Edge formation in Social Networks to Nurture Content Creators

Chun Lo
LinkedIn Corporation
chunlo@linkedin.com

Ankan Saha
LinkedIn Corporation
asaha@linkedin.com

Emilie De Longueau
LinkedIn Corporation
elongueau@linkedin.com

Shaunak Chatterjee
LinkedIn Corporation
shchatterjee@linkedin.com

ABSTRACT

Social networks act as major content marketplaces where creators and consumers come together to share and consume various kinds of content. Content ranking applications (e.g., newsfeed, moments, notifications) and edge recommendation products (e.g., connect to members, follow celebrities or groups or hashtags) on such platforms aim at improving the consumer experience. In this work, we focus on the creator experience and specifically on improving edge recommendations to better serve creators in such ecosystems.

The audience and reach of creators—individuals, celebrities, publishers and companies—are critically shaped by these edge recommendation products. Hence, incorporating creator utility in such recommendations can have a material impact on their success, and in turn, on the marketplace. In this paper, we (i) propose a general framework to incorporate creator utility in edge recommendations, (ii) devise a specific method to estimate edge-level creator utilities for currently unformed edges, (iii) outline the challenges of measurement and propose a practical experiment design, and finally (iv) discuss the implementation of our proposal at scale on LinkedIn, a professional network with 645M+ members, and report our findings.

KEYWORDS

Social Network, Statistical Modeling, Content Marketplace, Network Measurement

ACM Reference Format:

1 INTRODUCTION

Social networks like Facebook, LinkedIn, Instagram and Twitter have had a profound impact on how people interact with each other on the internet. This is amply demonstrated by their ever-growing popularity [25]. The other popular internet products which facilitate interactions are messaging (Facebook Messenger, WeChat, WhatsApp) are some of the most popular platforms and email.

Social networks are distinct from messaging and email platforms because the primary mode of communication is “broadcasting” content to a group of people (often to the broadcaster’s network, or to a platform-specific group). Messaging and email platforms are more commonly used for member to member interactions. Many platforms now offer both functionalities in some form, but we make the distinction in the two modes of communication: the public broadcasting mode, and the private messaging mode. In the broadcasting use case, there are two distinct roles that people play. The broadcaster (content creator) and the audience (the viewer or content consumer). In the private messaging use case, the roles are more symmetric and everyone is a participant (although in group conversations, there can be dominant parties).

Our focus in this paper is on the broadcasting use case in social networks. In this setting, the platform is a content ecosystem with creators and consumers. A social network grows by growing both players in the ecosystem. Creators come to the platform and broadcast content to reach their audience. As more creators create, the consumers have more (and better) liquidity to consume from, hence they get more value and visit more often. This, in turn, increases the creators’ audience and encourages them to create more. This virtuous cycle is an important path to growth for most social networks.

This ecosystem naturally creates two key consumer-facing recommendation problems. Firstly, how to rank the available liquidity on products like feed, moments and notifications to maximize a member’s (i.e., consumer’s) engagement [3, 4]. This liquidity comes from the creators in the consumer’s network. Secondly, how to help a consumer grow their network in size and also to enable them to find more engaging content on the platform [6, 30]. Most social platforms have a lot of focus on solving each of these problems, and have found considerable success via solving them.

The content creators also benefit from the improved consumer experience. However, there may be significantly more upside if we consciously try to improve the creator utility in various recommendation applications. For instance, [34] showed the benefit of estimating how feedback from audience affects a creator’s intent to create again, and then using that estimation to rank content for consumers with an additional creator utility term. There have been several other examples of improving the creator experience and incentivizing them to create new content as well [12, 20, 31].

In this paper, we study the problem of how to use creator utilities in edge recommendation products that grow the network. To the
best of our knowledge, all existing approaches optimize for the consumer utilities in network growth recommendation products. Two utilities often used are number of connections [24] and expected engagement value [30]. The solution relies on estimation of creator utility from an edge, and combining viewer and creator utility in a unified recommendation strategy.

The key contributions of the paper are as follows:

- A general framework to incorporate creator utility in edge recommendation problems
- A proposed approach for an edge-specific creator utility estimation, and its usage in edge recommendation
- A practical experiment design and implementation for the dynamic network setting, accounting for the competition effect among creators, to measure the effectiveness of the approach
- A successful deployment of our solution on LinkedIn, a social network with 645 million users, and showing significant benefit to high-value creators.

In Section 2, we review various related work which provides additional context. This leads into the formal problem formulation and the solution framework, which are covered in Section 3. We discuss certain choices for creator utility estimation and (a)symmetric edge formation and their implications in Section 4. Experiment design requires special consideration in this setting, and we spend Section 5 outlining our proposed solution, and why existing methods may not be a good fit. Section 6 provides details of the deployed use case at LinkedIn and online A/B test results and Section 7 summarizes the paper.

