jobs, which is a great vehicle for the job browsemap. An A/B split test that displays the job browsemap at the end of the application process versus one that does not, showed a dramatic 500% lift in the job application rate.

One hand washes the other.

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

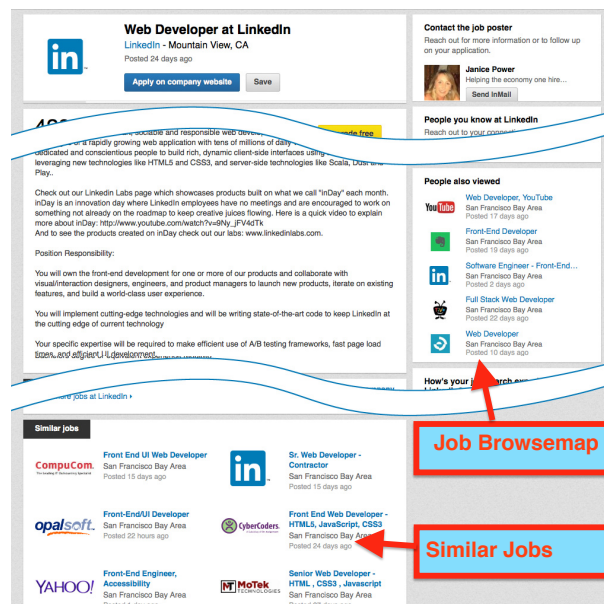


Figure 14: Job description page has both collaborative filtering and content-based recommendations. The two recommendation types can coexist on the same page without cannibalization of engagement.

The job entity page, as shown in Figure 14, 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 performed a true multivariate test, showing both recommendations, showing only one, adjusting locations and the number of recommendations. We found that these recommendation types can coexist without cannibalization of engagement. In fact, they actually amplify conversions because each module’s conversion rate is almost independent of the other, and they independently show different facets. That is, collaborative filtering fulfills members’ curiosity to learn from other people, and content-based recommendation allows the 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 [12]. 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 only sparse activity. Desparsification is vertical-specific and the platform provides techniques that can leverage the social graph or latent properties from other entities [13]. 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 has previously viewed merged through a reduction function. A/B testing has found that this technique can provide high coverage with virtually the same recommendation quality, measured by the click-through rate.

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: browsemaps are beholden to instrumentation on frontend services and the robustness of LinkedIn’s data pipeline. The result was some 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 an 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 [15]. We added robust auditing to ensure correct per-hop reliable data transfer, from the frontend all the way to our relevance systems. The Browsemap platform also includes auditing as part of its run to compare input and output coverage and offline metrics, alerting if there is significant deviation. Further, we have 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 we put these systems into place.

6. CONCLUSION

In this paper, we present Browsemaps, the item-based collaborative filtering platform at LinkedIn. A hybrid of offline/online system, the system computes a latent co-occurrence graph in batch and serves results to users with low-latency. The system’s usability and its quick onboarding procedure have enabled many behavior-based recommendation products at LinkedIn in the past few years. The various datasets Browsemaps produces are also used in many hybrid recommender systems that combine collaborative filtering and content-based methods. In addition to case studies on how LinkedIn uses the Browsemap platform, we presented lessons learned in the field over the several years this system has been in production.

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] K. Ali and W. van Stam. Tivo: Making show recommendations using a distributed collaborative filtering architecture. *KDD*, pages 394–401, 2004.
- [3] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW*, pages 271–280, 2007.
- [4] 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.
- [5] M. A. Hasan, N. Parikh, G. Singh, and N. Sundaresan. Query suggestion for e-commerce sites. In *WSDM*, pages 765–774, 2011.
- [6] M. Hayes and S. Shah. Hourglass: A library for incremental processing on Hadoop. In *BigData Conference*, pages 742–752, 2013.
- [7] N. Koenigstein, G. Dror, and Y. Koren. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *RecSys*, pages 165–172, 2011.
- [8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.
- [9] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.
- [10] H. Liu, M. Amin, B. Yan, and A. Bhasin. Generating supplemental content information using virtual profiles. In *RecSys*, pages 295–302, 2013.
- [11] A. Reda, Y. Park, M. Tiwari, C. Posse, and S. Shah. Metaphor: a system for related search recommendations. In *CIKM*, pages 664–673, 2012.
- [12] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260, 2002.
- [13] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, Jan. 2009.
- [14] 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.
- [15] R. Sumbaly, J. Kreps, and S. Shah. The “Big Data” ecosystem at LinkedIn. In *SIGMOD*, pages 1125–1134, 2013.
- [16] T. White. *Hadoop: The Definitive Guide*. O’Reilly Media, 2010.