2 RELATED WORK

Network formation algorithms lie at the heart of any social network. These algorithms generate recommendations for all users to build out their edge network which, in turn, facilitate the flow of content on the social network feed for these users from the edges that are formed via the recommendations.

Edge recommendation algorithms can be both symmetric or asymmetric in nature. A product like People-You-May-Know (PYMK) [1] on LinkedIn or Friend Request on Facebook recommends members whom a user would want to connect to. Once the user sends out a connection request, the member receiving the connection recommendation needs to accept the request to enable the edge to be formed. This creates an edge between the two members in the LinkedIn graph and allows updates performed by one of the members to potentially show up on the feed of the other user, if they are chosen to be relevant by the feed ranking algorithm.

On the other hand, Follow recommendations on LinkedIn, Twitter and numerous other social networks enable users to directly click on these recommendations and follow a set of diverse entities. This allows content (updates) from these entities to possibly show up on the user’s feed. However note that unlike PYMK (which results in a symmetric edge between the members), follow recommendations forms an asymmetric edge between the user and the entity, in the sense that only the user is able to see content from the entity and not the other way around. Most of the ideas in the paper can be extended to both sets of edge recommendation algorithms.

In this section, we explore some of the related work in the space of follow recommendations which has been a core problem at the center of most social networks. There have been several lines of work predicting follow recommendations using a combination of collaborative filtering and content based signals, with features obtained from tweets of both user and recommended entity, exploring real time signals from Twitter [18] SocialCollab [13] used a neighbor-based collaborative filtering algorithm that improves on traditional collaborative filtering by explicitly handling the attractiveness of the recommended entity. The authors of [32] explored the topology of the subgraph by using genetic algorithms to look at connected people up to three degrees of separation from a user. Separately there have been connection recommendation algorithms generating aggregate information from candidates based on a combination of viewer features, candidate features as well as combination features generated from the pair - using a set of simple features to choose a manageable candidate set from the set of all unconnected users, which are then ranked to obtain the list of recommendations [17]. There has also been work predicting tie strength on existing edges in a social network and using information based on that as a predictor for making edge value recommendations [15].

There have been other ideas exploring supervised random walks in social networks combining information from network structure with node and edge level features [9] and using it to recommend links to form for members. Similarly other link prediction methods based on local similarity measures between the nodes of the graph [36] can also be used to generate nodes to follow in the graph.

Separately there is an entire body of work to identify influencers in a network, topic based [27], extending PageRank algorithm to topic-based similarity scores [19], using influence maximization techniques to identify authors who are most influential in a social network [14, 23]. The influencers identified via all of these methods can be used as candidates for follow recommendations on the graph.

3 PROBLEM FORMULATION

The general edge recommendation problem has two primary considerations:

- The probability of the edge being formed, \( p_{\text{Edge}} \)
- The value of the edge, if formed, \( V(\text{edge}) \)

Let \( u \) denote the user who is being shown the edge recommendations, and \( e \) denote the specific entity with whom \( u \) is being recommended to form an edge. Let \( x_{u,e} \) denote the probability of recommending the edge \((u, e)\) and \( c_u \) denote the number of recommendations that can be shown to user \( u \). The objective of an edge recommendation application, without any loss of generality, can be expressed as:

\[
\max \sum_{u,e} x_{u,e} \cdot p_{\text{Edge}}(u, e) \\
\text{s.t., } \sum_{u,e} x_{u,e} \cdot p_{\text{Edge}}(u, e) \cdot V(\text{edge}) \geq V \\
\sum_{e} x_{u,e} \leq c_u, \quad \forall u \\
0 \leq x_{u,e} \leq 1, \quad \forall u, e
\]  

Equation 1 translates into maximizing the number of edges formed, while ensuring a certain amount of value (\( V \) to be specific)
is delivered as a result of the formed edges. The other constraints express the number of recommendations available for each member, and restrict $x_{u,e}$ to be probabilities. Using the Lagrangian dual formulation [11], we can obtain the optimal dual variables using historical data (which is representative of the ecosystem). Let $\alpha$ denote the optimal Lagrangian dual corresponding to the value constraint, then the mathematical form for the score of the pair $(u,e)$ is¹:

$$\text{Score}_{u,e} = p\text{Edge}(u,e) \cdot [1 + \alpha \cdot \text{Value}(u,e)] \quad (2)$$

The actual serving probabilities (i.e., $x_{u,e}$) are determined by ranking the entities for a given user according to the score in Equation 2. Mathematically, this can be written as:

$$x_{u,e} = \begin{cases} 1, & e = \arg\max_{e' \in \text{EntityCandidates}(u)} \text{Score}_{u,e'} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where EntityCandidates($u$) represents the set of edge formation candidates for user $u$, from which we choose the entity $e$.

An alternate (and actually equivalent) interpretation of Equation 2 is that $\alpha$ determines the relative importance of edge value compared to number of edges. It is equivalent because as we increase $V$ in Equation 1, which is indicative of increasing importance of value, $\alpha$ increases as well. Hence, the linear combination of the two utilities is the optimal way to balance the two utilities as long as our exact optimization problem is expressible via Equation 1.

### 3.1 Edge utility to a consumer

In the previous formulation, the edge-specific value function is left unspecified. Some possible consumer centric utilities, which would be used as Value in Equation 2, include:

- A global constant. This is equivalent to 0, since it doesn’t affect ranking (as per Equation 2)
- Probability of an interaction on the edge
- Expected number of interactions on the edge

Understandably, the $\alpha$ values would be very different depending on the utility that we choose.

### 3.2 Edge utility to a creator

A creator, as mentioned before, relies on the evolution of the network to build a growing audience and increasing reach. This improves their motivation to create content on the platform, which in turn produces an increased volume and also (possibly) better content liquidity for the consumers. We will expand on possible creator utilities and details of how to estimate them in Section 4.

Let $\text{Value}_{\text{consumer}}$ and $\text{Value}_{\text{creator}}$ represent the value functions to the consumer and creator respectively. Then, we can introduce consumer and creator specific constraints in Equation 1, and let’s assume $\alpha$ and $\beta$ are the optimal Lagrangian dual variables corresponding to the consumer and creator value constraints. In that case, the score of the pair $(u,e)$ becomes:

$$\text{Score}_{u,e} = p\text{Edge}(u,e) \cdot [1 + \alpha \cdot \text{Value}_{\text{consumer}}(u,e) + \beta \cdot \text{Value}_{\text{creator}}(u,e)] \quad (4)$$

### 3.3 Notes on optimality

There are a few different conditions which may affect the optimality of the decisions made using Equation 4. We list these to make the discourse more complete, and also raise the readers’ awareness on these factors.

**Accuracy in value estimates.** The recommendations’ effectiveness will depend very heavily on the individual accuracy of the component estimation models, namely $p\text{Edge}$, $\text{Value}_{\text{consumer}}$ and $\text{Value}_{\text{creator}}$. If (any or all of) the estimators have high variance, then that will increase the variance of the overall recommendation process.

**Past and present disparity.** The proposed solution makes the standard assumption of feature distribution invariance between the past and present. The optimal Lagrangian duals can guarantee optimality when the distributions are the same.

**Correlated value functions.** If the different value functions are correlated (either positively or negatively or not at all), it doesn’t affect the accuracy of the method. However, highly correlated value functions can reduce the variance of the solution by utilizing the accurate estimate of one value function to estimate another.

**Dependence in the value function.** One particularly challenging scenario is when the decisions made by the recommendation system affects the value estimates of future opportunities. This makes Equation 1 non-linear and most likely non-convex. For the rest of this paper, we will assume that this is not the case.

We next delve into more details on possible ways of estimating the creator value function, and how the asymmetric nature of an edge affects the problem formulation.

### 4 CREATOR UTILITY ESTIMATION

In this section, we discuss how to estimate the value of an edge to a creator. As posited before, we assume that creators are incentivized by the audience that they can reach and engage with. Possible creator value functions include:

- **Feedback sensitivity:** The effect of feedback on the creator’s future creation behavior
- **Audience size sensitivity:** The effect of a growing audience on the creator’s behavior

#### 4.1 Feedback Sensitivity Score

We want to model how feedback affects a creator’s creation activity. Hence, we define the feedback sensitivity score as the change in probability of creation of a creator associated with an increase of feedback² and denote it by $FS_e$ for creator $e$. A recent work by [34] looks at this specific problem and we present a brief summary of their proposal since it is critical to our work. The core component of the proposal is to build a “pCreate” model that estimates a creator’s creation behavior conditioned on many covariates including feedback volume, and uses this model to derive feedback sensitivity.

---

¹For a detailed derivation, please refer to [5]

²We would ideally like to estimate the causal effect on the creator’s behavior due to the feedback change. This is an orthogonal topic, which we do not further address here. Causal models for creator utility will have improved accuracy, but still be used in network recommendation in the same way as we propose
4.1.1 pCreate Model. The pCreate model predicts, at any given time \( t \), the probability of creator \( e \) creating at least one content piece in the following time window \([t, t + w]\) given all features \( a_e \) and \( S_e \), i.e., \( P(Y_e > 0 | a_e, S_e) \), where \( a_e \) is the number of feedback items that creator \( e \) received in the previous time window \([t - w, t]\), \( S_e \) is the set of static features of creator \( e \) at time \( t \), and \( Y_e \) is the number of creations of creator \( e \) in the time window \([t, t + w]\). A linear logistic model is learnt (partly to keep online serving latency low) using historically observed data, to model the probability of creations as follows:

\[
p_e = \frac{1}{1 + e^{-(\mu + \gamma T_S + \lambda a_e + \beta Y_e + (a_e \times S_e) + \delta)}},
\]

(5)

where \( \mu \) is the intercept, \( \lambda, \beta, \gamma \) are the coefficients for \( a_e, S_e, \) and \( (a_e \times S_e) \) respectively. \( (a_e \times S_e) \) are the interaction between the feedback features and the static creator features, which allow us to better distinguish how feedback affects future creation among member cohorts.

Note that the relationship between feedback \( a_e \) and \( logit(P(Y_e > 0) \) is non-linear. To better capture this non-linearity, the proposal bucketizes the numeric feedback feature \( a_e \) (and as well the feature \( a_e \) in the interaction feature \( (a_e \times S_e) \) into a \( k \)-level categorical feature, \( a_e = \{v_1, \ldots, v_k\} \). The bucketization intervals can be determined by a correlation analysis (between feedback received and probability of creation).

4.1.2 Estimating the Feedback Sensitivity Score. The Feedback Sensitivity Score of creator \( e \), denoted as \( FS_e \) is defined as the delta probability of content creation given an increase in feedback. Let \( \hat{p}_e = \hat{P}(Y_e > 0 | a_e, S_e) \) be the estimated probability of content creation and \( \delta \hat{p}_{e,k} \) be the estimated feedback sensitivity measure of creator \( e \) with received feedback \( a_e = v_k \). \( \delta \hat{p}_{e,k} \) can be calculated by:

\[
\delta \hat{p}_{e,k} = \frac{\hat{P}(Y_e > 0 | a_e = v_k, S_e) - \hat{P}(Y_e > 0 | a_e = v_{k-1}, S_e)}{v_k - v_{k-1}}
\]

(6)

A large number of features can be used in estimating \( \hat{p}_e \), and this is desirable\(^3\) to increase the accuracy of \( \hat{p}_e \) and especially that of \( \delta \hat{p}_{e,k} \), which is a rather personalized measure. Using bucketized version of the feedback feature \( a_e \), instead of the raw feature allows us to have a more stable \( \delta \hat{p}_{e,k} \) estimate. We use \( k \) to refer to the most common bucket corresponding to feature \( a_e \) and represent the Feedback sensitivity as \( FS_e = \delta \hat{p}_{e,k} \).

4.2 pResponse Model

Let \( R_{u,e} \) be the number of feedback pieces provided by consumer \( u \) to creator \( e \) during window \([t, t + w]\). The pResponse model, and more specifically \( pResponse_{u,e} \), measures the probability of consumer \( u \) providing at least one feedback to creator \( e \) given all creator \( e \)’s features \( X_e \) and consumer \( u \)’s features \( X_u \). The binarized version of the label is often chosen as that retains the most critical information (i.e., whether or not the consumer will provide feedback to the creator), while also leading to lower variance estimates.

The value to the creator \( Value_{creator} \) is represented as \( pResponse_{u,e} \times FS_e \), which provides an estimate of the (directional) edge value between consumer \( u \) and creator \( e \), for incentivizing creator \( e \) to create content due to feedback provided by consumer \( u \).

4.3 Audience size sensitivity

Creators could also be incentivized by the volume of followers (i.e., increase in the size of audience or reach) that they acquire. This depends on the specific feedback loops in the platform in question. For instance, some platforms notify a creator when anyone follows them. The appropriate way to handle such a scenario would be to extend the aforementioned “pCreate” model to include the number of edges formed (or followers acquired) as a signal, and quantify the “Audience growth sensitivity” score, or \( AGS_e \) of creator \( e \). The general form of Equation 4 which incorporates both \( FS_e \) and \( AGS_e \) is shown in Equation 7. We will not pursue \( AGS_e \) further in this work, and hence that term will also not be considered in our scoring function to keep the narrative easy to follow.

\[
Score_{u,e} = pEdge(u,e) \times (1 + \alpha Value_{consumer} + \alpha AGS_e \times pResponse_{u,e} \times FS_e)
\]

(7)

4.4 Symmetric vs asymmetric edge formation

As described in section 2, edge recommendation algorithms in a social network can result in both symmetric as well as asymmetric edges. The entire framework for \( Value_{creator} \) works for both these types of networks and can prioritize feedback sensitive creators in both these cases. However, in case of an asymmetric network, both the viewer and creator effects are independent and insulated compared to the case of symmetric networks. This is largely due to the fact that the recommendation candidate does not see any content from the viewer in case of Follow Recommendations and \( Value_{creator} \) and \( Value_{consumer} \) are separate and apply to only the appropriate parts of the marketplace.

In a symmetric network (like PYMK on LinkedIn or Friend Recommendations on Facebook), the recommended candidate is affected by \( Value_{creator} \) as well as \( Value_{consumer} \) (since she gets to see content from the viewer as well) and this affects the optimization objective to modify the final list of recommendations. This makes it more complicated to attribute and measure the exact effect on the viewer and creator sides. For brevity, we focus on Follow recommendations as a case study and choose to look into the details of the algorithm, challenges and online experimentation in the problem setting of follow recommendations at LinkedIn in section 6.

5 EXPERIMENT DESIGN

The impact on the consumer side can be measured by a standard online A/B test. This is because the consumers as the units of experimentation, satisfy the Stable Unit Treatment Value Assumption (a.k.a. SUTVA [28]), which states that the effect of treatment on any unit of the experiment is independent of the treatment allocation to the rest of the experimental units. This is a standard assumption
Figure 1: Achieving target boosting of feedback sensitive entities with proper boosting weight $\beta$ and attention balancing constant $c$. (a) Small perturbation: Scoring with a very small $\beta$ and no attention balancing. (b) Biased perturbation: Scoring with proper $\beta$ with no attention balancing leading to cannibalization of creators in Control. (c) Attention-balanced perturbation: Scoring with proper $\beta$ and with an optimal attention balancing offset $c$. The shift in the origin of the arrows indicate the effect of the offset. (d) Unbiased perturbation: The desired goal, significant boosting while balancing the attention received by creators in Treatment and Control.

made in most experimentation carried out in the internet industry today⁴.

The impact on the creator side requires a more careful experiment design since the SUTVA is very likely to be violated. The proposed scoring function (see Equation 4) boosts up feedback sensitive creators. When we launch an A/B test, this boosting may cannibalize creators in control and thus give over-optimistic measures. To understand the cannibalization effect, let’s consider the illustration in Figure 1. Each circle represents a creator, with the height representing the position in ranking due to value of the scoring function. When creators are partitioned in to Treatment and Control groups, we use white circle to represent creators in Control and green circles to represent creators in Treatment. The saturation of green color represents the magnitude of the creator utility, $\text{Value}_{\text{creator}}$. The higher the saturation, the larger the utility magnitude.

The creators in Control are ranked using the function without the creator utility (i.e., equation 2). The creators in Treatment are ranked using Equation 4, which gives each such creator a boost of $\beta \times p\text{Response}_{u,e} \times FSE$ in the score. This is illustrated by the arrows in Figure 1(a). Figure 1(b) shows an example of creator ranking, with respect to a consumer, under the original scoring function and the proposed scoring function with a big enough $\beta$ to change the ranking of creators in Treatment. Note that both $p\text{Response}_{u,e}$ and $FSE$ are non-negative. The creators in Treatment will have a higher average score than creators in Control under the new scoring function. As consumers will only observe the ranking position, the creators in Control are being effectively boosted down or cannibalized. This violates the stable unit treatment value assumption (SUTVA), which requires the experience of a creator to be unaffected by the particular assignment of treatment to the other creators. As a result, a simple A/B test is not able to report accurate measurement of the new scoring function on creators.

5.1 Network A/B testing

Scenarios where SUTVA are violated are not very uncommon in experiments in the internet industry. This is an active area of research [7, 8, 16, 21, 22, 33, 35].

All methods that we are aware of, treat the graph as a static entity, and then perform operations to design experiments. This includes:

- The simplest approach of identifying large sub-graphs in the network, which are largely isolated from the rest of the graph (e.g., countries like New Zealand and Netherlands are popular choices). Since this leads to very few data points, there is follow-up work on estimating the value of other cohorts from those observed data points.
- Creating a much larger number of small sub-graphs by minimizing the number of inter-cluster edges (which are dropped from consideration). Each subgraph has an “ego” and the
network of the ego (referred to as the “alter-egos”). One alter-ego can only belong to one graph. More details can be found in [29].

- For a specific kind of exposure redistribution experiment (our use case would also be appropriate for this), [26] propose an optimization formulation to identify a modified experiment that provides the same experience to the nodes on which measurement will be done.

All of these methods take a static graph to then define a particular experiment design that minimizes the network effect or leakage or cannibalization. In an edge recommendation application, the graph and its set of edges are not static. One possibility is that we consider the candidate set of edges periodically (say every day). However, this has two potential issues:

- We have to re-identify ego-clusters or an edge-specific treatment definition every day. This may be computationally infeasible for large graphs.
- Compounding the issue, we will have to introduce consistency constraints across assignments from one day to another. For instance, we will want to maintain the same egos. This will start affecting the quality of the results as the time window of the experiment increases.

5.2 Node-level effect

Instead of trying to use one of the static network experiment design methods, we consider an alternate possibility. We try to mimic the effect of the entire universe being in Treatment on a creator node (this is akin to [26]) without instantiating the actual candidate edge recommendations to the creator. Instead we consider a distribution of those edges and consider a creator level adjustment to counter the cannibalization effect.

When the entire universe is in Treatment, instead of boosting up all creators in Treatment, only the creators with high creator utility will be boosted up, and the ones with low creator utility will be boosted down. So can we modulate the scoring function for creators in Treatment to mimic the situation when all creators (and hence the entire universe) are in Treatment? Let \( c_e \) be a constant specific to creator \( e \). We consider a modified scoring function shown in Equation 8. In order to compute the value of such a \( c_e \), we would have to consider the distribution of \( \beta \cdot \text{pResponse}_{u,e} \times \text{FS}_e \) for creator \( e \) (i.e., the entity \( e \)) over the incoming users \( u \).

\[
\text{Score}_{u,e} = \text{pEdge}(u,e) \times (1 + \alpha \cdot \text{Value}_{\text{consumer}} + \beta \cdot \text{pResponse}_{u,e} \times \text{FS}_e - c_e)
\] (8)

5.3 Balancing creator populations

Learning a creator specific weight \( c_e \) can be quite noisy, since the variance in the distribution of \( \beta \cdot \text{pResponse}_{u,e} \times \text{FS}_e \) for an individual creator \( e \) can be quite high. Another alternative is to consider a single constant \( c \) (see Equation 9) which would be applied to all creators in Treatment (as shown in Equation 9) in the scoring function. We would compute \( c \) by considering the distribution of \( \beta \cdot \text{pResponse}_{u,e} \times \text{FS}_e \) for all creators \( e \) in Treatment. Since this is a distribution over a much larger set of nodes (and their edges), the variance is much lower.

\[
\text{Score}_{u,e} = \text{pEdge}(u,e) \times (1 + \alpha \cdot \text{Value}_{\text{consumer}} + \beta \cdot \text{pResponse}_{u,e} \times \text{FS}_e - c)
\] (9)

Computing the value of \( c \). The balancing constant, \( c \) is determined using an offline simulation mechanism. First, instead of balancing the score, in practice, we balance the proportion of creators from Treatment and Control across all consumers. For example, if the population consists of 40% of creators in Treatment and 60% of creators in Control, under the new scoring equation (9), there should be, on average over all edge recommendations, 4 creators from Treatment and 6 creators from Control in the top 10 positions. The average ranking of creators in Treatment is monotonically decreasing with the balancing constant \( c \). therefore, we can use binary search to find the optimal \( c \). As boosting weight \( \beta \) directly affects the magnitude of the attention balancing \( c \), we need to re-calculate \( c \) whenever \( \beta \) is changed.

In summary, we choose the last approach to lower the variance of our estimation of \( c \). The middle approach (\( c_e \)) has lower bias, since we can correct at the level of each creator, but incurs higher variance. Methods which use the whole graph information and estimate edge-level adjustments for experiment design are the most expressive and hence have the least bias. However, they also have the highest variance since we have to set the parameters with very little data per parameter. There are also other challenges in using methods that require a static graph.

6 FOLLOWS RECOMMENDATION
APPLICATION

Follow recommendations at LinkedIn (figure 2) form a unique heterogeneous edge formation tool, with followed entities comprising of members, companies and more recently hashtags. These recommendations are an important piece of the LinkedIn content ecosystem, averaging with a significant number of daily interactions on content coming from followed entities. We want follow edges to act as a strong indicator of what a user is interested in, which can be leveraged to show more engaging updates to the user. Currently the number of people seeing follow recommendations and content from follow edges on their news feed exceed tens of millions on a daily basis. It goes without saying that follow recommendations form an integral part of the LinkedIn Ecosystem.

The first iterations of the problem would optimize purely for the probability of the follower following an entity. We have had different iterations of these models with the latest model involving a combination of viewer features, entity features (node features) and interaction features between them (edge features), which are all fed into an XGBoost model. As expected, we use miscellaneous information about the users including the size of their network and their behavior (interaction patterns on the LinkedIn feed as well as previous follows behavior) to generate node features, while edge features revolve around how the viewer interacted with segments (like industry, company, region) that the entity is associated with.

Subsequently, akin to equation (2), we modified the objective to take the downstream value of the consumer into account. We can have different choices for downstream value including the liquidity of inventory generated by following an entity or the amount of
Figure 2: Examples of Follow Recommendations on LinkedIn: (a) Member follow recommendations grouped by a specific interest. (b) Personalized member follow recommendations on mobile. (c) Personalized member follow recommendations on desktop homepage right-rail. (d) Company page follow recommendations, grouped by consumer interests (company pages are currently not considered in the creator side utility).

Figure 3: The dual A/B testing setup for viewer and producer side experiments

6.1 Incorporating Valuecreator

We extend the objective in equation (10) in the format akin to equation (4) which leaves us with the following objective:

$$Score_{u,e} = Pr(u \text{ follows } e | \text{impression}) \times (1 + \alpha \times EDSA_{u,e} \times Value_{consumer} + \beta \times pResponse_{u,e} \times FS_{e} - c)$$

where $u$ refers to the user, $e$ refers to an entity and the probability is conditioned on the entity $e$ being impressed on $u$. For building the $EDSA_{u,e}$ model, we rely on the set of features from the pEdge model, including features about how the viewer interacts on the feed (prior Click-Through-Rate features) as well as features relying on the number of past feed interactions by the viewer on segments related to the entity (industry, region, company).

Note that the parameter $\alpha \in [0, \infty)$ controls the trade-off between the immediate probability of follow and the downstream utility that the edge provides. Larger values of $\alpha$ tend to boost entities which have a higher potential downstream interaction value in the recommendation list. For our experiments, we choose $\alpha$ through hyperparameter optimization [2].

pResponse Model. We build a logistic regression model to calculate $pResponse$ (described in section 4.2 which considers binarized versions of feedback (currently all likes and comments) provided as positive labels (as described in section 4.2) and trains the model on viewer, entity as well as interaction features.

The details of the $FS_{e}$ are similar to the description in Section 4. We implemented the new scoring function and launched online A/B testing experiments on LinkedIn Follows application. These experiments target a subset of member entities who are avid content creators.
producers and are also keen to grow their follower base, indicating the latter by opting in to allow other members to follow them. The final list of recommendations are obtained by implementing the attention balancing constant \((\beta, c)\) described in section 5. The impact on viewer side and entity side are discussed at the end of this section.

6.2 A/B testing for followee and followers

As mentioned previously, the new ranking function is expected to have trade-offs between followers’ interests and followees’ interests. We used a dual A/B testing setup to measure both viewer and producer impacts simultaneously. This dual A/B testing consists of two independent random user assignments which partition members into control and treatment groups. As shown in Figure 3, the left random user assignment partitions viewers into control \((C_u)\) and treatment \((T_u)\) groups while the right random user assignment partitions producers into control \((C_e)\) and treatment \((T_e)\) groups. In follow recommendations, only the edges between \(T_u\) and \(T_e\) are scored by the proposed scoring equation (11). The remaining edges are scored by the original scoring equation (10). To measure the experiment impacts on viewers, we only compare the metrics between \(T_u\) and \(C_u\). Similarly, we compare the metrics between \(T_e\) and \(C_e\) to measure the experiment impacts to producers.

6.3 Handling Impression Skewness

The balancing constant calculation method described in section 5 is based on the assumption that followees in Treatment and Control have the same average score under the Equation (10). However, when we analyze historical scoring data, we found there are a few “super” followees (or influencers) consistently have higher ranking score than normal followees, i.e., they appear in most followers’ recommendation. Among these influencers, the degree of dominance also varies widely. These influencers heavily skewed the scoring distribution. Even we randomly partition followees into Treatment and Control, the average ranking of followees from Treatment and Control has a high probability to be different, under the original scoring equation (10). Consequently, the proposed method no longer give a valid balancing constant.

The impact of influencers on the balance of ranking is summarized in Figure 4. We analyzed impression (which is a direction representation of ranking scores) received by the followees in Treatment \((T_e)\) and Control \((C_e)\) with respect to the inclusion of influencers. As we want to understand the skewed issue in the existing scoring system, only impression given by the followers from Control \((C_u)\) is used. We rank all followees (without separating into Treatment and Control) according to their total number of received impressions. Then the impression data is analyzed under different percentiles of followees with increasing number of received impressions. The left \(y\)-axis on Figure 4 shows the difference of average number of impressions received by followees in Treatment \((T_e)\) and Control \((C_e)\), \(\text{Imp}(T) - \text{Imp}(C)\). The average number of impressions is the total number of impressions received by the followees in \(T_e\) or \(C_e\) divided by the corresponding total number of followees in the group. The \(x\)-axis in Figure 4 shows the percentile of the followees used in the calculation of \(\text{Imp}(T) - \text{Imp}(C)\). The right \(y\)-axis represents the percentage of total impressions that is received by the corresponding percentile of followees. The value of impression coverage is censored because it is sensitive data to LinkedIn. To give a rough scale of the impression distribution, an influencer can easily receive 100 times more impression than a normal followee.

As can be seen in Figure 4, followees in Treatment and Control has similar average number of impressions when the percentile is smaller than 99.36. After 99.36 percentile, the discrepancy, \(\text{Imp}(T) - \text{Imp}(C)\), increases rapidly. Although these influences are only a small percentage of followees, they yielded a significant number of impressions. We further examine the feedback sensitivity, \(\delta \beta_{i,k}\), and confirmed that they have much smaller feedback sensitivity than the rest of followees. This is expected as they are likely popular creators and have already received plenty of feedback from viewers. As these upper 0.34 percentile followers skews impressions but their feedback sensitivity utility is negligible, we decided to treat them as outliers and excluded them before our balancing constant calculation and online A/B test analysis. Although removing these influencers reduces significant amount of impression coverage, we still have enough impression data to get high power A/B test results. Here we assume the distribution of impressions given by viewers from \(C_u\) to \(T_e\) and \(C_e\) is consistent with the distribution of impressions given by viewers from \(T_u\) to \(T_e\) and \(C_e\).

6.4 Online A/B testing

After we removed the outliers and balanced the attention given to followees in Treatment and Control, we launched orthogonal A/B tests to measure the effect of creator boosting separately. Note that boosting up feedback sensitive entities might come at the cost of recommending the most relevant entities in terms of immediate engagement. This trade-off is controlled by the parameter \(\beta\) in Equation (11). We first perform offline analysis to select a few candidates \(\beta\), which are big enough to have moderate changes in Follows recommendation. For example, in Figure 1(a), \(\beta\) is too small to change the relative position of entities in Treatment, while with a larger \(\beta\) (Figure 1(b)), the third entity is ranked higher than the second entity which has a lower creator utility. Then, for each \(\beta\), we launched an online experiments to understand the true tradeoffs and select the largest \(\beta\) which positively impact the creator side metrics while having minimal negative impact on the standard follower metrics. Table 1 lists the metrics that we are interested in. It consists of both producer metrics and viewer interaction metrics.
Table 1: Metrics of Interest

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributions</td>
<td>Actions by users that generate or distribute content in the ecosystem (includes public ones: shares, likes, comments; and private ones like messages).</td>
</tr>
<tr>
<td>Sessions</td>
<td>Number of member visits to the site</td>
</tr>
<tr>
<td>(Public/Private)</td>
<td>Unique number of users who have (public/private) contribution activities.</td>
</tr>
<tr>
<td>Contributors with</td>
<td>Unique number of contributors who receive a response within a time window.</td>
</tr>
<tr>
<td>Response</td>
<td></td>
</tr>
<tr>
<td>Retained Creators</td>
<td>Unique number of content creators who are retained from a previous time window.</td>
</tr>
<tr>
<td>(Member) Follow</td>
<td>Number of clicks on Follow (member) recommendations.</td>
</tr>
<tr>
<td>Clicks</td>
<td></td>
</tr>
<tr>
<td>L5d Interactions</td>
<td>Number of downstream interactions (e.g., click, comment, share, etc.) performed by a follower on content generated by the followee in the last 5 days</td>
</tr>
<tr>
<td>From Algorithmic</td>
<td></td>
</tr>
<tr>
<td>Follows</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Viewer Side Metrics Impact

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Delta % Effects (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Member Follow Clicks</td>
<td>-4.22% (&lt; 1E - 4)</td>
</tr>
<tr>
<td>All Follow Clicks</td>
<td>-1.81% (&lt; 1E - 4)</td>
</tr>
<tr>
<td>L5d Interactions From Algorithmic Follows</td>
<td>-4.48% (&lt; 1E - 4)</td>
</tr>
</tbody>
</table>

Table 3: Creator Side Metrics Impact

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Delta % Effects (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily Unique Contributors</td>
<td>+1.13% (0.05)</td>
</tr>
<tr>
<td>Macrosessions</td>
<td>+1.8% (4E - 3)</td>
</tr>
<tr>
<td>Public Contributions (Monthly)</td>
<td>+9.04% (0.01)</td>
</tr>
<tr>
<td>Unique Contributors (Re-share)</td>
<td>+2.31% (0.03)</td>
</tr>
<tr>
<td>Unique Contributors (React)</td>
<td>+0.87% (0.04)</td>
</tr>
<tr>
<td>Unique Contributors With Response</td>
<td>+1.24% (0.02)</td>
</tr>
<tr>
<td>Unique private Retained Contributors</td>
<td>+1.83% (0.03)</td>
</tr>
<tr>
<td>Unique Private Contributors With Response</td>
<td>+1.43% (0.01)</td>
</tr>
</tbody>
</table>

Impact of incorporating creator utilities in edge recommendations on both consumers (e.g., followers) and on creators.

In the currently implemented solution, it is a bit tedious to estimate the pair of parameters ($\beta, c$), which are part of the attention balancing mechanism as described in Section 5. Every modification in the traffic assignment to a creator-side model requires re-estimating these parameters, which is a data and time-intensive operation. This process can be largely automated by hyper-parameter optimization (See [2, 10] and references therein).

While the introduction of the orthogonal random user assignments allow us to measure the effect on both follower and creator side, these results can still be diluted by second order effects, where it is difficult to attribute content creation specifically to feedback provided by followers in treatment as opposed to other viewers in such a follower’s network, who might have been allocated to control. One on-going line of research is to more accurately measure the treatment effect in such two sided marketplaces [26] through better experiment design.

To the best of our knowledge, this is the first reported work on introducing creator utility in the edge recommendation problem. We leverage the creator utility estimation work in [34] and present a novel application that complements the newsfeed ranking application introduced in that paper. The baseline tested is what exists in most edge recommendation systems today (i.e., no creator utility at all). While we have shown the feasibility of our approach, there could be other interesting ways of incorporating creator utility into growing networks. Our main objective is to propose one methodologically sound and practically feasible approach that practitioners can adopt, and researchers can build upon (or compare against).

ACKNOWLEDGEMENT

We would like to especially thank Ye Tu for several insightful discussions, and enabling us to implement our current idea on top of some of her prior work. The internal LinkedIn reviewers – Jinyun Yan, Wei Lu and Rakesh Malladi – as well as the anonymous reviewers also provided some great feedback on technical points and writing style to help improve the paper.

REFERENCES
